Verifying the correctness of structural engineering calculations

by

Douglas William Brown

Submitted for the degree of Doctor of Philosophy in

Structural Engineering

School of Engineering

University of Surrey

Guildford, Surrey, UK.

Examiners

Professor I. A. MacLeod, University of Strathclyde

Professor H. Nooshin, University of Surrey

Verifying the correctness of structural engineering calculations

Abstract

In 1997, The American Society of Civil Engineers published a report prepared by their Task Committee on *Avoiding Failures caused by Computer Misuse*, the self-checking procedures developed in this research have been designed to prevent such misuse. In 2002, The Institution of Structural Engineers, published *Guidelines for the use of computers for engineering calculations*, which commence "These guidelines have been prepared in response to growing concern regarding the appropriate use of computers for structural calculations" and end with "Ten Top Tips to help get things right". The IStructE guidelines give definitive technical management advice which the writer advocates. This research deals with engineering matters not covered by the IStructE guidelines, the target audience is engineers who develop and support software for the production of engineering calculations.

Verifying the correctness of structural engineering calculations considers calculations for both the structural analysis of frameworks and the structural design of components such as beams, slabs & columns, and develops a unified approach for the development of *Verified Models* for both types of calculation. In this thesis, **verifying** means establishing the truth or correctness of software models by examination or demonstration. Each model to be verified incorporates a self check, **verification** is the process of generating a thousand or more discrete sets of engineered data providing high coverage for the model, running the model with each set of data, computing the average percentage difference between key results produced by the model and its self check, averaging the key results for each run, averaging for all runs and when the average percentage difference for all runs is within an acceptable value, typically 3% for models for structural analysis, then the model is said to be a **verified model**. Tools used for assisting verification are discussed including: benchmarking, flow charts, check lists and aids, help, generating sets of test data, self checking software, checking against known solutions, conversion of parametric files to numeric files, cross referencing of variables.

Approximately 50% of calculations submitted to building control departments for approval are now produced by computer. Engineers say that due to the pressure of work in the design office, checking is not as thorough as they would like. From the starting position that the data has been checked, this research develops an extensive set of models which are self checking and have each been verified with sets of automatically generated data providing extensive *coverage* for each model. All systems are described in sufficient detail such that they may be used by others.

The systems developed for verifying the correctness of structural engineering calculations, based on:

- the inclusion of an automatic self-check in every structural model
- the development of a parameter specification table permitting
- the automatic generation of engineered sets of test data for each model
- the automatic running of the sets of test data for a thousand runs for each model
- the automatic reporting of the results giving a statistical summary are all **new to the field of structural engineering**.

Declaration

The content of this research is the work of Douglas William Brown and includes nothing which is the outcome of work done in collaboration or work copied from others, except where that work is quoted herein and given proper acknowledgement.

This work was carried out in the Engineering School of the University of Surrey. It has not been submitted previously, in part or in whole, to any University or Institution for any degree, diploma, or other qualification.

The full length of this thesis is 105,000 words including appendices.

Acknowledgements

The writer acknowledges the giants of structural engineering theory on whose work this work depends, particularly Castigliano, Timoshenko, Cross & Hrennikoff, the structural equivalents of Bach, Mozart, Beethoven & Holst; or if preferred Armstrong, Ellington, Brubeck & Davis.

The writer thanks his supervisors: Professor Gerard Parke & Dr Peter Disney for their abundant encouragement, their interest in the subject, for the provision of papers for review, for reminding the writer to include subjects which he would have omitted otherwise, and for steering this research towards a unified approach covering both the structural analysis of frameworks and the structural design of components.

The writer thanks Jennifer, Ian & James, his wife and sons for their help and support.

Contents

Chapter 1

Introduction

Verifying the correctness of structural engineering calculations is to help engineers who spend a significant proportion of their professional time in the preparation of calculations for the structural analysis of frameworks and for structural component design. Although the examples given herein are structural, they have been kept simple so that civil, mechanical, electrical, refrigeration, heating & ventilation engineers may follow them and thereby see if their discipline can make use of the same principles for verifying the correctness of their computer produced calculations.

Proforma calculations written in Praxis (1990), parametric models for structural analysis written in the NL-STRESS language, and tables are shown throughout this document in the Courier font which has a fixed spacing, enabling the text to be *lined up*.

In 1994, the tenth report of the Standing Committee on Structural Safety, (SCOSS, 1994), highlighted the need for guidance in the use of computers in the construction industry, this thesis provides such guidance by developing a system to ensure the correctness of structural engineering calculations produced by computer. In 1997, The American Society of Civil Engineers published an Interim report (ASCE, 1997) prepared by their Task Committee on Avoiding Failures caused by Computer Misuse, the self-checking procedures developed in this research have been designed to prevent such misuse. In 2002, The Institution of Structural Engineers, published Guidelines for the use of computers for engineering calculations (Harris *et al.* 2002), which commence "These guidelines have been prepared in response to growing concern regarding the appropriate use of computers for structural calculations" and end with "Ten Top Tips to help get things right". The IStructE guidelines give definitive technical management advice which the writer advocates. This research deals with engineering matters not covered by the IStructE guidelines, the target audience is engineers who develop and support engineering software.

Verifying the correctness of structural engineering calculations considers calculations for the structural analysis of frameworks and for the structural design of components such as beams, slabs, columns, walls & foundations, and develops a unified approach for the development of *Verified Models* for both types of calculation. Tools used for assisting verification are discussed including: benchmarking, flow charts, check lists and aids, help, sets of test data, self checking software and checking against known solutions.

Approximately 50% of calculations submitted to building control departments for approval are now produced by computer. Engineers say that due to the pressure of work in the design office, checking is not as thorough as they would like. From the starting position that the data has been checked, this research develops an extensive set of models which are self checking and have each been verified with sets of automatically generated data providing extensive *coverage* (Marick, 1995) for the model. Both types of calculation are parametrically written, the engineer need only change typically 10-20 parameters to obtain a set of self checked results; thereby avoiding the mistakes associated with starting with a blank sheet of paper. The systems for verification which have been developed in this research, are described in detail so that they may be used by others.

One key component of the verification process is the classification of structural engineering data and *engineering* that data into a table from which discrete sets of data are automatically generated and run to ensure that the model is tested over its design range.

A second key component of the verification process is *self-checking*. For the structural analysis of a framework self-checking is provided by an appropriate classical method for the model being tested, or by equilibrium, compatibility and energy checks developed as part of this research. For the structural design of components, it is recommended that the self-check be provided by:

- checking that a structural framework or component will safely carry the design loading
- checking against an alternative method *e.g.* classical elastic, Eurocode *etc.*
- providing an alternative model *e.g.* treating a beam as a structural analysis problem and comparing the stresses with the empirical results produced by the model which has been written in accordance with a code of practice *e.g.* BS 5950-1:2000, which is being checked.

1.1 History of structural design from 1900 to the present

In the first three decades of the twentieth century, structural engineering was the domain of Universities and steel manufacturers such as Dorman, Long & Co. (1924), and Redpath, Brown & Co. (1924). Both companies provided a wealth of structural engineering information in their handbooks, which were given freely to engineers and architects. During this period, concrete was reinforced with a variety of steel sections including angles, channels and rolled steel joists, the concrete being provided for fire protection and to give a flat surface, the steel sections being designed to carry the loading. Theory of Structures (Morley, 1912) and Elementary Applied Mechanics (Morley & Inchley, 1915) supplemented the structural information available from the steel manufacturers. Heyman (1983) provides *Some notes for a historical sketch of the development of the plastic theory 1936-48*. The Reinforced Concrete Designer's Handbook (Reynolds) first published in 1932, was a major step forward for the design of reinforced concrete, providing charts and tables and other design information for reinforced concrete design just as Dorman, Long & Co. and Redpath, Brown & Co. had provided for structural steel design a decade before.

BS 449 (BS 449, 1969) for the structural design of steelwork was first introduced in 1932 and CP 114 (CP 114, 1969) for the structural design of reinforced concrete was introduced under another name in 1934. Both these codes continued in use until well beyond the introduction of limit state design for reinforced concrete, codified as CP 110 (CP 110 1972). The ultimate load design of steelwork had been in use since the London blitz when steel shelters over beds became the first example of the plastic design of structural steelwork; plastic design was later popularised by the BCSA Black Books (BCSA, 1965-1975). Livesley (1983) discusses early uses of computers for carrying out structural analysis in *Some aspects of structural computing: 1943-1983*. Structural calculations for the four decades prior to the introduction of limit state design were characterised by simple design principles and formulae, but included many arithmetic mistakes due to the misuse of slide-rules. Structural calculations since the introduction of limit state design are characterised by increasing complexity and consequent reliance on computers. Since the introduction of BS 449 & CP 114, engineering calculations have always been brief and to the point.

Older structural engineers will remember their concerns when CP 3 Chapter 5 (CP 3, 1952) was revised nearly doubling wind pressures for Exposures A to D, apparently making all our previous designs *unsafe*; the considerable number of changes to BS 6399 (BS 6399, 1997) over recent years proves that there is still uncertainty concerning the magnitude of the forces we should be considering in our designs. Today, engineers assume that the results produced by computer will be arithmetically correct and that the complicated semi-empirical formulae given in the codes are being applied correctly from an engineering standpoint; perhaps engineers are too trusting on both counts. With more and more firms registering to ISO 9001:2000 for quality management systems and the advent of the Eurocodes, the subject of verification is of considerable interest.

1.2 Longhand and computer produced calculations

From the nineteen fifties to the seventies nearly all calculations were produced longhand, the moment distribution method devised by Professor Hardy Cross was undoubtedly the most widely used pre-computer method for the analysis of indeterminate structures. Known in the US as *Hardy Cross* and in the UK as *moment distribution*, the method was intuitive and easy to apply. In the early 1960's, structural design offices were referred to as *drawing offices*, a misnomer as twice as many engineers were employed in structural analysis and structural component design than in drawing. Although continuous beams were by far the biggest workload for engineers with the ambiguous title of *reinforced concrete engineers*, each year one or two statically indeterminate frames - with the complication of sway - would be tackled.

Prior to the advent of the IBM PC in 1981, calculations were generally produced without computer assistance, for the cost of so called mini-computers was of the order of 5 man-years' salary *cf.* today's 2 man-days. A further hindrance to the widespread use of computers for the production of structural engineering calculations before 1981, was that each computer manufacturer had their own operating system/s; thus programs designed to run on a DEC Vax, would not run on a Data General, Texas Instrument or a Prime mini-computer. This problem was compounded by the fact that manufacturers developed different operating systems for each computer they manufactured, thus the operating system for a Data General Nova mini-computer, was different to that for a Data General Eclipse mini-computer. Just as IBM brought order to hardware, so Microsoft has brought order to software; the result is that today's engineers rely on computers for the production of structural engineering calculations and when an engineer attends for interview with a potential new employer, invariably the method of production of calculations is discussed.

Before the advent of the IBM PC, engineers who had access to mini-computers used computers to produce design and checking aids, for example: Stanchion design charts (Brown, 1974), Reinforced concrete design charts (Brown, 1975), Autofab Handbook (Redpath Dorman Long, 1978) and Design tables for composite steel and concrete beams (Noble & Leech, 1978). Noble & Leech (1978) used software developed by the writer's firm; as their foreword states "the tables were computer set, using tapes developed directly from program output tapes". The advent of the IBM PC, meant that even sole practitioners could afford a computer, thus the use of mini-computers for the production of design and checking aids was replaced by packaged programs for structural analysis and design running on *IBM compatibles*. Over the last decade, Windows has replaced DOS as the standard operating system in the western world.

The last five decades have seen an immense increase in the speed of computers *e.g.* 1956 world record MIT TX-0 83 kFLOPS *i.e.* 83,000 floating point operations/second; 2006 world record Lawrence Livermore National Laboratory and IBM's Blue Gene/L 65,536 processor supercomputer can sustain 280.6 teraflops *i.e.* trillion floating point operations per second *i.e.* 280.6*10^12/83*10^3 =3.4 billion fold speed increase in 50 years. This increase in performance has been accompanied by over a hundredfold decrease in the size and cost of computers. During the same period there has been a sevenfold increase in the length and complexity of British Standards and other codes of practice such as Department of Transport memoranda, *e.g.* the Building Regulations (1972) have increased from 188 A5 pages to 6 cm thickness of A4 printed both sides with an additional 542 pages of explanation (Stephenson, 2000). The improvement in the cost-benefit of computers, combined with the reduction in the cost-benefit of more complex design procedures, has persuaded more and more firms to use computers for routine element design such as beam sizing, as well as for structural analysis. Today, approximately 50% of structural calculations are produced by computer, the remainder by longhand calculation.

1.3 Growing concern

SCOSS (1994) highlights the need for guidance in the use of computers in the construction industry. Prukl & Lopes (2001) show that the results of a finite element analysis of a simple beam can vary widely with as much as 100% error on the wrong side at the critical mid-section, depending on the choice of element, model and software package. Harris *et al.* (2002) commence "These guidelines have been prepared in response to growing concern regarding the appropriate use of computers for structural calculations *etc.*". From discussions with the writer, engineers agree that checking of calculations is important, but for various reasons especially the day to day pressure of the design office, checking is not as thorough as engineers would like.

1.4 Objectives

The paramount and first objective of this research is that computer produced calculations are correct. One erroneous set of output calculations is more than enough, one thousand erroneous sets will affect the credibility of the firms which are responsible. Errors arise from several sources *e.g.* incorrect data, bugs in the logic, inappropriate structural modelling (see 3.1.3 MacLeod, 2005), not understanding the design assumptions *etc.* Ensuring that the output calculations are sensible will involve further objectives being met.

Traditionally structural analysis & design were included in the same set of longhand calculations. When computers first became generally available, they were used for structural analysis. Livesley (1983) tells us that in this country Bennett (1952) carried out the first structural calculations to be done on a computer and that the matrix techniques used in that first program, based on Kron (1944) survive with very little change in many of the structural analysis programs in use today. The use of computers for structural analysis caused schism in the set of structural calculations, longhand calculations being bound separately to the computer produced calculations, which in the 1980's were usually fanfold. The set of data required for a structural analysis differs to that required for the production of a set of structural design calculations. Most items of data for a structural analysis program; whether integer *e.g.* for the number of joints, or real *e.g.* for applied loads, can vary uniformly over a wide range of values; furthermore within any set of data for a structural analysis, there is little dependence of any item of data on any other (material properties being one exception). The set of data for a structural design calculation, has a high dependency on the items of data among themselves; many items of data being code dependent, sometimes given in tables for which there is no sensible alternative to a table; sometimes requiring engineering judgment for: degree of quality control applied on site; load sharing factor; whether or not lateral restraints are provided at loading positions *etc.*

The second objective is to provide a unified method for dealing with calculations for the structural analysis of a framework and for structural component design, which will enable engineers to return to the traditional single set of calculations.

The third objective is to get to grips with the nature of the data *i.e.* classifying the different types of parameters used in structural modelling and the dependency between parameters. Some items of data are integer values *e.g.* a joint number, some items are real values *e.g.* coordinates, some items are dependent on other items *e.g.* Young's modulus and the modulus of rigidity are related by Poisson's ratio, some items belong to sets *e.g.* reinforcing bars of 9mm diameter are not manufactured neither are universal beams of 185 mm serial depth, some items can only vary within a fixed range *e.g.* the position of the start of a partial UDL on a beam cannot be negative and the position of the end must be greater than the start and not exceed the beam length, and so on. Thus we need a system, which in turn means we must classify the various types of data required and the dependency of any item of data on any others in the set of data, both for the analysis of structural frameworks and for calculations for structural components covered by the codes of practice: BS 5950, BS 5400, BS 8110, BS 8007, BS 8666, BS 5268, BS 5628, BS 6399, BS 8002, BSI STANDARDS CATALOGUE (2005).

The fourth objective is to ensure sustainability of software and systems devised as part of this research, which means that the self-checking and verification software must be easy to maintain, which in turn means that structural models should be written in plain text rather than computer code.

The fifth objective is identifying and applying tools to sets of *calculations* to increase the correctness of the *calculations*. Tools include: published worked examples, elastic design methods, engineering judgement, assessment of a structural component for its ability to carry its design loading, calibration against other codes of practice *etc.* As well as increasing the robustness of the output calculations, these tools may aid in verifying that a structural program is giving *sensible* results. The word *sensible* is the best that engineers can hope to achieve.

The sixth objective is to provide a simple system to satisfy engineers that the results of running any structural model are as expected, this means each model must include at least a self check. Accountants use the word *reconciling* to describe the check on their double entry book-keeping. The *self check* to be provided with every structural model *reconciles the calculations*. The complete collection of building standards is now comparable in length to a legal library, engineers (following Lord Denning's ruling) are classed as experts, and an error made by an expert constitutes negligence. From discussions with young engineers, many are feeling concerned about the level of responsibility they are taking for the complicated structures they are now designing. It is likely that in the event of structural failure, the engineer responsible for the design will be considered negligent if he/she has accepted the results of a structural analysis as being correct without checking those results. The incorporation of a self check within every model should satisfy engineers that the results of their calculations are in the right field for the data provided.

When the writer commenced work in the drawing office, now referred to as the design office, structural steel design, reinforced concrete design, roads, embankments and sewerage, were separated. Since the disbandment of the Property Services Agency, and the retirement of famous names from the civil and structural engineering profession, large consultancies have become larger, sometimes by absorbing smaller practices. Comparison of the Consultants File 1995 (NCE, 1995) with Consultants File 2005 (NCE, 2005), shows that, generally, large consultancies have doubled in size. Over the same period, the writer has noticed a considerable increase in sole practitioner consultancies, caused by takeovers, mergers and the privatisation by Local Authorities of their building control departments. Sole practitioners have a different view to that expressed by the large consultancies, but they are too busy with carrying out structural steel design, reinforced concrete design, roads, sewerage, attic room conversions, structural surveys *etc.* to spare the time to express their views. With this in mind, the Institution of Structural Engineers Informal Study group *Computing in Structural Engineering* (Seifert *et al.* 2000) identified a wishlist of desirable attributes, which the group called *Computer toolkit for small consultancies*. Thus the seventh objective is that the system developed should be capable of verifying the correctness of structural calculations which are included in the IStructE wishlist *viz.*

- Analysis: 2D frame, continuous beam, subframe, foundations
- Concrete: RC slab beam and columns to BS 8110, retaining walls
- Steel: beams and stanchions to BS 5950, composite construction, section properties
- Masonry: walls, pier, *etc.* to BS 5628, vertical load and wind
- Timber: floor joists, beams *etc.*
- General: geometric properties, loading data, material weights, construction weights,
- imposed loads, ability to customise/simplify calculation sheets.

The eighth objective is finding bugs in existing and any new software written as part of this research.

1.5 Outline

Chapter 2 considers the reasons for testing software, and advocates *self-checking* software; chapter 2 reviews the classical structural analysis methods and how they can be used to form bedrock beneath the modern methods of structural analysis.

Chapter 3 describes tools and techniques for verifying engineering calculations. Verification includes: checking input data with red & yellow pencils and filing the check of the input data with the check prints of the drawings; paying special attention to supports and the members framing into them; extracting and checking sub structures; identifying key point positions and checking the results at them; engineering assessment of neutral axis depths; comparison with the nearest match in a library of parametric data files; binding post processing calculations into data files *etc.* Chapter 3 presents an armoury which can be brought to bear on bugs.

Chapter 4 discusses the nature of data for a set of *calculations* and classifies the types of data, taking due consideration of the dependency among items of data in any set required for the production of structural engineering calculations, so that sets of data may be generated automatically and used for testing structural models. The likelihood of a set of random numbers providing sensible data for testing a structural model such as the rafter of a portal frame, is remote; chapter 4 explains why and develops a general system which may be applied to *engineer* sets of data to fit each *calculation* in an ensemble comprising several hundred *proforma calculations*.

Chapter 5 introduces Praxis, which is English with embedded logic, which is an algebra for the mundane and is used in this thesis for the parametric modelling of structural components, the models being referred to as *proforma calculations*. Although the algebra is simple, even quite trivial examples with less than ten programming structures, can have thousands of different paths through their logic. Any tools to increase the robustness of proforma calculations, will not guarantee that every single path through the *calculation* has been tested, nevertheless using all the tools described in this research will substantially increase the robustness of a set of *proforma calculations*. Chapter 5 advocates using a proforma calculation to check other proforma calculations.

As Womak & Jones (1996) and Hawken *et al.* (1999) tell us, sustainability is not just about saving fossil fuels, it is about the avoidance of all waste, especially human effort. Hawken *et al.* (1999) use a southeastern Brazilian city called Curitiba as a good example of "weaving the web of solutions which has been done not by instituting a few economic megaprojects but by implementing hundreds of multipurpose, cheap, fast, simple, homegrown, people-centred initiatives harnessing market mechanisms, common sense and local skills." Many engineering megaprojects have failed, few can be unaware of substantial software developments which have been abandoned. Chapter 6 considers both the sustainability of engineering megaprojects and homegrown engineering systems, and discusses systems which are needed to ensure that major software projects can be maintained and do not become abandoned.

Chapter 7 introduces NL-STRESS which is a language for describing a model for the structural analysis of a framework. The NL-STRESS language has been extended as part of this research to permit logic to be incorporated between the SOLVE and FINISH commands so that each model may include a self check. Chapter 7 develops the subject of *verified models for structural analysis*; verified models act as *checking models* as described by Harris *et al.* (2002), but include self checking, and additionally have been verified for correctness using automatically generated sets of engineered data providing extensive *coverage* (Marick, 1995). When classical analysis methods are available, they are incorporated into the models, when classical structural analysis methods are not available, then the self checking of the models must be provided by other methods, these are developed in chapter 8.

Chapter 9 gives the reasons for *benchmarking* and develops a system for automating and reporting on the results of hundreds of benchmark tests, including the provision of a benchmark audit trail. The systems developed in chapter 9 are for *benchmarking* any structural analysis software. In the 1980's the then Department of Transport (DOT), Highways Engineering Computer Branch, kept an index of approved programs for use for the structural analysis of bridges and other structures. The DOT provided ten structural analysis problems to commercial firms which had developed structural analysis programs, the firms returned values of: bending & torsional moments, shear forces, deflections and rotations at certain key locations. The DOT checked these values against their own values and if satisfied, issued an approval letter and reference. The service was suspended during the cut backs in the late 1980's and a replacement service has not been offered by any organisation. The key to such a service is the development of a library of *benchmarks* (test problems). No service can guarantee that a commercial program is 100% OK but it can guarantee that a commercial program has satisfied a library of benchmarks. The benchmarks presented were obtained from a trawl through all the published elastic, elastic-plastic, stability, static & dynamic solutions. Chapter 9 lists the set of benchmarks developed for the above types of structure.

Elastic methods of structural analysis are as popular today as they were fifty years ago. Surprisingly, elastic analysis is still permitted to be used with limit state section design to BS 5950, BS 8110 *etc.* as listed in chapter 14. In the 1950's and early 1960's elastic section design was taught and practised. Elastic section design has two advantages: it is simple and intuitive. It is not proposed that elastic section design be reintroduced as design for the ultimate limit state is now the accepted norm, but it is proposed that elastic design be used to design the sets of test data to be used for increasing the robustness of *proforma calculations* for the structural design of components. Chapters 7 & 10 incorporate checking methods into the set of structural models presented.

Over the last decade, in response to the increase in the size of commercial vehicles, bridge engineers have been involved increasingly in carrying out assessment work to find out if their bridges can carry the increased loads. Bridge engineers now separate their work into *design* and *assessment*. Assessment is *reverse engineering* in the sense of working back from the strength of the as-built structure to find the loading which may be supported safely. Assessment work for bridge engineers is at least as complicated as design work; each assessment must take into account: corrosion of steel, effect of de-icing salts on the strength of the concrete *etc.* Assessment work for new-build is more straightforward than that for as-built, for the permissible strength of all the materials will be known. Chapter 10 develops *assessment* for use in increasing the robustness of *proforma calculations*, thereby giving the engineer confidence in the output calculations. Chapter 10 also develops the subject of *self checks by alternative* *methods*. Modern computers are so powerful that adding a few lines to the end of each output calculation in accordance with a British Standard *e.g.*

```
Selected size to BS 5950:   254 x 102 x 25 UB
cf. Linear elastic:         254 x 102 x 28 UB
```

would be acceptable to the majority of engineers and positively welcomed by a significant proportion, for the extra information to help with their final choice of section size. In a real job, the final choice of just one section size can often affect the entire structural design thus Chapter 10 considers the subject of comparison of structural sizes with those sizes produced in accordance with alternative codes of practice. This subject was formerly known as *calibration*, but more recently has been referred to as *validation*. Comparison of output calculations against fully worked examples found in manuals and books from organisations such as: The British Constructional Steelwork Association, Steel Construction Institute, British Cement Association, Timber Research And Development Association, Brickwork Development Association & expert authors, is a good way to check for correctness of models. Although published fully worked examples are uncommon, it is usually possible to find one or two good examples for each frequently used type of *calculation*.

Chapter 11 gives an abbreviated discussion on the structural behaviour and results of testing each model, conclusions drawn from the testing are contained in chapter 12. Recommendations aimed at engineer-programmers who wish to verify the correctness of their own calculations are contained in chapter 13.

1.6 Overview

In 1981/82, the writer wrote SuperSTRESS as a joint venture with the C&CA (Cement & Concrete Association) for the linear elastic analysis of structures. Following reorganisation at the C&CA, which resulted in SuperSTRESS being marketed by Integer, the writer became interested in non-linear analysis and in conjunction with Professor Michael Horne developed the program known as NL-STRESS for: non-linear elastic, stability, elastic-plastic & non-linear elastic-plastic analysis of structural frameworks. Following attempts to produce structural calculations using early spread-sheet programs, in 1985/86 the writer developed a system called SCALE (Structural CALculations Ensemble) to show full structural calculations which could be checked in the traditional manner using red and yellow pencils. Both NL-STRESS & SCALE have been supported for the past twenty years and are mature and robust. For this reason NL-STRESS & SCALE have been chosen for use in the development of a system for verifying the correctness of calculations for the structural analysis of frameworks and the structural design of components.

Figure 1.1 gives a diagrammatic overview on the unified system for verification which combines two types of calculation used by structural engineers viz:

- the structural analysis of frameworks *e.g.* a multi storey frame
- the structural design of a components *e.g.* a reinforced concrete beam.

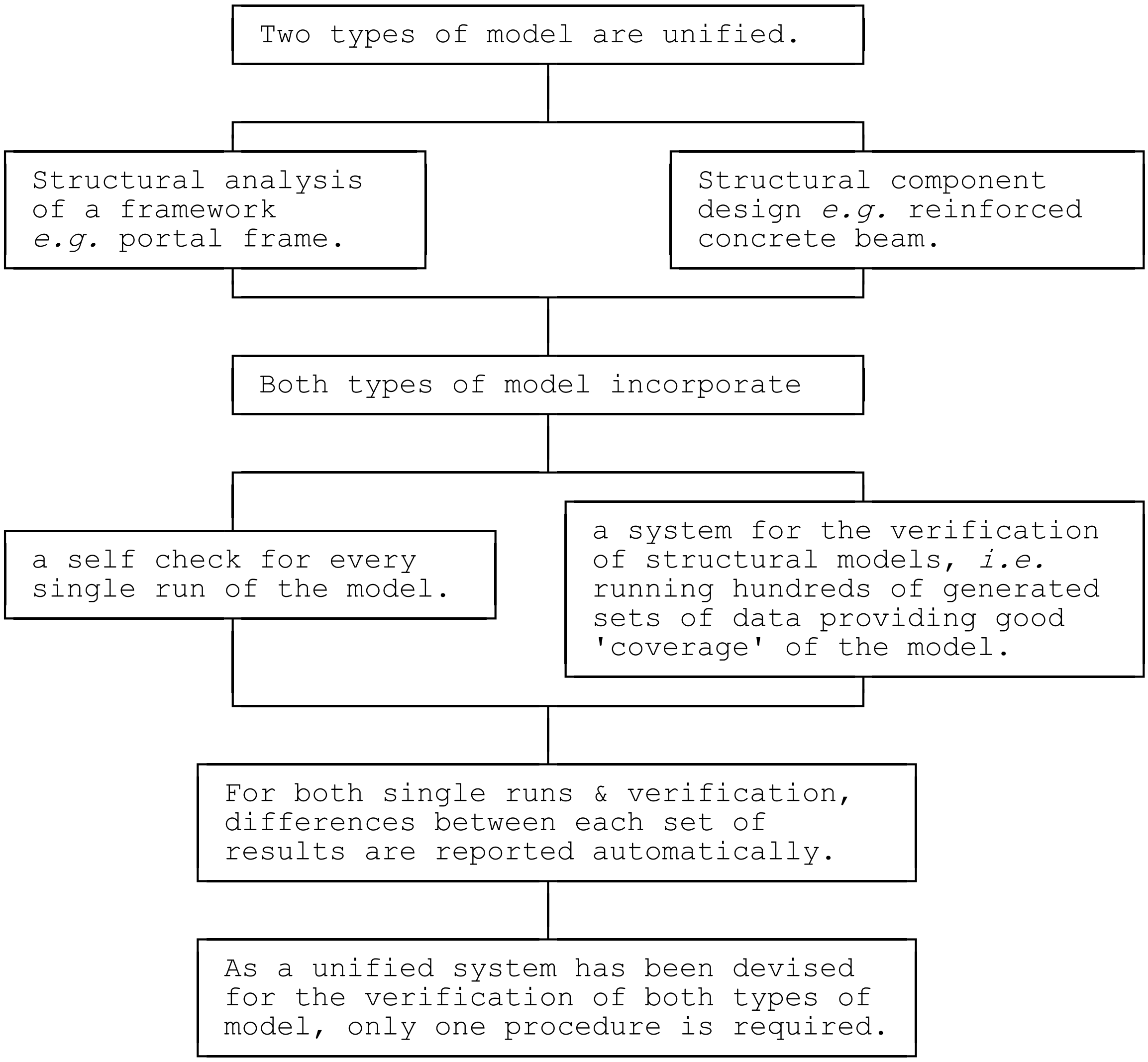


Figure 1.1 A Unified System.

Chapter 2

Literature review

A comprehensive library search has not revealed literature on the subject of verification of engineering calculations, consequently the search was widened to include testing and self checking software. (Ronald H Untch of STORM states that in searching for software testing literature, dissertations and theses are notoriously difficult to obtain, and that many libraries refuse Interlibrary Loan requests for dissertations and theses because of the expense. He invites anyone aware of an online version of a dissertation or thesis, whether an abstract or a complete text to contact him at storm@mtsu.edu.)

2.1 Testing software

Software is tested to detect bugs. Between June 1985 and January 1987, a computer-controlled radiation therapy machine, called Therac-25, massively overdosed six people. LEVESON (1995) lists causal factors as:

- overconfidence in the system
- confusing reliability with safety
- lack of defensive design
- failure to eliminate root causes
- complacency
- unrealistic risk assessments
- inadequate investigation or follow up on accident reports
- inadequate software engineering practices.

On 13 June 1994, Professor Thomas Nicely, whilst pursuing a research project in computational number theory noticed an incorrect count of primes < 2*10^13. This started a long process to pinpoint and eliminate the bug. On 4 October 1994 a new error was noticed. On October 30,1994, Dr Nicely, sent an email to several people regarding a bug in the Pentium divider. For example, he wrote, 1 divided by the prime number 824,663,702,441 (a twin-prime pair with 824,633,702,443) is calculated incorrectly (all digits beyond the eighth significant digit are in error). Dr Nicely provided several other values for which the Pentium produces an incorrect reciprocal, noting that the bug can be observed "by calculating 1/(1/x) for the above values of x. The Pentium FPU will fail to return the original x. Coe (1995) gives the full 36 page story entitled "Inside the Pentium FDIV Bug". The writer recalls Dell phoning in the summer of 1995 and then sending an engineer from Intel to change the Pentium processor in the writer's computer. On asking what would happen to the faulty processor, the engineer replied that the Pentiums with the bug would be supplied to the games industry. Dr Nicely's solution was to reverse-engineer the answer and compare it with the data. Although engineering calculations deal with discontinuities (*e.g.* joists, bars, bolts... are only available in certain sizes), such discontinuities arise as part of the design process and do not prevent reverse-engineering a calculation, indeed verification of a calculation should be easier than the production of the original calculation.

On 4 June 1996, the maiden flight of the Ariane 5 launcher ended in failure. Only about 40 seconds after initiation of the flight sequence, at an altitude of about 3700 m, the launcher veered off its flight path, broke up and exploded. The inquiry board for ARIANE 5 (1996), gives the cause of failure as "complete loss of guidance and attitude information due to specification and design errors in the software of the inertial reference system".

The Mars Climate Orbiter was launched on December 11, 1998, and was lost sometime following the spacecraft's entry into Mars occultation during the Mars Orbit Insertion manoeuvre, the spacecraft's carrier signal was last seen at 9:04:52 on September 23, 1999. The Mars Climate Orbiter Mishap Investigation Board (MCO,MIB 1999) gave the root cause for the loss of the spacecraft as failure to use metric units in the coding of the ground software file *Small Forces* used in trajectory models. Specifically, thruster performance data in English units instead of metric units was used in the software application code entitled SM_FORCES. A file called Angular Momentum Desaturation (AMD) contained the output data from the SM_FORCES software. The data in the AMD file was required to be in metric units per existing software interface documentation, and the trajectory modellers assumed the data was provided in metric units per the requirements.

Circa 1960-1970, structural engineering calculations and drawings were checked using red and yellow pencils. The checking engineer put a yellow pencil through every word and value with which the engineer agreed, and a red pencil and red correction for any word or value with which the checking engineer did not agree. Whereas in 1960-1970, each set of structural engineering calculations and drawings was likely to be used on one project; in 2006, software used in the production of structural engineering calculations may be used on thousands of projects, thus checking the software is essential. A wealth of information is available including:

- Standard for Software Component Testing, (BCS SIGIST), 2001.
- Best Practices Guide to Automated Testing, (AutoTester), 2002.
- BS 7925-1:1998, Software testing - Part 1: Vocabulary.
- BS 7925-2:1998, Software testing - Part 2: Software component testing.

- BS ISO/IEC 12207:1995 Information technology - Software life cycle processes.
- Computer Bulletin - March 2002.
- What is the SIGIST? Special Interest Group In Software Testing of BCS.
- IEEE Standard for Software Verification and Validation.

The Proceedings of the 17th International Conference, Edinburgh, July 2005 on **Computer Aided Verification** contain 54 papers. All papers are written by experts in informatics, *tools* deal with running a program and checking that the results are as expected, none deal with the verification of engineering software. Papers are grouped under the following headings:

- Invited Talks
- Tools Competition
- Abstraction and Refinement
- Bounded Model Checking
- Tool Papers I
- Verification of Hardware, Microcode, and Synchronous Systems
- Games and Probabilistic Verification
- Tool Papers II
- Decisions Procedures and Applications
- Automata and transition Systems
- Tool Papers III
- Program Analysis and Verification I
- Program Analysis and Verification II
- Applications of Learning.

Although one of the papers was entitled "Concrete Model Checking with Abstract Matching and refinement", the word *Concrete* was used metaphorically. None of the extensive searches on *self-checking* and *verification* revealed any papers concerned with verifying the results for the structural analysis of frameworks or the design of structural components. It follows that the verification of structural engineering software lags behind the verification of commercial software, and qualifies the need for this research.

In Britain, engineering is the poor relation of business; when engineers phone for help to move engineering software from their old computer onto their new computer, in response to the question "what version of Windows is on the new computer?", they frequently answer Windows 95, 98, ME or 2000; in response to the question why not XP?, they respond "well it's not really a new computer, the secretary has that so it's her old computer". Although the Government give seven figure contracts to consultants, with the exception of airframe manufacturers, few engaged in the production of structural engineering software are able to commission consultants who are specialised in testing. Furthermore, software for the analysis and design of all sorts of structures in all sorts of materials in all sorts of environmental conditions is complex; testing such software is best carried out by engineers.

Marick (1995) Testing Foundation, consulting in software testing, (marick@testing.com) extensively covers the subject of testing commercial software and lists **Some Classic Testing Mistakes**:

The role of testing:

- thinking the testing team is responsible for assuring quality
- thinking that the purpose of testing is to find bugs
- not finding the important bugs
- not reporting usability problems
- no focus on an estimate of quality
- reporting bug data without putting it into context
- starting testing too late.

Planning the complete testing effort:

- a testing effort biased towards functional testing
- under emphasising configuration testing
- putting stress and load testing off until the last minute
- not testing the documentation
- not testing the installation procedures
- an over reliance on beta testing
- finishing one testing task before moving onto the next
- failing to correctly identify risk areas
- sticking stubbornly to the test plan.

Personnel issues:

- using testing as a transitional job for new programmers
- recruiting testers from the ranks of failed programmers
- neither seeking candidates from the customer service staff nor technical writing staff
- insisting that testers be able to program
- a testing team that lacks diversity
- a physical separation between developers and testers
- believing that programmers can't test their own code
- programmers are neither trained nor motivated to test.

The tester at work:

- paying more attention to running tests than designing them
- unreviewed test designs
- being too specific about test inputs and procedures
- not noticing nor exploring irrelevant oddities

- checking that the product does what it's supposed to do, but not that it doesn't do what it isn't supposed to do
- test suites that are understandable only by their owners
- testing only through the user-visible interface
- poor bug reporting
- adding only regression tests when bugs are found (regression tests are those previously carried out prior to release of a previous version of the software)
- failing to take notes for the next testing effort.

Marick (1995) also lists mistakes under the headings *test automation & code coverage* where code is used in the sense of computer code.

Cohen *et al.* (1997), found that "most field faults were caused by either incorrect single values or by an interaction of pairs of values". It follows that we can get excellent coverage by choosing tests such that
- each state of each variable is tested
- each variable in each of its states is tested in a pair with every other variable in each of its states.

Section 5.8 in this thesis develops a matrix of patterns which ensure that every parameter is tested over its range from a start value to end value in combination with every other parameter varying from its start value to end value and also from its end value to its start value. In other words, successive test runs ensure that small values of each parameter are tested with both small and large values of every other parameter and any sensible number of intervals between.

Dickens (2004), a software tester at Microsoft, states that "it's clearly not the best use of my time to click on the same button looking for the same dialog box every single day. Part of smart testing is delegating those kinds of tasks away so that I can spend my time on harder problems. And computers are a great place to delegate repetitive work. That's really what automated testing is about. I try to get computers to do the job for me." Dickens discusses *Automated Testing Basics, Driving your program, Results Verification*. The writer echoes Dickens' "I try to get computers to do the job for me", most of this thesis is concerned with that aim.

Micahel (2004), another software tester at Microsoft, states that "One of the banes of my existence is test matrices. Everywhere I look it seems another matrix is lurking, impatiently waiting to lob another set of mind-numbingly boring test cases at me..."
Micahel says that to well and truly test the matrix of Microsoft products, you have to test every combination of: 9 operating systems with 10 browsers with 4 .Net frameworks with 12 versions of Office and at least 3 versions of your own application *i.e.* 12,960 different configurations on which you need to run all your tests.

Bolton (2004), states that "We can save time and effort and find bugs efficiently by testing variables and values in combination." In formal mathematics, the study of combinations is called *combinatorics*. Defects usually involve a single condition, but in some cases there are two or more bad dancers at a party; normally everything is all right but if one bad dancer encounters the other while in a certain state, they trip over each other's feet.

Global testing solutions - Testline (mail@testline.co.uk) are a commercial firm specialising in the testing of commercial software. They give the following reasons for testing software. "A software program is a complex item. This increases the chance for errors to occur, especially as some problems are not immediately visible but only become apparent under certain, sometimes rare conditions, thus there is wide scope for errors to be made. It is therefore advisable to test software, not only on completion but also during development to minimise the chance of problems occurring. Dealing with smaller more manageable sections of a program during development makes it easier to isolate and solve problems before they are incorporated in a program where they could be harder to find and cause many more problems. To allow problems to make it to the marketplace can have damaging, time consuming and expensive consequences. These often work out to be more costly than running a proper testing procedure during program development."

Dustin *et al.* (2005) deal extensively with the subject of testing commercial software, covering: automation, tools, strategy, risk-based testing, reporting and tracking defects, testability, working with developers, notorious bugs.

2.2 Knowledge based expert systems

Scott & Anumba (1999) define "Knowledge-based systems (KBSs) as interactive computer programs incorporating judgment, experience, rules-of-thumb, intuition and other expertise, to provide knowledgeable advice about a variety of tasks, and principally consist of a knowledge base, a context, an inference mechanism, and a user interface". Scott & Anumba (1999) describe their development of a knowledge-based system for the engineering management of subsidence cases. An advisory system is presented and shown to provide intelligent advice to engineers at all stages of the subsidence management process, from initial diagnostic and prognostic assessment, to the specification of successful remedial measures.

Knowledge based systems are discussed by Maher (1987) in "Expert systems for civil engineers" and by Wagner (1984) in "Expert systems and the construction industry". Mahmood (1989) describes "An expert system for checking the design of reinforced concrete elements to BS 8110". Gashnig *et al.* (1983) evaluate expert systems discussing issues and case studies. Allwood & Cooper (1990) use an expert system to select paint schemes to protect structural steelwork. Hanna *et al.* (1993) describes "A knowledge based advisory system for flexible pavement routine maintenance". Toll (1996) describes "A knowledge based system for interpreting geotechnical data for foundation design". Tizani (1990) describes a knowledge based system for the diagnosis of cracking in buildings. O'leary *et al.* (1990) describe tools and techniques for validating expert systems.

Gardner (1999) following a survey entitled *IT in engineering consultancies* states: "Expert systems are a specialist tool which some commentators predict will make a big impact in the future. The survey found a low use of expert systems, and there was evidence to suggest that the term was not widely understood".

Rafiq *et al.* (2000) tell us: "In the past, knowledge-based expert systems (KBESs) tried to model some of the activities of conceptual design. Owing to their restricted scope, the success of these systems was very limited. Artificial neural networks (ANNs) are applications of artificial intelligence (AI) which can imitate the activities of the human brain. Like human experts, ANNs are capable of learning and generalising from examples and experience, to produce meaningful solutions to problems, even when input data contain errors or are incomplete. This makes ANNs a powerful tool for modelling some of the activities of the conceptual stage of the design process."

2.3 Artificial neural networks

Chuang *et al.* (1997) have modelled the capacity of pinended slender reinforced concrete columns using neural networks. Jenkins (1997) in an introduction to artificial neural computing for the structural engineer, gives an example of a neural network having two input neurons, three neurons in a hidden layer, and two output neurons. Jenkins tells us: "At a particular instant in time, inputs to the neuron are summed and the sum is passed to a transfer or *thresholding* function which produces an output. The output signal is then processed either by another layer of neurons or as a final output signal. Connections between the neurons are individually weighted, so the input to a single neuron depends on the weightings and the values passed from previous neurons."

The transfer function, as with *fuzzy logic* (Mamdani & Gaines, 1981) provides decisions based on approximate information, thus an artificial neural network has applications for monitoring, such as:

- loss of strength due to crack propagation in a multi storey reinforced concrete frame
- leakages in fresh water systems beneath a metropolis
- bacterial activity in hospitals.

Jenkins (2001) identifies a number of desirable developments provided by a neural network based reanalysis method for integrating with structural design. Jenkins (2004) tells us that: "Just over a decade ago, the genetic algorithm (GA) arrived on the scene. It has gained in popularity but not, to any significant extent, in structural engineering. The reasons are clear. It appears quite foreign to what we know as practical structural design; furthermore it requires a considerable amount of unusual mathematics with uncertain prospects of useful practical application. Things have changed, we now call it *evolution* and it looks different. What has happened is that the genetic algorithm has *evolved* into a more practical, engineered orientated, style and it is worth having another look at it." It is noted that spawning neurons in the hidden layer as the network *learns*, appears to be at variance with the principles of *structured programming* as propounded by Dijkstra *et al.* (1972).

Engineers who are familiar with critical path programmes will be able to visualise a large network of activities with earliest and latest start and end dates for each activity. Such a programme has similarity with a flow chart for a computer program or a neural network. Critical path programmes usually have several activities going on at the same time. With the exception of *parallel processing computers*, computer programs have just one flow line going through the imaginary network. At each neuron, a decision has to be made on which direction the *flow* will go. In a computer program, the direction is dependent on numbers, and that means dependent on the data. For a given computer program, a different set of data will generate a different path through the network. For even a modest computer program, there will be many millions of different paths through the network. Software is available for converting a computer program into a flow chart, but staring at a flow chart with a million neurons for 1000 hours will not help in finding bugs; it is just not possible to spot an incorrect connection. The only way of testing the *flow switching at each neuron* is to provide data, one practical experiment is worth a thousand expert opinions. This research uses data to investigate the correctness/incorrectness of structural engineering models. It is expensive to employ engineers to devise sets of data, so this research develops a system to produce sets of *engineered data* automatically.

2.4 Checking models

MacLeod (1995) and Harris *et al.* (2002) advocate *checking models* which are based on a simplified version of the conceptual/computational model, for results verification purposes. Harris *et al.* (2002) warn: "People tend to take an optimistic view and when they find results are close, they are quick to accept this as proof of accuracy. A single apparently favourable correlation does not provide a full verification. It is necessary to treat all results with suspicion and not to jump to conclusions."

Checking models, which are trusted, would highlight bad choices of finite element in the global modelling software. The warning by Harris *et al.* (2002) that *a single apparently favourable correlation does not provide a full verification*, is good advice but does not imply that many correlations between the conceptual model and the checking model provides verification. If the software used for both the conceptual model and the checking model has the same flaw, then both may agree precisely and be wrong. For example: if both ignore shear deformation then for members which are continuous at one end and pinned at the other, having a span:depth ratio less than 10:1, then span moments can be under-estimated by 30% or more.

The question arises *who checks the checking model?*, the answer must be that the checking model incorporates its own check, in other words be *self checking*. The checking model for the structural analysis of a continuous beam is most likely to be based on the stiffness matrix method, so a good self check would be to use a classical analysis method *e.g.* moment distribution. *Self checking* for structural analysis models which use the stiffness method method, could automatically compare the results of each and every run using a classical structural method, tabulating percentage differences for the engineer's consideration. Before any first release of a checking model for the structural design of a framework, several hundred runs should be carried out comparing each run with an appropriate classical method so that an assurance can be given for the checking model and thus avoid problems such as those highlighted by Harris *et al.* (2002). The set of verified models in appendix A, developed as part of this research, meets this requirement.

2.5 Self checking software

Dr Nicely's self check of the correctness of the division process by calculating the value 1/(1/x) and checking that it equalled x (section 2.1), showed that great corporations such as Intel, are not infallible; hitherto software authors had assumed that hardware was always arithmetically correct. The incident gave impetus to the subject of *self checking*, and the use of computers to tackle abstract problems, one of which is known as the *four colour problem* for the colouring of maps. It is important because researchers used computers to verify that the *four colour theorem* was true; the work described in this thesis uses computers to verify that engineering calculations produced by computer are true or not true.

On 19 April 2005, NewScientist.com electronic headline reads *Computer generates verifiable mathematics proof*. A computer-assisted proof of a 150-year-old mathematical conjecture can at last be checked by human mathematicians. The Four Colour Theorem, proposed by Francis Guthrie in 1852, states that: "any four colours are the minimum needed to fill in a flat map without any two regions of the same colour touching". A proof of the theorem was announced by two US mathematicians, Kenneth Appel and Wolfgang Haken, in 1976. But a crucial portion of their work involved checking many thousands of maps - a task that can only feasibly be done using a computer. So a long-standing concern has been that some hidden flaw in the computer code they used might undermine the overall logic of the proof.

Last doubts removed about the proof of the Four Colour Theorem, Devlin's Angle MAA Online (The Mathematical Association of America). Devlin (2005) tells us "Gonthiers employed a proof assistant, called Coq, developed at the French research centre INRIA, where Gonthiers was formerly employed, and where much of his work on the Four Colour Theorem was done. A mathematical assistant is a new kind of computer program that a human mathematician uses in an interactive fashion, with the human providing ideas and proof steps and the computer carrying out the computation and verification."

Of course such a mathematical assistant will know neither about torsional buckling nor about the effect that different types of aggregates have on the creep coefficient of concrete, but paraphrasing the above: "An engineering assistant is a new kind of computer program that an engineer uses in an interactive fashion, with the engineer providing ideas and proof steps and the computer carrying out the calculations and verification." From the writer's experience, engineers appreciate interactive software but they would expect others to provide the ideas and proof steps, for they have many other things to worry about in addition to calculations, thus the above definition of an engineering assistant should be shortened to: "An engineering assistant is a new kind of computer program that an engineer uses in an interactive fashion, with the computer carrying out the calculations and verifying the output". Such a description loosely fits several UK systems, though none presently carry out verification; verification is the purpose of this research.

Were an engineer to be asked to verify the four colour theorem, the engineer would reach for a pencil and paper in preference to a *proof assistant*. The writer tackled the problem by looking for unusual situations. Rather than check the overall logic of the proof, a starting point for an engineer would be to look for an abnormal situation. Starting with the knowledge that South Africa encompasses Lesotho, encompassed countries seem a worthwhile limit state to consider as a possible exception to the four colour theorem. In figure 2.1 below, (A) represents a large country which encompasses a group of four small countries denoted B, C, D & E. Assuming four colours for countries A, B, C & D, then if E is replaced by either B, C or D it gives two adjacent countries having the same colour, thus E can only be replaced by A. If this is done it implies that the territory in the centre is part of the territory of country A; if this is not the case then the territory in the centre should have a fifth colour and Guthrie's theorem is disproved.

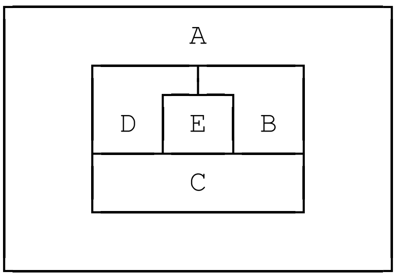


Figure 2.1 Four colour theorem.

Just as problem solving is at the heart of engineering **and the reason why applied mathematics should be at the heart of engineering training**, the principle of looking for the exception, the unobvious, looking for snags or bugs, seeking the one that has gone astray, should be at the heart of software testing, just as it is at the heart of engineering and caring. *Caring* is used in the sense of caring about the quality of workmanship, *i.e.* not accepting that *any calculations output by a computer will do*. MacLeod (1995) tells us "In the past, the main problem in structural analysis was the difficulty in achieving solutions. This is now the easiest part of the process. Now the difficult part of the process is in creating the model and in **checking that the results are correct**".

Limit state design uses statistics to derive partial safety factors for materials and loadings. It can be argued that lowering the dead load safety factor from 1.4 to 1.35 (used in Eurocode 2) has no stastical basis as the increase in errors due to the increase in complexity of Eurocode 2 is not taken into account as a partial safety factor. The principal of putting the onus on engineers to consider all possible limit states, remains valid; for map makers, encompassed countries are a state that must be considered. When a model needs to be verified for a thousand or more different conditions, whether it be for engineering or map making, then using computers to save human effort is desirable. It follows than **every single run of every model should include a self check with a highlighted warning when variance is found**. Each of the 108 models included in appendix A, which were developed as part of this research, include a self check either by the use of classical structural theory or by compatibility, energy and equilibrium considerations, described in chapter 8.

2.6 Classical structural theory

Livesley (1968) writes "in the 1950's it became possible for the first time to present these (matrix) principles in a manner independent of the traditional classification of structures according to their engineering form, so that beams, frames, trusses and arches could all be dealt with by a unified approach.".

On the subject of structural theory in use in the decades following WWII, Harrison (1973) in the University of Sydney, writes "Difficulties in grasping the concepts of structural theory have been amplified by the bewildering array of manual methods - each with its specific application. For example, moment distribution is applicable to rigid frames when axial and shear strains are insignificant. Energy and virtual-work methods are acceptable for portal frames and arches and for computing deformations in trusses. The distinction between determinate and indeterminate structures is of prime importance in the latter methods but not so significant in the slope-deflection technique, which, nevertheless, is of limited application when the order of equations to be solved becomes large. Static analysis of trusses by tension coefficients and of continuous beams by the three-moment equation appear to be isolated manual methods with seemingly little in common. Odd schemes for calculation, such as conjugate beam or column analogy, have always been difficult to classify... Many of these methods are by no means rendered obsolete by the computer techniques, because a large part of the preliminary design of quite complex structures will always be based on desk calculations pertaining to small assemblages of beam and column elements."

The above brief summary of classical structural theory, has two phrases which are as relevant today as they were when written over thirty years ago: "when axial and shear strains are insignificant" and "preliminary design... will always be based on desk calculations".

Each of these phrases poses a question. How is the engineer to know when axial and shear strains are insignificant, especially when detailed calculations have been replaced by modelling? Which of the classical methods should engineers use for their desk calculations? Such questions are part of *validation*, (IStructE 2002). The answer to "How is the engineer to know when axial and shear strains are insignificant, especially when detailed calculations have been replaced by modelling?" is that the same data for an analysis by the matrix stiffness method should be used for an analysis by a classical method *e.g.* moment distribution, which ignores axial and shear strains. Obviously today's engineer is unlikely to carry out moment distribution for say a four span continuous beam with pattern loadings to BS 8110 using *longhand calculations*, he/she will use a computer program. It follows that it will save time if both the matrix stiffness method & classical method are contained within the same analysis; as part of this research, verified models have been developed to combine the matrix stiffness method with classical methods.

The answer to "Which of the classical methods should engineers use for their desk calculations?" depends on which of the classical methods were taught to the engineer, if by *desk calculations* (Harrison, 1973) meant *longhand calculations*. Assuming Harrison meant *longhand calculations*, then today he would be referring to *scheme design stage* or *Initial design* as described in the *Manual for the design of reinforced concrete building structures* (Lee *et al.* 1985) and for this design stage, classical design methods are seldom applied, attention being given to: loading, material properties, structural form & framing, fire resistance and durability *etc.* with areas of reinforcement determined from charts, such as those given in appendix D to the manual, both the span and support bending moments being computed from W.L^2/10 *i.e.* engineering judgment. It is this *initial design* that is normally carried out by senior engineers, often at home where they have less interruptions and can *think on, work out all the angles, get a feel for the problem*, in other words bring their experience to bear on the matter. Such engineering judgment, colloquially *fag packet calcs*, sometimes appear in the design office. By the production of such calculations, senior engineers verify the computer calculations produced by junior engineers in the design office; and when senior engineers find mistakes made in the design office, they enjoy the situation, the stuff of life, and casually remark "I'm sure you would have checked the support steel but it looks extremely economical to me"; behaving in much the same way as consultants in the medical profession behave towards junior doctors.

Today there is a powerful computer on every structural engineer's desk, so today *desk calculations* would likely mean the engineer typing in a dozen numbers to a continuous beam model and *rationalising* the results. By *rationalising* is meant making practical decisions such as: spans 2 & 3 will have the same support & span reinforcement & link hanger bars; spans 1 & 4 should have support steel continued throughout their spans to control deflection.

Harris *et al.* (2002) attribute issues of inappropriate models and computer assisted error being due to the change from detailed calculations towards *modelling*. It follows that there is a need to return to the production of detailed calculations, as still produced by the thousands of structural engineers in sole practice; a need to extract structural sub-frames and components from the overall model and compare the forces and displacements in the sub-frames and components with the equivalent forces and displacements in the overall model. If engineers do not use the modelling process properly they cannot minimise the risk in doing structural analysis, MacLeod (2006).

2.7 Moment distribution

The writer remembers his first year of employment with the contractor Trollope & Colls in 1962, as mainly carrying out moment distribution for reinforced concrete continuous beams and slabs having up to six spans; those with more spans were tackled by splitting them into two shorter spans with at least a single span overlap to model structural continuity. Although the analysis of continuous beams and slabs was by far the biggest workload for reinforced concrete designers, each year two or three statically indeterminate frames - with the complication of sway - were also tackled using moment distribution.

Bhatt (1999) devotes chapter 7 in its entirety to moment distribution, describing the method as a simplified stiffness approach for manual calculations. Coates *et al.* (1988) give a thorough treatment in their chapter 6; they describe moment distribution as "a most powerful tool for the analyst without computing equipment and provides a convenient conceptual mechanism and much of the terminology in everyday usage". Gennaro (1965) devotes chapter 8 to moment distribution, covering the subject from a programming point of view, concluding with a Fortran program for multi-storey frames. Ghali & Neville (1988) add academic rigour to the method especially frames with sway, and give an example of the method applied to a Vierendeel girder.

Grassie (1957) in his chapter 2 gives many examples for both continuous beams and rigid frames, the treatment is easy to follow. Morley (1948) in his chapter 2 gives a simple introduction. Pippard & Baker (1957) in their chapter 10, thoroughly cover the method for continuous beams, sway frames & frames with semi-rigid joints, and secondary stresses. Reynolds (1957) in table 16A gives a simple introduction; Reynolds & Steedman (1988) in tables 66 & 67 give a simple introduction and include frames with sway. The Steel Designers Manual (1966) in chapter 13 gives numerous examples including frames with sidesway. Cross (1929) suggested the application of the moment distribution method to the determination of secondary stresses, a worked example is provided by Thompson & Cutler (1932).

Verified models vm120 and vm260 in appendix A, compare the matrix stiffness method with moment distribution. Verified model vm120 for a continuous beam, carries out an analysis using the matrix stiffness method including a mixture of dead and imposed: concentrated, distributed and linearly varying loads for pattern loading in accordance with BS 8110. The data for vm120 will be found in appendix A. The five lines within the label :880 and IF ncyc<32 GOTO 880, which provide self-checking for the matrix stiffness method, carry out the entire moment distribution, previous lines being for computing the fixed end moments and distribution factors for the various types of load and patterns. Although now 75 years old, the moment distribution method remains elegant, and still in use today. Verified model vm120 in appendix A, devised as part of this research, self checks the matrix stiffness method with the moment distribution method. The basis of the self check is that results of the model and the self check are within a few percent. Professor Michael Horne estimates that 3% error is inherent in the finite element method when compared to a classical elastic solution. Accordingly the basis of acceptance is taken as 3%.

2.8 Column analogy

As well as *moment distribution*, Cross (1929) also gave engineers *Column analogy* - another tool for their design repertoire. The method only applies to singly connected frames, such as portals or continuous beams; typically pre-computer reinforced concrete designers used column analogy once or twice a year for the analysis of frames for which the cross-section of members varied *e.g.* bents with tapered legs. Pippard & Baker (1957) give a good introduction to column analogy. Grassie (1957) gives numerous examples of the application of column analogy. Ghali & Neville (1997) give a formal definition of column analogy, but no worked examples.

Column analogy is a subtle method, certainly not intuitive *e.g.* hinged feet have no stiffness so that for a short column distance the column section has infinite width, which means that the centroid must be midway between two columns. Verified model vm210 in appendix A, for the analysis of a rectangular portal frame or bent, self checks the matrix stiffness method with column analogy.

2.9 Kleinlogel

Kleinlogel (1952) in Rigid Frame Formulas, translated from the German *Rahmenformeln*, provides explicit formulae for 114 rigid frame shapes. Kleinlogel was popularised in Britain by inclusion of some of his frames in the Third edition of the Steel Designer's Manual (1966), mentioned in the Fifth Edition (1992) as being in chapter 11, but confusingly appearing in the appendix. The use of Kleinlogel's frame formulas was not restricted to steel frames, in the sixties concrete portal frames were popular for economic reasons.

In the foreword to the first American edition of Kleinlogel, Professor I.F. Morrison states "the rigid frames, themselves statically indeterminate, can be used as units in adopting a primary structure dealing with cases of more highly indeterminate frames, and so bring such structures within the range of easy computation by means of the Maxwell-Mohr work equation, or if one prefers, the slope-deflection equations." This statement anticipates the use of sub-frames; sub-frames provide a useful tool for checking. Verified model vm220 in appendix A, for the analysis of a two bay ridged portal frame, self checks the matrix stiffness method with Kleinlogel's frame formulas.

2.10 Hetényi

The classical analysis for ground beams on elastic foundations is that provided by Hetényi (1948) who provided formulae for deflection, slope, bending moment & shearing force for a distributed load or concentrated force at an arbitrary point on a long ground beam elastically supported. Verified model vm130 in appendix A, for the analysis of a ground beam on an elastic foundation subjected to a train of loads, self checks the matrix stiffness method with Hetényi.

2.11 Flexibility

The Flexibility Method also known as the Force Method or Argyris Force Method, became popular in the sixties as mechanical calculators provided sufficient accuracy to enable indeterminate structures having three to eight redundancies to be analysed manually, albeit some taking a day. In the flexibility method, redundant reactions are released and the resulting determinate frame analysed, then values of released redundant actions are calculated so that continuity is restored. McMinn (1962), provides a good introduction to flexibility giving numerous worked examples for indeterminate plane trusses, continuous beams, rigid plane frames, and initial strains. Ghali & Neville (1997) in their chapter 3 give a five step approach to the method. Genarro (1965) in his chapter 4 gives a good introduction to flexibility with particular emphasis on indeterminate trusses; Grassie (1957) in his chapter 4 covers indeterminate trusses using a non-matrix treatment. Harrison (1973) devotes his chapters 2 to 4 to the method covering the application of the force method to: indeterminate plane trusses, rigid plane frames, initial strains, giving many worked examples. Baker *et al.* (1972) in their chapter 2 cover the force method with particular reference to shells. Verified model vm131 in appendix A, for the analysis of a ground beam supported on elastic piles and subjected to a train of loads, self checks the matrix stiffness method with the flexibility method; theory, derived by the writer, is included in vm131.

2.12 Influence lines and Müller-Breslau

Morley (1948) extensively covers the subject of influence lines for: cantilever bridges, continuous beams, spandrel-braced arches, suspension bridges, swingbridges and trusses. Reynolds (1957) gives graphs for influence lines for continuous beams having equal spans and various span ratios. Grassie (1957) extensively covers the subject of influence lines for: single span beams with various end fixities, continuous beams, fixed portal frames, spandrel-braced arches (top boom normally straight, bottom boom curved), parabolic fixed arches and two-hinged arches, long span bridges, internally redundant frames. Bhatt (1999) covers the subject of influence lines for: axial force in pin jointed trusses, indeterminate structures, three-pin arches, and discusses the Müller-Breslau principle. Coates *et al.* (1988) cover influence lines for single span and continuous beams, and discuss the Müller-Breslau principle. The Steel Designers' Manual (1992) gives influence lines for bending moments, shear forces and reactions in two, three and four span continuous beams. Pippard & Baker (1957) devote their chapter 15 to the subject commencing with the Müller-Breslau theorem which they derive from Castigliano's first theorem, and covering: continuous beams, trusses, and two pinned arches. Ghali & Neville (1997) devote their chapters 12 & 13 to the subject of influence lines & the Müller-Breslau theorem for: arches, beams, grids, plane frames and trusses. Verified model vm140 in appendix A, for the production of influence lines, self checks the matrix stiffness method with Müller-Breslau.

2.13 Castigliano's first theorem method

Castigliano's first theorem (1879) states "If the total strain energy expressed in terms of the external loads be partially differentiated with respect to any one of the external loads, the result gives the displacement of that load with its own line of action". Modern programs for structural analysis - which include shear deformation - rely as much on Castigliano's first theorem as they do on matrix arithmetic. Verified models vm290 & vm291 in appendix A, for an outrigged frame and braced outrigged frame respectively, self check deflections computed by the matrix stiffness method with those found from applying Castigliano's first theorem.

2.14 Unit load method

Grassie (1957) derives the Unit load Method from first principles, and subsequently notes that the working formula for the determination of the deflection at any section of a straight beam is the same form as that derived by Castigliano's First Theorem Method. Verified models vm113 & vm114 in appendix A, for prismatic and tapered cantilevers respectively, compare deflections computed by the matrix stiffness method with those found using the unit load method.

2.15 Method of joints

Lattice girders and portals offer a lightweight and architecturally interesting alternative to heavy long span beams; the latticing permits building services to be incorporated within the depth of the lattice. The *Method of Joints* assumes that all members are pinned at their joints *i.e.* PLANE TRUSS, and that the truss is statically determinate. The method is the traditional method for the analysis of pin-jointed trusses in which the engineer first computes the reactions by equilibrium *i.e.* applying ΣX=0 ΣY=0 ΣMZ=0, and then proceeds from the left support such that only two unknown member forces occur at each joint *i.e.* the same sequence that an engineer would follow in the manual solution of a truss. A good description of the *Method of Joints* is given by Gennaro (1965). Verified models vm156-181 in appendix A, self check member forces computed by the matrix stiffness method with those computed by the method of joints.

2.16 Pierced shear walls

Magnus (1968) derives the differential equations for pierced shear walls, applies the boundary conditions and solves to give bending moments and shears on the walls and lintels (coupling beams). Magnus ignores axial deformation due to applied vertical loading and only considers deformation due to the shears in the lintel. Verified model vm270 in appendix A, self checks member forces computed by the matrix stiffness method with forces computed by Magnus' formulae.

2.17 Roark's Formulas

Roark's "Formulas for Stress and Strain" have been a mainstay for engineers since the first edition was published in 1938. The fourth edition (Roark, 1965) and the seventh edition (Roark, 2002) were published in paperback for students. In the fourth edition, Roark had the courtesy to provide a *name index* as well as a *subject index*; alas the seventh edition under authors Young & Budynas, no longer contains a name index. The following verified models use Roarks' formulas in their self check: vm115 Cantilevered beam with tie down span; vm630 Spherical shell; vm640 Torque on I-section; vm641 Torque and biaxial bending on rectangular hollow section; vm642 Torque and bending on T-section; vm643 Torque and bending on channel section; vm644 Torque on angle section; vm620 Circular balcony; vm718 Natural frequency of built-in plate; vm830 Stability of circular ring/pipe.

2.18 Reynolds

Six years before Roark's "Formulas for Stress and Strain" was published, Reynolds published the first edition of "The Reinforced Concrete Designer's Handbook", providing charts and tables and other design information. Verified model vm211 uses Reynolds' treatment, for a rigid pile cap with many piles see the discussion for vm211 in chapter 11.

2.19 Arches & bow girders

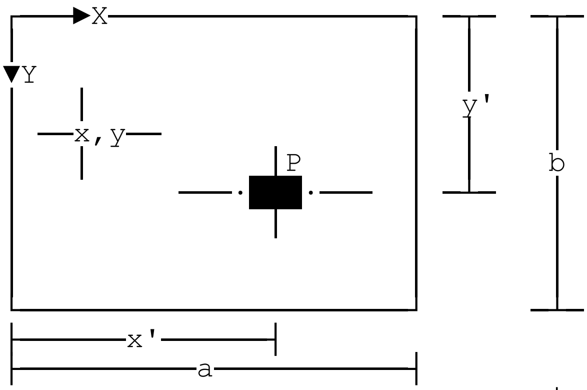
Pippard & Baker (1957) provide the classical solution for arches and bow girders. The following verified models use the formulae derived by Pippard & Baker: vm280 Two pinned circular arch; vm281 Encastré circular arch; vm282 Two pinned parabolic arch; vm283 Encastré parabolic arch; vm301 Circular arc cantilever; vm302 Circular arc bow girder. See the discussions for vm280-vm283 and vm301-vm302 in chapter 11.

2.20 Cables and suspension bridges

In addition to providing classical solutions for arches & bow girders, Pippard & Baker (1957) also provide the classical solution for cables and suspension bridges. The following verified models use the formulae derived by Pippard & Baker: vm950 Hanging cable with flexible platform; vm951 Suspension bridge with three pinned stiffening girder; vm952 Suspension bridge with two pinned stiffening girder. See the discussions for vm950-vm952 in chapter 11.

2.21 Plates and grillages

When a plate can be represented in the form of a double trigonometric series then a solution for the plate can be obtained using Navier. Timoshenko & Woinowsky-Krieger (1959) in their section 29 entitled *Further Applications of the Navier Solution* derive their equation 133 for the deflection at any point on a simply supported rectangular plate due to a single point load anywhere on the plate. The calculation below shows the expression for the deflection at any point on a rectangular simply supported plate due to a point load at any position on the plate.



For a concentrated load P located at (x',y') on a rectangular simply supported plate having sides of length a & b, plate thickness h, with a point of interest located at (x,y), having Young's modulus E & Poisson's ratio nu, having

$$\text{plate stiffness } D = \frac{E.h^3}{12(1-nu^2)}$$

$$\text{Deflection } w = \frac{4P}{\pi^4 abD} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\sin \frac{m\pi.x'}{a} \sin \frac{n\pi.y'}{b}}{\left[\frac{m^2}{a^2} + \frac{n^2}{b^2}\right]^2} \sin \frac{m\pi.x}{a} \sin \frac{n\pi.y}{b}$$

Expression for the deflection at any point on a rectangular plate.

Although the equation above looks formidable, inspection of vm601 & vm602 in appendix A shows that the thirteen lines following the SOLVE command in vm601 give an accurate self check for the deflection due to a uniformly distributed and any number of concentrated loads; that the seven lines following the SOLVE command in vm602 give an accurate self check for the deflection due to a uniformly distributed area load. Chapter 13 recommends that further research be carried out using Navier's approach for rectangular plates having any mixture of: free, simply supported, built-in or partially restrained edges.

Pilkey & Chang (1978) also use the Navier approach for which the deflection of a simply supported grillage of beams can be represented in the form of a double trigonometric series to obtain a solution for a grillage of beams. See the discussion for vm310 in chapter 11. Section 13.1 recommends that further research be carried out using Pilkey & Chang's Navier approach for a grillage of beams to generalise their expression for deflection at any beam intersection on the grillage due to a unit load applied at any other beam intersection point.

Ghali & Neville (1997) in their Example 16-4 for a square plate with three simply supported edges and one free edge, use the finite difference method to derive expressions for:

- deflection at the centre of the free edge
- moment at the centre of the free edge
- moment about X at the centre of the plate
- moment about Y at the centre of the plate.

These expressions they compare to exact expressions derived by Gere (1963). Verified model vm610 compares equivalent results from NL-STRESS with values from the finite difference method & exact formulae.

2.22 Circular tanks

Timoshenko & Woinowski-Krieger (1959) in their section 117, substitute the force on the wall of a tank of *Z=depth times weight per unit volume* into the fourth order differential equation for a cylindrical shell, which they note is the same form as that of a prismatical bar with a flexural rigidity D, supported by a continuous elastic foundation and submitted to the action of a load of intensity Z. From this they obtain a particular solution which represents the radial expansion of a cylindrical shell with free edges under the action of hoop stresses. Substituting this into the general solution of the fourth order differential equation for a cylindrical shell, gives the complete solution involving four constants of integration. Assuming that the wall thickness is small in comparison with both the radius and the depth of the tank, *i.e.* the shell is infinitely long, two of the constants are equal to zero. The remaining two constants are obtained by applying the conditions at the base of the tank *i.e.* wall deflection and rotation are both zero. Back substitution of the two values gives the deflection at any level on the wall. The full theory will be found in verified model vm650.ndf in Appendix A, omitted here for reason of space.

2.23 Natural frequency

Classical methods used in the self check are dependent on the type of structure. For determining the natural frequency of a multi storey frame, it is usual to model the structure as a vertical cantilever beam, lumping the masses of the floors at floor levels. For cantilevers, the flexibility matrix may be determined using moment area theorems, chapter 5, McMinn (1962) then determining the largest latent root (lambda), chapter 9, McMinn (1962) using the power iteration method, finding the period from $T=2\pi\sqrt{(lambda/g)}$ and hence frequency from 1/T. Theory will be found in verified model vm710.ndf in Appendix A, omitted here for reason of space.

For a floor plate, form the flexibility matrix for inside joints using Navier's solution in the form of a double trigonometric series giving the deflection at any position on a simply supported rectangular plate for a load applied at this position or any other position, section 2.21, then determining the largest latent root (lambda), chapter 9, McMinn (1962) using the power iteration method, finding the period from $T=2\pi\sqrt{(lambda/g)}$ and hence frequency from 1/T. Theory will be found in verified model vm720.ndf in Appendix A, omitted here for reason of space. For determining the natural frequency of plates, Warburton (1964) equates the strain energy in a plate to its kinetic energy, using a non dimensional frequency factor =36 for the fundamental mode, noting that the natural frequency computed using this factor agrees to within 0.06% of values obtained by Rayleigh-Ritz, chapter 3 in Warburton (1964).

2.24 Stability

Many structural text books deal with stability of structures and structural components. Pippard & Baker (1957) give Euler's classical treatment, noting "It should be noticed that in the absence of an external disturbing force all perfect struts, whether slender or stocky, will fail by direct compression". Pippard & Baker (1957) cover the Modified Smith formula, Perry strut formula, a variety of strut problems including combined bending and tension, polar diagrams. McMinn (1964) in his chapter 10 covers the stability of rigid frames from a computational viewpoint. Horne & Merchant (1965), as their preface states, "give a clear picture of phenomena affecting the stability, both in the elastic and in the partially plastic range, for plane, rigid-jointed, triangulated and non-triangulated frames". Horne & Morris (1981) cover local stability, Rankine-Merchant method, sway instability, and mention that it is "necessary to introduce, by means of stability functions, the effects of axial thrust on the stiffness of members". (NL-STRESS, with Michael Horne's blessing, avoids the use of stability functions by segmenting members and monitoring the bow for each member as the loading is applied.) Coates *et al.* (1988) in their chapter 9, entitled "The instability of struts and frameworks", commence with the Euler buckling load, and cover the design of steel struts, complex struts, lateral-torsional buckling, virtual work approach, post buckling behaviour, stability functions and the calculation of critical loads on plane frames. Ghali and Neville (1997), consider the effect of axial compression on structural members, the elastic stability of frames, and use moment distribution to calculate the buckling load for frames. Roark (1965 & 2002) includes formulas and tables for the elastic stability of bars, rings, plates and shells. For structures for which Roark's formulas are available, they provide a simple and elegant self check for elastic critical loads computed by non-linear matrix methods.

This chapter commenced by showing that *testing software* and *self checking* is compelling and urgent. The chapter also shows that there is a wealth of classical methods for the analysis of structural frameworks, which may be harnessed for the self checking of the majority of structures routinely analysed.

Chapter 3

Tools

Harris *et al.* (2002) in *Guidelines for the use of computers for engineering calculations* give good advice on the management of design projects within the design office; this research supports the guidelines by identifying an armoury which can be used to find errors in calculations. Harris *et al.* (2002) advocate *checking models* and describe how to use a checking model. This chapter proposes that in addition to the stiffness method of structural analysis, each checking model should include an appropriate classical structural method or other method as a self check, bound within it, so that the self check verifies the results of each structural analysis or design to within 3%, and thereby avoids uncertainty when the checking model and conceptual model do not agree. The classical methods of analysis, incorporated into the self checking models developed as part of this research, are reviewed in chapter 2.

3.1 Definition of terms used

A **benchmark** is defined as a set of results: deflections, stresses, forces, reactions, frequencies, member sizes, area of reinforcement *etc.* which is used as a reference to check that a new calculation using the benchmark data gives the same result as that previously. In general all engineering calculations are compromised by the omission of one or more effects such as: shear deformation, non-linear material properties, finite displacements, stability, fatigue, seismicity and other ambient conditions *etc.* hence the need for benchmarks. Changes to software give another need for benchmarks, colloquially *benchmarking*, to ensure that previous behaviour of the software has not changed, or has changed as expected.

Conformance checking, every structural design should be produced in accordance with, and checked for conformance against the appropriate standards, Thomson *et al* (2000).

Correctness when used to qualify a benchmark test, is used in the sense of true or false, *i.e.* if the result of using the same set of data with the same benchmark gives identical results, then the check is *correct*, if the result is different then the check is *incorrect*. Although fuzzy reasoning (Mamdani & Gaines, 1981) is applied to engineering systems such as steel making, fuel economy *etc.*, the engineering calculations considered in this research are essentially arithmetical, thus a benchmark test will be correct or incorrect. Saying the benchmark test or check (both are used in the sense of seeing if a set of results has changed when the data has not changed) is correct, does not imply that results are correct. For example a test on a benchmark for a portal frame subjected to vertical and horizontal loading, ignoring axial deformation, may be correct but the results will be incorrect if axial deformation has been ignored.

Coverage (Marick, 1995), when applied to engineering software, is a measure of how thoroughly a set of benchmarks, test the software, *e.g.* a set of benchmarks covering all of Kleinlogel (1952) would give low coverage for a structural analysis program which took into account: 2D/3D elastic, finite displacement, stability, plastic, collapse or finite element analysis, high level command language including logic and looping providing parametric data, selection of printed results from joint displacements, member forces, stresses, reactions at supports, segmented members giving the moments, shears & deflections along the length of each member, computation of elastic stresses, minimisation of band-width of the stiffness matrix, shear deformation treated rigorously, use of symmetry to reduce data preparation, joint & member end springs, member properties computed from geometry, selection of member properties from a steel section library, tension or compression-only members, application of load to members referred to global axes, self weights of members computed automatically, enveloping and combination of load cases, temperature changes in members, member distortions (lack of fit), length coefficients (creep and shrinkage), unloading plastic hinges *etc.*

Checking is the process of examining the quality of a calculation using engineering methods.

Checksum is the summation of a *key-value i.e.* one non-trivial real-value taken from each of a set of benchmarks providing high coverage, of the model being tested. Checksum provides a simple tool for verifying that no changes in the results have taken place, or if changes have taken place then provides a focus for the changed results to be investigated and explained. Checksum has an accuracy of 15+ decimal digits, the limit of accuracy of double precision arithmetic. The question arises "Is one key-value sufficient to check the correctness of a run or do we need to check more than one result, if so how many nodes and points between the nodes do we need to check?". Such matters are discussed in chapter 9.

A **discrete benchmark** is defined as the combination of an engineering model and a set of associated data. A model in which the data is intrinsic provides one discrete benchmark. A model in which the data is provided parametrically *e.g.* a multi-storey frame in which the number of storeys, columns, section properties... may vary, provides a discrete benchmark for each discrete set of data run with the model.

NL-STRESS is a mature computer program for the analysis of structural frameworks, extended as part of this research to permit logic to be included between the SOLVE and FINISH commands to provide a self check for the analysis.

Moment of inertia, used colloquially but incorrectly for the second moment of area of a section.

Pippard and Baker (1957) restate the **Principle of Saint Venant** as: *forces applied to one part of an elastic structure will induce stresses which except in a region close to that part, will depend almost entirely upon their resultant action, and very little on their distribution*. The question arises, if a force has changed on a member of a structure, will a key value chosen to provide a benchmark, show that the data has changed. Modern engineering calculations by computer, use double precision arithmetic with 15+ decimal digits of accuracy. UNIX & Linux Alpha systems support REAL(16) arithmetic, where the 16 refers to the number of bytes, providing typically 33 decimal digits of accuracy. With such arithmetic accuracy, the *very little* in the restated Principle of Saint Venant is always going to be sufficient to pick up a change in result when a benchmark is retested, thus when a benchmark is re-tested, inspection of 16 significant decimal digits of a key-value will be sufficient to show if the results or the data have changed. It follows that if the *checksum* of say 100 discrete benchmarks equals that obtained on the previous test, then it is reasonable to conclude that the results of the tests prove that all the runs are as for the previous test. This conjecture will satisfy most engineers but is unlikely to satisfy mathematicians; the fact that hundreds or thousands of benchmark tests are needed to provide adequate coverage (Marick, 1995), of an engineering problem, mitigates this conjecture. For all engineering problems, a balance must be struck between simplicity, in this case having a simple checking audit trail based on a key-value in each set of results, and rigour *i.e.* having certainty that bugs introduced by program changes will be found by the benchmark tests. If the above conjecture is accepted, then it follows that: increasing the number of discrete benchmarks in a set will be preferable to increasing the number of values checked in each discrete benchmark; the more the variety of benchmarks, the greater the coverage.

Praxis (1990) is a mature computer program for reproducing calculations in a paginated and tidy layout suitable for submission to a checking authority.

Self checking software is software included at the end of a model which independently checks that the results produced by the model are correct, not correct, or differ by a percentage.

BS EN 1990:2002 clause 3.4 states that the limit states that concern:

- the functioning of the structure or structural members under normal use
- the comfort of people

- the appearance of the construction works

shall be classified as **serviceability limit states**.

Experience shows that shear cracking is a serviceability limit state *e.g.* NCE, 8 December 2005 reports *A £200M shopping complex in Bournemouth is being closed indefinitely* due to shear cracking and diagonal spalling at the ends of long span beams, see section 10.7.

BS EN 1992-1-1:2004 (E) states linear elastic analysis of elements based on the theory of elasticity may be used for both the serviceability and ultimate limit states. For the determination of the action effects, linear analysis may be carried out assuming:

- uncracked cross sections
- linear stress-strain relationships
- mean value of modulus of elasticity.
-

Stack a set of parameters and associated values *e.g.* b=0.3 d=0.6 w=34.5 l=6.2 held within the computer's memory; **stack file** when held as a file on disk.

The **Uniqueness Theorem** (Coates *et al.* 1988), applicable to models for structural analysis, states that: *If, in addition to the body forces, either the surface forces or the surface displacements on the boundary of an elastic body are specified, then there exists one, but only one, solution for the stresses (and strains) in the body.* The Uniqueness Theorem implies that checking just one computed displacement, force or stress, against the corresponding displacement, force or stress from a previous set of results (the benchmark), will be sufficient to show if there has been a change in: the model, data, arithmetic, compiler, processor, disk *etc.* Excluding contrived exceptions *e.g.* when a UDL on a span is replaced by two partial loads to model the full UDL on that span; in general a change in the data for a continuous problem will change the displacement at every non-fixed nodal point or the member forces at every non-released member end. Practical exceptions occur when discontinuities are present in the model *e.g.* if, in a three bay by three storey frame, for structural reasons the centre span at 2nd floor level carries a UDL with moment releases at each end, then the bending moment at the centre of this span would not be a key-value for substantial changes in the behaviour of the remainder of the model will not affect the centre bending moment.

Verifying means establishing the truth or correctness of software models by examination or demonstration. Each model to be verified incorporates a self check, in this thesis **verification** is the process of generating a thousand or more discrete sets of engineered data providing high coverage for the model, running the model with each set of data, computing the average percentage difference between key results produced by the model and its self check, averaging the key results for each run, averaging for all runs and when the average percentage difference for all runs is within an acceptable value, typically 3% for models for structural analysis, then the model is said to be a **verified model**.

3.2 Software maintenance

Kasper & Godfrey (2006) tell us that "Code duplication, or code cloning, is a well-documented problem in industrial software systems. An example of how problems can arise is when multiple copies of one piece of code must be modified to fix a single bug. This leads to wasted effort in both finding and fixing the clones. Fanta & Rajlich (1999) report on a process for eliminating function clones and class clones from industrial object-orientated code; such clone removal can decrease system code size and facilitate maintenance. To answer the question *Which software modules have faults which will be discovered by customers?* Khoshgoftaar *et al.* (1999) describe a system of decision support tools used by software designers and managers at Nortel to assess risk and improve software quality and reliability. They conducted a case study of a large telecommunications system in the maintenance phase to predict whether each module will be considered fault-prone.

Software systems with a million lines or more of code, are designed to have an elegant structure. When users report a bug, section 2.1, additional code is added to fix the bug. As the years pass, the elegance of the original structure is compromised by *patches* and sometimes *patches on patches*. Gurp *et al.* (2005) present two case studies in the Netherlands, to identify *design erosion*. They address the problem by a practice they call *design preservation* and analyse the problems the systems had and provide remedies in the form of *design preservation practices*.

As software systems evolve over a series of releases, it becomes important to know which components are stable compared to components which show repeated need for corrective maintenance. Andrews *et al.* (2000) track faults over multiple releases and adapt a *reverse architecting technique* to defect reports of a series of releases. Fault relationships among system components are identified based on whether they are involved in the same defect report, and for how many defect reports this occurs. Comparisons across releases makes it possible to see whether some relationships between components are repeatedly fault prone, indicating an underlying systemic architecture problem.

Mayrhauser & Zhang (1999) tell us "Regression testing is an important activity in software maintenance. Current regression testing strategies can be categorised into two groups: *retest all* and *selective regression* testing. In industrial practice, regression testing procedures vary widely. Sometimes several regression testing techniques are used in combination. Mayrhauser & Zhang (1999) use a test generation tool based on domain-based testing; they explain the rules from retest all strategies to selective regression testing strategies.

Schach & Tomer (2000) present a process for software construction that recognises maintenance as an essential aspect of the entire life cycle of the software product. The process may be used in conjunction with any software development or maintenance methodology. The process consists of two components: a procedure that is uniformly applied at every step of the chosen methodology, whether development or maintenance, and a data structure, the propagation graph, which is updated at every step. When requirements change, the propagation graph is used to determine which artifacts of the software product are impacted by the change in requirements. Niessink & Vliet (2000) investigate the differences between software maintenance and software development; they argue that software maintenance can be seen as providing a service, whereas software development is concerned with the development of products. Consequently customers judge the quality of software maintenance differently from how they judge the quality of software development. They discuss two overall approaches to achieving a high quality service.

3.3 Flow charts

Over 723,000 lines of structural engineering calculations and libraries of component details, written in Praxis (1990), are currently in use in Britain. Praxis is popular with engineers as they can view each *proforma* calculation in its entirety. Some calculations contain several thousand lines, others just a hundred. Proforma calculation 370, which follows, gives the first part of a flow chart which identifies by >>> an authentic bug caused by a missing ENDIF before the start of the first procedure.

```
Proforma No. 370
Title Shape limitations for circular hollow sections.
Based on BS 5400 : Part 3 : 2000 and DTp Standard BD 13/04
Amendments November 2005 First issue
@scale.sta
! +ZZZZZ1=0 +ans1=1 +$4000=BS5400: Part 3: 2000
START
┌IF ZZZZZ1=0
│%Shape limitations for circular hollow sections. Clause 9.3.6.
│%───────────────────────────────────────────────────────────────
│%The calculations are in accordance with BS 5400: Part 3: 2000
│%" Code of Practice for Design of Steel Bridges " as implemented
│%by Departmental Standard BD 13/04.
│%The proforma checks for compliance with the shape limitations to
│%Clause 9.3.6.
│%
│%Would you like a set of defaults to be provided. You can use
│%the default values as references and type your own values
│%beneath to replace them.
│!Answer ( 1=Yes, 0=No ) +ans1=????
│! +maximum=3 +minimum=0 +value=ans1
│MAXMIN
│┌IF ans1=1
││! +$3000=Tube X
││! +ans2=1 +sys=355 +od=170 +wt=5 +ans3=1 +syc=355
│└ENDIF
│┌IF ans1=2
││! +$3000=Tube X
││! +ans2=1 +sys=300 +od=240 +wt=5 +ans3=1 +syc=355
│└ENDIF
│┌IF ans1=3
││! +$3000=Tube X
││! +ans2=1 +sys=355 +od=300 +wt=4 +ans3=1 +syc=355
│└ENDIF
└ENDIF!
```

```
+ZZZZZ1=1
Shape limitations for circular hollow sections. Clause 9.3.6.
The calculations are in accordance with BS 5400: Part 3: 2000
" Code of Practice for Design of Steel Bridges " as implemented
by Departmental Standard BD 13/04.
The proforma checks for compliance with the shape limitations
to Clause 9.3.6.
Location +$3000=????
────────
Circular hollow sections. Clause 9.3.6
The ratio of outside diameter to wall thickness of a circular
hollow section should not exceed 60√(355/σ).
STRES1
Nominal yield stress σy +sys=???? N/mm2
Outside diameter of section +od=???? mm
Wall thickness +wt=???? mm
Factor 60√(355/σ) +p=60*SQR(355/sys)
Factor od/wt +q=od/wt
┌IF q>p
│.Ratio of outside diameter to wall thickness +q exceeds
│.60√(355/σ)= +p
│.Section does not comply with Clause 9.3.6
│.Lesser value of stress. Clause 9.3.1
│.For compliance +syl=(60/q)^2*355 N/mm2
├ELSE
│.Ratio of outside diameter to wall thickness +q is less than
│.60√(355/σ)= +p
│.Section complies with Clause 9.3.6
└ENDIF
%Check for compact section
!Answer ( 1 = Yes, 0 = No ) +ans2=????
┌IF ans2=1
│.Compact sections. Clause 9.3.7.4
│! +C1=0
│STRES2
│STRES3
│Factor 46√(355/σ) +r=46*SQR(355/syc)
│┌IF q>r
││.Ratio of outside diameter to wall thickness +q exceeds
││.46√(355/σy)= +r
││.Section does not comply with Clause 9.3.7.4
││! +C1=1
│├ELSE
││.Ratio of outside diameter to wall thickness +q is less than
││.46√(355/σy)= +r
││.Section complies with Clause 9.3.7.4
│└ENDIF
│┌IF C1=1
││.Section is not compact
│├ELSE
││.Section is compact
│└ENDIF
│%To produce only a summary, respond C/S at the next prompt go
│%through the calculation again. To reset back to normal
│%operation after producing a summary, respond C/N.
│STOP
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
```

Start of a flow chart for a proforma calculation with a bug.

3.4 Comments in the data

For complicated structural analysis, the ability to include comments in the data, aids the checking process for both the author of the data and the checker. Some files of data exceed 100 pages; the NL-STRESS data which follows shows the start of an authentic set of JOINT COORDINATES extracted from a complicated set of data for a non-linear problem. Lines starting with an asterisk are comments, text following an exclamation mark are comments, without the ability to include comments in a model, an engineer *taking over* the job would have great difficulty in understanding the design assumptions.

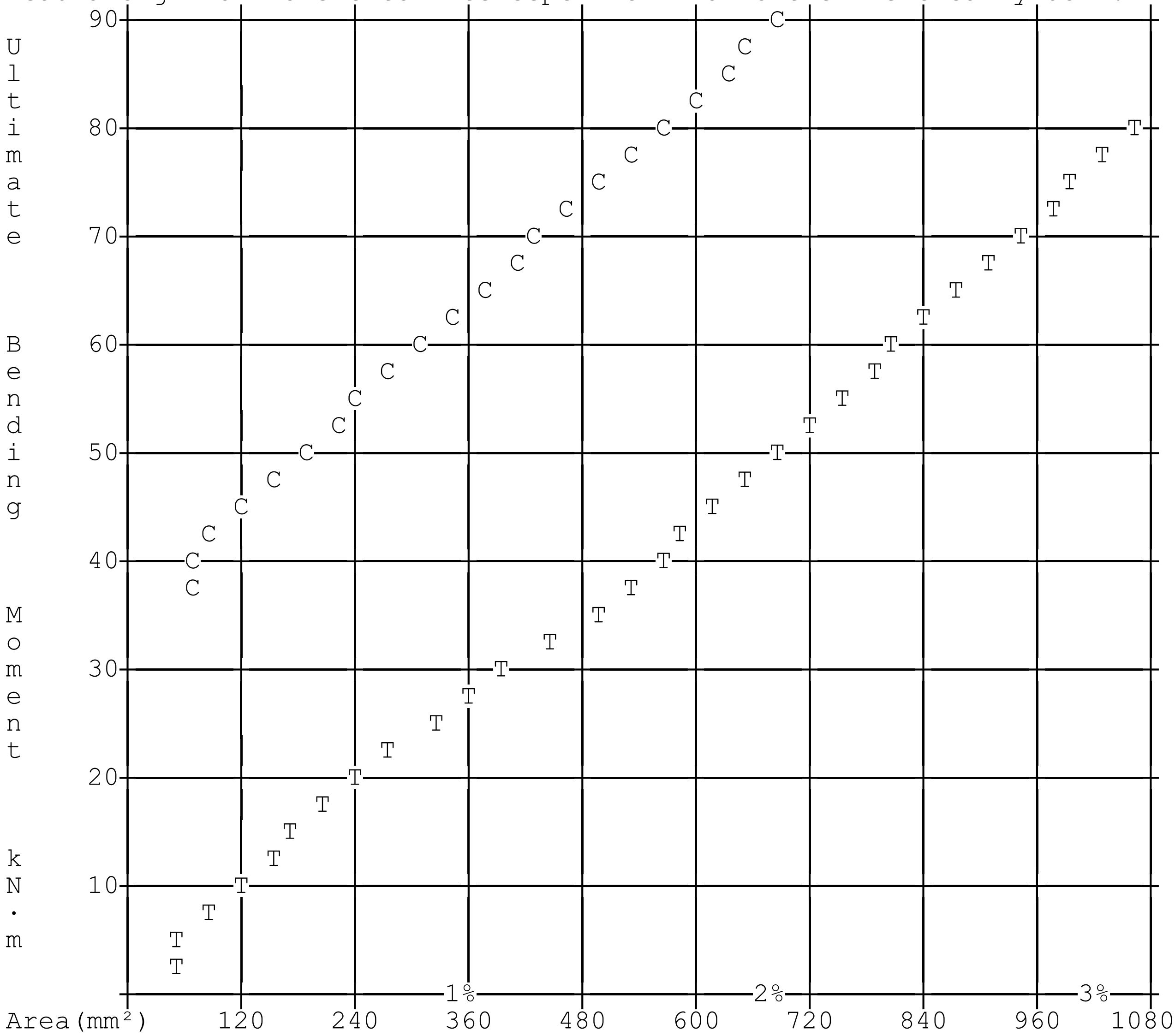
```
JOINT COORDINATES
bx=2100 b1=bx, b2=bx, b3=bx b4=bx, b5=bx, b6=bx
hj=2000 ! jack extension
hx=2000 h1=hx, h2=hx h3=hx, h4=hx, l=hj+h1+h2+h3+h4
a=1240.0, e1=54, e2=100, e3=100
*
* ALLOW FOR IMPERFECTIONS
* Bow Imperfection (Eurocode 5.2.4.4)
* x-k*SIN((PI/l)*y)+tan*y y
* k=the maximum imperfection at half height=kf*L/500 where
* L=span (height), kf=[0.2+1/nr]^0.5, nr=No. of members adjacent
* across the facade nr=2 hence kf=[0.2+0.5]^0.5=0.837,
* kz=0.837*l/500=l/597
* parallel with facade, nr=7, kf=[0.2+0.1428]^0.5=0.586,
* kx=0.586*l/500=l/854 hence for 8 m height
* kz=10000/597=16.73 kx=10000/854=11.71
*
* SWAY Imperfection (Eurocode 5.2.4.3)
* tan=kc*ks*l/200, where ks=[0.2+1/ns] where
* ns=number of storeys=4, ks=0.671
* kc=[0.5+l/nc]^0.5 where nc=as for nr above
* across the facade kc=1,
* parallel with the facade kc=[0.5+1/7]^0.5=0.802 hence
* tanz=0.671*1.000*(l/200)=l/298=0.003356
* tanx=0.671*0.802*(l/200)=l/371=0.002695
*
tanx=0.002695 tanz=0.003356
*
* kxx=11.71*SIN((PI/l)*y)*c1+tanx*y*c2
* kzz=16.73*SIN((PI/l)*y)*c3+tanz*y*c4
*
* Apply imperfection formulae in both z and x directions, and
* also c1, c2, c3, c multiplier coefficients for bow and SWAY
* imperfections for the X and Z global directions so that
* they may be applied, deleted or modified in either dirn.
c1=0, c2=0, c3=1, c4=1
```

3.5 Checking aids

Stanchion design charts (Brown, 1974) give an immediate check on the size of a stanchion for the satisfaction of bi-axial bending moments and axial load. Reinforced concrete design charts (Brown, 1975) give an immediate check on the amounts of tension, shear and compression reinforcement for the satisfaction of known bending moment and associated shear force. On a real job there may be perhaps a dozen different beam sizes, and perhaps half a dozen different column sizes. If charts are produced for each size of beam & column for the material strengths for the job; then a glance at the appropriate chart gives an independent check on the amount of reinforcement needed.

Example: Ultimate bending moment of 40 kNm & ultimate shear of 40 kN.
For a simply supported beam spanning 4 m, carrying factored DL & LL (imposed) of 20 kN/m the bending moment =w.L^2/8=20*16/8 =40 kNm, the shear=4*20/2=40 kN. Consult the chart in figure 3.1 for a beam width 150 mm & effective depth of 225 mm, reinforced using imported non-standard steel of yield stress =425 N/mm². For compression reinforcement, read horizontally from 40 kNm on the left until the intercept with *C* and then vertically down to give the area of compression steel approximately 70 mm². Continue reading down to select the number of bars required and the bar diameter, noting that 2/8mm will give a slightly higher area of compression reinforcement than required. For tension reinforcement, read horizontally from 40 kNm on the left until the intercept with *T* and then vertically down to give the area of tension steel approximately 570 mm². Continue reading down to select the number of bars required and the bar diameter, noting that 3/16mm will give a slightly higher area of compression reinforcement than required. For shear reinforcement, consult the short table at the top of the chart for the ultimate shear of 40 kN, inspection of which shows that 2/8 mm legs at 150 mm centres will provide a shear resistance of 49 kN and therefore be adequate.

```
REINFORCED CONCRETE BEAM DESIGN/CHECKING CHART TO BS 8110 (1997)
Parameters:  fcu=30 fy=425 fyv=250 b=150 d=225 d'=30
Assumptions: simplified stress block, redistribution <= 10%.
Shear (kN) carried by link legs at various spacings assuming minimum
percentage of tension steel provided (0 = non-compliance with BS 8110).
No/dia   2/8   4/8  2/10  4/10  4/12  6/12  4/16  6/16  6/20  8/20 10/20
100 mm   67    121   98    0     0     0     0     0     0     0     0
150 mm   49    85    70    126   0     0     0     0     0     0     0
In the chart below: T denotes Tension & C denotes Compression steel.
Read along from moment to intercept with T or C then vertically down.
```




```
Number of
bars reqd   2  32   2    3   2         32                  2
    /       /  //   /    /   /         //                  /
Diameter        1   1    1   1         12                  2
of bar(mm)  8  80   2    2   6         60                  5
```

Figure 3.1 Design and checking aid for reinforced concrete beams.

Charts are one example of many such aids which could be developed for checking the calculations produced by integrated design software. Using engineers' arithmetic *e.g.* WL^2/10 to give the design bending moment on a span, from a chart such as that shown above the reinforcement may be read directly and compared to the reinforcement detailed by integrated design software used.

3.6 Proactive help versus reactive help

Accessibility of help is important. Proactive help systems are those for which the engineer is *presented* with help and does not have to look for it; proactive help systems have high functionality. The example of proactive help which follows, provided to the engineer to help with his/her response to the prompt for *cover to all reinforcement*. The engineer does not have to consult a separate document.

```
   Nominal cover to all reinforcement for durability

 Conc. Grade          C30      C35      C40      C45      C50
 MILD exposure         25       20       20       20       20
 MODERATE exposure              35       30       25       20
 SEVERE exposure                         40       30       25
 VERY SEVERE exposure                    50       40       30
 EXTREME exposure                                 60       50
 MILD: Concrete surfaces protected against weather or aggressive
 conditions.  MODERATE: Concrete surfaces sheltered from severe
 rain or freezing while wet, concrete subject to condensation,
 concrete surfaces continuously under water, concrete in contact
 with non-aggressive soil.  SEVERE: Concrete surfaces exposed to
 severe rain, alternate wetting and drying, or occasional
 freezing and severe condensation.  VERY SEVERE: Concrete
 surfaces exposed to sea water spray, de-icing salts (directly
 or indirectly), corrosive fumes or severe freezing conditions
 whilst wet. EXTREME: Concrete surfaces exposed to abrasive
 action, e.g. sea water carrying solids or flowing water with
 pH<=4.5, or machinery or vehicles.
Cover to all reinforcement       +cover=???? mm
```

Example of proactive help.

Reactive help systems, are those for which the engineer is not automatically presented with help for providing appropriate data in response to prompts, but has to:

- know which icon to click to access the help system
- know how to move around the help system
- have perseverance.

Reactive help systems have low functionality.

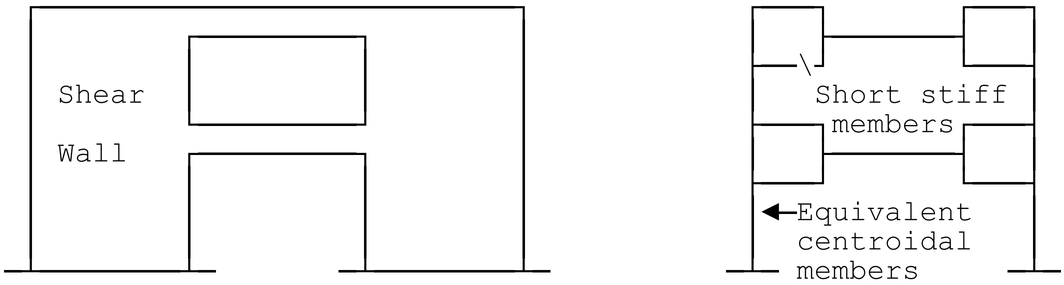
3.7 Worked examples

Most engineers prefer practical worked examples and a fundamental description to theory. Suppliers of structural engineering software know that when an engineer says for example "I haven't analysed a concrete bridge before, where do I start?", being able to refer the engineer to a worked example similar to his/her problem saves time and makes the customer happy, and if the engineer is following a well trodden path, he/she is far less likely to make an error. As an example, in modern structural analysis, bending moments, rotations and moments of inertia are *about the X, Y & Z axes*. For a bridge deck analysed as a PLANE GRID lying in the XY plane, IX refers to the torsional constant for a member and not the moment of inertia; if the engineer was following a worked example this would be made clear, thus time would be saved by both the engineer and by technical support. Clear definitions and worked examples are needed.

3.8 Guidance for modelling

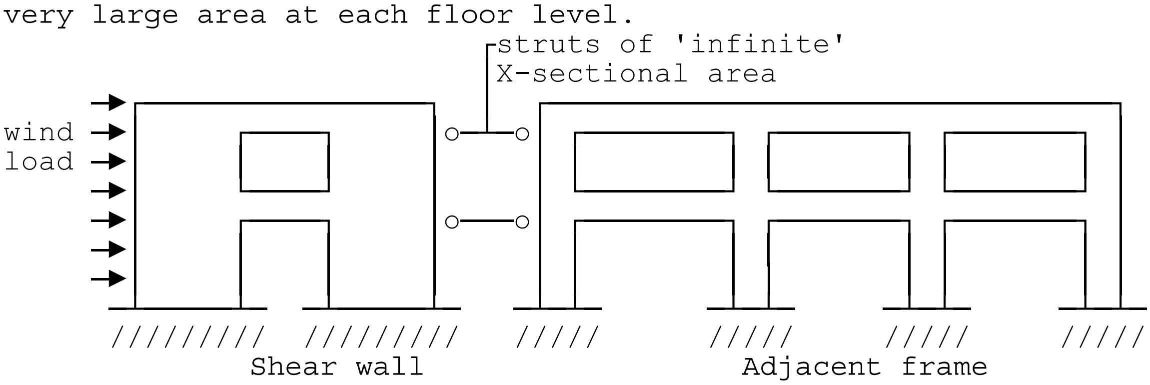
Check lists perform the same function as worked examples, they attempt to keep the engineer on a well trodden path. The following is an extract from the notes accompanying verified model vm270.NDF, such notes act as modelling guidance.

This analysis models coupled shear walls taking into account the shear deformation of the stiff storey-deep members which join the coupling beams to the centre lines of the shear walls. A shear wall may be modelled as a member running along the centre line of the wall, connected to short stiff members parallel to end faces.



To conform to the assumption that *plane cross-sections remain plane*, the short stiff members should have (ideally) infinite stiffness. In the kind of model illustrated, it should be sufficient to give values of AX and IZ say a hundred times greater than corresponding section properties of the equivalent centroidal members. The engineer may care to try different ratios of properties; for example making the area and inertia of the short stiff member ten times those of the centroidal member - then a hundred times. Assuming the larger values do not cause *overflow* in the computer, there would probably be no significant difference in the results. The word *significant* is used in the practical sense. If the results were plotted the differences would not be noticeable.

Distribution of forces between shear walls and skeletal frames lying parallel to one another presents another problem. The obvious solution is to treat the building as a space frame, but a reasonable model may be constructed by joining the shear wall and adjacent frame end to end - joined by a doubly hinged strut of very large area at each floor level.



The technique assumes symmetry in plan - not a building with its

```
lift shaft and stair wells tucked away at one end - ref. 'Analysis
of Shear Walls Using Standard Computer Programs' by Schwaighofer
and Microys - supplement to ACI Journal, Title 66-89. A digest
appeared in Proceedings V 66 No 12, December 1969, pp 1005-7.
See also 'Lateral Stiffness of Shear Walls with Openings' by I A
MacLeod, presented at the symposium on Tall Buildings at the
University of Southampton in April 1966.

NL-STRESS automatically considers shear deformation for all
members whose shear area is given, so the special calculations
given by Schwaighofer and Microys need not be carried out. The
shear area of rectangular sections is taken as 5/6 of the cross
sectional area, see 'Formulas for Stress and Strain' by Roark,
published by McGraw Hill.
```

3.9 Self checking engineering software

As stated in section 2.5, the proof of the Four Colour Theorem employed a proof assistant called Coq, developed at the French research centre INRIA. Such a mathematical assistant would need to be programmed so that it could apply *e.g.* The Uniqueness Theorem. Such programming would not be straightforward, for as shown in section 3.1, the term *body* does not embrace discontinuities introduced into structures, the term *body* is used as part of the language of engineering and as such it is part of a natural language. Zadeh (1977) has developed a meaning representational language for natural languages in which generally, a proposition, **p**, translates into a procedure, **P**, which returns a possibility distribution; with **P** representing the meaning of **p**, and the possibility distribution representing the information conveyed by **p**. Thus, there is a pathway for defining the uniqueness theorem and other theorems in terms of a possibility distribution and then using a mathematical assistant to carry out engineering reasoning. Engineers, by the nature of their work, are pragmatists; as such, engineers would prefer to work directly with their equations and avoid further levels of abstraction and the consequent increase in the probability of error. (Typically the current levels of abstraction of an engineering problem include a specialist language *e.g.* Formex (Nooshin, 1984) or Praxis (1990), Fortran (1999), C++ & C (Microsoft, 2003), assembly language, system software *e.g.* Windows XP, and at least three levels of software to manufacture the processor.)

Self checking engineering software, in this research, means self checking using an engineering method *e.g.* using the *method of joints* to check the results of a PLANE TRUSS analysed by the matrix stiffness method. Verification of structural engineering software, in this research, means confirming that a self-checking model, when run with a thousand sets of data providing extensive coverage is OK, as confirmed by average percentage differences between the results of the model and its self check.

3.10 Checking against known solutions

Real examples are great, the more the better, unfortunately engineers are either too busy or are reluctant to make their work available for fear of criticism from other engineers.

3.11 Engineered sets of test data

Within a specialist language are many commands and tables, *e.g.* the syntax (structure of statements in a computer language) for describing a MEMBER PROPERTIES table is given below, in which: capital letters indicate keywords; words in pointed brackets < > describe the kind of data needed; vertical bars | say *or*; [ ] say include one or more items from within square brackets; <members> may be expressed *e.g.* 6 or

```
6 THRU 15   or   6 THRU 15 STEP 3   or   6 12 13 132 INCLUSIVE.
```

```
MEMBER PROPERTIES
<members> [ <property> <value> ]
<members> <shape> [ <dimension> <value> ]
<members> AS <other member>
where: <property> is:
AX|AY|AZ|IX|IY|IZ|C|CX|CY|CZ|BETA|FXP|MXP|MYP|MZP
and: <shape> is:
RECTANGLE|CONIC|OCTAGON|ISECTION|TSECTION|HSECTION
and: <dimension> is D|DY|DZ|T|TY|TZ|C|CX|CY|CZ|BETA.
```

It will be clear that even for this single table, the number of combinations of data exceeds a million, and in consequence, it is not possible to test every combination of data, therefore exhaustive testing (where exhaustive means fully comprehensive) is not achievable, and extensive testing is the best that can be achieved. It will also be clear that even for this single table, test data cannot be provided by a random number generator *e.g.* <members> requires a meaningful arrangement of integer numbers and associated keywords, a beam depth DY of 1E-24 or -1E+24 is not practical, and so on. Thus, sets of data have to be engineered; but engineers are human and will tire if asked to produce a thousand unique sets of data, for testing a program. The production of a thousand unique sets of data, each of which need to be *engineered* and run automatically, is covered in the next chapter.

3.12 Symmetry

Wherever possible, start with a symmetrical structure and at least one symmetrical loading case, for which the results will be symmetrical, if not then something is wrong.

3.13 Avoiding information overload

When researching the behaviour of a type of structure, tools are needed to prevent *information overload*, including:

- presenting stresses to an engineer rather than bending moments and shear forces
- managing the tens of thousand of pages of results by reducing comparisons to percentage
- differences and then finding average percentage differences
- distilling summaries to get their essence.

Chapter 9 gives examples of the use of such tools to avoid information overload and save the engineer time.

3.14 File conversion

Conversion of files prepared parametrically to data files with numbers is useful as engineers can more quickly check a numerical value than they can check an assignment.

Parametric data files are easy to use, the data which follows shows a verified model for the structural analysis of ground beams in which the data is given parametrically. Generally parametric data is contained in lines which have an exclamation mark with a space before and after. The exclamation mark tells the NL-STRESS analysis program to ignore text which follows but include such text in the results. For brevity the self-checking component of the verified model, which is normally contained between the SOLVE and FINISH commands, has been omitted. On entering the name of the file required, in this case vm131.ndf, the program displays the data, and the engineer clicks the Edit button and changes the parameters as required. The editor works in a similar manner to that used for editing email. When the cursor is in a line which contains parameters, then the line is highlighted. On leaving the editor, the program carries out the analysis & displays the results, plotting bending moments, shear forces, deflected shape *etc.* as required by the engineer.

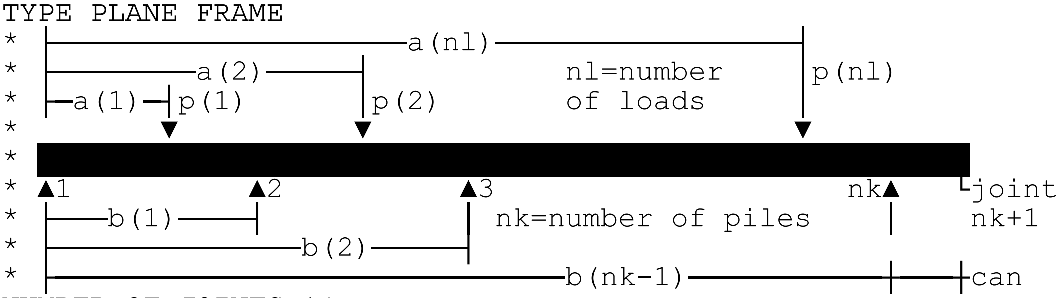
Parametric data files have considerable flexibility, but the embedded logic *e.g.* IF-GOTO is off putting to some. NL-STRESS offers a choice of output, the data file is shown before conversion and then after conversion *i.e.* with parametric data and logic removed. Consideration was given to removing the diagram during the conversion process but it was left to the discretion of the engineer, the converted data gives the engineer the ability to submit the data to the checking authority in a form which conforms with the original STRESS (A User's Manual, 1964) and STRESS (A Reference Manual, 1965). For those who are unfamiliar with STRESS, an overview is provided in Appendix B.

```
TITLE GROUND BEAMS ON ELASTIC PILES SUBJECTED
TITLE TO TRAIN OF MOVING LOADS; CHECKING OF RESULTS
TITLE AGAINST FLEXIBILITY, USING SIMPSON'S RULE & FOX'S
TITLE METHOD FOR INVERSION; AND BY ENGINEERS' ARITHMETIC.
MADEBY DWB ;DATE 08.02.05 ;REFNO vm131 ;METHOD ELASTIC JOINTS
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE FRAME run=0 ;*/7
* ├────────────────────────a(nl)──────────────────┐
* ├──────────a(2)────────┤           nl=number     │p(nl)
* ├─a(1)─┤p(1)           │p(2)       of loads      │
*          ▼              ▼                         ▼
* ████████████████████████████████████████████████████████████
* ▲1          ▲2          ▲3                      nk▲   └joint
* ├───b(1)────┤            nk=number of piles       │    nk+1
* ├─────────────b(2)──────┤
* ├───────────────────────b(nk-1)───────────────────┼────┼can
br=1 d=0.5 can=1.524 ! Beam breadth, depth, cantilever len (m).
nk=13 nl=13 ! Number of piles & load points in train.
e=4709.856E6 nu=0.2 ! Young's modulus kN/m² & Poisson's ratio.
nls=2 si=1 ! No. of load positions; spacing of steps.
! Distances from left end (m) to each pile/joint in order follow.
```

```
b0=VEC(0,1.8288,4.8768,7.9248,10.973,14.021,17.069,20.117,23.165)
b9=VEC(26.213,29.261,32.309,35.357,38.405)
! Distances from left end (m) to each load in order follow.
a1=VEC(0,2.6426,6.1874,8.8636,12.421,15.088,18.645,21.488,25.045)
a10=VEC(27.712,31.266,33.924,37.49)
! Load magnitudes (kN) from leftmost load in order follow.
p1=VEC(-619.76,-2146.2,-1607.2,-1534.5,-1534.5,-1474.7,-1381)
p8=VEC(-1476.7,-1444.8,-1534.5,-1534.5,-1474.7,-2052.6)
ks(1)=VEC(158689.85)*nk ! Vertical stiffnesses, left-right kN/m.
kr(1)=VEC(0)*nk ! Rotational stiffnesses, left-right kNm/rad.
#cc924.stk ! Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=nk+1 nj ;NUMBER OF MEMBERS nm=nj-1 nm nsg=1
NUMBER OF SUPPORTS nk ;NUMBER OF LOADINGS nls
JOINT COORDINATES ;i=0 b(nk)=b(nk-1)+can ;:10 ;i=i+1
i b(i-1) 0 SUPPORT ;IF i<nk GOTO 10 ;nj b(nk) 0
JOINT RELEASES ;i=0 ;:15 ;i=i+1 i FORCE Y ks(i) MOMENT Z kr(i)
IF i<nk GOTO 15 ;MEMBER INCIDENCES ;1 THRU nm RANGE 1,2 nk,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
ay=br*d*5/6 ;IF nu=1E-12 THEN ay=0
1 THRU nm AX br*d AY ay IZ iz=br*d^3/12 iz n=0 ;REPEAT ;n=n+1 m=0
LOADING TRAIN OF MOVING POINT LOADS (DOWN IS NEGATIVE) POSITION
MEMBER LOADS ;:200 ;lc=0 m=m+1 ;:300 ;lc=lc+1 x=a(lc)+(n-1)*si
IF x=0 THEN x=1E-6 ! Fix to get left end load on the span.
IF x>b(m-1) AND x<=b(m) THEN m FORCE Y CONCENTR P p(lc) L x-b(m-1)
IF lc<nl GOTO 300 ;IF m<nm GOTO 200 ;UNTIL n=nls ;ENDREPEAT ;SOLVE
FINISH
```

NL-STRESS data file before conversion.

```
TITLE GROUND BEAM ON ELASTIC PILES SUBJECTED
TITLE TO TRAIN OF MOVING LOADS; CHECKING OF RESULTS
TITLE AGAINST FLEXIBILITY USING SIMPSON'S RULE & FOX'S
TITLE METHOD FOR INVERSION; AND BY ENGINEERS' ARITHMETIC.
MADEBY DWB
DATE 08.02.05
REFNO vm131
METHOD ELASTIC JOINTS
PRINT DATA RESULTS FROM 1
TYPE PLANE FRAME
```



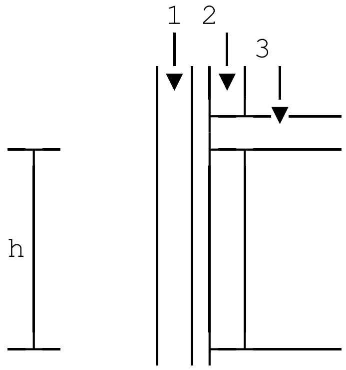

```
NUMBER OF JOINTS 14
NUMBER OF MEMBERS 13
NUMBER OF SUPPORTS 13
NUMBER OF LOADINGS 2
JOINT COORDINATES
1 0 0 SUPPORT
2 1.8288 0 SUPPORT
3 4.8768 0 SUPPORT
4 7.9248 0 SUPPORT
5 10.973 0 SUPPORT
6 14.021 0 SUPPORT
7 17.069 0 SUPPORT
8 20.117 0 SUPPORT
```

```
9 23.165 0 SUPPORT
10 26.213 0 SUPPORT
11 29.261 0 SUPPORT
12 32.309 0 SUPPORT
13 35.357 0 SUPPORT
14 36.881 0
JOINT RELEASES
1 FORCE Y 158690 MOMENT Z 0
2 FORCE Y 158690 MOMENT Z 0
3 FORCE Y 158690 MOMENT Z 0
4 FORCE Y 158690 MOMENT Z 0
5 FORCE Y 158690 MOMENT Z 0
6 FORCE Y 158690 MOMENT Z 0
7 FORCE Y 158690 MOMENT Z 0
8 FORCE Y 158690 MOMENT Z 0
9 FORCE Y 158690 MOMENT Z 0
10 FORCE Y 158690 MOMENT Z 0
11 FORCE Y 158690 MOMENT Z 0
12 FORCE Y 158690 MOMENT Z 0
13 FORCE Y 158690 MOMENT Z 0
MEMBER INCIDENCES
1 THRU 13 RANGE 1 2 13 14
CONSTANTS E 4.70986E9 ALL G 1.96244E9 ALL
MEMBER PROPERTIES
1 THRU 13 AX 0.5 AY 0.416667 IZ 0.0104167
LOADING TRAIN OF MOVING POINT LOADS (DOWN IS NEGATIVE)    POSITION
MEMBER LOADS
1 FORCE Y CONCENTR P -619.76 L 1E-6
2 FORCE Y CONCENTR P -2146.2 L 0.8138
3 FORCE Y CONCENTR P -1607.2 L 1.3106
4 FORCE Y CONCENTR P -1534.5 L 0.9388
5 FORCE Y CONCENTR P -1534.5 L 1.448
6 FORCE Y CONCENTR P -1474.7 L 1.067
7 FORCE Y CONCENTR P -1381 L 1.576
8 FORCE Y CONCENTR P -1476.7 L 1.371
9 FORCE Y CONCENTR P -1444.8 L 1.88
10 FORCE Y CONCENTR P -1534.5 L 1.499
11 FORCE Y CONCENTR P -1534.5 L 2.005
12 FORCE Y CONCENTR P -1474.7 L 1.615
LOADING TRAIN OF MOVING POINT LOADS (DOWN IS NEGATIVE)    POSITION
MEMBER LOADS
1 FORCE Y CONCENTR P -619.76 L 1
2 FORCE Y CONCENTR P -2146.2 L 1.8138
3 FORCE Y CONCENTR P -1607.2 L 2.3106
4 FORCE Y CONCENTR P -1534.5 L 1.9388
5 FORCE Y CONCENTR P -1534.5 L 2.448
6 FORCE Y CONCENTR P -1474.7 L 2.067
7 FORCE Y CONCENTR P -1381 L 2.576
8 FORCE Y CONCENTR P -1476.7 L 2.371
9 FORCE Y CONCENTR P -1444.8 L 2.88
10 FORCE Y CONCENTR P -1534.5 L 2.499
11 FORCE Y CONCENTR P -1534.5 L 3.005
12 FORCE Y CONCENTR P -1474.7 L 2.615
SOLVE
FINISH
```

NL-STRESS data file after conversion.

3.15 Cross referencing of variables

Consider the section of calculation which follows; the ** marks the start of a bug.




```
External cavity wall
____________________

This calculation is in accordance
with BS 5628-1:1992, the Code of
practice for use of masonry.
Part 1: Structural use of
unreinforced masonry.

Masonry details
_______________

Thickness of outer leaf              t(1)=100 mm
Density of outer leaf                den(1)=12.4 kN/m3
Thickness of inner leaf              t(2)=100 mm
Density of inner leaf                den(2)=12.4 kN/m3
Clear height of wall
between lateral restraints           h=2.425 m
Effective thick of cavity wall       tef=2*(t(1)+t(2))/3
                                        =2*(100+100)/3
                                        =133.33 mm
Support Restraint factor             rf=0.75
Effective height of wall             heff=rf*h=0.75*2.425=1.8188 m
Slenderness ratio                 SR=heff*1000/tef=1.8188*1000/133.33
                                    =13.641
Length of wall panel                 L=4 m      **
Mortar designation to table 1        mortar=3
Partial Safety Factor Material       gammam=3.1
```

Masonry wall calculation with a bug.

The ** marks a line which prompted the engineer for the length of the wall. For the calculation above, the engineer responded with 4 m, which was used to work out the area of the wall which was not used. Thus the length of the wall was not considered for assessing the wind load on the wall, but prompting for the length implied that the length was being taken into account. Two points arise in connection with the above:

- only an engineer familiar with masonry design could find this bug, a lay person would be likely to assume that the area calculated was at the end of a *chain*
- a facility for printing out a cross-reference for every variable with every other variable should help to locate such bugs.

A table listing every numerical variable with every other associated variable was found to be of limited value as it was necessary to keep looking through the proforma calculation to find where the associated variable was located and the context in which it was used. Thus, rather than displaying the names of associated variables, it was found preferable to list the line numbers where associated variables were located. This lead to the development of table 3.1 which shows an extract from the *occurrence of variables* table for proforma calculation sc075.pro as listed in appendix C. With any editor, it is straight forward to jump to any line numbers referenced and view both the associated variables and the context of association. To save having to jump to the line numbers referenced, each line was printed in the order following the list of line numbers, as in table 3.2. The final form of the table is that shown in table 3.3 for which the line numbers are printed at the start of each of the collected lines. For reason of space, tables 3.1 to 3.3 only contain variables *vc, vc04, vlim, vs* & *WtC.* To highlight these variables, they are included within , *e.g.* vlim .

Table 3.1 Occurrences of variables.

```
► vc ◄
1166   1171   1173   1178   1179   1181   1248   1254
► vc04 ◄
1248   1249   1250   1253
► vlim ◄
1149   1152   1179   1181
► vs ◄
327    328    329
► WtC ◄
684    727
```

Table 3.2 Collected lines containing variables.

```
► vc ◄
1166   1171   1173   1178   1179   1181   1248   1254
Design shear stress in concrete   vc=0.79*pcnt^(1/3)*f00d^.25/1.25 N/mm²
Modified design shear stress      vc=vc*(fcu/25)^(1/3) N/mm²
Modified design shear stress      vc=vc*(40/25)^(1/3) N/mm²
enhanced value of                 vc=vc*2*d/av N/mm² (see Cl.3.4.5.8).
IF vc>vlim
vc=vlim N/mm²
  vc04=vc+0.4
minimum spacing of links          sv=Asv*fyv/(gammaS*bv*(v-vc)) mm
► vc04 ◄
1248   1249   1250   1253
  vc04=vc+0.4
IF v<=vc04
As v ( v N/mm² ) does not exceed (vc+0.4) ( vc04 N/mm² ),
As v ( v N/mm² ) exceeds (vc+0.4) ( vc04 N/mm² ),
► vlim ◄
1149   1152   1179   1181
and thus limiting shear stress    vlim=0.8*SQR(fcu) N/mm²
and thus limiting shear stress    vlim=5 N/mm²
IF vc>vlim
vc=vlim N/mm²
► vs ◄
327    328    329
 dia2=dia/2 d=1.0E39 d=h-cover-dial-dia2 vs=hagg/1.5
IF dia>vs THEN vs=dia ENDIF
 d2=h-cover-dial-dia-vs/2
► WtC ◄
684    727
 WtT=.00785*Aspr WtC=.00785*As'pr
                                  Weight of steel provided WtC kg/m
```

Table 3.3 Cross referencing of variables.

```
► vc ◄
1166 Design shear stress in concrete   vc=0.79*pcnt^(1/3)*f00d^.25/1.25 N/mm²
1171 Modified design shear stress      vc=vc*(fcu/25)^(1/3) N/mm²
1173 Modified design shear stress      vc=vc*(40/25)^(1/3) N/mm²
```

```
1178 enhanced value of                vc=vc*2*d/av N/mm² (see Cl.3.4.5.8).
1179 IF vc>vlim
1181 vc=vlim N/mm²
1248   vc04=vc+0.4
1254 minimum spacing of links          sv=Asv*fyv/(gammaS*bv*(v-vc)) mm
► vc04 ◄
1248   vc04=vc+0.4
1249 IF v<=vc04
1250 As v ( v N/mm² ) does not exceed (vc+0.4) ( vc04 N/mm² ),
1253 As v ( v N/mm² ) exceeds (vc+0.4) ( vc04 N/mm² ),
► vlim ◄
1149 and thus limiting shear stress    vlim=0.8*SQR(fcu) N/mm²
1152 and thus limiting shear stress    vlim=5 N/mm²
1179 IF vc>vlim
1181 vc=vlim N/mm²
► vs ◄
327   dia2=dia/2 d=1.0E39 d=h-cover-dial-dia2 vs=hagg/1.5
328 IF dia>vs THEN vs=dia ENDIF
329   d2=h-cover-dial-dia-vs/2
► WtC ◄
684   WtT=.00785*Aspr WtC=.00785*As'pr
727                                    Weight of steel provided WtC kg/m
```

The most useful table from tables 3.1 to 3.3 was found to be table 3.3. As mentioned previously, each table is a short extract covering just 5 variables i.e. *vc* to *WtC* only. The full table contains 172 variables and 1387 lines *cf.* 2083 lines for the proforma calculation itself, see *sc075.pro* in appendix C. The cross referencing provides:

- an alternative perspective on a proforma calculation enabling all usages of each variable to be read together rather than spread over a thousand lines
- a table which may be checked *e.g.* to ensure that when just one line follows the variable enclosed within , then the line contains the *result*
- pinpointing unintentional reassignment of any variable
- identifying inconsistencies between assignments
- a means of getting to grips with the equations used
- the juxtaposition of the equations aids checking of units e.g. that when working in *mm* that lengths input in *metres* are multiplied by 1000
- assistance in the elimination of clones; Fanta & Rajlich (1999) Kasper & Godfrey (2006), section 3.2.

Of the help listed above, provided by the *cross referencing of variables*, a typical section of which is given in table 3.3, the most useful was found to be *a checking regime*.

This chapter commenced with a definition of terms in general use for verifying commercial software. Terms more familiar to structural engineers, which are used elsewhere in this thesis, are defined and discussed. Sections 3.3 to 3.15 give authentic examples of tools which support the process of *verifying the correctness of structural engineering software*.

Chapter 4

The nature of data

As shown in section 3.11, the production of say a thousand unique sets of data for extensively testing an engineering program, requires that the data be engineered, and that software be written to produce the sets of data. Before the sets of data can be produced, a classification of the various types of data is needed.

Building a set of a hundred self-checking models for verifying the correctness of a program for the structural analysis of frameworks, takes several man years. Before commencing on such a task, it is recommended that the software to be verified should be *benchmarked*. *Benchmarking* is comparing the results produced by a program with results which have been produced by at least one other program or with published results in text books and papers, see chapter 9.

Structural calculations include those for the structural analysis of a framework and those for the design of structural components: beams, slabs, columns *etc.* The set of data required for the structural analysis of a framework differs from the set of data required for the production of a set of structural calculations. Most items of data for a structural analysis program; whether integer *e.g.* for the number of joints, or real *e.g.* for applied loads, can vary uniformly over a wide range of values; furthermore within any set of data for a structural analysis, there is little dependency of any item of data on any other. A set of data for a structural design calculation differs to that for the analysis of a structural framework in that there is a high dependency on the items of data among themselves, many items of data being dependent on a code of practice, sometimes given in tables for which there is no sensible alternative to a table, sometimes requiring engineering judgment for example for the degree of quality control applied on site, appraisal of a load sharing factor, whether or not lateral restraints are provided at loading positions *etc.*

This chapter shows that the items of data required for the structural analysis of a framework have a higher degree of dependency among themselves than would be expected, and that a unified classification of types of data for both the structural analysis of frameworks and the structural design of components, is both worthwhile and achievable.

The phrase *engineer the problem* which is used in this chapter, means *develop a pragmatic system* after grasping the problem's salient features, or get to grips with the problem, Americans would say get down to the *nitty-gritty*, *i.e.* act in the way that engineers do throughout their life with every problem they come across. Theoreticians will frown at the system developed in this and the next chapter; they will launch into matters such as *periodicity* and *adequacy of coverage*, argue about the way that the type of data is classified, and so on. The two facts overriding all others are that:

- the automatic generation of sets of data for testing engineering software must be easy to specify and
- the data must be limited to practical ranges appropriate to each parameter alone and in combination with all other parameters which have a shared dependency.

The reader is asked to *go with the flow*; if after grasping the principles, the reader disagrees with, for example, the way that the *type* of data is classified, and considers that there is a simpler way, then he/she should develop their own system.

4.1 Data for structural analysis

It would appear to be straightforward to test a linear elastic structural analysis program for the robustness of its own logic by writing a program to pick random numbers from within preset ranges for: number of joints, number of members *etc.* and generate hundreds of files of data which an analysis program could run in batch mode. Such an approach would fail for the following reasons:

- some joints would not be connected into the framework
- injudicious joint and member releases would cause mechanisms
- injudicious selection of member properties would cause axial and shear deformation strain energy to swamp the bending strain energy
- injudicious positioning of loading on the members would cause loading applied to members to come off the end of members
- a randomly produced connectivity table would produce rubbish
- for the rare occasions when results would be produced, if they were plotted, it would be rare for them to look sensible *e.g.* a plot of deflections which looks like a spike could be due to an unsupported joint having a deflection of say 100 km when all the other deflections were less than a metre
- if the modulus of rigidity were very low in comparison to the modulus of elasticity, then the resulting deflection plot would give the impression that all the members were pinned at the joints rather than fixed *etc.*

For the above reasons, it is necessary to *engineer* every set of data before it can be run successfully by a model for the structural analysis of a framework. The test data should be realistic.

4.2 Data for structural design

As mentioned at the beginning of this chapter, structural design calculations have a high dependency of items of data on other items of data in the set of data for input to a structural design program, using structural timber for example:

- for a beam of width 50 mm, the minimum depth is typically 50 mm and the maximum depth is typically 300 mm
- for a simply supported beam of span 3.6 m, the minimum depth of the beam is typically 150 mm and the maximum depth is typically 300 mm
- permissible bending and tension stresses *pbat* parallel to the grain vary from typically 3 N/mm² to 20 N/mm²
- permissible compression stresses parallel to the grain vary from typically 69% of *pbat* to 84% of *pbat*
- permissible compression stresses perpendicular to the grain vary from typically 14% of *pbat* to 32% of *pbat*
- permissible shear stresses parallel to the grain vary from typically 9% of *pbat* to 14% of *pbat*.

Thus beam parameters: width, depth, span, permissible stresses *etc.* may not vary independently from each other, it follows that each set of data to be used for testing a program for the design of a structural timber component, must be engineered.

It is necessary to formalise the various types of data, and design a system for providing the dependencies between the parameters so that logic may be written for the automatic generation of sets of data which in turn may be used to test the logic of a model. This chapter gives examples of the different types of data required for the production of a set of design calculations or for the production of a set of data for the structural analysis of a framework.

It is not possible to devise sets of engineered data without incorporating logic such as "A>B" (Boole, 1847) named a Boolean expression after the English mathematician and logician George Boole. Praxis (1990), is shown throughout this document in Courier which has a fixed spacing which enables calculations to be lined up, for example:

```
The grade tension stresses apply to members assigned to a strength
class and having a width of 300 mm.  For other widths of members,
the grade tension stresses should be multiplied by the width
modification factor.
IF d>b
Largest section dimension       +h=d mm
ELSE
Largest section dimension       +h=b mm
ENDIF
IF h<=72
Width modification factor       +K14=1.17
ELSE
Width modification factor       +K14=(300/h)^0.11
ENDIF
```

There are two occurrences of the IF-ELSE-ENDIF programming structure in the above example of Praxis and four assignments commencing with a plus sign, which is a programming device, in this case to tell the computer that an assignment follows. Another programming structure is: REPEAT-UNTIL-ENDREPEAT; yet another programming device is a set of four question marks ???? to tell the computer to prompt the engineer for data. Generally such programming structures and devices are described as they are introduced. Lines commencing with keywords such as IF are omitted from the output calculations. Assignments are copied to the output calculations omitting the leading plus, optionally the right side of the assignment is repeated but with numerical values substituted for any *symbolic variables* (henceforth just *variables* for brevity), finally the new value assigned is shown followed by any units.

```
Width modification factor     K14=(300/h)^0.11
                                 =(300/350)^0.11
                                 =0.98319
```

When space permits, all three lines are concatenated onto one line *e.g.*

```
K14=(300/h)^0.11=(300/350)^0.11=0.98319
```

Appendix C includes a calculation for the design of a flanged reinforced concrete beam section in bending with optional shear, bar curtailment, lap length and span/effective-depth checks. The first calculation in appendix C is of average complexity and length (approximately 2000 lines); all of the examples in this section are taken from this calculation. Of course design calculations vary greatly, nevertheless the two typical calculations included in appendix C do give some idea of the variety of items of data in a set of data for component design. Text in the following examples, has been extracted from the calculation for the flanged reinforced concrete beam with simplification to the text where possible. There follows some of the different characteristics that an item of data for either the structural analysis of a framework, or the structural design of a component, may have.

4.3 Regular sets of integer data

```
Types of bar (see table 3.26 in Code) are as follows:
  0. Plain bars
  1. Type 1 deformed bars
  2. Type 2 deformed bars
Type of bar (0-2)                +Type=????
```

The engineer may respond to the ???? prompt with 0, 1 or 2. Other values will produce a warning and require the engineer to revise the data to be 0, 1 or 2 before proceeding. The characteristics of this set of data:

```
       the response must be   integer
                first value   0
                 last value   2
number of values in the set   3
```

These four characteristics are sufficient to define all values in the set. There is no requirement for the first value to be the minimum value, it is necessary that the software should be written to generate the 3 values required whether the range of integers is increasing or decreasing. Obviously *increasing* or *decreasing* for this range is a further characteristic but there is no need to specify it, as the direction of increase is established by the two end values of the range.

The commonest set of integer data is the response to Yes/No using the digits 1 or 0 respectively. The 1 and 0 have become popular on electric on/off switches, they have two advantages over Y & N:

- they are independent of language
- they do not require the user to question whether an upper or lower case response is required.

There are many prompts for a Yes/No answer, there are thirteen in the concrete example in appendix C, for example:

```
Comp.bars to control defln (1=Yes/0=No)  +ans5=????
```

Another example for a regular set of integer data, may be used to specify the set of data which can be given in response to the prompt:

```
Characteristic concrete strength +fcu=???? N/mm²
```

Although BS 8110 (BS 8110) does not restrict designed concrete mixes to a set of strengths, it gives guidance. In no circumstances may the concrete strength be less than 15 N/mm². The grade of concrete for reinforced concrete must not normally be less than 25 N/mm², but strengths down to 15 N/mm² may be used for concretes made with lightweight aggregates.

The characteristics of normal weight concrete strengths are:

```
       the response must be   integer
          name of parameter   fcu
                first value   25
                 last value   40
number of values in the set   4
               type of data   4
```

which will be used to generate concrete strengths: 25, 30, 35 & 40 N/mm². We can classify the *type* of data as 4 when we require four values to be specified between the first and last value of the range, with equal intervals between.

4.4 Irregular sets of integer data

An example of a small set of irregular integer data is that for reinforcing bar diameters viz: 6 8 10 12 16 20 25 32 40 & 50 mm. Steel section sizes such as Universal Beams, provide examples of large sets of irregular integer data. There are various strategies for dealing with small sets of irregular data such as that for reinforcing bars:

(a) Specify all the values in the set, give the set a reference number and provide a means of accessing all or part of the set as appropriate to the application.

(b) Provide a table (as in appendix C) and fault data which is not valid.

(c) *Engineer the problem.*

(a) Is cumbersome; as there are only a few sets of such data in any engineering discipline, it has been found simpler to have just one set containing the bar diameters, if other such sets are needed then they can be added, and values selected by reference to the location in the set of the first and last locations required for the application.

(b) Is already present in the *calculation*, a different strategy is preferable for checking.

(c) Bar diameters are used for different purposes, 6 8 10 & 12 cover the most popular choices for distribution steel; 12 16 20 & 25 cover the most popular choices for the main bars in slabs; 25 32 & 40 cover the most popular choices for the main bars in beams.

Each of these three sub-sets of data can be handled as for the regular sets of integer data.

```
Diameters 6 8 10 & 12 need to be defined
                     name of parameter   dia
                           first value   6
                            last value   12
          number of values in the set    4
                          type of data   4

Diameters 12 16 20 & 25 need to be defined
                       name of parameter   dia
                             first value   12
                              last value   25
            number of values in the set    4
                            type of data   -4
```

The selected sizes would be: 12 16.3 20.7 25 for type of data =4; to switch on integer rounding, we say the type of data =-4 which will specify bar diameters: 12 16 20 & 25 as required.

```
Diameters 25 32 & 40 need to be defined
                    name of parameter   dia
                          first value   25
                           last value   40
         number of values in the set    3
                         type of data   -3
```

The selected sizes would be: 25 32.5 & 40 for type of data =3, to switch on integer rounding, we say the type of data =-3 which will specify bars of diameter: 25 32 & 40 as required. When it is required that all or nearly all bar sizes are appropriate to a parameter, then in the system devised, the values should be saved in a vector za() and the first and last values should refer to the first and last locations in za().

```
Diameters 6 8 10 12 16 20 25 32 40 & 50 need to be specified.
                    name of parameter   dia
                          first value   3
                           last value   9
         number of values in the set    7
                         type of data   100
```

In the forgoing we have defined the type of data =100 as *looking up a set of values held in za()* and selecting a value from the third in the set *i.e.* 10, to the ninth *i.e.* 40, in the set inclusive. The selected sizes would be: 10 12 16 20 25 32 40, assuming that the assignment:
za(1)=VEC(6,8,10,12,16,20,25,40,50) which assigns za(1)=6, za(2)=8 and so on, has been made. It will be clear from the foregoing that for small sets of irregular integer data, with a bit of initiative, it is possible to engineer subsets from irregularly spaced sets and yet still keep the definition of the data simple *i.e.* specifying just: type of data, first & last value. Although the number of values has been declared in the foregoing, there is no need to declare it, as the number of values may be deduced from the *type of data*.

The system allows for 26 regularly occurring sets of data to be stored in za() to zz(), with access dependent on the *type of data* being selected by references 100 to 125.

For a large set of integer values *e.g.* rectangular hollow section sizes for stainless steel, a different strategy is needed, we need to call a procedure. It is desirable to keep the specification to the same structure as the forgoing so that a complete PARAMETER specification for any model can be contained in just one table.

```
A procedure named tri needs to be invoked
                                 name   tri
                          first value   3
                           last value   72
                         type of data   1E40
```

In the above, declaring the type of data =1E40 means that the name given is that of a procedure which is to be invoked, the first and last values being *arguments* for that procedure *i.e.* values passed to that procedure. The name *tri* is apposite to steel section sizes which generally have a section designation formed from: serial depth, breadth and mass/metre for open sections excluding angles; or serial depth, breadth and thickness for angles and closed sections with the exception of circular hollow sections. Section 5.12 describes the procedure **tri** and how it is included in the parameter table.

4.5 Sets of real values as data

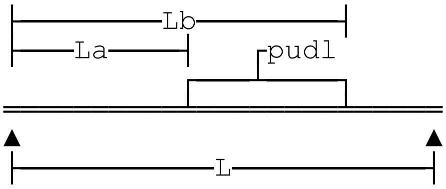
Although sets of real values in look-up tables are common in British Standards, generally sets of real values for input data are rare in *calculations*, reals as data items tend to be limited only by a range *e.g.* the prompt:

```
Moment before redistribution        +Mbef=???? kNm
```

invites the engineer to type a bending moment. Obviously a bending moment of 1E20 kNm would be unrealistic, so the engineer author of the *calculation* chooses values for the maximum and minimum bending moment and checks any input data for being within range. The choice of maximum and minimum bending moments depends on the section size, this introduces the subject of *dependency*, as the width and depth of a beam and the maximum permissible percentage of reinforcement fix the maximum permissible bending moment, some engineers' arithmetic is required. Although sets (as distinct from ranges) of real values are rare, they do occur, *e.g.* in timber K values (BS 5268), by using the method described in section 4.4 for reinforcing bar diameters, it is possible to engineer a set or subset of values, yet still keep the definition of the data simple *i.e.* specifying just: type of number, first & last values and number of values in the set.

4.6 Dependency

As an example of dependency of one item of data on another item of data, consider a simply supported beam of length *L*, subjected to a partial uniformly distributed *pudl*, which commences at a distance *La* from the left end of the beam and ends at a distance *Lb* from the left end.



Data required for a partial UDL.

```
Beam span                              +L=???? m
Partial uniformly distributed load +pudl=???? kN/m
Distance to start of UDL               +La=???? m
Distance to end of UDL                 +Lb=???? m
```

All four items of data are real, thus each can be defined by a start value, end value and the number of equally spaced values in the set, say 3 for this simple explanation but the number of values in a set may take any sensible value *e.g.* 166.

Variable	*Start & end values*		*The set of data*		
L	*1*	*10*	*1*	*5.5*	*10*
pudl	*-0.5*	*-20*	*-0.5*	*-10.25*	*-20*
La	*0*	*9.9*	*0*	*4.95*	*9.9*
Lb	*0.1*	*10*	*0.1*	*5.05*	*10*

To test for robustness of the logic, we must run every value in a set of data against every other value in every set for:

- both sets increasing
- one set increasing, the other decreasing.

Software to automatically generate sets of data containing all such combinations will generate invalid sets of data such as:

L	*La*	*Lb*	*Reasons for invalidity*
1	*4.95*	*10*	*La & Lb must come within span L*
5.5	*9.9*	*5.05*	*Lb must be greater that La*

For this simple example we need to restrict any values of *La* in each set to be less than say *L-0.1*, and to restrict any values of *Lb* to be greater than say *La+0.1* and less than or equal to *L*. Thus one parameter is dependent on the value of another parameter. Even for this simple example, it is evident that, to avoid generating invalid sets of data, *expressions* must be provided rather than *numerical values*.

4.7 Subscripted variables

In all but the simplest of calculations, there will be several occurrences of loading data, requiring variables to be subscripted *e.g.* La(1), La(2) *etc.* with the calculation containing subscripted variable names such as: La(i), where i takes values from zero upwards.

For the previous example, we can write:

```
IF La(i)>=L-0.1
STOP La(i) must be less than L-0.1 i.e. < +L-0.1 m
ENDIF
```

which, for *L=5.5* & *La(i)=9.9*, would cause display of the message:

```
La(i) must be less than L-0.1 i.e. < 5.4 m
```

and then stop.

For a general procedure which can be used for checking the logic of many different *calculations*, the procedure must be able to accept data in the form of expressions such as: *La(i)+0.1* To explain the significance of this statement, requires that the difference between *compilers* and *interpreters* be explained.

4.8 Compilers vs. interpreters

In a Fortran computer program which has been compiled, we can have a statement such as:

```
READ(*)A
```

which will read a value **A** typed on the keyboard, say 2.3026. As the program has already been compiled, the location for the storage of **A** will already have been assigned and therefore the numerical value read from the keyboard can be stored at the location reserved for the storage of **A**. In the program we may have an assignment such as:

```
B=2*A
```

which will cause the current value of **A** and the constant **2** to be copied to the *arithmetic unit* of the computer for multiplication, following which the result =4.6052, will be stored in the location reserved for **B**.

In a similar manner a string of characters may be read from the keyboard and stored *e.g.*

```
READ(*)KS
```

assuming KS has been declared as a character string, then characters **B=2*A** may be stored in the location reserved for KS, but this will not carry out the assignment implied, it will merely store the characters.

Praxis (1990), the notation used for writing SCALE proforma calculations does not have the limitations of a compiled program, for Praxis *interprets* the data it receives at *run-time i.e.* when the program is being run. When Praxis finds a prompt *e.g.* **+A=????** the engineer may respond with an expression of any complexity. The expression will be evaluated, the result stored in the numerical stack of values, which is associated with the stack of names of the variables. If the left side of the assignment *i.e.* **A**, is not already on the *stack*, then it is added. A single procedure named sc924.pro, currently containing a modest 944 lines, builds all the sets of data required for verifying models for the structural analysis of frameworks and for the structural design of components.

The line in sc924.pro which builds the parameter assignment follows:

```
! Build assignment +$32000= +$(zp'+27000) = +zva(zp')
```

In the above line, zp' is the current parameter number, in the example above for the data for a partial UDL, the 3'rd parameter is **La** which is read from a parameter table and saved in a string having the parameter number plus 27000, *i.e.* $(27003) holds **La**. The current value of **La** will have been calculated from the limits imposed on **La** and its value will have been stored in zva(zp') *i.e.* zva(3), we will assume zva(3)=1.2. The *Build assignment* line starts with an exclamation mark which is interpreted as *evaluate that which follows the exclamation mark but do not write anything to the output*. String $32000 is a special string which substitutes **La** for **$(27003)** and **1.2** for **zva(3)** and concatenates the text, after the assignment $32000 contains **La=1.2**, which is subsequently added to a set of data to be run by the required model to be tested.

Summarising, the ability of Praxis (1990) to accept expressions as data, means that one proforma calculation may be written to read numbers and expressions from a table and build assignments so that hundreds of sets of data may be generated and run automatically by the model to be tested. Although compiled languages allow the input, manipulation and output of *strings* (strings of characters), they do not permit the assignment of a string (which itself varies) to a numerical variable. In a compiled language we may write: Lb(i)=La(i)+0.1 which if La(i)=5.5, would assign the value of 5.6 to the variable Lb(i), but we may not write: Lb(i)='La(i)+0.1' where the single quotes denote that a character string is enclosed. A compiled language cannot evaluate an expression which has been read in as a text string at run time, for the location of the variables in that expression will not be known.

An engineer sitting at a computer to design a multi-storey frame, can prepare a figure, number up the joints and members, provide dimensions to the beams and columns, and type in typically a hundred items of data required. The results of running the analysis must then be verified. Alternatively a model for multi-storey frame may be built parametrically so that instead of typing in hundreds of numbers, the engineer need only type typically a dozen values to provide the parameters; because a self check is included in the model, inspection of the percentage difference between the model's results and its self check, provides verification. It is the ability of NL-STRESS and Praxis to process expressions as data which enables tabular descriptions of parameters to be produced from which a thousand or more sets of data may be produced and run with a self check to confirm that the model has been verified.

Chapter 5

Logic to check logic

It is not possible to discuss modern engineering calculations without mentioning logic. Praxis (1990), a language for writing proforma calculations, was developed by Alcock & Brown in 1984, the software was written by the writer. As this research is about verifying the correctness of *calculations* rather than programming, a minimum of programming structures and devices are included. Considerations such as: formatting, fonts, printer control, pop-up help, screen display, saving data, building sets of output calculations into a document, accessing files, passing *arguments* to procedures, system calls *etc.* are all omitted.

Praxis (1990) is used throughout this research as: it avoids the need for writing dialogues such as: click on this and that; Praxis is specifically designed for the production of structural engineering calculations; the *calculations* can be written, read and used by engineers; the *calculations* do not need a computer specialist for their maintenance. A brief introduction to Praxis, follows.

5.1 Special character usage

! means do not copy the rest of the line (or the !) to the output calculations, but carry out any assignments contained in the line.
? (conventionally ????) means display the line and wait for the engineer's response, which then replaces the ????
+ says the word which follows is usually the name of a variable but may be a constant at the start of an expression *e.g.* +12*b. This name may stand alone or be part of an assignment. Examples are +a +a*b +b=12.5 +I=b*d^3/12 +e=TABLE(26,Grade) +a1=VEC(1.0,1.2,1.6).

5.2 Expressions

Expressions comprise terms; each term may be a number, the name of a variable, or a function. The terms are bound together with operators ^ * / + -. Operators are shown in order of precedence, the order can be overridden by the use of brackets: 2*3+4 is 10, 2*(3+4) is 14.

5.3 Functions

ABS Absolute value, INT Integral part, SQR Square root, LOG Natural log,
EXP Natural exponent, DEG Degrees from radians, RAD Radians from degrees.
RAN Random No., SGN 1, 0, -1 if positive, zero, negative respectively.
SIN,COS,TAN Sine, Cosine & Tangent respectively of an angle in radians.
ASN, ACS, ATN Arcsine, Arccosine & Arctangent respectively.
SNH, CSH, TNH Hyperbolic sine, cosine & tangent respectively.

Other functions can be created from the above *e.g.* to find the remainder when *a* is divided by *b* use +mod=a-(INT(a/b)*b). For log to base 10 of 2 say, use: +logof2 =LOG(2)*0.4342945 =0.69315*0.4342945 =0.30103. To convert between EXP & antilog base 10 reverse above thus +alg=EXP(0.30103/0.4342945) =EXP(0.69315) =2.

5.4 Storage of data

The dollar sign with an integer suffix *e.g.* $43 or subscript *e.g.* $(a) is used as a variable name for storing a string of characters *e.g.* assuming a=123 then +$(a)=*Electrodes to comply with BS 639 grade E43* will store *Electrodes to comply with BS 639 grade E43* in $(123); for brevity $123 may be written in place of $(123).

VEC is short for VECtor, where vector is used in the programming sense rather than the mathematical sense, *e.g.* +a(12)=VEC(3.2,b,-5.7)*n causes a(12) to be assigned the first value =3.2, a(13) the second =b, a(14) the third =-5.7, the optional *n for n=2 would cause the assignments to be continued for a second time thus a(15)=3.2, a(16)=b, a(17)=-5.7. When the VEC function is printed in the output calculations, a minor rearrangement takes place to make interpretation more intuitive
e.g. +dia(1)=VEC(6,8,10,12,16,20,25,32,40,50) is printed
as: dia(1) *etc.*=(6,8,10,12,16,20,25,32,40,50).

Tables of values may be stored by: STORE *reference rows columns* where *reference* is the reference number for the table, *rows* is the number of rows, *columns* is the number of columns. Follow with a line of column headings, omit if only one column; follow with rows of values, preceding each with a numerical heading, omit this heading if only one row. To make the computer look up a value, include in an assignment: TABLE(ref,row,col) where *ref* is the table reference number, *row* is the row heading, *col* is the column heading; all may be integer or real values or variables holding integer or real values. For a table with one row or one column, leave out the corresponding 1. The row or column headings need not be matched precisely, the looked up value is established by linear interpolation.

5.5 Control

Stop normally:

```
STOP  message
```

stops the *calculation* and outputs the optional *message.*

Process conditionally:

```
IF condition/s
lines
ELSE
alternative lines
ENDIF
```

First evaluates *condition/s*: true or false. If *true*, process *lines*, ignore *alternative lines*. If false, ignore *lines*, deal only with *alternative lines*.

ELSE *alternative lines* may be omitted where nothing is to be done when *condition/s* evaluates to false.

Repeat lines:

```
REPEAT
lines
UNTIL condition/s
lines
ENDREPEAT
```

Processes all lines between REPEAT and ENDREPEAT again and again. If, on evaluating *condition/s*, the result is true, leave the loop and process the line following ENDREPEAT.

Procedures:

```
DEFINE name
lines
ENDDEFINE
```

Whenever *name* appears at the start of a line in the *calculations*, substitute and process *lines*. Procedures should be placed at the end of a set of *calculations*, between STOP and FINISH.

5.6 Devising sets of test data

It will be clear from the classification of data in chapter 4, that no single *switch* can be set to test the logic of a model for the design of a structural component or a model for the analysis of a structural framework, as the testing procedure needs to take into account the *type*, *range* and *dependency* of each variable. The remainder of this chapter describes how to prepare the data using simple examples; once the data has been prepared, a standard procedure may be invoked to test the logic of a model for robustness. Experience in maintaining the parametric description of a model when written using Praxis (1990), see table A.1 in appendix A, is that a tabular format is preferable. It has been found simpler to prepare the data as a table and let a standard procedure convert the parametric description data to the form required. The tabular form of the data is identical, whether it has been prepared for testing a model for a structural component or a structural framework.

5.7 Example calculation

Let us imagine the following five lines constitute a complete *calculation* which we wish to test for all combinations of data:

```
Default values (1=Yes, 0=No)          +ans=????
Axial load                            +P=???? kN
Continuous/not-continuous (1 or 0)    +con=????
Bar diameter (12,16,20,25)            +dia=???? mm
Exposure condition (1 to 5)           +xpo=????
```

Obviously we could run the *calculation* and type in each bar diameter, for each continuous or not-continuous; for each exposure conditions 1 to 5. Such an exercise would take a considerable amount of dedication when the data is not just 5 numbers, as above, but extends to say 50 numbers. The process of providing 1000 sets of data for a *calculation* containing 50 prompts requires 50000 items of data to be *coded*. It is considered that such an exercise would be too onerous for most engineers, so a shorthand method has to be devised. Usually engineers prefer *examples* to *syntax & formality*, so there now follows a discourse on the preparation of data for the example given above.

It is convenient to include the data for generating the sets of test data within the *calculation* itself. To avoid conflict with names of variables used in the *calculation*, each numeric variable will commence with a *z*. Four types of string are required:

- storage of parameter names: ans,P,con,dia,xpo in $27001 to $27005 say
- storage of expressions giving minimums, where required, in $28001 to $28005
- storage of expressions giving maximums, where required, in $29001 to $29005
- storage of expressions giving overriding expressions in $30001 to $30005.

The last set of strings is for expressions which are totally dependent on previously defined parameters *e.g.* in a reinforced concrete beam having just one layer of tension reinforcement, the effective depth **d** will be defined as **h-cov-20** where **h** is the overall depth, **cov** is the cover, and 20 is an allowance for bar diameters up to 40 mm. It would be bad engineering to vary the effective depth over the range of the overall depth.

First we need to specify the number of increments for each variable; this can vary between 2 and 166 so that the product of number of increments and 6 patterns (section 5.8) of loading is below 1000, thereby keeping the number of runs in each batch manageable. For this simple explanation take the number of runs as three *i.e.* the first & last value and one in the middle thus:

```
! +zni=3
```

Next specify the number of parameters, for the example there are five parameters viz: ans, P, con, dia, xpo, thus:

```
! +znp=5
```

Next provide the symbolic names for all the prompts thus:

```
! +$27001=ans +$27002=P +$27003=con +$27004=dia +$27005=xpo
```

Next provide a start and end numerical value for each prompt thus:

```
! +zst(1)=VEC(0,1,1,12,1) +zen(1)=VEC(0,400,0,25,5)
```

which says that the start & end value for the first parameter are both zero; for the second parameter the start and end values for the axial load are 1 kN and 400 kN; for the third parameter the start and end values are 1 and zero; for the fourth parameter the start and end values are 12 mm & 25 mm; for the fifth parameter the start and end values are 1 and 5.

So far we have coded data to say that: the name of the first variable is *ans* which starts and ends with zero, the name of the second variable is *P* which starts at 1 and ends at 400; the name of the third variable is *con* which starts at 1 and ends at 0; the name of the fourth variable is *dia* which starts at 12 and ends at 25; the name of the fifth variable is *xpo* which starts at 1 and ends at 5. Although this is not the complete story, the exercise has been made straightforward.

Next we need to say how each parameter may vary between its start value and its end value so that the procedure which builds the sets of data to be run by the model, can *engineer* the sets of data such that the data will be appropriate. Some parameters may vary uniformly *e.g.* axial load, other parameters must be constrained to be integer *e.g.* continuous/not-continuous can only take values 1 or 0, reinforcing bar diameters have to be integer and have to be obtainable, although 18 is an integer number, bars of diameter 18 mm are not available, and so on. The word *code* is ambiguous, so we will call the definition of how the value of a parameter may vary its **type**. The **type** is held in the array zty(1:znp), where znp is the number of parameters in a model.

The first prompt in the example calculation asks if the engineer wishes to be offered a set of *default values*, default values are useful the first time a *calculation* is run as they allow the engineer to see what the *calculation* does without the need to worry about the data. For subsequent runs, it is usual to respond to the prompt with zero to use the engineer's own data or automatically generated sets of data for which the variable *ans* should contain zero whatever the number of increments. For the first variable *ans*, by interpolation, three increments of: 0, 0, 0 would be sensible, we can say that the **type** of the variable *ans* =0 (aide-memoire 0=Ordinary interpolation).

For the second variable *P*, by interpolation, three increments of: 1 kN, 200.5 kN, & 400 kN would be sensible, so we could again say that its type =0, but the production of a set of test data loads to the nearest kN will be more convenient for the engineer, so we will say that the type of variable *P* =1 (aide-memoire 1=Integer); by linear interpolation with Integer rounding, three increments of: 1kN, 200kN, 400kN would be generated.

For the third variable *con*, by interpolation, three increments of: 1, 0.5, 0 would not be sensible, but if we say that its type=2 (aide-memoire 2, or more, means cycle as in bi-cycle), cycling the 2 values 1 & 0 would be sensible, so for three increments, **con** would take the values: 1,0,1. For seven increments *con* would take the values: 1,0,1,0,1,0,1.

For the fourth variable *dia*, by interpolation, three increments of: 12mm, 18.5mm, 25 mm would not be sensible as 18.5 mm bar diameters do not exist; but if we say that its type=4 (*i.e.* cycle 4 values), bar diameters of: 12mm, 16.33mm, 20.67mm, 25mm would be computed and would be sensible if their integer value were taken for each bar diameter; we can say its type=-4, where the 4 says cycle just 4 values and the minus tells the program to round the 4 values to integers. (Bar diameters 12mm to 25mm are a sensible range for the main steel of a suspended reinforced concrete floor slab.)

For the fifth variable *xpo*, by interpolation, three increments of: 1, 3 & 5 would be OK but not thorough, if we say that its type=5, cycling the values 1,2,3,4,5 would be sensible, but with only 3 increments, only exposure conditions 1,2,3 would be considered. The implication of five exposure conditions is that we must increase the number of increments *zni* to at least 5 to test for exposure conditions 4 & 5, but this is not essential as later described in *Patterns of variation*, section 5.8. The following will suffice for describing the type for the 5 parameters in this example:

```
! +zty(1)=VEC(0,1,2,-4,5)
```

To those who would enquire *why not use different increments for each variable and test all increments for all variables?* The response is that the choice of three increments, used in the above explanation, is only to describe what is going on; to verify each model, 166 increments are normally specified, as the subcycling: 2, 4 & 5 respectively for *con, dia* & *xpo* is very small in comparison to 166, each parameter will be tested many times over its full range, furthermore each of the 166 increments is tested in combination with 6 patterns of variation, producing 166*6=996 sets of data, to ensure that every parameter on a rising range of values is tested against every other parameter on both a rising and falling range.

Reinforcing bar diameters may be considered separately for beams and slabs, for large beams use zst()=25mm zen()=40mm with zty()=-3 which will compute bar diameters 25, 32 & 40 mm diameters (the -3 means cycle 3 values making each value an integer). For smaller beams use zst()=16 zen()=25mm with zty(n)=-3 which will compute bar diameters 16, 20 & 25mm.

As an alternative to selecting bar diameters as described above, there will be occasions where it is desirable to include all the bar diameters *e.g.* in the testing of bar scheduling in accordance with BS 8666. To do this, type=100 means interpret the zst() & zen() as pointers to the start and end elements in the special set za() containing:

+za(1)=VEC(6,8,10,12,16,20,25,32,40,50..) thus if zst()=3, zen()=8 & zty()=100, bar diameters 10mm to 32mm will be used in a set of 8-3+1=6 values and cycled.

There are occasions when it is desirable to round computed values to the nearest 5 or 10 or whatever; *e.g.* in response to the prompt:

```
Characteristic concrete strength   +fcu=???? N/mm²
```

although BS 8110 does not prohibit concrete strengths of say 43 N/mm², concrete suppliers normally supply strengths from 20 to 50 N/mm² in steps of 10 N/mm², these strengths can be specified in the test data by setting the type to 4 which will cause strengths of 20 30 40 50 N/mm² to be cycled, therefore if the engineer has set the number of increments to 7, the test data will be: 20 30 40 50 20 30 40. There are occasions when the engineer may require that the concrete strength be increased uniformly from the first increment to the last; this can be achieved by specifying the type as: 200 to round to the nearest 2, 300 to round to the nearest 3 *etc.* 1000 to round to the nearest 10 and so on. Thus if the engineer specifies the type as 1000, for the number of increments =7, the strengths would be prorated initially to: 20 25 30 35 40 45 50, then rounded to the nearest 10 as: 20 30 30 40 40 50 50.

To those who would enquire *how is periodicity avoided* e.g. *when two parameters of type=3 are included in the same parameter table?* The response is that: instances 1 2 3 of parameter A will be tested with instances 1 2 3 of parameter B and also 3 2 1 of parameter B; furthermore inspection of the parameter tables for 108 verified models for the structural analysis of frameworks in Appendix A, shows that when parameters have the same *type*, then periodicity does not affect any of the logic within the model. For models for the structural design of components, which exhibit periodicity, judicious engineering *i.e.* changing the number of instances for say parameter A, can be used to avoid the problem of periodicity. The writer feels sure that changes will be made to the system invented for verification, and *solution space* theories will be developed.

Summarising: the vector zty(1:znp) holds a number referred to as *type* for each of *znp* parameters, which describes how intermediate values between the start *zst(1)* to *zst(znp)* and end *zen(1)* to *zen(znp)* values for each parameter are formed; in other words *type* is a reference for modifying linear interpolation between the start and end values of each parameter.

0 means no modification *i.e.* zst(n)-zen(n) in zni equal increments for parameter *n*.

1 means as for 0, but make all values integer (aide-memoire by 1=I).

2 means split into 2 increments and cycle with the 2 increments; similarly 3 means cycle with 3 increments, and so on. When it is required to both cycle and make all values integer, then *type* is -2,-3 *etc.*

100 means interpret the zst(n) & zen(n) as pointers to the start and end element in the za() set, if +za(1)=VEC(6,8,10,12,16,20,25,32,40,50) and if zst(n)=3, zen(n)=8 & zty(n)=100 then bar diameters 10mm to 32mm will be used as a set of 8-3+1=6 values to be cycled. Types 101 to 125 are similar to 100 but refer to vectors zb() to zz() respectively.

126 means store the numbers following the 126 and use them *e.g.* if zst=1 and zen=5 and the numbers following the *type* are 20 7 20 23 26, then rather than assigning values within the range 1 to 5, values 20 7 20 23 26 are assigned to the parameter, thus *type=126* provides a means of accessing an irregular set of integer or real numbers.

200 means make the parameter value exactly divisible by 2; when the type=300, it is taken as an instruction to make the parameter value exactly divisible by 3; when the type=1000, it is taken as an instruction to make the parameter value exactly divisible by 10; and so on up to 20000 which is taken as an instruction to make the parameter value exactly divisible by 200.

5.8 Patterns of variation

The foregoing treatment will produce test data for running every parameter over a range of values specified by the engineer; but does not run the initial range of one parameter against the reversed range of another parameter. As an example, a rectangular beam of width *b* range 200-800mm, and depth *d* range 200-800mm, with the number of increments zni=7, will run a width of 200mm with a depth of 200mm, width of 300mm with a depth of 300mm, and so on. To run a width of 200mm with a depth of 800mm, or vice-versa, the procedure needs to reverse the start and end values of every single parameter with respect to every other parameter. This can be achieved by having a pattern for direction of the range, *cf.* a pattern for considering live load on a continuous beam.

For five variables, a pattern of 1 1 1 1 1 means that all ranges go in the initial direction, where we define the initial direction as that specified by the engineer when setting zst(n) & zen(n), respectively the start and end values of the range for the n'th parameter. In the example in section 5.7, *dia* increases whereas *con* decreases (starts at 1 for continuous, and ends at 0 for non-continuous), but as the engineer has specified them, we say that 1 represents the start value for *con* and 0 represents the end value. A pattern of 1 0 1 0 1 means that even numbered parameters go in the reverse direction.

To run each parameter with its reverse, a mathematician would use a *unity* matrix, such as that shown below:

1 0 0 0 0	where the first row says run the Initial value of the
0 1 0 0 0	first parameter with the reverse of parameters 2 to 5;
0 0 1 0 0	the second row says run the initial value of the second
0 0 0 1 0	parameter with the reverse of the first and parameters
0 0 0 0 1	3 to 5; and so on. Five parameters would require five
	patterns additional to the basic 1 1 1 1 1 ...

It follows that fifty parameters would require fifty additional patterns. Engineers *engineer* problems by opting for simplicity, practicality at the expense of rigour *e.g.* using one depth for beams, rather than 16 different depths were the beams to be designed for 16 different sets of bending moments, shear forces and deflections. It is desirable to:

- limit the number of runs to avoid producing over a million pages of calculations
- run the initial range of each parameter with the reverse of every other parameter
- run the initial range of each parameter with the reverse of every pair of parameters
- run the initial range of a pair of parameters with the reverse of every other parameter
- and so on.

It is also desirable to keep the engineering to the forefront for:

- one parameter e.g. bending moment, may be increasing whereas another e.g. exposure condition may be decreasing, the former is non-cyclic, the latter is cyclic
- the overall depth of a concrete beam may be increasing but the effective depth of the beam should not decrease
- the distance to the start of a partial load on a span may be increasing, but the distance to the end should not decrease such that the udl due to gravity is flipped
- and so on.

Considerations need to be given to patterns of variation which do not loose sight of the engineering. For simplicity let us assume that we require a minimum number of patterns which will guarantee that every parameter which is increasing, need only be considered with every parameter which is decreasing, and that a minimum number of patterns is required.

Parameter No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	...
Pattern 1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	...
Pattern 2	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	...
Pattern 3	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	...

Reading pattern 1 in association with the parameter numbers above, where 1 indicates a parameter is going in the original direction, 0 indicates the parameter is reversing, adjacent parameters go in opposite directions, parameters which step by odd numbers *e.g.* 1 4 7 10 13... 2 5 8 11 14... go in opposite directions as do 1 6 11 16... 2 7 12 17... 1 8 15... 2 9 16... and so on. An easy way to check the veracity of this is to use the first & little finger of the left hand as pointers to the figures in pattern 1 and check the parameter numbers above. By mathematical induction, pattern 1 shows that all parameters which have an odd step between, go in opposite directions. For patterns which have an even step between, repeat the exercise but reading patterns 2 & 3 in conjunction. Again by mathematical induction, patterns 2 & 3, read in conjunction for parameter numbers having an even step between, can provide opposite directions. When pattern 2 does not provide opposite directions, then pattern 3 will. To provide opposite directions for a parameter step of 2 *i.e.* parameter Nos. 1,3,5,7,9,11... pattern Nos. are: 2 2 3 2 2 3 2 2 3 2... For step equals 4 *i.e.* parameter Nos. 1,5,9,13,17... pattern Nos. are: 3 2 2 3 2 2...

Patterns 2 to 3 above, do not quite cover for the case of adjacent spans going in the original direction *e.g.* parameter numbers 3 & 4 always go in opposite directions. Obviously patterns 2 and 3 are the same but out of alignment by 1 parameter number, it seems sensible to add another pattern, shifted by one parameter to complete the set of 3 patterns of type: 1 1 0 1 1 0 ... which will provide for all adjacent pairs of parameters going in the original direction, prefaced & followed by parameters going in the reverse direction. Pattern 1 only considers even parameter numbers going in the original direction with odd parameter numbers going in the reverse direction. To provide for the case when adjacent parameters are of different types, the opposite to pattern 1 will be added, *i.e.* odd parameters going in the original direction. When building the definition table for parameters, the engineer has control over the primary direction of each parameter. All the patterns discussed above, are for varying the engineer's original choice of direction. For good *functionality*, a pattern which keeps to the original directions specified by the engineer will be added. Collecting all six patterns together and rearranging them into the familiar order for dealing with dead & live (imposed) load patterns on a continuous beam, we have:

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ... The 6 patterns shown to the left will cover
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 ... for all parameters ranging over the original
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 ... direction specified, and for each parameter
1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 ... considered with the reverse of every other
0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 ... parameter, and for all sets of: 1 1 0 1 1 0
1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 ... patterns where 1=initial direction, 0=reverse.

Number theory and *coverage* are important when devising patterns for engineering systems, but engineering is of paramount importance when dealing with sets of data for engineering models which are beset with discontinuities and interdependency between the parameters.

5.9 Dependency conditions

So far we have developed a compact treatment to define the start and end for a range of values for each and every parameter for the structural analysis of a framework or the structural design of a component; we have defined how each parameter may vary within its range, classifying it as a **type** 0, 1, 2 *etc.*; we have provided a system for dealing with the small percentage of parameters which have irregular ranges; we have identified patterns to be applied to vary each parameter going from the start of its range to the end of its range in association with every other parameter going from the end of its range to the start. From inspection of the parameter tables for the 108 verified models in Appendix A, approximately 20% of the parameters are dependent on the current values of one or more other parameters. As an example of dependency, consider the reinforced concrete Tee beam shown in figure 5.1.

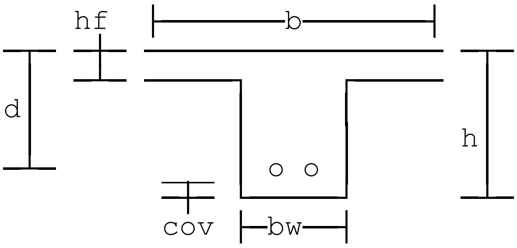


Figure 5.1 Tee beam.

```
Dependency conditions
bw>=b/5 bw<=b/2
hf>=h/5 hf<=d/2
     d=h-cov-20
Minimums Maximums
stored in stored in
$(28000+n) $(29000+n)
Specials stored in $(30000+n).
where n=parameter No. & current
parameter value is zva(n).
```

Firstly a table for defining the parameter specification will be given showing the symbolic names of computer variables used throughout this research, then the final form of the table which was adopted in this research will be presented. As shown in section 4.8, character expressions such *=h-cov-20* cannot be stored as an integer or real number, they must be stored as a string of values, Praxis (1990) uses a *dollar* sign to say that the symbolic name of a *variable* is that for a *character string*, where *variable* means the string of characters *held* in the symbolic name, may vary; where *held* means *associated with*. (Fortunately engineers are familiar with the concept of symbolic names metaphorically *holding or storing values*, an early example being Pythagoras' *a^2=b^2+c^2* where *a* is the length of the hypotenuse of a right angled triangle.) The string numbering used for dependency conditions is defined in section 5.7. ***z*** at the start of array zst() *etc.* is to distinguish the name of the variable from other variable names used in each model which contains a parameter table.

Table 5.1 Parameter table with dependencies *coded.*

```
                          The end value
The start value           for the parameter
for the parameter                    The type of parameter, 1=integer,
                                           500 means divisible by 5.
Parameter Name zst() zen()  zty()  Dependency
number                             conditions
   1      cov    20     75    500     Make d equal to
   2      h     200   3000    500      following expression
   3      d     150   2900      1  $30003=zva(2)-zva(1)-20
   4      b     300   3000    500
   5      bw    150   1500    500  $28005=zva(4)/5 $29005=zva(4)/2
   6      hf     75   1450    500  $28006=zva(2)/5 $29006=zva(3)/2

                  Make hf greater than     Make hf less than
                  following expression     following expression
```

In the table above, *cov* varies from zst(1)=20 to zen(1)=75mm, zty(1)=500 rounds the current increment to 5mm; *h* varies from zst(2)=200 to zen(2)=3000mm; *d* is computed from the special formula stored in $30003 which takes the current value of *h i.e.* zva(2), subtracts the current value of *cov i.e.* zva(1) and subtracts 20mm then checks that it is in the range zst(3) to zen(3) adjusting for compliance; *b* is computed in the range zst(4)=300 to zen(4)=3000 according to the increment number, then rounding to the nearest 5mm; *bw* is computed in the range zst(5)=150 to zen(5)=1500 according to the increment number, then rounding to the nearest 5mm then checks that it is in the range zva(4)/5 to zva(4)/2 adjusting for compliance; *hf* is computed in the range zst(6)=75 to zen(6)=1450 according to the increment number, then rounding to the nearest 5mm, then checks that it is in the range zva(2)/5 to zva(3)/2 adjusting for compliance. When the foregoing data is run by a standard procedure, the following sets of test values are produced, rearranged below for clarity. The table below includes 6 patterns and 4 increments, all values for the parameters are given as numbers. Table 5.2 shows the sets of test data when all *dependencies* are suppressed, table 5.3 includes the *dependencies*. Errors caused by ignoring the dependencies are shown *starred*. Were 3 increments chosen then the test data for Increment 2 would show that all six patterns are identical, this follows from the fact that with just three increments, the middle increment values will always be computed as average of the start and end values whatever the direction of parameter increase. In a proper test for verifying the correctness of the logic of a *calculation*, the engineer who is carrying out the verification, when he/she is preparing the parameter definition table, would normally set the number of increments to be greater than 100.

Table 5.2 Sets of test data ignoring dependencies.

	INCR 1	INCR 2	INCR 3	INCR 4
	SET 1	SET 2	SET 3	SET 4
	cov=20	cov=40	cov=55	cov=75
	h=200	h=1135	h=2065	h=3000
PATTERN	d=150	d=1066	d=1983	d=2900
1	b=300	b=1200	b=2100	b=3000
	bw=150	bw=600	bw=1050	bw=1500
	hf=75	hf=535	hf=990	hf=1450
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 5	SET 6	SET 7	SET 8
	cov=20	cov=40	cov=55	cov=75
	h=3000	h=2065	h=1135	h=200
PATTERN	d=150*	d=1066	d=1983*	d=2900*
2	b=3000	b=2100	b=1200	b=300
	bw=150*	bw=600	bw=1050*	bw=1500*
	hf=1450	hf=990	hf=535	hf=75
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 9	SET 10	SET 11	SET 12
	cov=75	cov=55	cov=40	cov=20
	h=200	h=1135	h=2065	h=3000
PATTERN	d=2900*	d=1983*	d=1066*	d=150*
3	b=300	b=1200	b=2100	b=3000
	bw=1500*	bw=1050	bw=600	bw=150*
	hf=75	hf=535	hf=990	hf=1450
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 13	SET 14	SET 15	SET 16
	cov=20	cov=40	cov=55	cov=75
	h=200	h=1135	h=2065	h=3000
PATTERN	d=2900*	d=1983*	d=1066*	d=150*
4	b=300	b=1200	b=2100	b=3000
	bw=150	bw=600	bw=1050	bw=1500
	hf=1450*	hf=990*	hf=535	hf=75*
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 17	SET 18	SET 19	SET 20
	cov=75	cov=55	cov=40	cov=20
	h=200	h=1135	h=2065	h=3000
PATTERN	d=150	d=1066	d=1983	d=2900

5	b=3000	b=2100	b=1200	b=300
	bw=150*	bw=600	bw=1050*	bw=1500*
	hf=75	hf=535	hf=990	hf=1450
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 21	SET 22	SET 23	SET 24
	cov=20	cov=40	cov=55	cov=75
	h=3000	h=2065	h=1135	h=200
PATTERN	d=150*	d=1066*	d=1983*	d=2900*
6	b=300	b=1200	b=2100	b=3000
	bw=1500*	bw=1050*	bw=600	bw=150*
	hf=75*	hf=535	hf=990*	hf=1450*

Table 5.3 Sets of test data considering dependencies.

	INCR 1	INCR 2	INCR 3	INCR 4
	SET 1	SET 2	SET 3	SET 4
	cov=20	cov=40	cov=55	cov=75
	h=200	h=1135	h=2065	h=3000
PATTERN	d=160	d=1075	d=1990	d=2900
1	b=300	b=1200	b=2100	b=3000
	bw=150	bw=600	bw=1050	bw=1500
	hf=75	hf=535	hf=990	hf=1450
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 5	SET 6	SET 7	SET 8
	cov=20	cov=40	cov=55	cov=75
	h=3000	h=2065	h=1135	h=200
PATTERN	d=2900	d=2005	d=1060	d=150
2	b=3000	b=2100	b=1200	b=300
	bw=600	bw=600	bw=600	bw=150
	hf=1450	hf=990	hf=530	hf=75
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 9	SET 10	SET 11	SET 12
	cov=75	cov=55	cov=40	cov=20
	h=200	h=1135	h=2065	h=3000
PATTERN	d=150	d=1060	d=2005	d=2900
3	b=300	b=1200	b=2100	b=3000
	bw=150	bw=600	bw=600	bw=600
	hf=75	hf=530	hf=990	hf=1450
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 13	SET 14	SET 15	SET 16
	cov=20	cov=40	cov=55	cov=75
	h=200	h=1135	h=2065	h=3000
PATTERN	d=160	d=1075	d=1990	d=2900
4	b=300	b=1200	b=2100	b=3000
	bw=150	bw=600	bw=1050	bw=1500
	hf=80	hf=540	hf=535	hf=600
	INCR 1	INCR 2	INCR 3	INCR 4
	SET 17	SET 18	SET 19	SET 20
	cov=75	cov=55	cov=40	cov=20
	h=200	h=1135	h=2065	h=3000
PATTERN	d=150	d=1060	d=2005	d=2900
5	b=3000	b=2100	b=1200	b=300
	bw=600	bw=600	bw=600	bw=150
	hf=75	hf=530	hf=990	hf=1450

```
          INCR 1     INCR 2     INCR 3     INCR 4
          SET 21     SET 22     SET 23     SET 24
          cov=20     cov=40     cov=55     cov=75
          h=3000     h=2065     h=1135     h=200
PATTERN   d=2900     d=2005     d=1060     d=150
   6      b=300      b=1200     b=2100     b=3000
          bw=150     bw=600     bw=600     bw=600
          hf=600     hf=535     hf=530     hf=75
```

5.10 Tabular form

So far we have developed a system for automatically producing sets of engineered data which should give *good coverage* assuming the number of increments is set to be over 100. An informal definition of coverage, for those who remember the BT advert featuring Maureen Lipman & Richard Wilson could be "... all of the colours in all of the sizes...". All of the colours in all of the sizes gives 100% coverage within the range of sizes and colours. *Good coverage* for this research requires significantly more increments than the 4 used in tables 5.2 & 5.3 above. In excess of one hundred increments has been found from experience to provide good coverage; above 100 increments few anomalies in the structural behaviour of the model are found *e.g.* when depth to compression reinforcement approaches the neutral axis depth.

The character strings for the storage of expressions have been defined as:
 $27001, $27002 and so on for storing the names of parameters
 $28001, $28002 and so on for storing an expression for limiting their minimum value
 $29001, $29002 and so on for storing an expression limiting their maximum value
 $30001, $30002 and so on for storing an expression defining the value.
To assign these values is not a long process, and many models were so coded, but with hindsight, there is a quicker way *i.e.* write the parameters and their properties in a table. The following example is taken from verified model vm112.NDF which is contained in Appendix A.

Table 5.4 Parameter table for the structural analysis of a framework.

```
PARAMETER  Start   End     Type    Dependency conditions
No. name   zst()   zen()   zty()
  1 sp     2       8       0
  2 nsg    16      64      3
  3 dy     0.2     1.2     0       >sp/12       <sp/6
  4 dz     0.2     1.2     0       >dy/2        <2*dy
  5 e      10E6    30E6    3
  6 nu     0.1     0.2     2
  7 w      -12     -60     0
  8 v      -100    0       0
  9 prop   0       1       2
```

The keyword PARAMETER starts table 5.4, a blank line terminates the data. Some parameters are dependent on other parameters; in the table above, the depth of the beam *dy* is dependent on its length *sp* and limited to >*sp/12* and <*sp/6*. Similarly the breadth of the beam *dz* is dependent on the beam depth *dy* and limited to >*dy/2* and <*2*dy*. All such dependencies are shown to the right of the table and commence with > = or <, usually followed by an expression. When the expression, which may be dependent on many parameters, is too long to be contained on the same line as the parameter to which it refers, the > = or < are provided as flags to say: the expression is given on a line by itself following the table, in the order of occurrence (l to r, t to b) of any isolated > = or < flags given in the table. The > and < are used in the algebraic sense *e.g.* a<b<c or c>b>a meaning b is bounded by the limiting values of a & c.

The reinforced concrete Tee beam in section 5.9, table 5.1, would be recast as below in table 5.5. Comparison of tables 5.4 and 5.5 reveals that the same format may be used for the tables for testing models for either the structural analysis of a framework or the structural design of a component.

Table 5.5 Parameter table for the structural design of a component.

PARAMETER No.	name	Start zst()	End zen()	Type zty()	Dependency conditions	
1	cov	20	75	500		
2	h	200	3000	500		
3	d	150	2900	1	=h-cov-20	
4	b	300	3000	500		
5	bw	150	1500	500	>b/5	<b/2
6	hf	75	1450	500	>h/5	<d/2

5.11 Variable ranges and redundant data

When giving section properties, occasionally it is necessary to specify that a start and/or end value in the range of a parameter be computed from an expression; to do this give its value as 1E20. When zst(n)=1E20, it causes the numerical expression contained in $(28000+n) to be evaluated and used for the start value instead of 1E20; similarly when zen(n)=1E20 it causes the numerical expression contained in $(29000+n) to be evaluated and used for the end value instead of 1E20. Care must be taken when using this device as the limits will change dependent on the parameters contained in the expressions. Note that this is a different treatment to that used when the strings: $(28000+n) and/or $(29000+n) are found but zst(p) and/or zen(p) do not contain 1E20; for this case the value of the expression contained in the strings is used to act as a limit to the zva(n) computed from the numbers held in zst(n) & zen(n). Generally the computed value zva(n) lies within the range zst(n) and zen(n), the exception being when either or both contain 1E20.

In the previous example for a reinforced concrete Tee beam, test data was generated for all the prompts. There are occasions when sometimes a prompt is given and sometimes omitted *e.g.* the depth to compression reinforcement will only be required if the *calculation* finds that compression reinforcement is required; if compression reinforcement is not required then the logic will by-pass the prompt. As the above treatment assumes values have been provided for all the prompts, then data may have been provided for cases when the prompt is within a programming structure and should not have been provided for the set. This problem of redundant data only occurs with the structural design of components *i.e.* models which have been written as proforma calculations. When SCALE has run a set of data for the design of a structural component, it builds a stack of values, containing only those parameters and their values which have been provided in response to prompts, thus if redundant data has been supplied then it will be omitted from the stack; thus the solution to the problem is to rerun the problem using the stack of parameters and their values which have been created after the first run, *i.e.* letting SCALE filter out redundant data. The following gives an example, the double asterisk marks the bug which would pass the first run but not the second.

```
Rectangular (1) or Tee-beam (2)  +type=????
Overall depth of beam            +h=????
IF type=2
Depth of top flange              +hf=????
ENDIF
  ...
IF hf>h/2 **
STOP Depth of top flange exceeds half of beam depth.
ENDIF
```

Obviously if values were provided for all prompts *e.g.* type=1, h=400, hf=100, then the above would work; but if type=1 then *hf* would not be input and in consequence the Boolean could not be evaluated. The second run would pick up the bug, the first run would not. The juxtaposition of the three prompts with the line which contains the bug, make this seem so obvious that it should have been avoided; when the ellipses represent 1000 missing lines, and the engineer who is writing the proforma calculation is juggling with 50 variables in his head, such a bug is less obvious.

Another consideration: if the calculation has terminated before the end, because, say, a stress combination has exceeded unity, the engineer must not reason "the stress combination check has reported correctly, so the calculation would have worked if the loads had been reduced, thus the test data provided a successful test". Values given in response to prompts which follow the *stress combination exceeds unity* message, would not have been input; the logic which follows the message would not have been tested for the set of test data. Thus modifying the data slightly to reduce the stress combination below unity, will not guarantee a successful second run free from errors. For this reason it is recommended that time be spent on choosing ranges of values for the parameters so that at least 70% of every set of test data automatically generated, runs through to the end of the calculation.

5.12 Section property dependency

Section property dependency is common to both the structural analysis of a framework and the structural design of a component. Even experienced engineers forget that the theory of structures is predicated on section properties such as: cross-sectional area, shear area, moment of inertia *etc.* being compatible. Assigning arbitrary numbers to the AX, AY & IZ triad for the members of a plane frame, will generally result in an impossible structure as AY and AZ are functions of AX. As an example, the section properties of a: 254 x 102 x 22 UB, would be correctly specified in kN & m units, for all 7 members of a plane frame by: 1 THRU 7 AX 28.0E-4 AY 14.5E-4 IZ 2840E-8 where the shear area AY has been taken as the product: depth of section times thickness of web. Were the section properties to be specified as:

1 THRU 7 AX 28.0E-8 AY 14.5E-4 IZ 2840E-8 *i.e.* the engineer assuming a small cross sectional area to avoid the members taking axial load, then axial strain energy would be predominant rather than bending strain energy, and results could be grossly in error. Were the section properties to be specified as: 1 THRU 7 AX 28.0E-4 AY 14.5E-8 IZ 2840E-8 *i.e.* the engineer assuming a very small shear area, then shear strain energy would be predominant rather than bending strain energy, plotted results frequently appearing as if all members had pinned ends rather than fixed. Even if the engineer had correctly specified the section properties as: 1 THRU 7 AX 28.0E-4 AY 14.5E-4 IZ 2840E-8, but had inadvertently given Young's modulus as 205E3 instead of 205E6, then deflections would be 1000 times larger than expected; had the engineer inadvertently given Young's modulus as 205E-6 instead of 205E6 then deflections would be so large that they could not be printed in the field width provided, thus resulting in a set of asterisks being displayed.

It will be clear from the foregoing, that creating hundreds of sets of test data for testing a model, using section properties ranging from say a 203 x 133 x 25 UB up to a 914 x 419 x 388 UB, the system must avoid testing some properties of one beam with those of another beam.

The following lines are taken from the parameter table for proforma calculation sc385.pro which is described in detail in section 10.6, where sd12,sb12,st12 is the serial depth, breadth & thickness respectively and L is the length of the member in metres, **tri** is the name of a procedure which is called to choose a suitable section designation.

```
PARAMETER   Start   End     Type    Dependency conditions
No. name    zst()   zen()   zty()   and notes.
  ...
  9  sd12   40      400     1       >L*1000/24          <L*1000/8
10  sb12    20      200     1       =sd12/2 say for RHS
11  st12    2       15      1       >sd12/66.667+1  <sd12/12+1
12  tri     3       72      1E40    Calls procedure tri.
  ...
```

Those engaged in using structural engineering design models, will know that for hot-finished rectangular hollow sections a 200x100x5 RHS is available but a 200x100x16 RHS is not available. We are trying to select a valid section designation, this we can do in two stages, firstly we choose a serial depth dependent on typical span/depth limits, then a serial breadth which for a stainless rectangular hollow section is taken as half of the depth, then a serial thickness which varies from serial depth/66.67+1 to serial depth/12+1. The addition of 1 mm to both these limits is to compensate for subsequent rounding down. It will be apparent that any section designation formed from the limits given in the above table, is unlikely to exist as the designation has been compiled from practical size considerations; in consequence we need to use this *estimate* of the section to provide us with the best section match available in the required steel section tables. Rather than trying to match the three components of the section designation, it is computationally more efficient to form a compound number from the estimated section designation and compare it with a list of compound numbers formed from available section designations. To form a compound number for a section designation, assuming the serial depth is parameter *sd12*, breadth is *sb12* and thickness *st12* we can form a variable thus: triad=sd12*1E6+sb12*1E3+st12 and compare the value of *triad* with the set of equivalent compound section designations which are available, colloquially *rolled*, selecting the nearest match, which then can be separated into its three components, thus providing an authentic section designation which will not be rejected by the model.

When the procedure **tri** is called, zp'=12 *i.e.* the current parameter is the twelfth in the above table and zva(12-3) *i.e.* zva(9) holds the estimated serial depth within the ranges 40 to 400 and also in the range >L*1000/24 and <L*1000/8 where L is the member length in metres, zva(10) holds an interpolated value for the serial breadth in the range 20 to 200 and of value =half the serial depth, zva(11) holds an interpolated value for the serial thickness in the range 2 to 15 and in the range >serial depth/66.667+1 to <serial depth/12+1. When **tri** is called, zen=zen(12)=72, thus rectangular hollow section designations tri(0) to tri(72) are assigned in the procedure **tri** below; the loop which follows, finds the nearest compound number holding the section designation which matches the trial value and separates out the authentic components into zva(9), zva(10) & zva(11) which become the values describing the steel section which are included in the current set of data.

```
DEFINE tri ! Procedure for fixing triad preceding parameter zp'.
! +triad=zva(zp'-3)*10^6+zva(zp'-2)*10^3+zva(zp'-1)
IF zen=58 ! Compound section designations for SHS.
! +tri(0)=VEC(0,40040002,40040003,50050002,50050003,50050004)
! +tri(6)=VEC(60060002,60060003,60060004,60060005,80080002)
! +tri(11)=VEC(80080003,80080004,80080005,100100003,100100004)
! +tri(16)=VEC(100100005,100100006,100100008,125125003,125125004)
! +tri(21)=VEC(125125005,125125006,125125008,150150003,150150004)
! +tri(26)=VEC(150150005,150150006,150150008,175175004,175175005)
! +tri(31)=VEC(175175006,175175008,175175010,200200004,200200005)
! +tri(36)=VEC(200200006,200200008,200200010,250250005,250250006)
! +tri(41)=VEC(250250008,250250010,250250012,300300005,300300006)
! +tri(46)=VEC(300300008,300300010,300300012,350350006,350350008)
! +tri(51)=VEC(350350010,350350012,350350015,400400006,400400008)
! +tri(56)=VEC(400400010,400400012,400400015)
ENDIF
IF zen=72 ! Compound section designations for RHS.
! +tri(0)=VEC(0,50025001.5,50025002,60030002,60030003,80040002)
```

```
! +tri(6)=VEC(80040003,80040004,100050002,100050003,100050004)
! +tri(11)=VEC(100050005,100050006,150075003,150075004,150075005)
! +tri(16)=VEC(150075006,150075008,150100003,150100004,150100005)
! +tri(21)=VEC(150100006,150100008,200100004,200100005,200100006)
! +tri(26)=VEC(200100008,200100010,200125004,200125005,200125006)
! +tri(31)=VEC(200125008,200125010)
! +tri(33)=VEC(250125006,250125008,250125010,250125012,250125015)
! +tri(38)=VEC(250150006,250150008,250150010,250150012,250150015)
! +tri(43)=VEC(300150006,300150008,300150010,300150012,300150015)
! +tri(48)=VEC(300200006,300200008,300200010,300200012,300200015)
! +tri(53)=VEC(350175006,350175008,350175010,350175012,350175015)
! +tri(58)=VEC(350200006,350200008,350200010,350200012,350200015)
! +tri(63)=VEC(400200006,400200008,400200010,400200012,400200015)
! +tri(68)=VEC(400250006,400250008,400250010,400250012,400250015)
ENDIF
! +i=0
REPEAT
! +trista=tri(i) +i=i+1 +triend=tri(i)
UNTIL triad>trista AND triad<=triend OR i>=zen
ENDREPEAT
! +trinew=triend Save revised values.
IF triad-trista<triend-triad THEN trinew=trista ENDIF
! +zva(zp'-3)=INT(trinew/10^6+.5) +trinew=trinew-zva(zp'-3)*10^6
! +zva(zp'-2)=INT(trinew/10^3+.5)
! +zva(zp'-1)=trinew-zva(zp'-2)*10^3
ENDDEFINE
```

Occasionally there is a requirement that a parameter *p* has to be the lesser (or greater) of two previously assigned parameters. In a program, this is straightforward; e.g. to assign the lessor of *a* & *b* to *p*, we would simply write: p=a ;IF b<a THEN p=b. When building the PARAMETER table, such programming logic can be included as described above for procedure *tri*. Although a procedure could be provided containing:

```
p=a
IF b<a THEN p=b
```

for such simple logic it is easy to avoid using a procedure. This may be achieved by devising one expression which replaces the above so that it can be used directly in the parameter table, and another similar expression which finds the larger of two values. A little bit of thought and the usage of SGN (Signum) will suffice where: SGN (signum) returns 1 if the argument is positive, -1 if negative, 0 if zero. For example: SGN(0.01) returns 1, SGN(-270) returns -1. The reader may confirm by substituting values for *a* and *b* that the:

minimum of a,b =b-SGN(SGN(b-a)+1)*(b-a)

maximum of a,b =b-SGN(SGN(a-b)+1)*(b-a)

where SGN(SGN(b-a)+1) is a switch which evaluates to 1 when b>=a or 0 when b<a.

In the parameter table below, *L* takes the lesser of values *a* and *b*. Although, so far, it has not been found necessary to set a parameter to be the least (or greatest) of three parameters; by the use of *L* as a dummy parameter it can be done. Because *L* already holds the lesser of *a* and *b*, then *p* will hold the least of *a*, *b* & *c*.

```
PARAMETER
No. Name   Start   End    Type    Dependency conditions
 1  a      0       12     0       'a' varies from zero to twelve
```

```
 2  b       2         19      0         'b' varies from two to nineteen
 3  L       0         19      0         =b-SGN(SGN(b-a)+1)*(b-a)
 4  c       2         16      0
 5  p       0         19      0         =c-SGN(SGN(c-L)+1)*(c-L)
```

If the least of four values is required, then a second dummy parameter will be required. The above is one example of the use of functions in the parameter table. Engineers enjoy devising such tools.

Summarising, it is possible to achieve considerable flexibility in the specification of dependency conditions, yet maintain the form of the parameter table. For procedures, such as the selection of serial sizes for structural steel components, a procedure may be *called* from the parameter table. To do this, the type of parameter is set to 1E40 to declare that the name given is the name of a procedure which has to be called. The current limit is 500 procedures which may be included in each parameter table. One program is used for generating sets of data for any model, working from the parameter table prepared by the engineer. This is preferable to writing a separate program for each of several hundred models and having to provide maintenance for each.

Praxis (1990) permits the assignment of a string of characters to a numerical variable, thus the engineer may write:

```
+a(i)= +$(j)
```

where $(j) contains any numerical expression. Assuming i=3 & j=20 and the string $(20) contains: L-0.1, then for L=5; +a(3)=+$(20) would assign +a(3)=L-0.1=4.9. This may seem trivial but it is profound, for it is the assignment of a string which contains *any* numerical expression to a subscripted numerical variable which permits the same procedure to be used for checking the logic of hundreds of different *calculations*. The word *any* is apposite, note that the assignment +a(3)=$(20) would be faulted, the +$(20) first causes the substitution of the data held in the string $(20) and then carries out the assignment.

The probability of picking numbers at random, to provide say twenty numbers for the data, which results in a sensible output calculation, is on par with winning the national lottery. Just as Praxis (1990) has to be engineered, so does each set of test data.

Each *calculation* written in Praxis (1990) is both a model and a computer program, engineers like writing programs but experience has shown they do not like the considerable chore of testing their programs. Modern structural calculations which include stability calculations, are not simple, thus a system which can check proforma calculations and identify any bugs, is desirable. Experience has shown that thorough checking of a proforma calculation, does not always shake out the bugs - unless the logic is simple. As stated in section 3.11, the only way of extensively testing the pathways through a program is to provide many sets of data to do the switching between paths. This chapter has described how sets of engineered data may be produced for testing models.

Chapter 6

Sustainability of systems

The systems described and developed in this thesis for verifying the correctness of structural engineering calculations, based on:

- the inclusion of an automatic self-check in every engineering model
- the development of a parameter specification table permitting
- the automatic generation of engineered sets of test data for each model
- the automatic running of the sets of test data for a thousand runs for each model
- the automatic reporting of the results of the tests giving a statistical summary, are all new to the field of structural engineering. The writer is *past his sell by date*, so for this work to survive, the new systems described herein were designed to be sustainable.

The writer hopes that over the next two decades, all firms which write engineering software will adopt the above principles for verifying the correctness of their structural engineering calculations. After four decades as a chartered structural engineer, during which thousands of man years' programming work has been abandoned, the writer makes no apology for introducing the subject of *sustainability*, which is the ability to meet the needs of the present without compromising the ability of future generations to meet their own needs (Brundtland *et al.*, 1987). As Womak & Jones (1996) and Hawken *et al.* (1999) tell us, sustainability is about the avoidance of all waste, *especially human effort*. In this respect, Government IT projects have poor track records.

Weizsäcker *et al.* (1997) use *Consumption* (tuberculosis) as a metaphor for the inefficient use of our resources (which they call the wasting disease) and quote a study for the US National Academy of Engineering which found that about 93% of materials we buy and *consume* never end up as saleable products at all. Moreover, 80% are discarded after a single use, and many of the rest are not as durable as they should be. The authors tell us "The cure for the wasting disease comes from the laboratories, workbenches and production lines of skilled scientists and technologists, from the policies and designs of city planners and architects, from the **ingenuity of engineers**, chemists and farmers, and from the intelligence of every person. Motivation needs to be experienced as *compelling and urgent* by a critical mass of people, otherwise there won't be enough momentum to change the course of our civilisation." SOS (Saving Old

Software), discussed in section 6.7, is compelling and urgent and will ensure the sustainability of pre-Windows software languishing in the nooks & crannies of the engineering departments of Universities and engineering consultancies.

Hawken *et al.* (1999) tell us that Sustainability is not just about the environment it is about "creating a healthy economic, social, and ecological system that develops both better people and thriving nature". A key chapter entitled *Human Capitalism* starts with "What destination does our society want to reach, and how will it get there? Lessons in what not to do can often be found in cities, where most officials, overwhelmed by a flood of problems, try to cope by naming them and solving them one at a time. If they are faced with congestion, their answer is to widen streets and build bypasses and parking garages. Crime? Lock up offenders, Smog? Regulate emissions *etc.* Communities and whole societies need to be managed with the same appreciation for integrative design as building, the same frugally simple engineering as lean factories, and the same entrepreneurial drive as great companies *etc.* Social systems have a dual role. They provide not only the monetised *human resource* of educated minds and skilled hands but also the far more valuable but unmonetised *social system services* - culture, wisdom, honour, love, and a whole range of values, attributes, and behaviours that define our humanity and make our lives worth living."

Lean Thinking: Banish Waste and Create Wealth in Your Corporation (Womack and Jones, 1996), espouses the doctrine of Taiichi Ohno - the father of the Toyota production system (Ohno, 1988). Ohno defined waste as "any human activity which absorbs resources but creates no value". Abandoned systems which have taken many man-years to develop is waste on a grand scale.

Older engineers warn that young engineers are becoming too dependent on computers. Few programmers will forget that the computing profession was near to *meltdown* in 1998/99 due to the so called *Millennium Bug*. Just about anyone who could *get into program code* put in long hours to avoid what the media headlined as: planes dropping out of the sky, pacemakers dying, white goods exploding *etc.* Some lessons can be learnt.

6.1 Systems which have been abandoned

When access to those who have been associated with the development of a system over several decades is lost, then maintenance becomes extremely difficult with entire systems being abandoned. Five examples of megaprojects which were abandoned follow. To ensure sustainability, lessons must be learnt and applied.

- In AD 570, the Great Marib Dam in southern Arabia (in what is now called the Yemen) failed. The dam was 600m long and 18 metres deep and supported a population of 30,000. When the dam failed, the skills which had been developed to build the dam had long since been forgotten. The agriculture which depended on the dam collapsed so the dam was abandoned; within a year the desert had returned.

- It is three decades since man last put foot on the moon, the system that was developed to make moon exploration happen, has long since been abandoned. President John F. Kennedy's "to land a man on the moon and return him safely to earth before the end of this decade" is unlikely to be repeated by the West in this decade.
- In 1969-73 the PSA (Property Services Agency) invested heavily in developing a system called Genesys for the integrated structural analysis and design of buildings. Elegant concepts of setting up a 3D model, defining loads as objects, and applying those objects to the model, were not popular with engineers; Genesys was abandoned in the late seventies, the PSA suffered the same fate a decade later.
- From 1975 until the late eighties, the C&CA worked on the development of a system called BARD for the bar-scheduling and detailing of reinforced concrete beams, slabs & columns; BARD was abandoned in the nineties.
- A recent news report stated that the true reason why Concorde was grounded was that the team which worked on the wiring no longer exists and today's engineers are not able to get to grips with the wiring even after long periods of study.

6.2 Large software systems

The Daily Telegraph of 22.04.06 News Bulletin, states "The Government bowed to pressure yesterday to conduct an independent review of a £6.2 billion computerised online booking system for the National Health Service. This month 23 computer experts wrote an open letter to MPs calling for an independent audit for the National Programme for IT". As quoted in section 2.1, Micahel (2004), a software tester at Microsoft, states to well and truly test the matrix of Microsoft products, you have to test every combination of: 9 operating systems with 10 browsers with 4 .Net frameworks with 12 versions of Office and at least 3 versions of your own application *i.e.* 12,960 different configurations on which you need to run all your tests. Very large software systems such as the NHS system, require a level of testing at least as comprehensive as that at Microsoft.

When software systems contain just one million lines, it becomes difficult for even the most experienced programmer to hold an overview of the entire process, especially if the code is heavily interleaved and heavily patched. Microsoft's $55,000,000,000 credit balance allows Windows to be maintained and improved; alas a million other programs in daily use in: engineering, every type of industry, science, research, production *etc.* (some programs with patches on the patches) are used by firms which do not generate sufficient profits to pay for necessary maintenance on their software. We should be concerned about this situation. The lesson from Concorde's grounding is that rigid QA procedures noting changes, and storing the notes in filing cabinets full of extensive documentation, are useful if one engineer has a detailed knowledge of the *internals* of the entire system, but not useful if no one has a detailed knowledge of the *internals* of the entire system.

6.3 Genesys & Lucid

Genesys and Lucid are two systems for structural engineers. The development of both systems was based at Loughborough University in the seventies. Genesys was abandoned, Lucid is in daily use in many firms. Genesys was: financed by Government; designed by computing enthusiasts who were part-time engineers; used elegant concepts which had to be mastered by engineers; was expensive to maintain. Lucid was: financed by engineering firms; designed by engineers; used simple concepts which were familiar to engineers; was cheap to maintain.

6.4 Engineering shareware

The last three decades of the twentieth century saw considerable investment in structural engineering software, mainly in the seventies and eighties before the severe recession in the nineties. During the nineties, some robust structural systems were shelved because funds were not available to convert the software to run under Windows; BARD for BAR-scheduling & Detailing of reinforced concrete was one such system. Computer managers do not like the inconvenience of supporting non-Windows applications; engineers who have mastered the intricacies of Windows are not easily persuaded to use a non-Windows program.

In the nineteen seventies and eighties, authors of software could not be persuaded to make their software available to others. The situation has now changed, some authors now make their software available for a small charge and recognition of the author; this change in thinking has been brought about entirely by the Internet.

A centre for the distribution of engineering shareware would be accepted more favourably by engineers were it to be associated with a University rather than a commercial firm. There is no requirement for a building or premises of any sort; the centre need only exist as a web site.

6.5 On a personal level

The writer spent 1969-1973 working on Genesys RC/1 Slabs, and in 1973 financed and wrote the Genesys subsystems: Composite Construction/1 and Subframe/1. Both Genesys subsystems were financial disasters as engineers like engineering design to be ***in-house***. The situation is different now because computers and engineering software are cheap and can be purchased for in-house use, thus there is now a market for engineering software.

In 1981/82 the writer wrote a linear elastic analysis program called Super-STRESS in partnership with the Cement & Concrete Association. When the Operational Research Department at the C&CA was closed in the cutbacks in the late eighties, Integer inherited Super-STRESS. By email dated 14.3.03 Integer informed the writer that there was a bug in the member distortions, the writer produced longhand calculations proving there was not a bug, and on 15.3.03 replied:

"In response to your request for help, I have unearthed Version 2.7 of SuperSTRESS as handed to you and Hugh Duncan on 25.9.85 (*i.e.* nearly eighteen years ago) and after fiddling with the I/O, I have managed to compile and link it. I enclose an annotated printout for an unrestrained space frame member subjected to member distortions in all six degrees of freedom, and a second analysis when the member is restrained; in both cases the results are as expected."

Integer subsequently found out that the bug was introduced by one of their own programmers. The reasons that Super-STRESS has survived for over 20 years are that:

- the program code is manageable
- the expertise still exists to go back to the source of the software and carry out longhand checks to prove the correctness, or otherwise, of the output.

Just as the expertise which last landed man on the moon thirty years ago has atrophied, so has expertise in many other fields of engineering, *e.g.* nuclear, railways, mining, shipbuilding *etc.* Much of Britain's industrial atrophy has been caused by poor management by Governments. Cancelling the Blue Streak missile project was a far too important decision for a Minister to make. In 2006 there is an active debate on the subject of building new nuclear power stations, concern has been voiced because the engineers which designed the last generation of nuclear power stations 25 years ago, which included Torness, with which the writer was involved, have now retired. Such important engineering decisions need a National Engineering Vote, which nowadays means a vote by internet. This is one useful job that the reorganised Engineering Council could organise; the pros & cons for developing or cancelling any scheme should be shown on their website and all chartered engineers should be encouraged to vote. Governments would not ignore such a collected body of wisdom. The various engineering institutions should feature impending votes in their publications.

6.6 Education

In the nineteen sixties and seventies, computer programming was included in engineering and mathematics syllabuses. Today, even when reading engineering at the so called prestigious universities, computer programming is either not compulsory or is unavailable. We should be concerned that computer programming is not an integral part of engineering syllabuses because engineering software needs to be maintained by engineers who are part-time programmers and not vice versa.

6.7 SOS - Save Old Software

It is proposed that old programs be converted to *black-boxes* for running as Windows programs. A simple practical example is given to be used as a model for *how to do the conversion*.

There are thousands of robust engineering programs still in use but not running under Windows. The cost of rewriting these programs to take advantage of Windows, is high. This section develops a simple system which will allow old software used in engineering design - in the main written in Fortran running under DOS - to run under Windows XP and all previous versions back to Windows 95. Desirable attributes of such a system are:

- The Saved Old Software should be accepted from any source, the Universities should prove to be particularly fruitful.
- The system should provide a common Windows' interface for the engineer, whether running: a reinforced concrete design program; an expert system for the choice of a shear-wall system; or a bridge assessment program.
- The system should be free and non-proprietary, e.g. not dependent on the engineer having to buy or use a particular word-processor.
- The system should be easy for engineers to use for running existing applications and for developing their own applications and adding them into the system.
- The system should require no longer than a day by an engineer (who may be unfamiliar with the subject matter) to do the conversion.
- User's Manuals and other documentation will be supplied in electronic form thus avoiding the cost of clerical assistance.
- The system should cost next-to-nothing to run, thus it needs to be fully electronic *i.e.* a *virtual centre* existing only on the Internet, but masquerading as a long established entity.

Persuasion

If you have old engineering software (written in BASIC or FORTRAN) which you would like converted to run under Windows, please email your telephone number and any documentation for the software to the *Structural Calculations Centre*. You will be phoned and by mutual agreement, your old software will be converted to run under Windows.

For those who wish to do their own conversion: firstly convert the old program to read from and write to named disk files *e.g.* in1.dat, in2.dat, out1.res & out2.res (these names may be any valid DOS filenames). The programming work for this only involves changing the format statements to read from and write text to disk rather than the screen. The programmer does not need to understand how the old program works; indeed the programmer should be discouraged from tampering with the program logic; *if it ain't broke, don't fix it*. Once the program has been converted to read its input data from a named disk file and write its output to another named disk file, a *black-box* has been produced. Next write a short proforma calculation to prompt for data and write a file of input data for the black-box, invoke the black-box, then read the output file written by the black-box, display it and optionally print it.

As a simple example, imagine an executable program called ADD.EXE which prompts for 2 integer numbers and writes the sum to the screen. Firstly convert the formatting to read the 2 numbers A & B from the 1st & 2nd lines of the file IN.DAT and write the sum to the file OUT.RES. We now have a black-box with FORTRAN similar to:

```
C Black-box to ADD two integer numbers.
      OPEN(UNIT=7,FILE='IN.DAT',STATUS='OLD')
      REWIND 7
      OPEN(UNIT=8,FILE='OUT.RES',STATUS='UNKNOWN')
      REWIND 8
      READ(7,10)I
   10 FORMAT(I9)
      READ(7,10)J
      K=I+J
      WRITE(8,10)K
      CLOSE(UNIT=7,STATUS='KEEP')
      CLOSE(UNIT=8,STATUS='KEEP')
      END
```

which when compiled & linked, produces an executable program called ADD.EXE. Next write the following logic-ENGLISH to: prompt for the numbers; pipe the numbers to the file IN.DAT; invoke ADD.EXE; include the results from the file OUT.RES in the output calculation.

```
START
First integer number      +A=????
Second integer number     +B=????
FILE IN.DAT ! Open the file IN.DAT to receive piped values A &
B.
% +A
% +B
FILE ! Close piped file.
WIN ADD.EXE ! Invokes ADD.EXE which writes result/s to OUT.RES
#OUT.RES    ! Include result/s from OUT.RES in the output calc.
FINISH
```

The above model can be copied to convert any existing or new FORTRAN or BASIC executable program into one which runs and outputs the calculations under Windows.
Requirements for participation with the system:

- the contributor must provide documentation for the black-box
- the black-box must have merit and not duplicate one already provided.

6.8 Text files and manageable proportions

Text files are easily read, text written a century ago can be understood today, text written today will be understood a century from now, thus ***text is good***. The systems listed in section 6.1 which have now been abandoned, were all integrated systems which required teams of people, no single person understood the whole system and was able to maintain it, thus ***large integrated systems are not good*** from a sustainability viewpoint.

The writer's SCALE (Structural CALculations Ensemble), contains 1,198,000 lines of software excluding documentation, of which 936,000 lines are text (models for the analysis of structural frameworks and for the design of structural components and details), 262,000 lines are procedural. SCALE includes 780 proforma calculations or models for the structural design of components and details, 338 models for the structural analysis of frameworks, 140 benchmarks *i.e.* data files mostly collected from published examples. All 1,258 files are text files, thus in a century from now, an engineer will be able to pick up any of these 1,258 files and follow it. Of the software developed as part of this research, approximately 95% was text, 5% was procedural *i.e.* computer code.

The SCALE proforma calculations are modular and have coupling to and from the NL-STRESS program and its utilities. SCALE proforma 924 is used for the verification of models (proforma calculations) for the structural analysis of frameworks and the structural design of components. For verification, SCALE calls proforma 924 which in turns calls SCALE which may or may not call NL-STRESS dependent on the proforma being verified. For a wide range of examples of verification, see chapter 2 for COMPUTER AIDED VERIFICATION, "Proceedings of the 17th International Conference, CAV 2005, Edinburgh", July 2005.

The ratio 95% text to 5% procedural mentioned above is not limited to structural engineering systems. The writer recalls that in the previous millennium, Windows was coupled by binary files. The writer, who admits bias, considers that Praxis is superior to spread sheets for which hidden formulae are difficult to follow. For *number crunching,* compiled computer code will always be more efficient than interpreted plain text, but for sustainability, plain text should be the way forward. As English has become the international language of business, wherever possible the plain text should be written in English.

Hot keys, buttons and mice may be treated parametrically; the system developed in chapter 5 may be applied to large coupled systems so that they may be tested by automatically generated sets of data. As quoted in section 2.1, Micahel (2004), a software tester at Microsoft, states to well and truly test the matrix of Microsoft products, you have to test every combination of: 9 operating systems with 10 browsers with 4 .Net frameworks with 12 versions of Office and at least 3 versions of your own application *i.e.* 12,960 different configurations on which you need to run all your tests. **Comprehensive testing of 12,960 different configurations can only be carried out by running automatically generated sets of test data.**

Chapter 7

Verified models for structural analysis

This chapter develops a system which permits a text file of data for a structural analysis to be extended with logic to compare the results of an analysis with those expected by an existing classical solution. All data will be written parametrically and the system will run up to a thousand different sets of data, highlighting any results from the matrix analysis which diverge from a classical solution appropriate to the model.

Baker (1968) writes "I agree entirely with the main thesis, that is it is absolutely essential to have verification, whether by model or, if the economy justifies it, full scale, of the various analysis techniques which the computer has made available to us." The IStructE Guidelines (Harris *et al.* 2002), list what to do when results of the checking model and global model do not agree. The set of verified models will ensure that if the results of the global model do not agree with the appropriate verified model, then it will be the global model which is in error.

7.1 Data input and checking

The results of a structural analysis will be wrong if the input data is wrong; it follows that the input data for a structural analysis must be checked. This research, from the starting position that the data has been checked, develops an extensive set of models which are self checking and have each been verified with a thousand sets of data providing extensive coverage, for the model. There are several methods for the preparation of data for a structural analysis including: language, graphical user interface and algebra.

In 1963, a group of professors and students at MIT developed a structural engineering language which they called STRESS (Fenves *et al.*, 1964). That same year, Ivan Sutherland at MIT drew a structure to be analysed by moving an electronic pen over a computer screen, he called his system: Sketchpad (Sutherland, 1963). Both were different but valid approaches to the preparation of structural data for analysis; STRESS was the first language for structural analysis, Sketchpad was the first GUI (Graphical User Interface) for structural analysis. In the two decades which followed, the language approach to the preparation of structural data was in general use in the bigger consultancies, but the GUI approach was rare, although Warwick University had a GUI in the early seventies. In the third and fourth decades, due to the fall in computer processing costs, general use of Windows and attractive advertising, GUIs became popular. This research considers both approaches using a cantilever beam typical of a simple structure.

7.2 Simple structure written in 1963 STRESS

```
STRUCTURE CANTILEVER BEAM
TYPE PLANE FRAME
NUMBER OF JOINTS 2
NUMBER OF SUPPORTS 1
NUMBER OF MEMBERS 1
NUMBER OF LOADINGS 1
METHOD STIFFNESS
JOINT COORDINATES
1 X 0. Y 0. SUPPORT
2 X 3. Y 0.
MEMBER INCIDENCES
1 1 2
MEMBER PROPERTIES
* IZ is moment of inertia about the Z axis, Z points out.
1 AX 0.24 IZ 0.0072
CONSTANTS E 28000000. ALL
LOADING 1, END POINT LOAD
JOINT LOADS
* Y axis is up, therefore a downward load is negative.
2 FORCE Y -2.
SOLVE
FINISH
```

The above data will be immediately recognisable to all structural engineers who have come across one of following varieties of the STRESS language: IBM 1130 STRESS, MISTRESS (MIni computer STRESS), Olivetti STRESS, New-STRESS, SuperSTRESS, STRESS-3, NL-STRESS; all but the first version were written in England. If the above data were to be run on any version of STRESS, the results would be identical, giving a maximum bending moment of 6 due to the point load of 2 on the end of the cantilever of length 3. The cantilever beam is invaluable for testing structural analysis software for linear elastic, sway and within-member stability, and plastic analysis. Even engineers who have never come across the STRESS language, will be able to follow the data; the language is intuitive to those who speak English. STRESS is ideal for teaching purposes, for as it is a language, it has syntax which may be taught, *e.g.* writing with chalk on a blackboard *A line starting with an asterisk is a comment which is included in the results, but otherwise ignored.*

7.3 Cantilever beam - data preparation by GUI

The writer's NL-STRESS program has a GUI as one of several methods of data input. A list of instructions for its operation would start: Click GUI, click File, click Open; select the name of the data file required or type a new name, click OK. Click Identification and complete the page headings. Click Output, click Tabulate, click Displacements, Forces, Stresses, Reactions as required... Click Joint Coordinates, click Draw Structure and draw your structure on the screen, click Help when required *e.g.* for changing the grid or switching snap on/off *etc.* For NL-STRESS, the data prepared by running the GUI would be as given in 7.2, optionally tables of joint, member and loading information, may be included in the results for checking. Other GUIs list tables of joint coordinates, member incidences, member properties, loading *etc.* for the engineer to check.

Few engineers would read such instructions containing so many *clicks*. The easiest way to master a GUI is to use it. It will be immediately obvious from the *screens* that it is easier to click *Type Plane Frame*, *Method Elastic etc.* than to type the data. A GUI is a quicker and more attractive way of preparing data for a simple structures than typing the data using the STRESS language.

Generally for simple structures, the GUI will always be the quicker and more attractive way of preparing data; but that is not the end of the matter. Searle (1987) tells us that Wittgenstein urged us to think of words as tools. The following simple words, when added as tools to STRESS, can save time: IF ENDIF THEN GOTO REPEAT UNTIL ENDREPEAT VEC. VEC (short for VECtor) is one of a set of functions: SIN COS *etc.* Examples of usage of these tools follow.

For the structural analysis of a framework, the data file will always have the structure of that given in section 7.2 *i.e.* comprising a mixture of commands and tables written in English, even if the GUI has been used to generate the data. The addition of the extra words given in the previous paragraph permit NL-STRESS data to be written parametrically. If the model is to be verified, it is necessary that data for the model and its self check, be written parametrically.

7.4 Parameters

There is no point including a statement such as: IF 11.2>3.6 THEN ... in data for the structural analysis of a framework; 11.2 and 3.6 are constants and the condition will always be true, but a statement such as: IF a>b THEN... gives control to do something if the Boolean is *true*, *i.e.* variable *a* is greater than the variable *b*, or to ignore that which follows the THEN if the Boolean is false.

Dijkstra (1972) tells us "Once a person has understood the way in which variables are used in programming, he has understood the quintessence of programming". Fortunately engineers are familiar with assignment and the use of variables. An engineer presented with the logic: a=3 b=5 c=a*b will instinctively know that *c* has been assigned the value 15, without having to think through: the computer sets up a box which it names *a* and stores the value 3 in it; sets up another box which it names *b* and stores the value 5 in it; sets up another box which it names *c*; extracts the values from the boxes marked *a* and *b* and sends them to the arithmetic unit for multiplication storing the result in box *c*. So far we have not said what a,b,c represent, mathematically they should be referred to as symbolic names, in computer parlance **variables**, as the values stored in their named boxes may vary. If *a* and *b* are the sides of a rectangle, then an engineer will immediately recognise that *c* is the area of the rectangle and that a,b,c are all *parameters* of a rectangle. Similarly if *a* is a velocity and *b* is a time, then *c* is the distance travelled and a,b,c are *parameters* of a journey.

7.5 Words used as tools

Engineers are comfortable with assignments such as nj=23, and conditional statements such as: IF j<nj GOTO 10, and will be able to follow such logic. Less familiar will be a VEC() assignment; VEC is short for vector, where vector is used in the computing sense, meaning a one dimension array. On occasions when using parametric data it is necessary to assign a sequence of subscripted variables *e.g.* k(6)=40000 k(7)=80000 k(8)=40000 k(9)=80000 k(10)=40000 k(11)=80000 k(12)=40000 k(13)=80000 k(14)=40000 k(15)=80000.

When the subscript is to be incremented by unity starting from the first subscript, 6 in the above example, then the data can be shortened to:
k(6)=VEC(4E4,8E4,4E4,8E4,4E4,8E4,4E4,8E4,4E4,8E4)

For regularly repeating values as in the above, it is permissible to add a multiplier after the closing bracket *e.g.* k(6)=VEC(4E4,8E4)*5 which causes the assignments to be continued for 5 loops, thus carrying out the same assignments as both previous examples. As a further example the assignment: p(1)=VEC(-1288)*nl would assign p(1) to p(nl) each with the value -1288. The VEC function shortens the amount of data to be typed and checked, from experience *the shorter the data the fewer the mistakes*.

The command prompt in Windows XP and versions of Windows as far back as Windows 95, recognises and runs batch files. Control within a batch file is by a conditional statement such as *IF a<b GOTO 125* where 125 is a label. This Windows' model for control has been incorporated into NL-STRESS; somewhere in the file of data there must be a line commencing :125 to which control is transferred if the Boolean *i.e.* a<b, is true.

The programming structure IF a<b THEN... causing assignments following THEN to be carried out if the Boolean is true, else ignored, has been described in section 7.4; the programming structure IF-ENDIF, which is used when several lines of assignments need to be included between IF and ENDIF, has been included in NL-STRESS.

For looping, engineers generally favour the intuitive:

```
i=0
:200
i=i+1                  The ellipsis represents, assignments
 ...                   tables, conditional statements, and
IF i<10 GOTO 200       further loops.
```

When several loadings are generated by including a loading within a loop, then the loading title needs to be changed for each pass. For this case the *structured* loop *REPEAT-UNTIL-ENDREPEAT* has been incorporated into NL-STRESS.

7.6 Other languages

The foregoing has described additions to 1963 STRESS, by that is meant the STRESS language. None of the 1963 STRESS program has survived as it was *machine dependent i.e.* written for operating on the IBM 709-7090-7094 system. The STRESS language has survived the past forty years as words such as: JOINT, COORDINATES, FORCE, CONCENTRATED, UNIFORM... are relevant today and will still be relevant forty years from now. STRESS was chosen for its universal appeal. STRUDL - STRUctural Design Language, Emkin (1977) is as close to STRESS as English is to American-English, and the additions described above would be appropriate. Many other structural languages are similar, but differing in choice of keywords *e.g.* using CONNECTIVITY in place if INCIDENCES. All structural languages which are similar to STRESS would benefit from the additions described.

One structural language which is very different to STRESS is Formex Algebra (Nooshin, 1984). Formex Algebra was designed for the representation of structural configurations and the automated generation of the related data. It will be clear from the following that formex algebra is a powerful tool when applied by an expert. Its language contains terms such as *a signet of grade three* and *two-plex cantle* which will be unfamiliar to the majority of structural engineers. An example of Formex algebra follows.

For a cylindrical barrel vault roof with a radius of 9 and basifactors of b1=1m, b2=c/36 and b3=0.9 m, Yassaee (1984) gives the formex representing the interconnection pattern of the barrel vault relative to the indicated basicylindrical normat as:

$$c = \sum_{i=1}^{5} C_i \text{ where}$$

c1=rinit(9,20,4,2)|lamit(2,1)|[9,0,0; 9,2,1]

representing the bracing elements of the barrel vault,

c2=rinit(10,20,4,2)|[9,0,0; 9,0,2],

c3=rinit(9,19,4,2)|[9,2,1; 9,2,3]

and

c4=lam(3,20)|rin(2,9,4)|[9,2,0; 9,2,1]

representing the elements which are parallel to the generatrix of the cylinder and finally,

c5=lam(3,20)|rin(2,18,2)|[9,0,0; 9,2,0]

representing elements along the normat lines U3=0 and U3=40.

Whereas STRESS (1964) extended by the *words used as tools* described in section 7.5 enables an engineer to build a structural model using commands and tables of joint coordinates *etc.*, Formex Algebra with Formian software enable an engineer to define complex structural problems in a few lines, from which the computer automatically builds a model, now that's structural magic.

7.7 Aims of verified models

As the years pass, engineers build up a design repertoire - dictated by the nature of the work they do. Some engineers design tens of multi-storey frames, others design hundreds of timber roof trusses, or portal frames, and so on. It is tiresome starting each time with a blank sheet of paper or screen and preparing the data for a similar structure to one previously analysed, or editing an old data file to change the joint coordinates, and topology (adding/removing members). Parametric data generation solves the problem, and at the same time provides answers to *What if?* questions; *e.g. What if we add another column?*

If the engineer writes the data in terms of the parameters for the problem, thus providing one file of data for solving a class of problems rather than just solving the current problem; then as the years pass, a library of parametric data files will be created covering the bulk of the engineer's analysis work. It will be heart-warming to pick up a parametric data file prepared a decade earlier, change a few parameters and immediately get a feel for the structural behaviour of a new frame.

The GUI (Graphical User Interface) approach to the preparation of data for structural analysis is recommended for simple structures having a few joints and members. For *one-off* structures - of average size of 10-50 joints - which do not belong to a structural class such as: portal frame, continuous beam *etc.*, the GUI approach is again recommended. For any structures which belong to a structural class and are included in (or can be added to) the library of parametric data files listed in section 7.8, the language approach can speed data preparation ten-fold. For large structures - having a few hundred joints to several thousand joints - the language approach is again recommended.

The GUI in NL-STRESS combines the best of both worlds as it takes a file of NL-STRESS data, which has been written parametrically, and automatically converts it into one containing only numerical values. This conversion from parametric data to numerical data is achieved by the GUI invoking the SCALE program to carry out numerical substitutions for all parameters and to remove keywords such as IF ENDIF REPEAT *etc.*

Advantages of having a text file of data written parametrically include:

- Once the parametric data file has been thoroughly tested and the parameters have been checked using the method described, there should be no errors in the data.

- It takes no longer than a few minutes to change the parameters and rerun the analysis, giving a ten fold increase in productivity.
- The ability to experiment with the parameters gives a feeling for the structural behaviour, *Morphing*.
- Developing a parametric data file is an attractive way of introducing structural programming to engineers and students.

Perhaps when the existing library of parametric data files, has been extended to a few hundred, the tiresome chore of numbering up joints, working out coordinates *etc.* will join Column Analogy and Moment Distribution as part of our structural heritage.

Each of the 108 verified models developed as part of this research constitutes a completed mini research project. A discussion and conclusion for each, will be found in sections 11.1 and 12.1 respectively.

The aims of the set of verified models follow.
- To avoid major disasters such as those described by Harris et al. (2002).
- To give assurance to the engineer that the numbers computed are OK.
- To provide results for any meetings to dispute the results provided by other modelling systems.
- To provide an expert engineer in the form of specific advice given with each model e.g. "See equilibrium check to ensure that applied loads include safety factors; run elastic critical load model for this configuration to ensure that sway and within-member stability criteria are satisfied."
- To provide a system which is easy to maintain after the enthusiasm present at its development has waned.
- To reconcile classical analysis with modern matrix methods for: linear elastic, stability, elastic-plastic analysis, i.e. to use classical methods of analysis to provide bedrock beneath modern stiffness matrix methods.
- To be useful to engineers & immediately recognisable as a structural form.
- The data required to be simple as engineers are rightly suspicious of complexity.
- The classical method chosen to verify the model should have been published, or the theory provided, so that interested engineers can obtain the knowledge.
- The Verification Data should be practical.
- Each model should be self contained and include all references.
- The set of verified models should have a common structure, so that any engineer who can use a text editor, can type: spans, section sizes, strengths... and run any model, thereby avoiding the time-waste and mistakes associated with starting with a blank sheet of paper.

7.8 List of verified models

All 108 models which were developed and verified as part of this research are listed below, the brief description following *cf.* describes the type of self-check which has been used to compare key values calculated by the self check with their equivalent values calculated by NL-STRESS. All the classical self check methods included in the following list are reviewed in chapter 2.

vm110 Deflection of beams including shear *cf.* Chebyshev polynomials.
vm112 Cantilevered beam cf. equilibrium, compatibility & energy.
vm113 Cantilevered beam with many loads *cf.* unit load method.
vm114 Tapered cantilevered beam *cf.* unit load method.
vm115 Cantilevered beam with tie down span *cf.* Roark.
vm117 Subframe, continuous beam + columns *cf.* equilibrium, compatibility & energy.
vm120 Continuous beam with pattern loadings *cf.* Hardy Cross.
vm122 Two member lean-to or Mansard beam *cf.* equilibrium, compatibility & energy.
vm123 Three member lean-to/Mansard beam *cf.* equilibrium, compatibility & energy.
vm124 Three member cranked beam *cf.* equilibrium, compatibility & energy.
vm130 Ground beam on an elastic foundation *cf.* Hetényi.
vm131 Ground beam on elastic piles *cf.* flexibility.
vm140 Influence lines *cf.* Müller-Breslau.
vm150 Pratt through truss *cf.* method of joints.
vm153 Pratt deck truss *cf.* method of joints.
vm156 Howe through truss *cf.* method of joints.
vm159 Howe deck truss *cf.* method of joints.
vm162 Warren through truss *cf.* method of joints.
vm164 Warren through truss with verticals *cf.* method of joints.
vm165 Warren deck truss *cf.* method of joints.
vm168 Warren deck with verticals *cf.* method of joints.
vm171 Two rafter with tie *cf.* method of joints.
vm172 Two rafters, post & tie *cf.* method of joints.
vm173 King post roof truss *cf.* method of joints.
vm174 Three segment rafters, Pratt internals roof truss *cf.* method of joints.
vm175 Three segment rafters, Howe internals roof truss *cf.* method of joints.
vm177 Trussed rafter, or Fink roof truss *cf.* method of joints.
vm178 Three segment trussed rafter, Warren internals roof truss *cf.* method of joints.
vm179 Three segment rafters, Warren internals roof truss *cf.* method of joints.
vm181 Mansard truss *cf.* method of joints.
vm202 Pipe tree having two branches *cf.* equilibrium, compatibility & energy.
vm203 Pipe tree having four branches *cf.* equilibrium, compatibility & energy.
vm204 Pipe tree having six branches *cf.* equilibrium, compatibility & energy.
vm207 One storey bent, vertical/raking columns *cf.* equilib., compatibility & energy.

vm208 Two storey bent, vertical/raking columns *cf.* equilib., compatibility & energy.
vm209 Three storey bent, vertical/raking columns *cf.* equilib., compatibility & energy.
vm210 Bent *cf.* column analogy.
vm211 Rigid pile cap *cf.* Reinforced Concrete Designers' Handbook.
vm215 Ridged portal frame, pinned/fixed feet *cf.* equilibrium, compatibility & energy.
vm216 Mansard portal *cf.* equilibrium, compatibility & energy.
vm217 Gable frame with inclined legs *cf.* equilibrium, compatibility & energy.
vm218 Portal with skewed corners *cf.* equilibrium, compatibility & energy.
vm219 Trapezoidal frame *cf.* equilibrium, compatibility & energy.
vm220 Two bay ridged portal *cf.* Kleinlogel.
vm223 Multi bay ridged portal, pinned/fixed feet *cf.* equilibrium, compatibility & energy.
vm225 Couple roof frame *cf.* equilibrium, compatibility & energy.
vm226 Couple close roof frame *cf.* equilibrium, compatibility & energy.
vm227 Collar-tie roof frame *cf.* equilibrium, compatibility & energy.
vm228 Collar-and-tie roof frame *cf.* equilibrium, compatibility & energy.
vm230 Attic roof frame *cf.* equilibrium, compatibility & energy.
vm232 Fink room roof frame *cf.* equilibrium, compatibility & energy.
vm233 King post roof frame *cf.* equilibrium, compatibility & energy.
vm234 Queen post roof frame *cf.* equilibrium, compatibility & energy.
vm235 Tied Mansard roof frame *cf.* equilibrium, compatibility & energy.
vm241 Vierendeel girder *cf.* equilibrium, compatibility & energy.
vm242 Vierendeel roof frame *cf.* equilibrium, compatibility & energy.
vm244 N/Pratt lattice portal/girder *cf.* equilibrium, compatibility & energy.
vm245 Howe lattice portal/girder *cf.* equilibrium, compatibility & energy.
vm246 Warren portal/girder end diags in tension *cf.* equilibrium, compatibility & energy.
vm247 Warren portal/girder end diags in compr. *cf.* equilibrium, compatibility & energy.
vm260 Multi-storey frame *cf.* Hardy Cross.
vm262 Multi storey frame *cf.* equilibrium, compatibility & energy.
vm270 Pierced shear walls *cf.* Magnus.
vm280 Two pinned circular arch *cf.* Pippard & Baker.
vm281 Encastré circular arch *cf.* Pippard & Baker.
vm282 Two pinned parabolic arch *cf.* Pippard & Baker.
vm283 Encastré parabolic arch *cf.* Pippard & Baker.
vm290 Outrigged frame *cf.* Castigliano.
vm291 Braced outrigged frame *cf.* Castigliano.
vm300 Cantilever or propped cantilever *cf.* equilibrium, compatibility & energy.
vm301 Circular arc cantilever *cf.* Pippard & Baker.
vm302 Circular arc bow girder *cf.* Pippard & Baker.
vm310 Grillage of beams *cf.* Pilkey & Chang.
vm311 Grillage of beams *cf.* equilibrium, compatibility & energy.
vm410 Plastic analysis of cantilever *cf.* equilibrium, compatibility & energy.

vm411 Plastic analysis of propped cantilever *cf.* equilibrium, compatibility & energy.
vm420 Plastic analysis of continuous beam *cf.* equilibrium, compatibility & energy.
vm430 Plastic analysis of rectangular portal *cf.* equilibrium, compatibility & energy.
vm435 Plastic analysis of ridged portal *cf.* equilibrium, compatibility & energy.
vm436 Plastic analysis of multi bay ridged portal *cf.* equilibrium, compatibility & energy.
vm440 Plastic analysis of multi storey frame *cf.* equilibrium, compatibility & energy.
vm501 Cantilever beam in space *cf.* equilibrium, compatibility & energy.
vm510 Four legged stool space frame *cf.* equilibrium, compatibility & energy.
vm520 Spiral stairs space frame *cf.* equilibrium, compatibility & energy.
vm601 Plate with out of plane point loading *cf.* Navier double trigonometric series.
vm602 Flat plate in flexure with area loading *cf.* Navier double trigonometric series.
vm605 Floor panel with hole *cf.* equilibrium, compatibility & energy.
vm610 Plate with free edge *cf.* finite differences & exact formulae.
vm618 Plate/wall in extension with hole *cf.* equilibrium, compatibility & energy.
vm620 Circular balcony *cf.* classical analysis & Roark.
vm630 Spherical shell *cf.* Roark's Formulas for Stress & Strain.
vm640 Torque on I-section *cf.* analysis by Roark & Timoshenko.
vm641 Biaxial bending and/or torque on rectangular hollow section *cf.* Roark.
vm642 Bending and/or torque on T-section *cf.* Roark.
vm643 Bending and/or torque on channel section *cf.* Roark.
vm644 Torque on angle section *cf.* Roark.
vm650 Circular tank *cf.* analysis by Timoshenko & Woinowski-Krieger.
vm710 Natural frequency of beam or frame *cf.* flexibility & latent root.
vm718 Natural frequency of built-in plate *cf.* Roark & Warburton.
vm720 Natural frequency of simply supported plate *cf.* Navier, flexibility & latent root.
vm802 Cantilever beam with large displacements *cf.* equilibrium, compatibility & energy.
vm810 Stability of columns with various supports *cf.* classical formulae by Euler.
vm830 Stability of circular ring/pipe *cf.* classical formulae by Roark.
vm850 Stability of cantilever with udl & end load *cf.* equilibrium, compatibility & energy.
vm852 Non-linear elastic analysis of multi storey frame *cf.* equilib., compat. & energy.
vm950 Hanging cable with flexible platform *cf.* Pippard & Baker.
vm951 Suspension bridge with three pinned stiffening girder *cf.* Pippard & Baker.
vm952 Suspension bridge with two pinned stiffening girder *cf.* Pippard & Baker.

Figure 7.1, which follows on pages 100 to 105, shows the structural framework for the verified models listed in above.

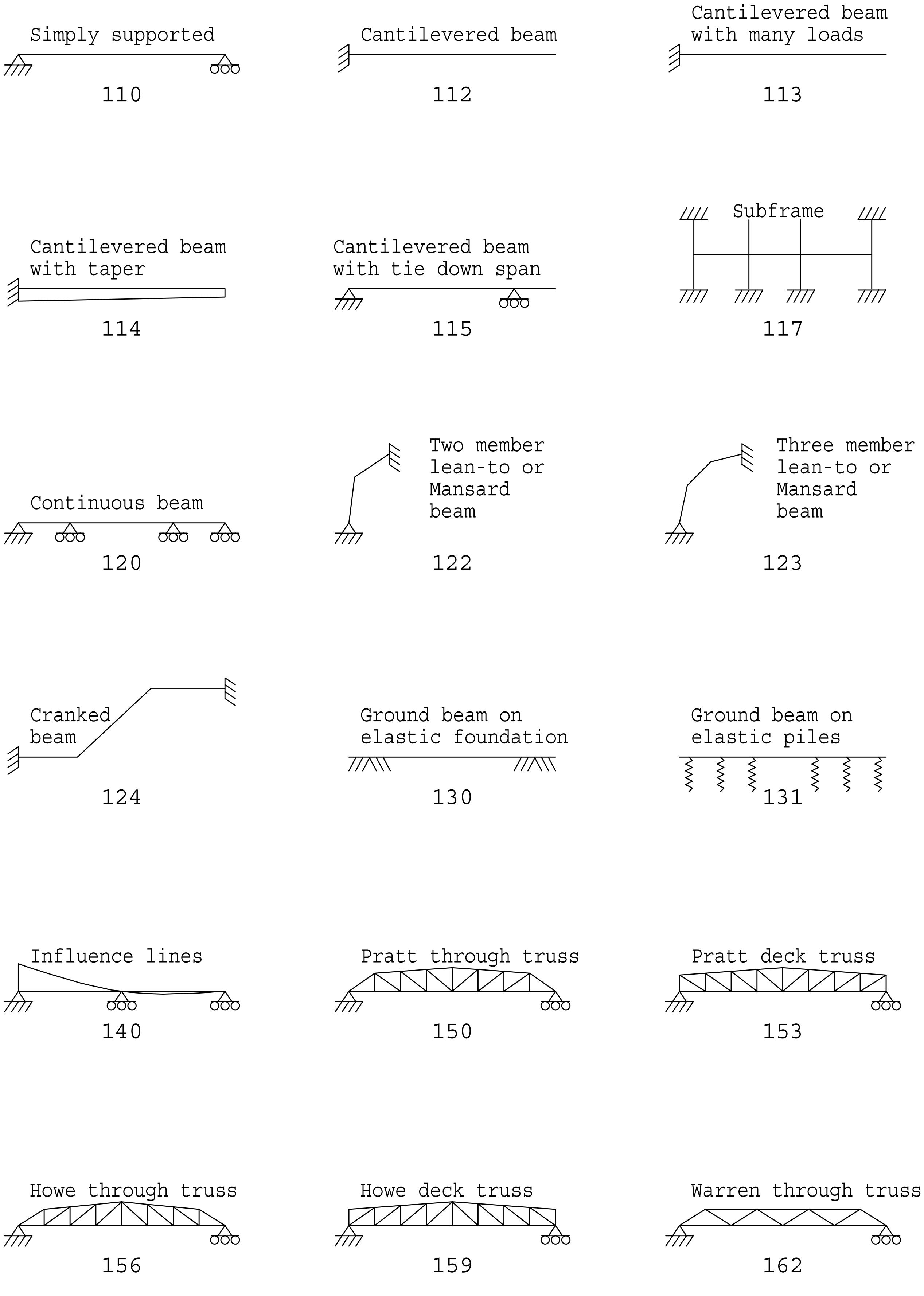


Figure 7.1 Verified models vm110 to vm162

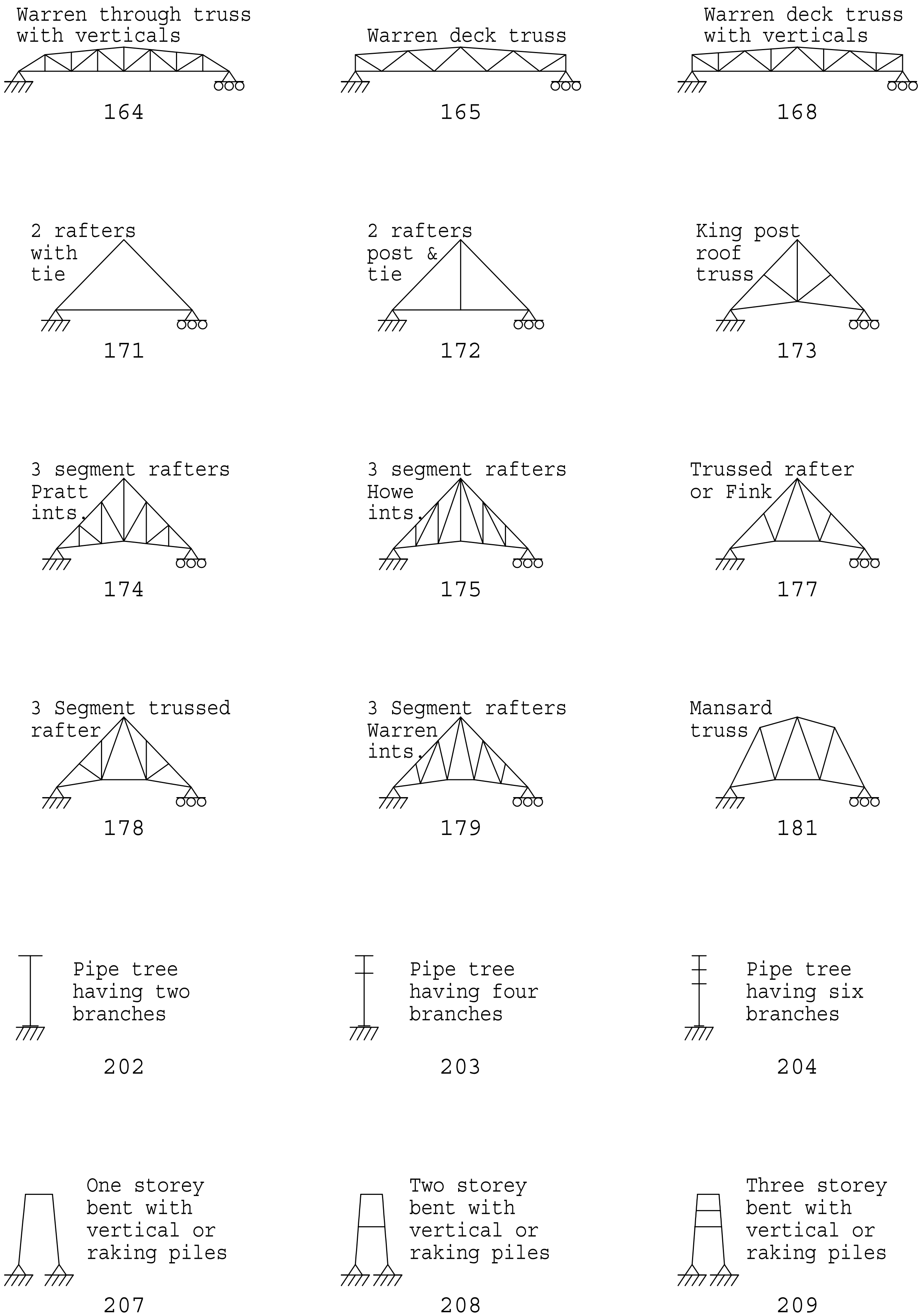


Figure 7.1 Verified models vm164 to vm209

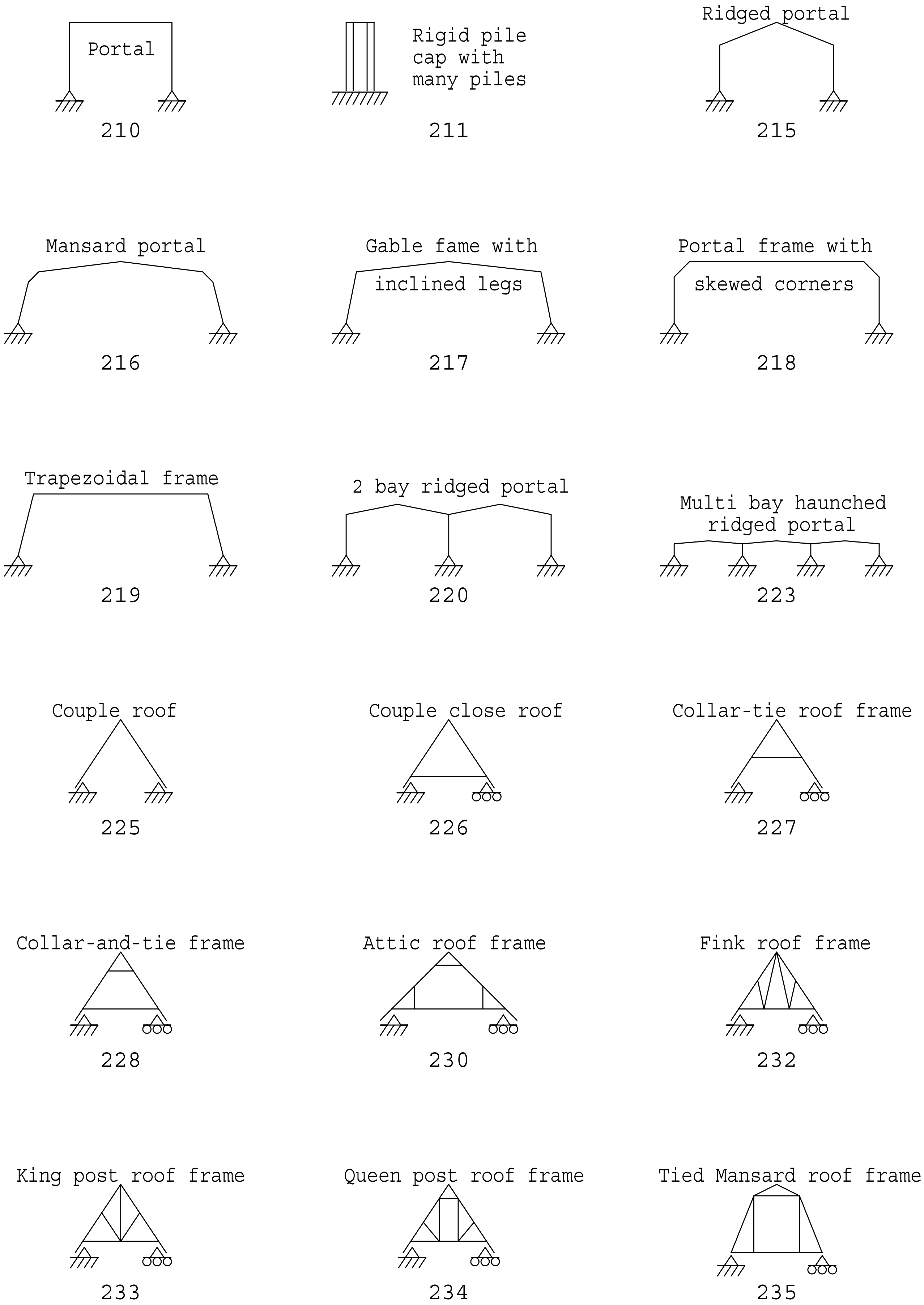


Figure 7.1 Verified models vm210 to vm235

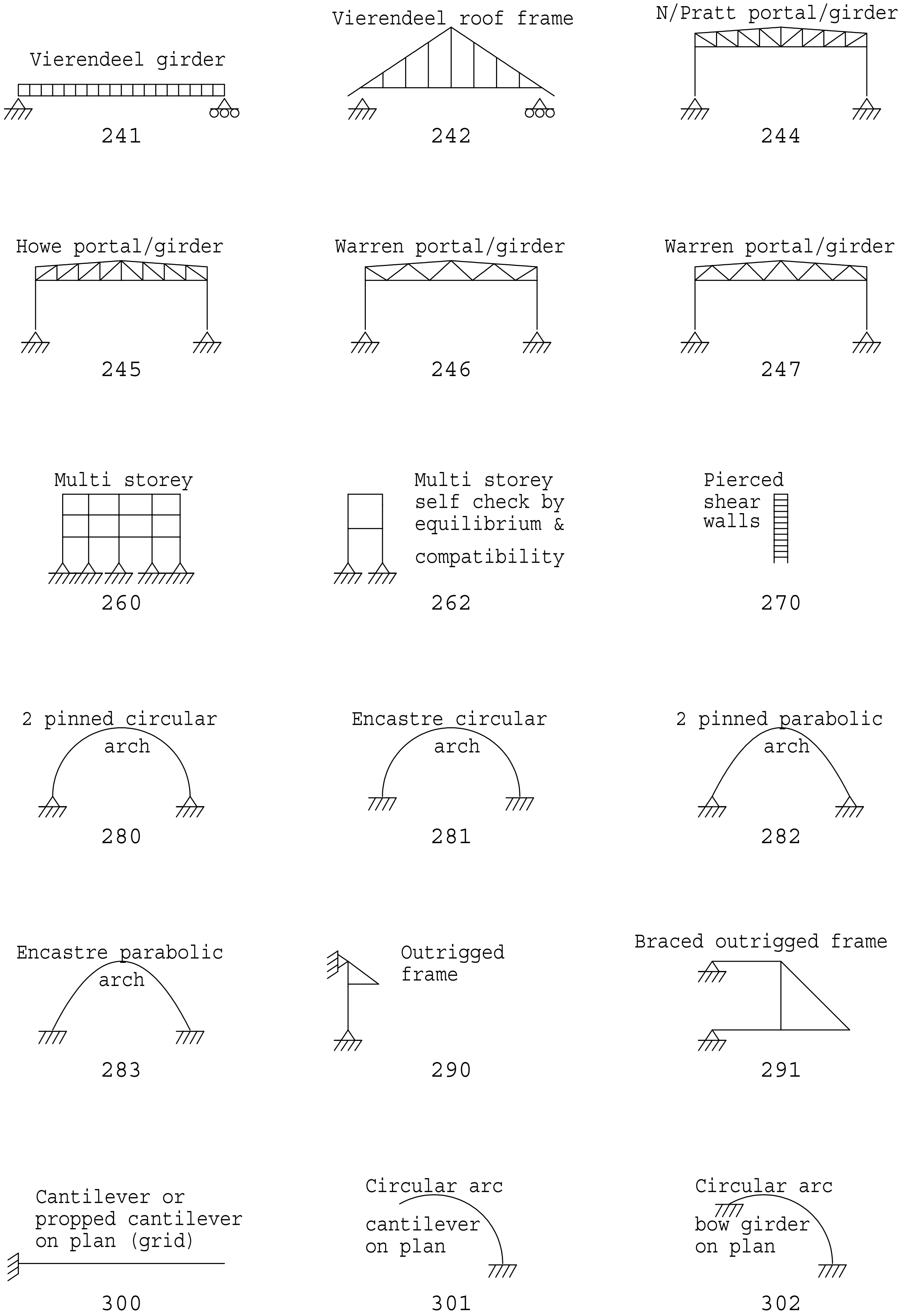


Figure 7.1 Verified models vm241 to vm302

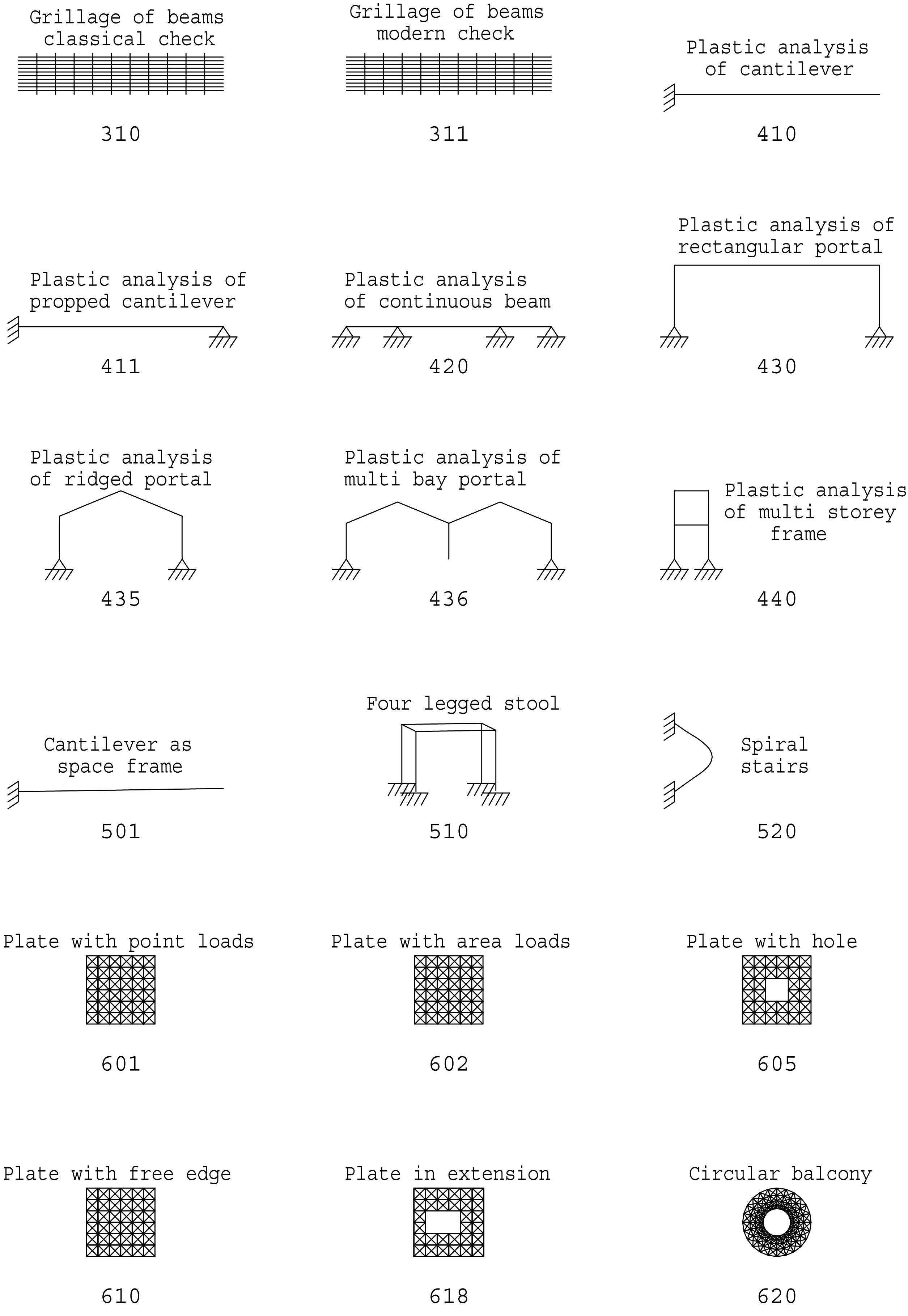


Figure 7.1 Verified models vm310 to vm620

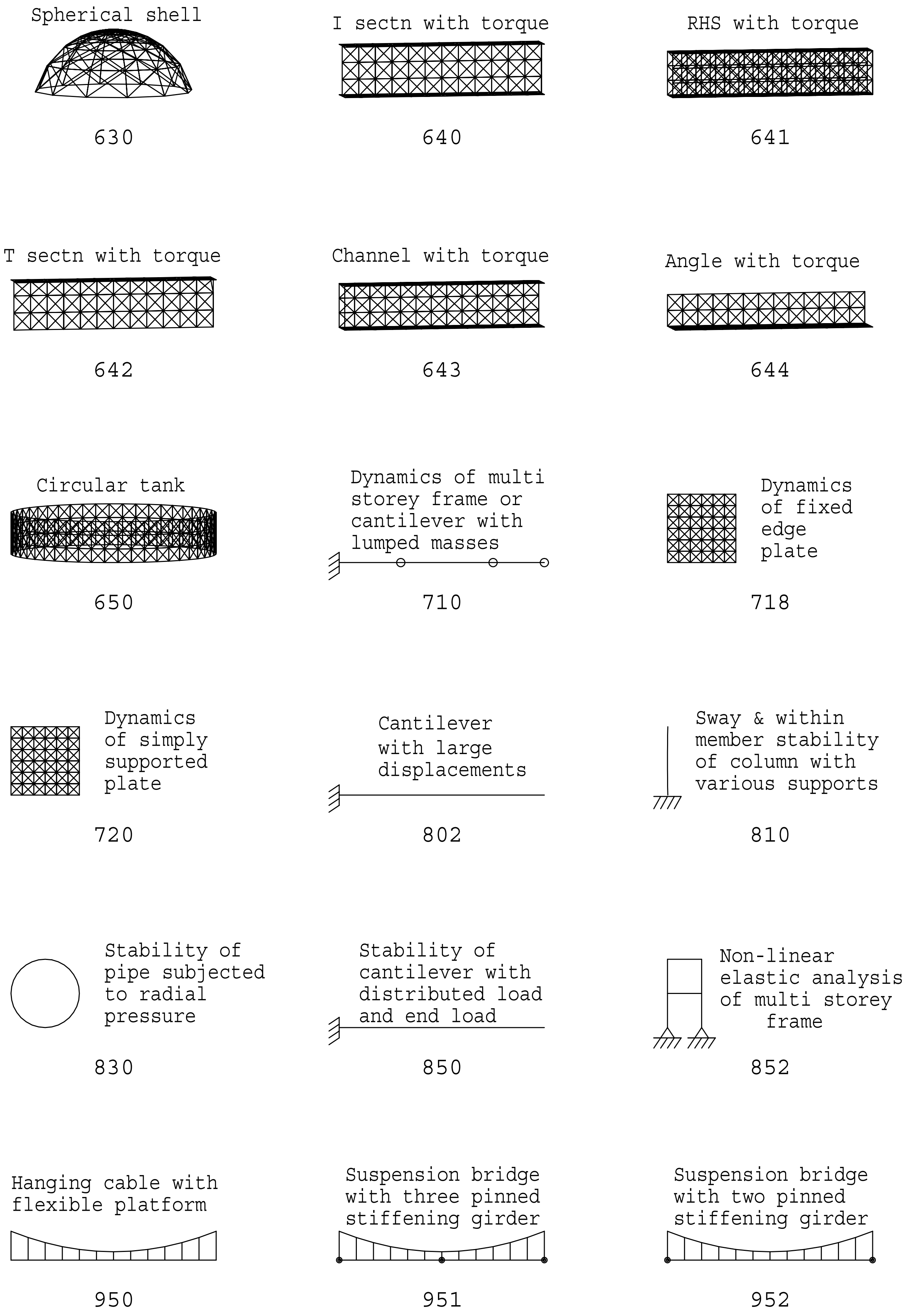


Figure 7.1 Verified models vm630 to vm952

7.9 Structure/form of each verified model

Figure 7.1 shows the structure/form of each verified model for the structural analysis of a framework. Figure 7.2 gives a diagrammatic overview on the self checks used for the structural analysis of frameworks viz:

- classical methods e.g. moment distribution
- modern methods *e.g.* compatibility and equilibrium checks for each member.

Figure 7.3 shows a typical self check using the modern method developed in this research.

Every model starts with some help. Help commences with an exclamation mark or an asterisk. Only help which starts with an asterisk is sent to the results file.

Next: page heading information, results, type of structure, method of analysis follow keywords: STRUCTURE, MADEBY, DATE, REFNO, TABULATE, PRINT, TYPE, METHOD.

Next: figure showing the structure, followed by parameter assignments using default values for each parameter.

#cc924.stk ! to import sets of data from file cc924.stk which overwrite the default values when the model is being verified *i.e.* run in batch mode for a thousand sets of data.

Commands defining the size of the structure *i.e.* NUMBER OF JOINTS, MEMBERS, SUPPORTS, LOADINGS, INCREMENTS, SEGMENTS.

Tables of material and geometric properties *i.e.* JOINT COORDINATES, JOINT RELEASES, MEMBER INCIDENCES, MEMBER RELEASES, CONSTANTS, MEMBER PROPERTIES.

Loading cases: for each case comes tables of JOINT LOADS & DISPLACEMENTS; MEMBER LOADS, DISTORTIONS, TEMPERATURE CHANGES, SELF WEIGHTS, LENGTH COEFFICIENTS; followed by loading combination commands: COMBINE, MAXOF, MINOF, ABSOF.

SOLVE starts the analysis after which results are available.

See The self check by classical methods in figure 7.2 also by equilibrium, compatibility & energy methods in Fig. 7.3.

FINISH marks the end of the structural data.

Figure 7.1 Structure/form of each verified model.

```
Self check for the structural
analysis of frameworks
```

```
Twenty classical methods
& reference to appendix A
for self checking model

Castigliano            vm290
Chebyshev & Rolfe      vm110
Column analogy         vm210
Euler                  vm810
Finite differences     vm610
Flexibility            vm131
Flex. & latent root    vm710
Hardy Cross            vm120
Hetényi                vm130
Kleinlogel             vm220
Magnus                 vm270
Method of joints       vm150
MΔller-Breslau         vm140
Navier                 vm601
Pilkey & Chang         vm310
Pippard & Baker        vm280
Reynolds               vm211
Roark                  vm620
Timoshenko             vm640
Unit load method       vm113
Warburton              vm718
```

```
Modern methods, tools & logic
enabling the self check, all
between SOLVE & FINISH commands

Compatibility
Local & global equilibrium
Clark Maxwell, Betti, Southwell
Strain energy

Assignments e.g. a=3*(b+c)^2
Functions e.g. SIN(a), EXP()
IF <condition/s> <lines> ENDIF
IF <condition/s> THEN ...
IF <condition/s> GOTO <label>
Looping   REPEAT
          <lines>
          UNTIL <condition/s>
          <lines>
          ENDREPEAT
#<filename> invoke procedure
ARR(No,row,col) selects result
          from the arrays file
Number of segments/member: to
give moments etc. along member
reqd. for strain energy check
```

Figure 7.2 The self check.

Equilibrium, compatibility & energy checks for a space frame			
Description of check for totals for all members	NL-STRESS result	Check value	%age diff.
Equilibrium - net moment y	1552.1686	1552.1524	
Equilibrium - net moment z	5446.5742	5441.9053	
Flexure - change in slope y	-0.1952	-0.1951	
Flexure - change in slope z	-0.2341	-0.234	
Change in displacement y	1.9217	1.9216	
Change in displacement z	2.5411	2.5409	
Axial - net length	59.9956	59.9976	
Axial - rotation	.1497	.1497	
Strain energy vs external work	131.7434	133.6321	1
Σ Forces X vs applied forces	-336	-336	
Σ Forces Y vs applied forces	331.84	331.84	
Σ Forces Z vs applied forces	-336	-336	
Σ Moments X vs applied moments	-3069.0802	-3068.8016	
Σ Moments Y vs applied moments	1883.3228	1883.3105	
Σ Moments Z vs applied moments	3211.2978	3211.2176	
Clerk Maxwell, Betti, Southwell	.3844	.3794	
Average difference .0625 %			

Figure 7.3 The self check by equilibrium, compatibility & energy.

Each model starts with the following comment lines to provide help, each comment line commences with an exclamation mark. For reason of space, these comment lines are omitted from the listing of the kernels of all models given in appendix A.

```
! This verified model is parametric. General notes follow FINISH.
! A semi-colon preceded by a space is used to separate statements.
! Parameters precede '!' help which follows, is copied to results.
! VEC (short for VECtor, a rank-one array), is used for multiple
! assignments: a(7)=VEC(3.2,b,-5.7) ≡ a(7)=3.2, a(8)=b, a(9)=-5.7;
! cs(1)=VEC(12,2.8)*2 ≡ cs(1)=12, cs(2)=2.8, cs(3)=12, cs(4)=2.8.
```

A typical verified model will be found in appendix B/6 to B/10. After the help, come up to four lines commencing with the keyword STRUCTURE or TITLE, both of which are treated the same, which are combined with information following keywords MADEBY, DATE and REFNO, to form a page heading for the results.

Commands PRINT & TABULATE define what is to be contained in the results; structural data starts with keywords METHOD (which may be omitted if linear elastic) and TYPE. The form of the data is the same as that for the well-known program called STRESS (1964) developed in the early 1960's; but there are additional features for dealing with non-linearity, control, looping, assignment, parametric data... all subjects not catered for in STRESS. Although it is over 40 years since the STRESS language was developed, keywords such as: JOINT COORDINATES, MEMBER INCIDENCES, JOINT LOADS, MEMBER LOADS are as applicable today as they were then. It makes little sense to change the words just for the sake of change; what has changed is the program itself; not a single line from the MIT STRESS program has survived.
Beneath the page heading information, comes a figure which contains structural information shown parametrically. Either beneath or at the side of the figure, the parameters are assigned with *default values*, which the engineer overtypes with specific data.
Following the last parameter assignment, is a line commencing #cc924.stk, the # directs NL-STRESS to import a stack of parameters and their values contained in the file cc924.stk and thus reset any preceding assignments. The import of data allows the model to be run hundreds of times, each time importing a different set of data and producing and filing the results for each set. Between the *import* line and the keyword SOLVE, comes all the data for analysing the structural problem by the matrix stiffness method. Between the keywords SOLVE and FINISH comes post-processing for which there are three components:

- a solution to the problem using the same data but based on a classical structural method/s such as moment distribution, Hetényi, Kleinlogel, flexibility, Roark... and thus independent from the matrix stiffness method a solution to the problem using the same data but based on self checks using equilibrium, compatibility and energy i.e. modern methods as summarised in figure 7.3.

- a comparison of the results of the matrix stiffness method with those of the self check whether by classical or modern methods, giving percentage differences for key values together with an average percentage difference for all key values.

The #vmper.ndf between the SOLVE & FINISH commands, invokes the following procedure.

```
zz=0 ! Compute %age diff. between d1 & d2 & write text message.
IF ABS(d1)<1E-8 AND ABS(d2)<1E-8 OR d1=d2 THEN zz=1 d1=0 d2=0
IF ABS(d1-d2)<1E-4 THEN zz=1 d1=0 d2=0
IF d1=0 AND d2<>0 THEN zz=2
IF d2=0 AND d1<>0 THEN zz=2 ;IF zz=0 THEN zz=d1/d2
IF zz<0 OR zz>2 THEN zz=2 ;IF zz<1 THEN zz=1/zz ;IF zz>2 THEN zz=2
nur=nur+1 per=INT(ABS(100-100*zz)+.5) d1=INT(per/10) d2=per-d1*10
IF d1=0 THEN d1=-1 ;IF per<100 THEN ok=(d1+1)*39+d2+1
IF per>99 THEN ok=$(>99) ;IF per<1 THEN ok=0 ;gtot=gtot+per
```

Following the FINISH command, comes a line containing GENERAL NOTES which tells the program to display help extracted from the program manual to aid the engineer in understanding the model whilst editing the data to change the default values to specific values. This help is neither sent to the results, nor to the edited data file.

The PARAMETER table lists the reference number and name of each parameter, associated *Start value*, *End value* and *Type* of parameter, expressed as a number to denote *e.g.* real, integer, set... The table contains all the data for building a thousand different sets of data for testing the model; satisfactory completion of all sets of data constitutes Verification. Each parameter table is unique to the model. Occasionally *Dependency conditions* specified to the right of the table need further explanation, when this is the case, the explanation will be found in a section entitled NOTES ON THIS VERIFIED MODEL, which also gives references, theory and formulae as appropriate to the model. To help the engineer during preliminary design, engineering help is included in the notes *e.g.* "Moduli of subgrade reaction". Finally come CONCLUSIONS from running the comparison tests between the matrix stiffness method and the included classical or modern self check.

The hope is that, one day, the set of verified models may be considered as a modern day Kleinlogel; with all due respect to the great man, the models encompass a wider range of structural frames than *Rahmenformeln* (Kleinlogel, 1952), include a self check, and are neither prone to arithmetic error nor error in copying values between the formulae. The verified models included in appendix A, are the start of a library. The development of the library, was characterised by many returns to the start as new facilities and improvements were made to the system to cater for new requirements identified as the work progressed *e.g.* spring supports, plastic hinges *etc.*

Chapter 8

Compatibility, energy & equilibrium

The IStructE Guidelines for the use of computers for engineering calculations, (Harris *et al.* 2002), in section 3.2 advise *Do not use a model that is more complex than necessary*, and in sub-section 3.5.2 advise *Check data and verify output*. This chapter assumes that the engineer has chosen a model which is not more complex than necessary and that the engineer has checked the data. This chapter is concerned with verifying the results of a structural analysis *i.e.* self-checking the structural analysis of a framework. Classical solutions provide bedrock beneath the matrix stiffness method of analysis, this chapter develops verification for structural frameworks for which no classical solutions exist. The method developed is general and is thus more powerful than any classical method.

After the structural analysis of a framework has been carried out, the member end forces and displacements for each member are known. Equilibrium and compatibility must be satisfied for each member. For simplicity, consider a plane frame member with ends A and B with the local z axis out of the media, the sum of the moments about z at ends A & B must balance the applied loading in the local y direction taking into account any change in length of the member. If the length of the member increases, applied distributed loads normal to the member must be scaled *down* and loading positions scaled up and vice versa so that the net loading on the member equals the original applied loading. The satisfaction of compatibility requires that the final length of a member should equal the original length plus the change in length due to the applied axial loading taking due account of signs, referred to as the **Net length check** in the subsequent discussion. The change in slope and change in displacement checks referred to as **Flexure checks** in the subsequent discussion are required also for compatibility. If the difference in any of the checks exceeds say 1%, then a non-linear analysis will be appropriate and the checks repeated. It is essential that both equilibrium and compatibility are satisfied *e.g.* the satisfaction of equilibrium checks for a guyed mast to less than 1% difference, but with compatibility checks showing a difference of say 7%, gives results which are in error. Non-linear elastic analysis should give all equilibrium and compatibility checks within 3% difference. Overall equilibrium checks are more straightforward, whether the analysis is non-linear or linear elastic, the applied loading in its displaced position should balance the support reactions to within 3% difference for: ΣX=0 ΣY=0 ΣZ=0 ΣMX=0 ΣMY=0 ΣMZ=0. Satisfactory equilibrium and compatibility checks for each member combined with overall equilibrium checks identify any problems with correctness of the calculations. MacLeod (2005) 3.5.3 warns **If the equilibrium is not to an accuracy that is close to that used in the solution (i.e. to 12 significant figures) then, provided the data is correct, ill-conditioning is likely. Surprisingly, the reverse is not true: satisfactory equilibrium checks do nor guarantee absence of ill-conditioning problems."** Other checks which are available and which are incorporated into the verified models include: ensuring that the external work done equals the internal strain energy plus energy stored in spring supports; reciprocity, described in section 8.9. The combination of all the checks described in this chapter are effective for identifying errors due to:

- **incorrect analysis** *e.g.* assuming linear elastic analysis for a sway frame
- **software shortcuts** *e.g.* lumping member self weights to member ends
- **incorrect derivation** of stiffness matrices particularly member end springs
- **strain energy imbalances** due to incorrect modelling *e.g.* plastic hinges
- **incorrect assessment** of elastic critical loads due to the use of approximate methods
- **algorithmic errors** *e.g.* in bandwidth minimisation, member distortions *etc.*
- forgetting Pippard & Baker's (1957) "In the absence of an external disturbing force, all perfect struts, whether slender or stocky, will fail by direct compression".

8.1 The particular solution

The most common structural model is that for a simply supported beam and will be used to discuss concepts.

```
             ┌w load/unit length
 Y│▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
  A════════════════════B ───X
  ▲       C            ▲
 7/7                  ooo
  ├──x──┤
  ├─────────L──────────┤
```

Taking moments to left of C.

Mc = x.wL/2 - wx^2/2

$$i.e.\ EI.\frac{d^2y}{dx^2} = \frac{wLx}{2} - \frac{wx^2}{2}$$

This equation relates the bending moment ($EI.d^2y/dx^2$) with the distributed load (w), length of beam (L) and the distance from the origin (x). Such an equation containing differential coefficients is called a differential equation. The general solution to this differential equation may be found by double integration to be:

$$y = -\frac{wx^4}{24EI} + \frac{wLx^3}{12EI} + Cx + D$$

where C & D are the two constants arising from the integrations. For the case of a simply supported beam, y=0 at x=0, thus D=0; also y=0 at x=L thus C=wL^3/(24EI) and the deflection of the simply supported beam at any distance x from the origin is:

$$y = -\frac{wx^4}{24EI} + \frac{wLx^3}{12EI} + \frac{wL^3.x}{24EI}.\ \text{For } x=kL,\ y = \frac{wL^4}{24EI}\left[-k^4 + 2k^3 + k\right]$$

Differentiate to obtain slope, equate to zero for a maximum:

$$\frac{dy}{dk} = \frac{wL^4}{24EI}\left[-4k^3 + 6k^2 + 1\right] \quad \text{therefore } -4k^3 + 6k^2 + 1 = 0$$

Solving this cubic, there are 2 real roots, k=1.366 and k=0.5, ignore the first as 1.366 is out of the range 0 to 1. Thus maximum deflection is when k=0.5 *i.e.* x=L/2 as every engineer will know. Substituting x=L/2, gives the well known: ymax = 5wL^4/384EI.

This simple model shares features with many classical structural solutions:

- the differential equation is devised
- the differential equation is solved to provide a general solution
- the boundary conditions are used to eliminate the constants arising from integrations, thus providing the particular solution.

Beams, including those on an elastic foundation, circular, square and rectangular plates all follow the above procedure, though the solution of the bi-harmonic for plates is more complicated. This research uses classical particular solutions, whenever these are available, for the verification of each structural analysis, when classical particular solutions are not available, verification is by the general procedure presented in this chapter.

The particular solution, in this case giving the deflection *y* at any point along a simply supported beam, will be obvious to most engineers, less obvious is that substituting values for the parameters in the particular solution will often give the wrong deflection:

- when the beam is timber and the short term E value has not been divided by 16 to allow for creep deflection
- when the beam is concrete and the short term E value has not been divided by 2.5 to 4 to allow for creep deflection
- when the beam is concrete and the moment of inertia I is the gross inertia rather than that for the transformed section
- when the beam is concrete, the transformed section has been used, the engineer has allowed for creep by estimating a creep factor of 2 but forgot that E should be divided by the creep factor +1 (to include for the short term deflection)
- when the beam is concrete, the transformed section has been used, the engineer has allowed for creep by estimating a creep factor of 2 and dividing E by 3, but has assumed the short term E taken from BS 8110 and omitted to take into account that this value is for Thames Valley aggregates and in other areas, aggregates give much lower values for E
- when the beam is steel and the span:depth ratio is less than 10, as shear deformation is not included in this particular solution. (For a UB of span:depth ratio =3.33, shear deformation doubles the deflection computed when shear deformation is omitted.)

The late Dr Fred Dibnah MBE, in his last programme in the series entitled *Made in Britain*, first screened by BBC2 on 17.05.05, reminisced about apprenticeships when "old men sat next to young apprentices". Check lists, included with each verified model, would help to pass on experience. Check lists, section 3.8, in memory of Fred, should form an integral part of every checking procedure.

Verified model vm110.NDF is for the analysis of a simply supported beam, and the deflections at the ends of each of *nsg* segments are checked against the particular solution, given above, but extended for the effect of shear deformation. Verification is not just checking the results of one analysis against the particular solution for the model, but providing extensive *coverage* by a thousand different sets of data checked against the particular solution for the model. The generation of a thousand sets of data, the structural analysis of the thousand sets of generated data, the checking of the sets of results against the particular solution for the model, the reporting of percentage differences between each set of results and the particular solution, is automated. When all percentage differences are within typically 3%, the maximum acceptable percentage difference being dependent on the design assumptions, then the model is said to be a *verified model*.

For statically determinate structures, such as pin-ended trusses, for which the applied loading is carried by axial forces in the members, rather than by bending, a particular solution may be found by the *Method of Joints*. The method of joints is the classical method for the analysis of pin-jointed trusses in which the support reactions are first computed by equilibrium *i.e.* applying ΣX=0 ΣY=0 ΣMZ=0, and then proceeding from the left support such that only two unknown member forces occur at each joint *i.e.* the same sequence that an engineer would follow in the manual analysis of a truss.

For statically indeterminate structures, such as continuous beams, portals, multi-storey frames, pierced shear walls, arches, bow girders, regular grillages of beams, rectangular & circular plates, spherical shells & circular tanks, suspension bridges, and the stability of: plates, pipes, columns & cantilevers, classical analysis methods exist; where appropriate they are used in this research to verify the results of an analysis.

Generally structural analysis involves 2D & 3D frames for which the applied loading produces: axial, bending, shear and torsional strain energy. For such problems, where a classical solution does not exist and a particular solution for the complete structure is intractable, a particular solution may be derived for each and every member in the structure by extending the particular solution derived above for bending in the beam model to include the additional axial, shear, and where appropriate, torsional effects.

8.2 The verified conjecture

If each and every member in a structure satisfies the particular solution for the member *i.e.* satisfies equilibrium and compatibility, and that the total strain energy stored in the structure equals the external work done, and that overall equilibrium (*e.g.* ΣX=0 ΣY=0 ΣMZ=0 for plane frames) is satisfied for a thousand sets of engineered data providing extensive *coverage* for each and every parameter over practical ranges, with each and every parameter both increasing and decreasing through its range with respect to every other parameter, then the model has been verified. The corollary is likely to have a higher degree of certainty, that is:

8.3 The incorrectness conjecture

If the results of any member in a structure do not agree with the particular solution for the member, or that the total strain energy stored in the structure does not equal the external work done, or that overall equilibrium is not satisfied for one or more of a thousand sets of test data providing extensive *coverage*, then the model will be in error.

8.4 Verifying the data

If engineers ran independent analysis checks on every structural analysis, or carried out equilibrium, compatibility and energy audit checks on the results of each analysis, then this facet of this research would be unnecessary. Occasionally, engineers carry out spot checks to ensure that bending moments or shears, balance at critical joints; when they believe that balance is not obtained, they phone the writer; always the phone call is due to confusion over direction of forces. The writer responds *"Think of the joints as separate from the members, the forces and their directions are what the joints do to the members"*. It is clear that engineers have neither the time nor the structural tools to verify the results of every analysis, thus yet more software is needed to do thorough checks on the results.

The IStructE Guidelines for the use of computers for engineering calculations, (Harris *et al.* 2002), in sub-section 3.5.2 advise *check data and verify output*. There is no way that a computer can check that the data provided by the engineer is correct, *e.g.* only the engineer can know that he/she intended to include an extra 20 kN for a water tank and lead bund, but forgot; thus the engineer must verify the data.

8.5 Verifying the output

This research advocates that *verifying the output* should automatically follow the results of the structural analysis and thus use identical data to that used in the analysis, and that the results of the checks continue on from the results of the analysis *i.e.* in the same document. The equilibrium, compatibility, energy audit and other checks to verify the output are not intended to replace current checks, but to supplement them. Wherever possible, and essential for *state of the art* structures, completely independent checks by an independent engineer are recommended.

Texts books do not advocate that members be checked for the satisfaction of equilibrium and compatibility (flexure and axial and shear loads) nor do they advocate that the total strain energy in a structure be checked to see that it equals the external work done. Some structural analysis programs claim to do an equilibrium check but assume that all deflections are negligible and consequently only check the accuracy of the arithmetic carried out by the computer. A proper overall equilibrium check must take into account the displaced positions of all the applied loads and in general must satisfy: $\Sigma X=0$ $\Sigma Y=0$ $\Sigma Z=0$ $\Sigma MX=0$ $\Sigma MY=0$ $\Sigma MZ=0$.

8.6 Plane frame verification

Procedures for carrying out checks on the results of an analysis, of necessity involve algebra and logic and symbolic names, colloquially *variables*. The following are those used.

```
Segment coordinates at start & end       xcs ycs      xce yce
      displacements at start & end       xds yds zrs  xde yde zre
             forces at start & end       fxs fys mzs  fxe fye mze
              nodes at start & end       nst          nen
Original & final length, dirn. cosines   lo lf        cx cy
Start/end displ. in global x, y & z: xds' yds' zrs' xde' yde' zre'
Start/end displ. in local x, y & z dirn:  xds yds zrs  xde yde zre
Elastic, shearing moduli, member props   e g          ax ay iz
Member springs stiffness at start & end  ks           ke
The various checks, numbered 1 to 8 follow:
Equilibrium check:  net moment nl1   balancing moment          ch1
Flexure:       change in slope nl2   balancing slope           ch2
        change in displacement nl3   balancing displacement    ch3
Axial:              net length nl4   should equal              ch4
Energy audit:    strain energy nl5   external work done        ch5
ΣX=0               ΣX reactions nl6   Σapplied loads in X dirn. ch6
ΣY=0               ΣY reactions nl7   Σapplied loads in Y dirn. ch7
ΣM=0            ΣM about origin nl8   ΣM from applied loads     ch8
```

For the last three checks, as Σreactions and Σapplied loads have opposite signs, the negative of ch6, ch7 & ch8 are taken when computing the percentage differences. Although in the following discussion, values are assigned directly to nl(1:8) & ch(1:8) for a single member or segment, to keep the summary of percentage differences short, the values for each effect are accumulated by *e.g.* nl1=nl1+... ch1=ch1+... nl2=nl2+... ch2=ch2+... and so on, thus avoiding thousands of pages of summary when the engineer has chosen 100 segments per member for a structure that has say 10,000 members, resulting in 8,000,000 lines *cf.* 8 lines when each effect is accumulated.

Let us consider a plane frame member number *mn*, having *nsg* segments, with start node number *j*, and end node number *k*, with its own local axes, subjected to a uniformly distributed load *wx* along the member *i.e.* in the direction of local *x*, and a uniformly distributed load *wy* normal to the member, *i.e.* in the direction of local *y*. We need to consider *wx,* as uniformly distributed loads applied to inclined members are frequently applied in the global Y direction thus generating components in both local *x* and local *y* directions.

```
local y
    |   ┌wy load/unit length
    |▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲
    j►►►►►►►►►►►►►►►►►►k ───local x
       └wx load/unit length
```

To check the results of the structural analysis for the member against the particular solution for the member, we need to extract: coordinates, member forces and displacement at ends *j* & *k* from the results. After the command SOLVE has been processed, all the results of the analysis are available in arrays. A formal description of the arrays' file is given in table B.1 in appendix B.

For member number *mn*, the start & end nodes, *nst* & *nen*, are found from:

```
nst=ARR(1,mn,1) nen=ARR(1,nm,2).
```

Having found the node numbers, the start & end coordinates are found from:

```
xcs=ARR(8,nst,3) ycs=ARR(8,nst,4) xce=ARR(8,nen,3)
yce=ARR(8,nen,4)
```

The member properties for a plane frame are:

```
ax=ARR(11,mn,1) ay=ARR(11,mn,2) iz=ARR(11,mn,6).
```

ARR(8,1,n) holds the joint number for node *n*, and ARR(8,2,j) holds the node number for joint *j*; we need these as the original joint numbers, *i.e.* the joint numbers used in the data and results, are mapped internally to a set of node numbers to minimise the bandwidth of the stiffness matrix. Array 6 holds the joint displacements in *npd* rows, where *npd* is the product of number of joints and number of degrees of freedom per joint, 3 for a plane frame. The row number in Array 6 for displacement in the global X direction, for node number *nst*, is **rn=3*(nst-1)+1**, the next row being for the Y direction, the next again being for: about Z. Thus the displacements at the start & end of member number *mn* in the global directions (denoted by ') for loading *nli* are:

```
rn=3*(nst-1)+1 xds'=ARR(6,rn,nli) rn=rn+1 yds'=ARR(6,rn,nli)
rn=rn+1 zrs'=ARR(6,rn,nli) rn=3*(nen-1)+1 xde'=ARR(6,rn,nli)
rn=rn+1 yde'=ARR(6,rn,nli) rn=rn+1 zre'=ARR(6,rn,nli).
```

Finally we need the forces at the member ends, these are held in Array 13, which has *nls*nm* rows, where *nls* is the product of the total number of loadings and the number of increments per loading. Thus the forces at start & end of member number *mn* of *nm* members, for loading increment *nli* (fxs fys mzs fxe fye mze) are given by:

```
nr=(nli-1)*nm+mn fxs=ARR(13,nr,1) fys=ARR(13,nr,2)
mzs=ARR(13,nr,3) fxe=ARR(13,nr,4) fye=ARR(13,nr,5)
mze=ARR(13,nr,6)
```

which completes the basic values needed to do an equilibrium & compatibility check and energy audit on member number *mn*.

Having found the member forces & displacements at the ends of each segment, equilibrium, compatibility & energy checks may be carried out. The names of variables given above are used in the following checks, the * as generally used in computer programs denotes multiplication. Firstly some necessary algebra; the original member length:

```
lo=SQR((xce-xcs)^2+(yce-ycs)^2); and final length:
lf=SQR((xce+xde'-xcs-xds')^2+(yce+yde'-ycs-yds')^2).
```

Direction cosines for the member:

```
cx=(xce+xde'-xcs-xds')/lf cy=(yce+yde'-ycs-yds')/lf
```

End displacements in local axes:

```
xds=cx*xds'+cy*yds' xde=cx*xde'+cy*yde' yds=-cy*xds'+cx*yds'
yde=-cy*xde'+cx*yde'.
```

Equilibrium check

Essentially this check is making sure that bending moments about the start end of each member balance.

Net moment `nl1=mzs+mze`

Balancing moment `ch1=fye*lf+wy(mn)*lo*lf/2`

i.e. the balancing moment is the end shear times the final length plus the bending moment from the total udl *i.e.* `wy(mn)*lo` times half the final member length.

Flexure checks

To take into account shear strain energy, a shear deformation coefficient *s* is defined as follows: for zero shear area **s=1**, else shear deformation coefficient `s=1/(1+12*e*iz/(lf^2*g*ay))`.

The derivation of *s* and the stiffness matrices for 2D & 3D structures are given in the NL-STRESS reference manual. For plane frames, from the stiffness matrix, where *mz* & *rz* denote moment & rotation respectively about the z axis; suffixes *s* & *e* denote the start & end of a segment, and *lo* denotes the segment original length:

$$\texttt{mzs} = \frac{\texttt{e*iz}}{\texttt{lo}} \left[\texttt{6*yds*s/lo+zrs*(1+3*s)-6*yde*s/lo+zre*(3*s-1)} \right]$$

$$\texttt{mze} = \frac{\texttt{e*iz}}{\texttt{lo}} \left[\texttt{6*yds*s/lo+zrs*(3*s-1)-6*yde*s/lo+zre*(1+3*s)} \right]$$

$$\texttt{Subtracting, mzs-mze} = \frac{\texttt{e*iz}}{\texttt{lo}} \left[\texttt{2*zrs-2*zre} \right]$$

```
thus change in slope (zrs-zre)=(mzs-mze)*lo/(2*e*iz).
```

As expected, shear deformation has no influence on the change in slope, however the change in slope is influenced by any loading within the member. Although simply supported & continuous beams are subjected to partial uniform, triangular & trapezoidal loads, general plane frame structures usually involve only distributed load components normal-to, and along the members. Obviously a udl along a member *wx(mn)* will not affect the change in slope between the ends of the member, the change in slope due to a udl *wy(mn)* normal to a member is `wy(mn)*lo*lf^2/(12*e*iz)` therefore:

```
change in slope nl2=zrs-zre
should equal ch2=lf*(mzs-mze)/(2*e*iz)+wy(mn)*lo*lf^2/(12*e*iz).
```

From the stiffness matrix, where *mz* & *zr* denote moment & rotation respectively about the z axis; suffixes *s* & *e* denote the start & end of a segment, and *lf* denotes segment *final length*.

$$\texttt{mzs} = \frac{\texttt{e*iz}}{\texttt{lf}} \left[\texttt{zrs*(1+3*s)+zre*(3*s-1)+6*s/lf*(yds-yde)} \right]$$

$$mze = \frac{e*iz}{lf}\left[zrs*(3*s-1)+zre*(1+3*s)+6*s/lf*(yds-yde) \right]$$

$$\text{Adding, } mzs+mze = \frac{e*iz}{lf}\left[6*s*(zrs+zre)+12*s*(yds-yde)/lf \right]$$

Rearranging: `lf*(mzs+mze)/(e*iz)-6*s*(zrs+zre) =12*s*(yds-yde)/lf`
therefore `yds-yde =lf^2*(mzs+mze)/(12*e*iz*s)-lf*(zrs+zre)/2,`
change in slope `(zrs-zre)=(mzs-mze)*lf/(2*e*iz)`, therefore
change in displacement `nl3=yds-yde`
should equal `ch3=lf^2*(mzs+mze)/(12*e*iz*s)-lf*(zrs+zre)/2.`

Net length check `nl4=lf`
should equal `ch4=lo+(fxe-fxs)/2*lf/(ax*e).`

Strain energy check

For a straight bar of length L, of axial stiffness Ax.E, subjected to a compressive axial load P, the change in length is given by Hooke as δ=PL/(Ax.E) ...(a).

Axially loaded linear elastic materials behave as a spring, thus as the load is applied, the axial force increases from zero to P, thus the work done =(0+P).δ/2. By the principle of conservation of energy, the external work done on the bar must be stored within the bar as strain energy. Let the strain energy due to axial load be denoted Ua, then we can write Ua=P.δ/2. Substituting for δ in (a), then Ua=P^2L/(2.Ax.E).

For a plane frame, the bending, shear and axial strain energy, must be considered in an energy audit. When the length of the bar is made sufficiently short (in the limit), we can write:

$$\text{Bending, } Ub=\int_0^L \frac{M^2.dl}{2.E.Iz} \qquad \text{Shear, } Us=\int_0^L \frac{F^2.dl}{2.Ay.G} \qquad \text{Axial, } Ua=\int_0^L \frac{P^2.dl}{2.Ax.E}$$

As the bending moment, shear force & axial load usually vary along a member, it is necessary to have values for all three strain energy components at various positions along each member. Although it is clear that bending moments & shear forces vary along members, axial loads also vary along members due to axial loading along members *e.g.* for two rafters meeting at a ridge subjected to distributed gravity loading on plan applied to the members, the axial compressive load in each rafter is a minimum at the ridge, increasing to a maximum at the eaves.

To obtain bending moments, shear forces & axial loads, at various positions along a member, NL-STRESS provides a command: NUMBER OF SEGMENTS <n>, where n may be any integer number in the range 1 to 100, at which forces & displacements at the ends of each segment of a member are evaluated. To save paper, all the additional results for the segment ends are not normally written to the results; for assurance, the engineer can output displacements and forces at the intermediate joints by adding the keyword TRACE to the end of the NUMBER OF SEGMENTS command.

The higher the number of segments, the better the strain energy audit, for the computation of external work done by the distributed loads assumes that half the load, from each segment on either side of each internal joint, is lumped to the joint position between adjacent segments; start and end joints receiving only one half of the segment load. When members are segmented, NL-STRESS automatically renumbers the members, thus if *nsg=4*, then *external* member number 1 becomes *internal* members 1 to 4 in order, external member number 2 becomes internal members 5 to 8, and so on. Henceforth *mn* refers to the internal member number.

Work done vs. strain energy

Work done `ch5=ch5+fx*(xds'+xde')/2+fy*(yds'+yde')/2`

`fa=(fxs-fxe)/2 fy=(fys-fye)/2 mz=(mzs-mze)/2`

St. energy `nl5=nl5+fa^2*lf/(2*e*ax)+fy^2*lf/(2*g*ay)+mz^2*lf/(2*e*iz)`

Finally, **overall equilibrium checks**. For a plane frame, checking ΣX=0 ΣY=0 ΣMZ=0, are required, these are available as:

```
ΣX for the applied forces                    =ARR(12,4,10)
ΣY for the applied forces                    =ARR(12,4,11)
ΣM for the applied forces about the origin  =ARR(12,4,12)
ΣX reaction computed by NL-STRESS           =ARR(12,4,13)
ΣY reaction computed by NL-STRESS           =ARR(12,4,14)
ΣM for the reactions about the origin       =ARR(12,4,15).
```

One hundred and eight structural models have been developed and verified for correctness. For each model, a thousand sets of data were generated using the system developed as part of this research; the results of running the data using the matrix stiffness method were compared with the results obtained by analysis using either a classical method or compatibility, equilibrium and energy checks discussed above.

The procedure for the above theory now follows; this procedure is contained in the file *vmecp.ndf* which is called from plane frame models which require it. The following should be read in conjunction with the forgoing explanation.

```
! va() udl along local x direction, vb() when times fac.
! vc() udl in local y dirn, vd() when times fac. ;meth=ARR(12,4,9)
! Check case 1 ;nli=ARR(12,4,1) lli=ARR(12,4,2) nsg=ARR(12,4,4)
nmo=ARR(12,4,6) njo=ARR(12,4,7) fac=1 ;! Prorata lds. ;IF lli<nli
fac=lli/nli ch9=0 ;ENDIF ;! Equil, com. & energy. ;mn=0
nm=nmo*nsg ;:698 ;mn=mn+1 k=INT((mn-1)/nsg)+1 vb(mn)=fac*va(k)
vd(mn)=fac*vc(k) ;IF mn<nm GOTO 698 ! Incr. seg. loads with conc.
IF status=1E-36 GOTO 697 ;nc=0 ;:697 ;IF nc<1 GOTO 700 ;i=0 ;:699
i=i+1 sno=nc(i) mn=INT(nsg*cs(i)/s(sno)+0.5) ;IF mn<1 THEN mn=1
IF mn>nsg THEN mn=nsg ;mn=mn+nsg*(sno-1) ;vb'=fac*ct(i)*nsg/s(sno)
vb(mn)=vb(mn)+vb' vd'=fac*cn(i)*nsg/s(sno)
```

```
vd(mn)=vd(mn)+vd' ;IF i<nc GOTO 699 ;:700 ;jn=0 ;:701 ;jn=jn+1
hjl(jn)=fac*hjl(jn) vjl(jn)=fac*vjl(jn) ;IF jn<njo GOTO 701
nl(1)=VEC(0)*8 ch(1)=VEC(0)*8 mn=0 ;:702 ;mn=mn+1 ;! End node Nos.
nst=ARR(1,mn,1) nen=ARR(1,mn,2) ;! Coordinates ;xcs=ARR(8,nst,3)
ycs=ARR(8,nst,4) xce=ARR(8,nen,3) yce=ARR(8,nen,4) ;! Sectn props.
ax=ARR(11,mn,1) ay=ARR(11,mn,2) iz=ARR(11,mn,6) e=ARR(11,mn,11)
g=ARR(11,mn,12) ;! Displ in global axes ;rn=3*(nst-1)+1
xds'=ARR(6,rn,lli) rn=rn+1 yds'=ARR(6,rn,lli) rn=rn+1
zrs'=ARR(6,rn,lli) rn=3*(nen-1)+1 xde'=ARR(6,rn,lli) rn=rn+1
yde'=ARR(6,rn,lli) rn=rn+1 zre'=ARR(6,rn,lli)
! The original member length ;lo=SQR((xce-xcs)^2+(yce-ycs)^2)
! Fin length ;lf=SQR((xce+xde'-xcs-xds')^2+(yce+yde'-ycs-yds')^2)
! Memb forces ;rn=(lli-1)*nm+mn xfs=ARR(13,rn,1) yfs=ARR(13,rn,2)
zms=ARR(13,rn,3) xfe=ARR(13,rn,4) yfe=ARR(13,rn,5)
zme=ARR(13,rn,6) xds=ARR(13,rn,13) yds=ARR(13,rn,14)
zrs=ARR(13,rn,15) xde=ARR(13,rn,16) yde=ARR(13,rn,17)
zre=ARR(13,rn,18) ks=ARR(1,mn,6) ke=ARR(1,mn,7) nlp5=0
IF meth=3 ;mzcs=ARR(10,rn,4) mzce=ARR(10,rn,8)
nlp5=(zrs'-zrs)*mzcs/2+(zre'-zre)*mzce/2 ;ENDIF
! Shear def. coeff. ;s=1 ;IF ay>0 THEN s=1/(1+12*e*iz/(lf^2*g*ay))
! Dir cosines ;cx=(xce+xde'-xcs-xds')/lf cy=(yce+yde'-ycs-yds')/lf
! Equilibrium check; Net mmt ;nl1=nl1+ABS(zms+zme)
ch1=ch1+ABS(yfe*lf+vd(mn)*lo*lf/2) ;IF iz=0 GOTO 703
! Change in slope ;nl2=nl2+zrs-zre
ch2=ch2+lf*(zms-zme)/(2*e*iz)+vd(mn)*lo*lf^2/(12*e*iz)
! Change in displacement ;nl3=nl3+ABS(yds-yde)
ch3=ch3+ABS(lf^2*(zms+zme)/(12*e*iz*s)-lf/2*(zrs+zre)) ;:703
nl4=nl4+lf ch4=ch4+lo+(xfe-xfs)/2*lf/(ax*e) ;! Work vs. strain en.
fx=cx*vb(mn)*lo/2-cy*vd(mn)*lo/2 fy=cy*vb(mn)*lo/2+cx*vd(mn)*lo/2
ch5=ch5+fx*(xds'+xde')/2+fy*(yds'+yde')/2 fa=(xfs-xfe)/2 fyse=0
fy=(yfs-yfe)/2 mz=(zms-zme)/2 ;IF ay>0 THEN fyse=fy^2*lf/(2*g*ay)
nl5'=0 ;IF iz>0 THEN nl5'=mz^2*lf/(2*e*iz)
nl5=nl5+fa^2*lf/(2*e*ax)+fyse+nl5'+nlp5
IF meth<>3 AND ks>0 THEN nl5=nl5+zms^2/ks/2
IF meth<>3 AND ke>0 THEN nl5=nl5+zme^2/ke/2
IF mn<nm GOTO 702 ;! Jnt w.d. ;jn=0 ;:705 ;jn=jn+1
nn=ARR(8,jn,2) rn=3*(nn-1)+1 xd=ARR(6,rn,lli) xsp=ARR(7,rn,1)
rn=rn+1 yd=ARR(6,rn,lli) ysp=ARR(7,rn,1) rn=rn+1 zr=ARR(6,rn,lli)
zsp=ARR(7,rn,1) ch5=ch5+hjl(jn)*xd/2+vjl(jn)*yd/2
! Jt spring energy ;IF xsp>0 THEN nl5=nl5+xsp*xd^2/2
IF ysp>0 THEN nl5=nl5+ysp*yd^2/2 ;IF zsp>0 THEN nl5=nl5+zsp*zr^2/2
IF jn<njo GOTO 705
! Eq. overall ;nl6=ARR(12,4,16) nl7=ARR(12,4,17) nl8=ARR(12,4,18)
ch6=ARR(12,4,10) ch7=ARR(12,4,11) ch8=ARR(12,4,12) ;*/11
* Description of check for        NL-STRESS    Check        %age
* totals for all members          result       value        diff.
status=1 gtot=0 nur=0 d1=nl1 d2=ch1 iret=720
:710 ;z=0 ! Compute %age dif. 'tween d1 & d2 & write text message.
IF ABS(d1)<1E-8 AND ABS(d2)<1E-8 OR d1=d2 THEN z=1 d1=0 d2=0
IF ABS(d1-d2)<1E-4 THEN z=1 d1=0 d2=0
IF d1=0 AND d2<>0 OR d2=0 AND d1<>0 THEN z=2 ;IF z=0 THEN z=d1/d2
IF z<0 OR z>2 THEN z=2 ;IF z<1 THEN z=1/z ;IF z>2 THEN z=2
nur=nur+1 per=INT(ABS(100-100*z)+.5) d1=INT(per/10) d2=per-d1*10
IF d1=0 THEN d1=-1 ;IF per<100 THEN ok=(d1+1)*39+d2+1
IF per>99 THEN ok=$(NOT OK) ;IF per<1 THEN ok=0 ;gtot=gtot+per
GOTO iret ;:720
* Equilibrium  - net moment       +nl1         +ch1         $ok
d1=nl2 d2=ch2 ;iret=730 ;GOTO 710 ;:730
* Flexure change in slope         +nl2         +ch2         $ok
d1=nl3 d2=ch3 ;iret=740 ;GOTO 710 ;:740
* Change in displacement          +nl3         +ch3         $ok
d1=nl4 d2=ch4 ;iret=750 ;GOTO 710 ;:750
```

```
* Net length                           +nl4          +ch4          $ok
d1=nl5 d2=ch5 ;iret=760 ;GOTO 710 ;:760
* Strain energy vs external work   +nl5          +ch5          $ok
d1=nl6 d2=-ch6 ;iret=770 ;GOTO 710 ;:770
* d React. X vs applied forces     +nl6          +-ch6         $ok
d1=nl7 d2=-ch7 ;iret=780 ;GOTO 710 ;:780
* d React. Y vs applied forces     +nl7          +-ch7         $ok
d1=nl8 d2=-ch8 ;iret=790 ;GOTO 710 ;:790
* d React. Z vs applied moments    +nl8          +-ch8         $ok
IF ch9=0 GOTO 810 ;nla=2*lli nlb=3*lli nl9=0 ch9=0 jn=0 ;:811
jn=jn+1 nn=ARR(8,jn,2) nr=(nn-1)*3+1 dela1=ARR(6,nr,nla)
delb1=ARR(6,nr,nlb) nr=nr+1 dela2=ARR(6,nr,nla)
delb2=ARR(6,nr,nlb) nr=nr+1 dela3=ARR(6,nr,nla)
delb3=ARR(6,nr,nlb) jn'=njo+1-jn nl9=nl9+jn*(delb1+delb2+delb3)
ch9=ch9+jn'*(dela1+dela2+dela3) IF jn<njo GOTO 811
d1=nl9 d2=ch9 ;iret=800 ;GOTO 710 ;:800
* Clerk Maxwell, Betti, Southwell  +nl9          +ch9          $ok
:810 ;IF ch10=0 GOTO 850 ;d1=nl10 d2=ch10 iret=820 ;GOTO 710 ;:820
:850
```

8.7 Plane grid verification

Procedures for carrying out checks on the results of an analysis require the use of *variables*; the following are those used.

```
Segment coordinates at start & end       xcs ycs      xce yce
      displacements at start & end       xrs yrs zds  xre yre zde
             forces at start & end       xms yms zfs  xme yme zfe
              nodes at start & end       nst          nen
Original length                          lo
Start/end displ in global x, y & z:  xrs' yrs' zds' xre' yre' zde'
Start/end displ in local x, y & z dirn:  xrs yrs zds  xre yre zde
Elastic, shearing moduli, member props   e g          ix iy az
Member springs stiffness at start & end  kxs kys      kxe kye
The various checks, numbered 1 to 8 follow:
Equilibrium check:  net moment nl1   balancing moment          ch1
Flexure:       change in slope nl2   balancing slope           ch2
        change in displacement nl3   balancing displacement    ch3
Twisting:            net twist nl4   should equal              ch4
Energy audit:    strain energy nl5   external work done        ch5
ΣX=0           ΣX about origin nl6   ΣX from applied loads     ch6
ΣY=0           ΣY about origin nl7   ΣY from applied loads     ch7
ΣZ=0              ΣZ reactions nl8   Σapplied loads in Z dirn. ch8
```

For the last three checks, as Σreactions and Σapplied loads have opposite signs, the negative of ch6, ch7 & ch8 are taken when computing the percentage difference. Although in the following discussion, values are assigned directly to nl(1:8) & ch(1:8) for a single member or segment, to keep the summary of percentage differences short, the values for each effect are accumulated, as outlined previously.

Let us consider a plane grid member number *mn*, having *nsg* segments, with start node number *j*, and end node number *k*, with its own local axes, subjected to a uniformly distributed torque *wx* about the member *i.e.* about the direction of local *x*, and a uniformly distributed load *wz* normal to the member.

```
local z
     │  ┌wz load/unit length
     │▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲
     j┌►┌►┌►┌►┌►┌►┌►┌►┌►k ───local x
      └ └ └ └ └ └ └ └ └
        └wx twisting moment (torque) per unit length
```

To check the results of the structural analysis for the member, against the particular solution for the member, we need to extract: coordinates, member forces and displacement at ends *j* & *k* from the results. After the command SOLVE has been processed, all the results of the analysis are available in arrays.

For member number *mn*, the start & end nodes, *nst* & *nen*, are found from:

```
nst=ARR(1,mn,1) nen=ARR(1,nm,2).
```

Having found the node numbers, the start & end coordinates are found from:

```
xcs=ARR(8,nst,3) ycs=ARR(8,nst,4) xce=ARR(8,nen,3)
yce=ARR(8,nen,4)
```

The member properties for a plane grid are:

```
ix=ARR(11,mn,4) iy=ARR(11,mn,5) az=ARR(11,mn,3).
```

ARR(8,1,n) holds the joint number for node *n*, and ARR(8,2,j) holds the node number for joint *j*; we need these as the original joint numbers, *i.e.* the joint numbers used in the data and results, are mapped internally to a set of node numbers to minimise the bandwidth of the stiffness matrix. Array 6 holds the joint displacements in *npd* rows, where *npd* is the product of number of joints and number of degrees of freedom per joint, 3 for a plane grid. The row number in Array 6 for displacement about the global X direction, for node number *nst*, is **rn=3*(nst-1)+1**, the next row being for about the Y direction, the next again being for the Z direction. Thus the displacements at the start & end of member number *mn* in the global direction (denoted by ') for loading *nli* are:

```
rn=3*(nst-1)+1 xrs'=ARR(6,rn,nli) rn=rn+1 yrs'=ARR(6,rn,nli)
rn=rn+1 zds'=ARR(6,rn,nli) rn=3*(nen-1)+1 xre'=ARR(6,rn,nli)
rn=rn+1 yre'=ARR(6,rn,nli) rn=rn+1 zde'=ARR(6,rn,nli).
```

Finally we need the forces at the member ends, these are held in Array 13, which has *nls*nm* rows, where *nls* is the product of number of loadings and the number of increments per loading. Thus the forces at the start & end of member number *mn* of *nm* members, for loading increment *nli* (xms yms zfs xme yme zfe) are given by:

```
nr=(nli-1)*nm+mn xms=ARR(13,nr,1) yms=ARR(13,nr,2)
zfs=ARR(13,nr,3) xme=ARR(13,nr,4) yme=ARR(13,nr,5)
zfe=ARR(13,nr,6)
```

which completes the basic values needed to do an equilibrium & compatibility check and energy audit on member number *mn*. Having found the member forces & displacement at the ends of each segment, equilibrium, compatibility & energy checks may be carried out. The variable names used above are used in the following checks, the * as generally used in computer programs denotes multiplication. Firstly some necessary algebra. In a plane grid, no account is taken of axial load, thus there is no change in the axial length of any member and in consequence the check on net length carried out for a plane frame is not appropriate to a plane grid member.

The member length `lo=SQR((xce-xcs)^2+(yce-ycs)^2)`.

Equilibrium check:

Essentially this check is making sure that bending moments about the start end of each member balance.

Net moment `nl1=yms+yme`

Balancing moment `ch1=zfe*lo+wz(mn)*lo^2/2`

i.e. the balancing moment is the end shear times the length plus the bending moment from the total udl *i.e*. `wz(mn)*lo` times half the member length.

Flexure check:

To take into account shear strain energy, a shear deformation coefficient *s* is defined as follows: for zero shear area **s=1**, else shear deformation coefficient `s=1/(1+12*e*iy/(lo^2*g*az))`.

The derivation of *s* and the stiffness matrices for 2D & 3D structures are given in the NL-STRESS reference manual. For plane grids, from the stiffness matrix, where *my* & *ry* denote moment & rotation respectively about the *y* axis; suffixes *s* & *e* denote the start & end of a segment, and *lo* denotes the segment length:

$$\mathtt{yms} = \frac{\mathtt{e*iy}}{\mathtt{lo}}\left[\ \mathtt{6*zds*s/lo+yrs*(1+3*s)-6*zde*s/lo+yre*(3*s-1)}\ \right]$$

$$\mathtt{yme} = \frac{\mathtt{e*iy}}{\mathtt{lo}}\left[\ \mathtt{6*zds*s/lo+yrs*(3*s-1)-6*zde*s/lo+yre*(1+3*s)}\ \right]$$

$$\mathtt{Subtracting,\ yms-yme} = \frac{\mathtt{e*iy}}{\mathtt{lo}}\left[\ \mathtt{2*yrs-2*yre}\ \right]$$

`thus change in slope (yrs-yre)=(yms-yme)*lo/(2*e*iy).`

As expected, shear deformation has no influence on the change in slope, however the change in slope is influenced by any loading within the member. Although simply supported & continuous beams are subjected to partial uniform, triangular & trapezoidal loads, general plane grid structures usually involve only distributed load components normal to, and about the members. Obviously a udl about a member *wx(mn)* will not affect the change in slope about the y axis of the member, the change in slope due to a udl *wz(mn)* normal to a member is **wz(mn)*lo^3/(12*e*iy)** therefore:

```
    change in slope nl2=yrs-yre
    should equal ch2=lo*(yms-yme)/(2*e*iy)+wz(mn)*lo^3/(12*e*iy).
```

From the stiffness matrix, where *ym* & *yr* denote moment & rotation respectively about the *y* axis; suffixes *s* & *e* denote the start & end of a segment, and *lo* denotes segment length.

$$\mathtt{yms} = \frac{\mathtt{e*iy}}{\mathtt{lo}}\left[\ \mathtt{yrs*(1+3*s)+yre*(3*s-1)+6*s/lo*(zds-zde)}\ \right]$$

$$\mathtt{yme} = \frac{\mathtt{e*iy}}{\mathtt{lo}}\left[\ \mathtt{yrs*(3*s-1)+yre*(1+3*s)+6*s/lo*(zds-zde)}\ \right]$$

$$\text{Adding, yms+yme} = \frac{e*iy}{lo}\left[\ 6*s*(yrs+yre)+12*s*(zds-zde)/lo\ \right]$$

Rearranging: `lo*(yms+yme)/(e*iy)-6*s*(yrs+yre) =12*s*(zds-zde)/lo`
therefore `zds-zde = lo^2*(yms+yme)/(12*e*iy*s)-lo*(yrs+yre)/2`,
change in slope `(yrs-yre)=(yms-yme)*lo/(2*e*iy)`,
therefore change in displacement `nl3=zde-zds`
should equal `ch3=lo^2*(yms+yme)/(12*e*iy*s)-lo*(yrs+yre)/2`.

Twisting moment: `net rotation nl4=xre-xrs`
should equal: `ch4=(xme-xms)*lo/(2*ix*g)`.

Strain energy check:
For a straight bar of length L, of torsional stiffness Ix.G, subjected to a torque T, the change in twist in radians is given by Θ=T.L/(Ix.G) (a).

Twisted linear elastic materials behave as a spring, thus as the load is applied, the torque increases from zero to T, thus the work done =(0+T).Θ/2. By the principle of conservation of energy, the external work done on the bar must be stored within the bar as strain energy. Let the strain energy due to twisting be denoted Ut, then we can write Ut=T.Θ/2. Substituting for Θ in (a), then Ut=T^2L/(2.Ix.G).

For a plane grid, the bending, shear and torsional strain energy, must be considered in an energy audit. When the length of the bar is made sufficiently short (in the limit), we can write:

$$\text{Bending, Ub} = \int_0^L \frac{M^2.dl}{2.E.Iy} \qquad \text{Shear, Us} = \int_0^L \frac{F^2.dl}{2.Az.G} \qquad \text{Torque, Ua} = \int_0^L \frac{T^2.dl}{2.Ix.G}$$

As the bending moment, shear force & torque (twisting moment) usually vary along a member, it is necessary to have values for all three strain energy components at various positions along each member. Although it is clear that bending moments & shear forces vary along members, torques also vary along members due to twisting moments applied within the length of the members.

To obtain bending moments, shear forces & twisting moments, at various positions along a member, NL-STRESS provides a command: NUMBER OF SEGMENTS <n>, where n may be any integer number in the range 1 to 100, at which forces & displacements at the ends of each segment of a member are evaluated. To save paper, all the additional results for the segment ends are not normally written to the results; for assurance, the engineer can output displacements and forces at the intermediate joints by adding the keyword TRACE to the end of the NUMBER OF SEGMENTS command.

The higher the number of segments, the better the strain energy audit, for the computation of external work done by the distributed loads assumes that half the load, from each segment on either side of each internal joint, is lumped to the joint position between adjacent segments; start and end joints receiving only one half of the segment load. When members are segmented, NL-STRESS automatically renumbers the members, thus if *nsg=4*, then *external* member number 1 becomes *internal* members 1 to 4 in order, external member number 2 becomes internal members 5 to 8, and so on. Henceforth *mn* refers to the internal member number.

Next comes the work done, where *va()* is the torsional moment about local *x* axis, *vb()* when times fac; *vc()* is the udl in the Z direction, *vd()* when times *fac*, where *fac* is the ratio of the number of loading increments safely carried/total number of loading increments.
Where `mx=vb(mn)*lo/2 fz=vd(mn)*lo/2` then the external work done:
`ch5=ch5+mx*(xrs'+xre')/2+fz*(zds'+zde')/2` should agree with the internal work done. For `mx=(xms-xme)/2 my=(yms-yme)/2 fz=(zfs-zfe)/2`
which are the average values of torque, shear & moment for the segment, to avoid division by zero when the torsion constant or shear area *az* is not given:
`mxse=0 ;IF ix>0 THEN mxse=mx^2*lo/(2*g*ix)`
`fzse=0 ;IF az>1E-12 THEN fzse=fz^2*lo/(2*g*az)`, cumulating strain energies, `nl5=nl5+mxse+fzse+my^2*lo/(2*e*iy)`.
When all such strain energy components have been accumulated for the members, then external work done by loads applied to the joints is added *e.g.* for joint number *jn:*
`nn=ARR(8,jn,2) rn=3*(nn-1)+1 xr=ARR(6,rn,nli) xsp=ARR(7,rn,1)`
`rn=rn+1 yr=ARR(6,rn,nli) ysp=ARR(7,rn,1) rn=rn+1 zd=ARR(6,rn,nli)`
`zsp=ARR(7,rn,1)`, where xsp, ysp & zsp are the spring stiffnesses. Cumulating the external work done at joint *jn*, `ch5=ch5+vl(jn)*zd/2`.

The strain energy stored in spring supports is added to the strain energy by:

```
IF xsp>0 THEN nl5=nl5+xsp*xr^2/2
IF ysp>0 THEN nl5=nl5+ysp*yr^2/2
IF zsp>0 THEN nl5=nl5+zsp*zd^2/2
```

Finally, **overall equilibrium checks**, for a plane grid checking ΣMX=0 ΣMY=0 ΣZ=0, are required, these are available as:

```
ΣMX for the applied moments                  ch6=ARR(12,4,10)
ΣMY for the applied moments                  ch7=ARR(12,4,11)
ΣZ for the applied forces                    ch8=ARR(12,4,12)
ΣMX reaction about the origin by NL-STRESS  nl6=ARR(12,4,16)
ΣMY reaction about the origin by NL-STRESS  nl7=ARR(12,4,17)
ΣZ reaction by NL-STRESS                     nl8=ARR(12,4,18).
```

The procedure for the previous theory now follows; this procedure is contained in the file *vmecg.ndf* which is called from plane grid models which require it. The following should be read in conjunction with the foregoing explanation. Array variables *va()*, *vb()*, *vc()* & *vd()* have their own stacks for quicker access. In the procedure below, the torque on the beams *ux()* in the foregoing is replaced by *va()* and *vb()* after multiplying by *fac*; the udl on the beams *wz()* in the foregoing is replaced by *vc()* and *vd()* after multiplying by *fac*.

```
! va() torsional moment about local x axis, vb() when times fac.
! vc() udl in Z direction, vd() when times fac.
! Check case 1 ;nli=ARR(12,4,1) nli'=ARR(12,4,2) nsg=ARR(12,4,4)
nmo=ARR(12,4,6) njo=ARR(12,4,7) fac=1 ;! Prorata lds. ;IF nli'<nli
fac=nli'/nli nli=nli' ch9=0 ;ENDIF ;! Equil, com. & energy. ;mn=0
nm=nmo*nsg ;:700 ;mn=mn+1 k=INT((mn-1)/nsg)+1 vb(mn)=fac*va(k)
vd(mn)=fac*vc(k) ;IF mn<nm GOTO 700 ;jn=0 ;:701 ;jn=jn+1
vl(jn)=fac*vl(jn) ;IF jn<njo GOTO 701
nl(1)=VEC(0)*8 ch(1)=VEC(0)*8 mn=0 ;:702 ;mn=mn+1 ;! End node Nos.
nst=ARR(1,mn,1) nen=ARR(1,mn,2) ;! Coordinates ;xcs=ARR(8,nst,3)
ycs=ARR(8,nst,4) xce=ARR(8,nen,3) yce=ARR(8,nen,4) ;! Sectn props.
az=ARR(11,mn,3) ix=ARR(11,mn,4) iy=ARR(11,mn,5) e=ARR(11,mn,11)
g=ARR(11,mn,12) ;! Displ in global axes ;rn=3*(nst-1)+1
xrs'=ARR(6,rn,nli) rn=rn+1 yrs'=ARR(6,rn,nli) rn=3*(nen-1)+1
xre'=ARR(6,rn,nli) rn=rn+1 yre'=ARR(6,rn,nli) ! Original memb len.
lo=SQR((xce-xcs)^2+(yce-ycs)^2) ;! Memb forces ;rn=(nli-1)*nm+mn
xms=ARR(13,rn,1) yms=ARR(13,rn,2) zfs=ARR(13,rn,3)
xme=ARR(13,rn,4) yme=ARR(13,rn,5) zfe=ARR(13,rn,6)
xrs=ARR(13,rn,13) yrs=ARR(13,rn,14) zds=ARR(13,rn,15)
xre=ARR(13,rn,16) yre=ARR(13,rn,17) zde=ARR(13,rn,18)
ksx=ARR(1,mn,6) ksy=ARR(1,mn,7) kex=ARR(1,mn,8) key=ARR(1,mn,9)
IF ksx>0 THEN nl5=nl5+xrs^2/ksx/2
IF ksy>0 THEN nl5=nl5+yrs^2/ksy/2
IF kex>0 THEN nl5=nl5+xre^2/kex/2
IF key>0 THEN nl5=nl5+yre^2/key/2
! Shear def. coeff. ;s=1 ;IF az>0 THEN s=1/(1+12*e*iy/(lo^2*g*az))
! Equilibrium check; Net mmt ;nl1=nl1+ABS(yms+yme)
ch1=ch1+ABS(zfe*lo+vd(mn)*lo^2/2) ;! Change slope ;nl2=nl2+yre-yrs
ch2=ch2+lo*(yme-yms)/(2*e*iy)+vd(mn)*lo^3/(12*e*iy)
! Change in displacement ;nl3=nl3+ABS(zde-zds)
ch3=ch3+ABS(lo^2*(yms+yme)/(12*e*iy*s)-lo/2*(yrs+yre)) ;! Twisting
IF ix>0 THEN nl4=nl4+ABS(xre-xrs)
IF ix>0 THEN ch4=ch4+ABS((xme-xms)*lo/(2*ix*g))
! Strain en. ;mx=vb(mn)*lo/2 fz=vd(mn)*lo/2
ch5=ch5+mx*(xrs'+xre')/2+fz*(zds+zde)/2 ! Udl torque & udl Z dirn.
mx=(xms-xme)/2 my=(yms-yme)/2 fz=(zfs-zfe)/2
mxse=0 ;IF ix>0 THEN mxse=mx^2*lo/(2*g*ix) ! St.en. due to mx.
fzse=0 ;IF az>1E-12 THEN fzse=fz^2*lo/(2*g*az) ! Shear st. en.
nl5=nl5+mxse+fzse+my^2*lo/(2*e*iy) ! Add bending st. en.
IF mn<nm GOTO 702 ;! Jnt w.d. ;jn=0 ;:705 ;jn=jn+1 nn=ARR(8,jn,2)
rn=3*(nn-1)+1 xr=ARR(6,rn,nli) xsp=ARR(7,rn,1) rn=rn+1
yr=ARR(6,rn,nli) ysp=ARR(7,rn,1) rn=rn+1 zd=ARR(6,rn,nli)
zsp=ARR(7,rn,1) ch5=ch5+vl(jn)*zd/2 ;! Jt spring energy
IF xsp>0 THEN nl5=nl5+xsp*xr^2/2 ;IF ysp>0 THEN nl5=nl5+ysp*yr^2/2
IF zsp>0 THEN nl5=nl5+zsp*zd^2/2 ;IF jn<njo GOTO 705
! Eq. overall ;nl6=ARR(12,4,16) nl7=ARR(12,4,17) nl8=ARR(12,4,18)
ch6=ARR(12,4,10) ch7=ARR(12,4,11) ch8=ARR(12,4,12) ;*/11
* Description of check for        NL-STRESS     Check        %age
* totals for all members          result        value        diff.
status=1 gtot=0 nur=0 d1=nl1 d2=ch1 iret=720
:710 ;z=0 ! Compute %age dif. 'tween d1 & d2 & write text message.
IF ABS(d1)<1E-8 AND ABS(d2)<1E-8 OR d1=d2 THEN z=1 d1=0 d2=0
IF ABS(d1-d2)<1E-4 THEN z=1 d1=0 d2=0
IF d1=0 AND d2<>0 OR d2=0 AND d1<>0 THEN z=2 ;IF z=0 THEN z=d1/d2
IF z<0 OR z>2 THEN z=2 ;IF z<1 THEN z=1/z ;IF z>2 THEN z=2
nur=nur+1 per=INT(ABS(100-100*z)+.5) d1=INT(per/10) d2=per-d1*10
```

```
IF d1=0 THEN d1=-1 ;IF per<100 THEN ok=(d1+1)*39+d2+1
IF per>99 THEN ok=$(NOT OK) ;IF per<1 THEN ok=0 ;gtot=gtot+per
GOTO iret ;:720
* Equilibrium  - net moment       +nl1         +ch1         $ok
d1=nl2 d2=ch2 ;iret=730 ;GOTO 710 ;:730
* Flexure change in slope         +nl2         +ch2         $ok
d1=nl3 d2=ch3 ;iret=740 ;GOTO 710 ;:740
* Change in displacement          +nl3         +ch3         $ok
d1=nl4 d2=ch4 ;iret=750 ;GOTO 710 ;:750
* Change in twisting rotation     +nl4         +ch4         $ok
d1=nl5 d2=ch5 ;iret=760 ;GOTO 710 ;:760
* Strain energy vs external work  +nl5         +ch5         $ok
d1=nl6 d2=-ch6 ;iret=770 ;GOTO 710 ;:770
* ΣMX React. vs applied forces    +nl6         +-ch6        $ok
d1=nl7 d2=-ch7 ;iret=780 ;GOTO 710 ;:780
* ΣMY React. vs applied forces    +nl7         +-ch7        $ok
d1=nl8 d2=-ch8 ;iret=790 ;GOTO 710 ;:790
* dFZ React. vs applied forces    +nl8         +-ch8        $ok
IF ch9=0 GOTO 810 ;nla=2*nli nlb=3*nli nl9=0 ch9=0 jn=0 ;:811
jn=jn+1 nn=ARR(8,jn,2) nr=(nn-1)*3+1 dela1=ARR(6,nr,nla)
delb1=ARR(6,nr,nlb) nr=nr+1 dela2=ARR(6,nr,nla)
delb2=ARR(6,nr,nlb) nr=nr+1 dela3=ARR(6,nr,nla)
delb3=ARR(6,nr,nlb) jn'=nj+1-jn nl9=nl9+jn*(delb1+delb2+delb3)
ch9=ch9+jn'*(dela1+dela2+dela3) IF jn<nj GOTO 811
d1=nl9 d2=ch9 ;iret=800 ;GOTO 710 ;:800
* Clerk Maxwell, Betti, Southwell  +nl9         +ch9         $ok
:810 ;IF ch10=0 GOTO 850 ;d1=nl10 d2=ch10 iret=820 ;GOTO 710 ;:820
:850
```

8.8 Space frame verification

The compatibility, energy, local & overall equilibrium and the Clerk Maxwell, Betti, Southwell checks are contained in the file *vmecs.ndf* in Appendix A, omitted here for reason of space. Space frames combine the behaviour of both plane frames and plane grids thus sections 8.6 & 8.7 may be read in conjunction with *vmecs.ndf*. For convenience a list of self-checks for space frames follows.

```
1 Equilibrium check: net moment y = balancing moment y
2                    net moment z = balancing moment z
3 Flexure:      change in slope y = balancing slope y
4               change in slope z = balancing slope z
5        change in displacement y = balancing displacement y
6        change in displacement z = balancing displacement z
7  Axial:              net length = balancing length
8  Rotation:         net rotation = balancing rotation
9  Energy audit:    strain energy = external work done
10 ΣX=0              ΣX reactions = Σapplied loads in X dirn
11 ΣY=0              ΣY reactions = Σapplied loads in Y dirn
12 ΣZ=0              ΣZ reactions = Σapplied loads in Z dirn
13 ΣMX=0         ΣMX about origin = ΣMX from applied loads
14 ΣMY=0         ΣMY about origin = ΣMY from applied loads
15 ΣMZ=0         ΣMZ about origin = ΣMZ from applied loads
16 Maxwell, Betti, Southwell.
```

For a 3D structure for which partial varying distributed loads may be applied to any member in or about any axis for BETA (the angle of rotation of the member about its centroidal axis) set to any value, the formulation of the data, although complicated, is treated rigorously by NL-STRESS. For a self check, it would be simple to pick up the loads after they have been distributed by NL-STRESS to the joints but this would impinge on the independence of the self check. After some deliberation, for the strain energy check it was decided to directly convert all loading on the members to the joints at the end of each segment, in much the same way as finite element analysis does, thus avoiding all the complications of fixed end moments *etc.* for partial distributed loads which vary in or about the 3 axes. As with the finite element method, the accuracy will be compromised by a coarse mesh, a suggested minimum number of segments per member is 16; the parametric formulation of the model makes it easy to experiment with variation of the number of segments.

8.9 Clerk Maxwell, Betti, Southwell

All the checks discussed in this chapter are for use as self checks when a classical method of structural analysis is not available. If two additional loading cases are added to a model for the structural analysis of a framework, then a further overall check may be provided by *reciprocity*. James Clerk Maxwell's reciprocal theorem may be stated thus, *"Suppose any elastic body, either solid or a framework, is supported in such a way that the reactive forces do no work when loads are applied to the body, then the displacement of B in the direction of W2 when a unit load acts at A in the direction of W1 is the same as the displacement of A in the direction of W1 when a unit load acts at B in the direction of W2."* A proof of this is given by Pippard & Baker (1957). In modern structural analysis, because *direction* means in or about the X, Y or Z axes, singly or in combination, the writer prefers the following words *the displacement of A due to unit load at B, is equal to the displacement of B due to unit load at A, both unit loads being in or about the same direction.* For a structure having either supported or unsupported edges which are either clamped or unclamped *i.e.* built-in or simply supported, then the reactive forces will do no work when loads are applied to the body.

Clerk Maxwell's theorem was extended by Betti and again by Southwell (1923). In the more general form due to Betti the reciprocal theorem may be stated as follows: Suppose that a number of forces P1,P2...Pn, act simultaneously upon a body which obeys Hooke's Law and that the displacements in the lines of action of these forces are respectively δ1,δ2...δn. If these forces are replaced by a second system P'1,P'2...P'n acting at the same points and in the same directions as those of the first system, the corresponding displacements being δ'1,δ'2 ...δ'n, then
P1.δ'1+P2.δ'2+...+Pn.δ'n =P'1.δ1+P'2.δ2+...+P'n.δn. As previously stated, *direction* means in or about the X, Y & Z axes, thus the above may be generalised further to combine *ndf* degrees of freedom at any joint appropriate to the type of frame being analysed; ndf=3 for a plane frame or grid, ndf=6 of a space frame. For example the logic to do this check for plane grids follows; the simple code tests loading in all directions of freedom at every joint. For non-linear analysis, the same procedure is applied unless the structure has collapsed. When the structure has collapsed, subsequent loading cases are ignored. As *reciprocity* is predicated on the basis of linear elastic behaviour, it will not apply to structures which form plastic hinges. From experiments with model vm850.ndf, changing the factor *a=1* in loading case 2 to *a=180*, thereby imposing loading which approached the buckling load, agreement between load cases 2 & 3 was within 14%. For loading below half the buckling load, the Clerk Maxwell, Betti, Southwell check agreed to within 8%, though verified model vm850.ndf, for a

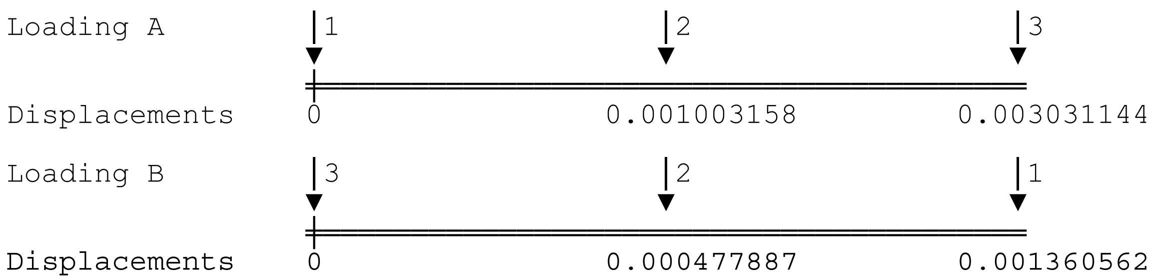
cantilever beam, is not typical of all structures, nevertheless the fact that any agreement was achievable was both surprising and interesting.

```
nla=2*nli nlb=3*nli nl9=0 ch9=0 jn=0 ;:811
jn=jn+1 nn=ARR(8,jn,2) nr=(nn-1)*3+1 dela1=ARR(6,nr,nla)
delb1=ARR(6,nr,nlb) nr=nr+1 dela2=ARR(6,nr,nla)
delb2=ARR(6,nr,nlb)
nr=nr+1 dela3=ARR(6,nr,nla) delb3=ARR(6,nr,nlb) jn'=nj+1-jn
nl9=nl9+jn*(delb1+delb2+delb3) ch9=ch9+jn'*(dela1+dela2+dela3)
IF jn<nj GOTO 811
```

In the above:

- nli=number of increments in which each loading is applied, =1 for elastic analysis.
- nla & nlb are pointers to loading cases 2 & 3 which respectively are the two loading cases a & b.
- nl9 cumulates the products of forces for loading a & displacements for loading b.
- ch9 cumulates the products of forces for loading b & displacements for loading a.
- the forces for loading a, are numerically equal to the joint number 1,2...nj.
- the forces for loading b, are numerically equal to the joint number nj,nj-1,nj-2...1.
- suffixes 1, 2 & 3 refer to directions: rotations about X & Y axes and displacement in direction Z.

A numerical example follows, for simplicity consider vertical loads and displacements only:



Product of loads of A with displacements of B

sigma= 1*0+2*0.000477887+3*0.001360562=0.0050371

Product of loads of B with displacements of A

sigmb= 3*0+2*0.001003158+1*0.003031144=0.0050373

As sigma=sigmb the Clerk Maxwell, Betti, Southwell check is OK. Where appropriate, the check is included as the ninth self check for plane frames and plane grids, and the sixteenth self check for space frames.

Chapter 9

Benchmarking

In the second paragraph of chapter 4 it was recommended that before use, engineering software should be benchmarked, *e.g.* comparing the results produced by the software with results which have been produced by at least one other program or published results in text books and papers. Another use for *benchmarking* is checking for changes in *results* caused by revisions to the software over a period of years. For example, the *benchmarking* described in section 9.6 is used to verify that the results of a structural analysis using modern matrix methods agrees with the results produced by classical methods. This involves running a batch of over a hundred models each with a thousand different sets of data. When differences occur in any of the 100,000 runs in the batch, an audit trail must be provided as described in section 9.3, so that the engineer can pinpoint any problems and draw conclusions. To do this, it is essential that intermediate results be provided and that the form of the intermediate results is text rather than binary.

In this research, benchmarks include: circular & rectangular flat, folded or curved plates; box girders & shear lag; square on square, square on diagonal space frames; guyed masts & transmission towers; beams & rafts on elastic foundations; single & multi-bay portal frames; lattice girders & lattice portals; Gangnail, Pratt, Howe, Warren, attic-room, collar-tie, collar-and-tie, couple, couple-close, Fink, Mansard, King & Queen post roof trusses; bents, trestles & pipe racks; beams & plates with trains of moving loads; coupled shear walls; circular & parabolic arches; beams curved on plan; multi-storey & multi-bay frames; continuous beams & sub-frames; bunkers, tanks & silos; highway & suspension bridges; hyperbolic paraboloid nets; post-buckling behaviour; snap through *etc.* The term *Benchmark* is defined in section 3.1 and qualified by:

9.1 The Inexact conjecture

In general all engineering calculations are compromised by the omission of one or more effects such as: axial or shear deformation, non-linear material properties, finite displacements, stability, fatigue, seismicity and other ambient conditions *etc.*, consequently all engineering calculations are at best, inexact.

The fact that engineering calculations are inexact, provides one need for benchmarks, as benchmarks provide a set of numerical results against which changes *e.g.* the inclusion of finite displacement effects in the model, may be assessed. Changes to software are another need for benchmarks, the process of *Benchmarking* is to ensure that the previous behaviour of software has not changed, or has changed as expected.

9.2 The Checksum conjectures

Section 3.1 defines the Uniqueness Theorem (Coates *et al.* 1988), the Principle of Saint Venant as restated by Pippard and Baker (1957); from these and engineering considerations such as discontinuities and common sense, the Checksum conjectures may be argued.

If the *checksum* of say 100-1000 discrete benchmarks equals that obtained on the previous test to 15 decimal digits of accuracy, then it may be concluded that the results of each discrete benchmark are identical to those of the previous test. Before issuing an update, it is recommended that the Checksum test be run on two different and unconnected computers with different versions of the system software using the previous version of the application software. The result will be referred to as the *previous pair of Checksum tests*. The exercise should be repeated using the new version of the application software, the result will be referred to as the *new pair of Checksum tests*. Ignoring trivial changes such as: altering the version number, formatting, pagination *etc.*, then conclusions may be drawn from the previous pair and new pair of Checksum values.

- If the previous pair of Checksum tests gave the same OK result AND if the new pair of Checksum tests are identical to the previous, then any alterations to the software have not compromised the results for the coverage provided by the range of benchmarks tested.
- If the previous pair of Checksum tests gave the same OK result AND if the new pair of Checksum tests are identical but different to the previous pair, then alterations to the software have compromised the results for the coverage provided by the range of benchmarks tested.
- If the previous pair of Checksum tests gave the same OK result AND if one of the new pair of Checksum tests is the same as the previous pair but the other differs, then alterations to the system software/hardware on the odd one out has compromised the result.
- And so on, to include for the previous pair of Checksum tests giving the same result which was subsequently found to be not OK.

Average percentage differences reported in chapter 12 are computed from summing the absolute values of the differences to yield a total, then dividing the total by the number of differences in the sample. *Checksum*, as used in benchmarking a set of results, has nothing to do with percentage differences, *checksum* is simply the sum of a set of key values, one from each model. Percentage differences are not involved, the main use of *checksum* is for testing to see if the value of *checksum* is unchanged following an update to the software. Consideration has been given to the use of absolute values in cumulating the *checksum* for sets of benchmarks. Taking absolute values of the key values could conceal a sign change bug between software releases. For this reason the use of absolute values for checksum for benchmarking is not recommended; though as stated at the start of this paragraph, absolute values are essential for computing the percentage differences reported in chapter 12.

9.3 Benchmark audit trail

If just one of Checksum's 15 decimal digits has changed, then there needs to be a meaningful explanation for customers and other interested parties such as those in technical support. An efficient way of doing this is by an annual newsletter supported by a website. Obviously, if Checksum has changed, then the first question will be "What has been effected?"; to answer this, a summary giving the key-values for each discrete benchmark has to be produced. It is then straightforward to look down the list and see if there is a pattern, or not, either result being salient. Of course if the summary contains only key-values, then a pattern will be difficult to discern. Thus key-values must be accompanied by an apt description.

It has been found convenient to batch benchmarks with approximately 100 benchmarks per batch. Three sets of benchmarks are given in sections 9.4 to 9.6, all three have a common audit trail commencing with *e.g.*

```
    Check of benchmarks for errors.
    OK \sand\BM01.BMK/b
    OK \sand\BM02.BMK/b
    and so on.
```

OK which denotes that a successful run has been completed or NOT OK... if a successful run has not been completed, is followed by the path to the file; the /b denotes that the run is in batch mode. For reason of space this first part of the audit trail is omitted. The second part of the audit trail is coded for brevity and tidiness to the eye when scanning the trail. It was found that scanning a hundred sets of benchmarks which had wordy notes such as:

```
Ref: BM01 Value 0.35166E-03 Load case 1 Joint 27 Joint Displacement
in X dirn.
```

was far more tiring than the shortened form finally adopted viz:

```
BM01 0.35166E-03 L1 27 JDX
```

Firstly comes the filename without its extension, this is followed by the key-value selected. The description of the key value then follows, coded thus:

- L prefixes the Loading (load case) number
- the integer number which follows the loading case number refers to either a joint number or a member number
- member forces commence with F denoting Force, or M denoting Moment, followed
- by a direction from X, Y or Z, followed by S denoting Start or E denoting End; selected from the set: FXS FYS FZS MXS MYS MZS FXE FYE FZE MXE MYE MZE
- member stresses commence with S denoting Stress, followed by F denoting Force, or

M denoting Moment, followed by a direction from X, Y or Z, followed by S denoting Start or E denoting End; selected from the set: SFXS SFYS SFZS SMXS SMYS SMZS SFXE SFYE SFZE SMXE SMYE SMZE

- joint displacement commence with J denoting Joint, followed by D denoting Displacement or R denoting Rotation followed by direction from X, Y or Z; selected from the set: JDX JDY JDZ JRX JRY JRZ
- support Reactions commence with R denoting Reaction, followed by F denoting Force or M denoting Moment followed by direction from X, Y or Z; selected from the set: RFX RFY RFZ RMX RMY RMZ
- natural frequencies commence with X, Y or Z denoting direction followed by HRZ denoting Hertz; selected from the set: X HRZ Y HRZ Z HRZ

9.4 Traditional Benchmarks

Traditional benchmarks give examples of data files for a wide range of engineering structures. Embedded in the data is part of the results, so that the problem may be run and the results obtained compared with those embedded in the data; therefore each/every data file may be used as a benchmark/s against which the results obtained from running the problem on a computer can be compared; references are also embedded in the data. The filename extension for each benchmark is given as .bmk (short for Benchmark) to distinguish the files from other files, and to facilitate the set being run in a batch. These benchmarks were collected over a number of years. All have been amended as part of this research to enable an audit trail, *i.e.* to process and report on the set automatically.

The first part of table 9.1 gives the filenames and a brief description, the second part of the summary gives an abbreviated version of the results obtained by running the benchmarks as a set.

Table 9.1 Traditional benchmarks.

Filename	TIMING BENCHMARKS
bm01.bmk	Plane frame with 27 joints, 38 members & 2 load cases.
bm02.bmk	Space frame with 66 joints, 99 members & 1 load case.
	DEPARTMENT OF TRANSPORT - HECB BENCHMARKS
dt01.bmk	Plane truss with varying relative stiffness.
dt02.bmk	Plane frame with displaced supports.
dt03.bmk	Plane frame problem (2).
dt04.bmk	Encastré segmental arch rib.
dt05.bmk	Grillage with applied displacements & elastic supports.
dt06.bmk	Grillage with shear deformation.

dt07.bmk	Skew deck of orthogonal grillage.
dt08.bmk	Circular-arc bow girder.
dt09.bmk	Space truss.
dt10.bmk	Space frame with varying stiffnesses & displaced supports.

DYNAMICAL BEHAVIOUR BENCHMARKS

dy01.bmk	Ex. from Fig 3.2, Warburton (1964).
dy02.bmk	Ex. from table 12.2, Steel Designers' Manual (1992).
dy03.bmk	Ex. from table 12.2, Steel Designers' Manual (1992).
dy04.bmk	Nat. freq. for point loads, Dunkerley method (Ryder, 1957).
dy05.bmk	Nat. freq. example 10.3-2 Coates *et al.* (1988).
dy06.bmk	Nat. freq. example in Fig 4.8, Warburton (1964).
dy07.bmk	Nat. freq. example problem 1 in chapter 1, Warburton (1964).
dy08.bmk	Nat. freq. example problem 7 in chapter 15, Ryder (1957).
dy09.bmk	Nat. freq. grid cl.12.15, Steel Designers' Manual (1992).

FINITE ELEMENT ANALYSIS BENCHMARKS

fe01.bmk	Plate in flexure, Yettram & Husain (1965).
fe02.bmk	Plate in extension, Yettram & Husain (1966).
fe03.bmk	Plates in extension & flexure, Yettram & Husain (1965/66).
fe04.bmk	Folded plate, figure 8.13, Rockey *et al.* (1983).
fe05.bmk	Torisphere with central hole.
fe06.bmk	Edge supported plate with only four elements.
fe07.bmk	Edge supported plate with sixty four elements.
fe08.bmk	Example Fig. 5.5, Rockey *et al.* (1983).
fe09.bmk	Example Fig. 6.3, Rockey *et al.* (1983).
fe10.bmk	Simply supported circular plate with UDL, Roark (1965).
fe11.bmk	Rectangular plate with UDL, Timoshenko *et al.* (1959).
fe12.bmk	RHS as fe03.bmk with span:depth ratio increased to 8:1.
fe13.bmk	Rect. plate with one free edge, Timoshenko *et al.* (1959).
fe14.bmk	Built-in rectangular plate, Timoshenko *et al.* (1959).
fe15.bmk	Channel section with destabilising end load, Roark (1965).
fe16.bmk	Circular shell from Fig 265, Timoshenko *et al.* (1959).
fe17.bmk	Circ. concrete tank from Fig 246, Timoshenko *et al.* (1959).
fe18.bmk	Modelling of element in extension & sign conventions.
fe19.bmk	Modelling of element in flexure & sign conventions.
fe20.bmk	Modelling of element in combined extension & flexure.

PLANE GRID BENCHMARKS

gr01.bmk	Bridge deck example, C&CA (1972).
gr02.bmk	Foundation raft, Sawko (1972).

gr03.bmk	Authentic bridge deck provided by Dr R. C. Slater.
gr04.bmk	Curved balcony member from Design Ex. 6, SCI (2001).
gr05.bmk	Member stresses for sections defined by props or geometry.

PLANE FRAME BENCHMARKS

pf01.bmk	Shear wall, MacLeod (April, 1966).
pf02.bmk	Box culvert.
pf03.bmk	Influence lines by Müller-Breslau, Coates *et al.* (1988).
pf04.bmk	Natural frequency determination, McMinn (1962).
pf05.bmk	Prestressed continuous beam, Lin (1963).
pf06.bmk	Shear deformation - Ex. 6.7-1 by Coates *et al.* (1988).
pf07.bmk	Member loads - Example 6.7-2 by Coates *et al.* (1988).
pf08.bmk	Symmetry - Example 6.10-1 by Coates *et al.* (1988).
pf09.bmk	Looping example, Problem 6.1 by Coates *et al.* (1988).
pf10.bmk	Looping across tables - Pr. 6.2 Coates *et al.* (1988).
pf11.bmk	Springs at supports - Pr. 7.18 by Coates *et al.* (1988).
pf12.bmk	Applied moments - Problem 8.5 by Coates *et al.* (1988).
pf13.bmk	DIAGRAMS example - Pr. 6.14 by Coates *et al.* (1988).
pf14.bmk	Propping force - Problem 6.16 by Coates *et al.* (1988).
pf15.bmk	Member distortions - Pr. 4.15 by Coates *et al.* (1988).
pf16.bmk	Temperature, self weights, length coefficients example.
pf17.bmk	Curved member, Design example 6, BCC 842, SCI (2001).
pf18.bmk	Temperature gradient, example from Emkin *et al.* (1977).
pf19.bmk	Stresses for sections defined by properties or geometry.
pf20.bmk	Member properties given by: AS other member properties.

PLASTIC ANALYSIS BENCHMARKS

pl01.bmk	Single bay portal frame, Morris & Randall, (1997).
pl02.bmk	Two storey frame, Horne & Merchant, Fig 5.14, (1965).
pl03.bmk	Plastic grillage, Example 1.0, Morris & Randall, (1977)
pl04.bmk	Elastic-plastic analysis of compression members.
pl05.bmk	Reversing plastic hinge example.
pl06.bmk	Built-in beam, Example 1.1, Morris & Randall, (1997).
pl07.bmk	Propped cantilever, Example 1.2, Morris & Randall, (1997).
pl08.bmk	Two span beam, Example 2.1, Morris & Randall, (1997).
pl09.bmk	Three span beam, Example 2.3, Morris & Randall, (1997).
pl10.bmk	Single ridged portal, Example 4.4, Morris & Randall, (1997).
pl11.bmk	Two bay ridged portal, Ex. 4.7, Morris & Randall, (1997).
pl12.bmk	Multi-storey frame, Example 6.2, Morris & Randall, (1997).
pl13.bmk	Test order of formation of plastic hinges.
pl14.bmk	Portal frame with out of plane loading.

pl15.bmk	Space frame - ring beam supported on RHS columns.
pl16.bmk	Example 14.6-1 from Coates *et al.* (1988).
pl17.bmk	Rect. portal, Example 14.6-4 from Coates *et al.* (1988).
pl18.bmk	Non-symmetric portal, Ex. 14.6-5 from Coates *et al.* (1988).
pl19.bmk	Non-symm. 2 bay portal, Ex. 14.7-1 from Coates *et al.* (1988).
pl20.bmk	Two storey portal, Example 14.7-2 from Coates *et al.* (1988).
pl21.bmk	Collapse load factor, Ex. 14.8-1 from Coates *et al.* (1988).
	PLANE TRUSS BENCHMARKS
pt01.bmk	Example 20, Gennaro (1965).
pt02.bmk	Example 22, Gennaro (1965).
pt03.bmk	Chapter 4 Problem 1, Gennaro (1965).
pt04.bmk	Chapter 4 Problem 4, Gennaro (1965).
pt05.bmk	Chapter 4 Problem 13, Gennaro (1965).
pt06.bmk	Example 31, Gennaro (1965).
pt07.bmk	Example 32, Gennaro (1965).
pt08.bmk	Example 33, Gennaro (1965).
pt09.bmk	Example 34, Gennaro (1965).
pt10.bmk	Example 4.10, Grassie (1957).
	SPACE FRAME BENCHMARKS
sf01.bmk	Cantilever stair.
sf02.bmk	Guide dolphin.
sf03.bmk	Example in figure 3-8, Weaver (1967).
sf04.bmk	Example in figure 3-9, Weaver (1967).
sf05.bmk	Example from UCC symposium Nov 1972.
sf06.bmk	Tapered beams example - equivalent to rect. section.
sf07.bmk	Cantilever with various loadings, Steel Designers' Man (1966).
sf08.bmk	S.S. beam with various loadings, Steel Designers' Man. (1966).
sf09.bmk	Built-in beam with various loadings, Steel Des. Man. (1966).
sf10.bmk	Ring beam on T columns to show need for BETA angle.
sf11.bmk	Curved balcony member for SCI design example 6.
sf12.bmk	Curved balcony member from Design Ex. 6, SCI (2001).
sf13.bmk	Member distortions for cantilever or built-in beam.
sf14.bmk	Temperature gradient, Emkin *et al.* (1977).
sf15.bmk	Stresses for sections defined by properties or geometry.
	SWAY FRAME BENCHMARKS
sw01.bmk	Column with axial load, figure 4.1, Horne & Morris (1981).
sw02.bmk	Column with axial load and lateral load.
sw03.bmk	Guyed mast analysis.

sw04.bmk	Two storey frame, figure 5.14, Horne & Merchant (1965).
sw05.bmk	Lateral displacement of tip of end loaded cantilever.
sw06.bmk	Suspension bridge, Chap. 13, Ex. 5, Pippard & Baker (1957).
sw07.bmk	Comparison between member end springs & pseudo springs.
sw08.bmk	Modelling imperfections by parabolic bow.
sw09.bmk	Example 9.11-1, Coates *et al.* (1988).
sw10.bmk	Problem 9.1, Coates *et al.* (1988).
sw11.bmk	Problem 9.2, Coates *et al.* (1988).
sw12.bmk	Problem 9.8, Coates *et al.* (1988).
sw13.bmk	Problem 9.9, Coates *et al.* (1988).
sw14.bmk	Problem 9.10, Coates *et al.* (1988).
sw15.bmk	Problem 9.11, Coates *et al.* (1988).
sw16.bmk	Problem 9.12, Coates *et al.* (1988).
sw17.bmk	Net, symmetrical loading, Elibiari *et al.*, Nooshin (1984).
sw18.bmk	Net, unsymmetrical loading, Elibiari *et al.*, Nooshin (1984).
sw19.bmk	Hyperbolic paraboloid net, Elibiari *et al.*, Nooshin (1984).
sw20.bmk	Stable/unstable post-buckling behaviour, Coates *et al.* (1988).
sw21.bmk	Snap through - Problem 9.8-1, Coates *et al.* (1988).

```
Check of benchmarks for errors.
bm01  0.35166E-03 L1 27 JDX
bm02  0.10935E+01 L1 1 JDX
dt01 -0.24902E+00 L1 1 JRZ
dt02  0.12262E+03 L1 2 JDX
dt03  0.13760E+02 L1 1 MZS
 ... 123 checks omitted for space reason
sw17 -0.13787E+00 L20 4 JDX
sw18  0.97837E+01 L20 8 JDX
sw19 -0.11139E-01 L20 10 JDX
sw20  0.13558E+03 L65 2 JDX
sw21 -0.68589E+03 L94 2 JDY
Checksum of selected value/file for 133 files =-263.471187989868E6
```

9.5 Parametric Benchmarks

Parametric benchmarks give examples of data files for a wide range of engineering structures. Each item of data is provided as a parameter and assigned a default value, a default value being that used if no replacement value is provided by the engineer. These benchmarks have all been amended as part of this research to enable an audit trail, and to process and report on the set automatically. The first part of table 9.2 gives the filenames and a brief description, the second part of the summary gives an abbreviated version of that obtained by running the benchmarks as a set.

Table 9.2 Parametric benchmarks.

Filename	PLANE FRAME PARAMETRIC DATA FILES
pf01.ndf	Cantilever beam, parametric data overview, editor.

pf02.ndf	As pf01 with inclined supports, use of syntax.
pf03.ndf	Cantilever beam with tie-down - including diagrams.
pf04.ndf	Multi-storey frame.
pf05.ndf	Natural frequency calculation.
pf06.ndf	Influence lines by Müller-Breslau method.
pf07.ndf	Influence lines by PRINT COLLECTION command.
pf08.ndf	Curved beams introducing trig. & other functions.
pf09.ndf	Coupled shear walls - up to 100 storeys.
pf10.ndf	Continuous beam.
pf11.ndf	Subframe - beams with columns below (and above).
pf12.ndf	Including post processing in the data file.
pf13.ndf	Overview of the stiffness method.
pf14.ndf	Data preparation of large structure by substructures.
pf15.ndf	Thermal strains.
pf16.ndf	Member eccentricities and end joint sizes.
pf17.ndf	Overlapping members and scissors.
pf18.ndf	Prestressed continuous beams - load balancing method.
pf19.ndf	Ground beam on elastic piles subjected to load train.
pf20.ndf	Box culvert.
pf21.ndf	Loads on piles in groups.
pf22.ndf	Circular/parabolic arch.
pf23.ndf	Pipe-tree rack.
pf24.ndf	Bents and pipe rack.
pf25.ndf	Two member lean-to or Mansard beam.
pf26.ndf	Three member lean-to or Mansard beam.
pf27.ndf	Dogleg or cranked beam.
pf28.ndf	Vierendeel girder.

PLANE GRID PARAMETRIC DATA FILES

gr01.ndf	Bow girder introducing PLANE GRIDS.
gr02.ndf	Ground slab with loads from racks.
gr03.ndf	Solid slab bridge.
gr04.ndf	Timber floor panel.
gr05.ndf	Primary beams supporting secondary beam.

FINITE ELEMENT PARAMETRIC DATA FILES

fe01.ndf	Finite element method for plates in flexure.
fe02.ndf	Finite element method for plates in extension.
fe03.ndf	Finite element analysis for extension & flexure.
fe04.ndf	Plate/wall with hole/window in extension.
fe05.ndf	Floor slab with opening in flexure, various supports.

fe06.ndf	Spiral stair - folded steel plate.
fe07.ndf	Rectangular tank.
fe08.ndf	Setting openings and stiffeners in plate and wall.
fe09.ndf	Boxbeam with side cantilevers.
fe10.ndf	Circular hopper.
fe11.ndf	Circular chimney.
fe12.ndf	Circular balcony - plate with opening.
fe13.ndf	Barrel vault roof.
fe14.ndf	Rings and pipes.
fe15.ndf	Spherical dome.
fe16.ndf	Angle in bending and torsion.
fe17.ndf	Channel in bending and torsion.
fe18.ndf	I section in bending and torsion.
fe19.ndf	T section in bending and torsion.
fe20.ndf	Flat slab subframe.

SPACE FRAME PARAMETRIC DATA FILES

sf01.ndf	3D multi-storey frame.
sf02.ndf	Circular tank.
sf03.ndf	Space structure - square on square example.
sf04.ndf	Shear (flexural) centre.
sf05.ndf	Conical roof.
sf06.ndf	Orange segment roof truss.
sf07.ndf	Spiral stair - reinforced concrete.
sf08.ndf	Temporary works - column outriggers.
sf09.ndf	Dynamical behaviour of 3D multi-storey frame.

ROOF FRAME PARAMETRIC DATA FILES

rf01.ndf	N or Pratt lattice girder or portal.
rf02.ndf	Howe lattice girder or portal.
rf03.ndf	Warren lattice girder or portal.
rf04.ndf	Warren lattice girder or portal.
rf05.ndf	Portal frame/s without haunches.
rf06.ndf	Portal frame/s with haunches.
rf07.ndf	Gangnail roof truss.
rf08.ndf	Attic room roof truss.
rf09.ndf	Collar-tie roof truss.
rf10.ndf	Collar-and-tie roof truss.
rf11.ndf	Couple roof truss.
rf12.ndf	Couple-close roof truss.
rf13.ndf	Fink roof truss.

rf14.ndf King post roof truss.
rf15.ndf Queen post roof truss.
rf16.ndf Mansard roof truss.
rf17.ndf Tied-Mansard roof truss.
rf18.ndf Plane/pitched Vierendeel roof truss.

SWAY/BUCKLING FRAME PARAMETRIC DATA FILES

sw01.ndf Longitudinal defln of cantilever with lateral load.
sw02.ndf Elastic critical load (E.c.l) - end loaded column.
sw03.ndf E.c.l. - end & distributed loaded column.
sw04.ndf E.c.l. - piles with lateral restraint pressure.
sw05.ndf E.c.l. - circular ring with lateral pressure.
sw06.ndf E.c.l. - circular arch.
sw07.ndf E.c.l. - rectangular beam under pure bending.
sw08.ndf E.c.l. - narrow rectangular cantilever beam.
sw09.ndf E.c.l. - narrow rectangular beam, centre load.
sw10.ndf E.c.l. - I beam under pure bending.
sw11.ndf E.c.l. - cantilever I beam with end load.
sw12.ndf E.c.l. - I beam with centre point load.
sw13.ndf E.c.l. - rectangular plate - all edges S.S.
sw14.ndf E.c.l. - rectangular plate - all edges clamped.
sw15.ndf E.c.l. - short edges simply supported, long clamped.
sw16.ndf E.c.l. - one long edge free others simply supported.
sw17.ndf E.c.l. - short edges s.s. one long clamped other free.
sw18.ndf E.c.l. - short edges clamped others simply supported.
sw19.ndf Non-linear analysis - the DIRECTION command.
sw20.ndf Non-linear analysis of wall with window, in extension.

```
Check of parametric files for errors.
pf01 -0.13757E-02 L1 2 JDY
pf02 -0.19550E-04 L1 2 JDX
pf03  0.16076E-02 L1 1 JRZ
pf04 -0.15023E-04 L1 4 JDY
pf05 -0.28571E+02 L1 1 MZS
 ... 89 checks omitted for space reason
sw16  0.19591E+03 L51 1 RFZ
sw17  0.62519E+02 L52 1 RFZ
sw18  0.10298E+04 L57 1 RFZ
sw19  0.10672E-03 L20 3 JDX
sw20  0.36893E-04 L20 16 JDX
Checksum of selected value/file for 100 files =-76643.7506640545
```

9.6 Verified models as benchmarks

Verified models, which have been developed as part of this research, are listed in section 7.8. As with the parametric benchmarks, verified models cover a wide range of engineering structures for which each item of data is provided as a parameter and assigned a default value, a default value being that used if no replacement value is provided by the engineer. The verified models will be found in Appendix A. All have been written as part of this research to enable an audit trail, and to process and report on the set automatically. Table 9.3 gives an abbreviated version of the second part of the summary obtained by running the benchmarks as a set. The first part of the summary is omitted for reason of space.

Table 9.3 Benchmarks for verified models.

```
Check of NL-STRESS Verified Models for errors.
vm110 -0.12054E-01 L1 1 JRZ
vm112 -0.33216E-01 L1 2 JDY
vm113 -0.24920E-01 L1 2 JDY
vm114 -0.29909E-04 L1 2 JDY
vm115 -0.89089E-03 L1 3 JDY
... 98 checks omitted for space reason
vm850 -0.33265E-02 L1 2 JDY
vm852 0.32985E-03 L1 3 JDX
vm950 0.66709E-05 L10 1 JDX
vm951 -0.69453E-04 L10 1 JDY
vm952 -0.90921E-02 L10 1 JDY
Checksum of selected value/file for 111 files =116.87753469972
```

9.7 Other checking matters

For testing, the writer uses Windows' *command shell*. Microsoft state *The command shell is a separate software program that provides direct communication between the user and the operating system. Any command may be included in a batch file and run from the command shell. Certain commands such as FOR, GOTO and IF enable the user to do conditional processing.* The command shell and support for batch files are integral parts of Windows XP, Windows XP Professional x64, and the forthcoming Windows Vista 64-bit. SCALE has been designed, tested, and is supported on all versions of Windows from Windows 95 through to Windows Vista 64-bit, beta version.

Section 2.1 defines *regression tests* as those carried out prior to release of a previous version of the software. If the old version of software is version 3.1, the current version 3.2, then regression testing refers to testing version 3.1 and those prior to it.

With the cost of CDs less than 20p then backing up all the software associated with SCALE is done on a monthly basis to CDs and on a daily basis to *1GB memory sticks*. For security, duplicated CDs are stored in a different building. When differences between *Checksum* are found, it takes minutes to locate when the change took place. Of course, adding a benchmark to a set will give a different *Checksum*, so when this happens it must be noted in the benchmark log for future reference.

Chapter 10

Models for structural design

Although the engineer must always be responsible for the correctness of the calculations, for only the engineer is party to all the special conditions obtaining on site, it is desirable to have an automatic self check contained within each model for the design of a structural component. Self checks are used to confirm that the results from running a model are acceptable. There are various strategies for providing a self check:

- compare the results with those of an alternative code of practice
- work back from the results to see if the design requirements are met
- carry out a structural analysis for the component e.g. elastic analysis, yield line etc.
- thoroughly check the serviceability limit states *e.g.* deflection and cracking.

The suitability of any strategy to be used depends on the type of model being checked:

- for steel beams and columns subjected to axial, bending & twisting moments, which require assessments to be made of restraints and support conditions for the satisfaction of a unity factor equation (*e.g.* Fc/Pc+Mx/Mbs+My/(pyZy)<=1, BS 5950-1:2000 clause 4.7.7), a non-linear elastic analysis for the component to check for the maximum stress at the serviceability limit states would be appropriate
- for long span rectangular and flanged reinforced concrete beams, a thorough check by the serviceability limit states would be appropriate.

Sections 10.6 & 10.7 develop self checks for models which design structural steel members and reinforced concrete beams respectively.

Chapter 4 shows that a unified classification of types of data for both the structural analysis of frameworks and the structural design of components is achievable; thus chapters 4, 5 & 6 are relevant to the design of structural components and in particular the tabular form of data presented in section 5.10 is appropriate to structural design models. Most structural engineers are familiar with the set of bar diameters: 6,8,10,12,16,20,25,40 & 50. This set gives rise to other parameters having discontinuous values *e.g.* reinforcement cover, lap lengths *etc.* which are defined in codes of practice in terms of bar diameter. Fire resistance requirement of 0.5, 1, 2 & 4 hours, is another parameter having discontinuous values. Strategies are developed for dealing with such parameters including both short and long tables *e.g.* steel section properties.

Before the advent of computers, the analysis of structural frameworks was considered difficult and the province of engineers, but the structural design of members was considered to be straightforward and accordingly was delegated to technicians. This division of work between engineers and technicians has now changed in some firms, with graduates preparing the data for global models, and senior engineers being rightly concerned with materials and the local design of structural components such as beams and columns. Today the longhand design of say a steel beam is protracted, and even when carried out using computer software, the concepts in the design are far from the concepts of BS 449. For manufactured components such as Universal Beams *etc.* further types of data are required to cater for say a 203x133x25 UB. Tools for handling such items of data are developed in this chapter.

10.1 Engineers' arithmetic

Engineers' arithmetic is used for establishing the dependency conditions in the parameter table. A large proportion of routine structural *calculations* is for the sizing of beams, slabs and columns. A large proportion of beams are simply supported. The maximum bending moment in a simply supported beam carrying a uniform total load of W kN is W*L/8 kNm; for the load W kN concentrated at the centre of the beam, the maximum bending moment is WL/4 kNm *i.e.* twice the bending moment for the load if it were uniformly distributed. For the approximate sizing of a beam, engineers work out W*L/8 and then allow a percentage increase depending on how the loading is distributed. Once the bending moment is known, it is divided by the lever arm (assumed as 75% of the beam depth for a reinforced concrete beam) to give the tension or compression force, which in turn gives the area of reinforcement, or steel beam flange size, for known permissible stress. Simple calculations, such as that described, are carried out by engineers for both initial design and for checking. Engineers' arithmetic relies on concepts such as: lever arm, modular ratio, load factor (typically 1.5) *etc.* Engineers' arithmetic has use in the design of the sets of test data for increasing the robustness of *proforma calculations*; this chapter uses engineers' arithmetic both in the development of self checks and in the development of the parameter table which contains the parametric description of the model.

One important requirement for software is that the logic be robust. To test the logic against a thousand sets of data for a structural program requires that the data be appropriate and non-trivial. An example of trivial data is a square hollow section beam 400x400x15, spanning 3 m and carrying a minimum distributed load of 1 kN/m and a maximum distributed load of 10 kN/m. To avoid such trivial data, cognizance of structural engineering is required *i.e.* the ability to produce back-of-envelope calculations, referred to in this thesis as engineers' arithmetic. The tools of engineers' arithmetic include:

- limiting maximum span:deflection ratio to say 384

- limiting span:depth ratio to be in the range 8 to 24
- limiting the maximum bending moment such that the design stress times area of steel in tension times depth of section is not exceeded
- assuming area of steel in tension = cross-sectional area/3 for a steel section
- assuming aspect ratio (depth:breadth) of beam varies from 1 to 3 with a typical value =2.

The order in which these tools are applied depends on the order of known values *e.g.* if the maximum bending moment and the span are known, then the equivalent distributed load may be found using M=w*L^2/8 rearranged as w=8*M/L^2. Rearranging delta=5*w*L*L^3/(384*E*I), span:deflection = L/delta =384=384*E*I/(5*w*L^3), knowing E for the material, I may be found from I=5*w*L^3/E. Knowing D=L/14 where D is the section depth, & I=At*D^2, where At is the area of steel in tension, then At=I/(L/14)^2 =196*I/L^2. Knowing A=3*At, then the cross sectional area A is known. Knowing D/B=2 then breadth of section B=0.5*D, and flange thickness T=At/B=196*I/(B*L^2).

The above reasoning provides expressions which in turn provide dependency conditions which may be used in the parameter table to avoid trivial data or data which exceeds the model's limits *i.e.* the range specified under the column headings *Start* and *End* in the parameter table.

10.2 Upheaval caused by changes to codes of practice

Examples of the frustrations imposed on structural engineers by bodies who develop codes of practice follow.

- In 2005 structural engineers were told that BS 8110 Part 1, for the structural use of concrete, would be withdrawn in 2010.
- On 30 November 2005, a new version of BS 8110 was published incorporating amendments which change both partial safety factors and permissible stresses for reinforcement, deleting clauses 3.3.5.1 - 3.3.5.3 and referring the engineer to BS 8500-1 and BS 8500-2.
- On 21 September 2005, a new version of BS 8666 for the scheduling of bars was published, which doubled the number of bar shapes, changed the notation for reinforcing steels, and has to be read in conjunction with a new version of BS 4449 which classifies new yield strengths for reinforcement. Structural engineers will remember the familiar BS 4466 for the scheduling or reinforcement was withdrawn in 2000 and replaced by BS 8666 halving the number of shape codes and causing upheaval in the reinforcement industry, within five years the exercise was repeated.

Engineers are bound to be concerned, for such a plethora of changes has a direct impact on the **correctness of calculations**, with which this work is concerned.

10.3 Commonality of analysis & design models

The principles and systems adopted for verifying the correctness of models for the structural analysis of a framework generally apply to verifying the correctness of models for the design of structural components, the two main systems being:

- the self check
- the automatic generation of a thousand sets of engineered data and the running and reporting of results using the same system.

Just as the self check for each model for the structural analysis of a framework is unique to each model, so the self check for each model for the design of a structural component is unique to each model. The unified approach means that the parameter table which contains the specification of the data required by the model, has an identical format for both types of structural model *e.g.* the *Type* classification for the parameters is the same for tables 5.4 and 5.5.

This research has been concerned mainly with models for the analysis of structural frameworks as described in chapters 7 to 9, as

- models for the analysis of structural frameworks are less complicated than models for the design of structural components and consequently the models are easier to introduce
- the analysis of structural frameworks is not dependent on codes of practice and therefore if this thesis is perused in 2040, the models presented will still be valid and immediately recognisable
- models for the structural design of components are code dependent and therefore subjected to frequent changes as described in section 10.2.

Although both models share the same format for their parameter table, the self check report differs between the two types of model. For the structural analysis of frameworks, differences between classical and modern methods of analysis are close and it is sensible to summarise the differences as percentages; however when models are dependent on codes of practice, percentage differences can be high especially when the serviceability limit states are compared to the ultimate limit state. For this reason, for models which are dependent on codes of practice, reporting is descriptive rather than numerical *e.g.*

```
Run=1
Run=2
Run=3
** Crack width exceeds permissible, 0.3586 > 0.3 mm
Run=4
Run=5
** Serviceability shear > link capacity, 2222.2 > 267.62 kN
Run=6
Run=7
** Crack width exceeds permissible, 0.36952 > 0.3 mm
Run=8
```

```
Run=9
Run=10
** Serviceability shear > link capacity, 755.53 > 755.14 kN
Run=11
Run=12
```

The above report was for 12 runs of proforma calculation sc075.pro as listed in appendix C. When no message follows the run number, it signifies that the check was satisfactory. The self check is not limited to *batch operation*, it is active for each and every single *interactive* run; the engineer may suppress the check thus producing a calculation in accordance with BS 8110:1, or may include the check thus producing a calculation in accordance with BS 8110:1 & BS 8110:2 *i.e.* checking both the ultimate and serviceability limit states. Only *the* engineer can properly assess the warnings provided *e.g.* the message given for run 5 above *i.e.*

```
** Serviceability shear > link capacity, 2222.2 > 267.62 kN
```

will be meaningful to *the* engineer. The italics usage of the definite article rather than the indefinite article is deliberate, for it is *the* engineer who will have a mental picture of the

- location
- number of times the structural component will be duplicated
- background to the design
- environment
- structural importance

for the section being designed. No automatic self check can bring the engineer's wealth of experience to bear on the component design, *the best the self check can do is to look at the design from another perspective.*

10.4 Classical and modern structural component design

There is an anomaly in current structural design *i.e.* the bending moments, shear forces, axial loads and torques (twisting moments) are almost invariably computed by linear elastic analysis and the sufficiency of strength in the members assessed on the basis of semi-empirical ultimate limit state equations, which are doctored from time to time by the authors of codes of practice.

One aim of the creation and verification of the models developed for the structural analysis of frameworks, was to lay some bedrock beneath modern matrix methods using classical structural analysis methods, and this has been done. We can say that *classical methods of structural analysis provide bedrock beneath modern matrix methods even though the bedrock contains fissures, due to the omission of axial and/or shear deformation effects from most of the classical methods*.

A reasonable goal for the verification of models for the structural design of components such as beams, slabs, columns, walls and foundations would be to lay some bedrock between the intuitive classical section design elastic theory methods and the modern non-intuitive semi-empirical semi-intractable section design methods.

After some recent rethinks by the cement industry following *concrete cancer* problems, followed by steel reinforcing bars providing insufficient ductility, *durability* has become a key issue. Concrete mix design is no longer based on just the 28 day strength; for durability, a minimum weight of cement in kg/m3 must now be specified.

When limit state design was first introduced, with the exception of shear reinforcement for reinforced concrete beams, the equations were adjusted to give similar answers to designs produced in accordance with elastic theory. Although there has been some minor tinkering to partial safety factors since CP 110 and BS 5950 were first introduced, there is merit in providing models for the structural design of components which have a self check based on classical principles. Accordingly, self checks based on classical linear elastic behaviour are being developed. See section 10.6 for a typical structural steel component and section 10.7 for a typical reinforced concrete component.

10.5 Differences between analysis & design models

Although a unified treatment has been derived for the analysis of structural frameworks and the structural design of components, there are some differences between the two types of model. The classification of types of structural data in chapter 4 for the automatic generation of sets of data to test the logic of proforma calculations is equally applicable to the automatic generation of sets of data to test the logic of a model for the structural analysis of a framework *i.e.* the parameter table is appropriate to both. Nevertheless, the data required for the structural design of a component is more *disjointed* than that required for the structural analysis of a framework, *e.g.* the following extract is taken from SCALE proforma calculation sc385.pro for the design of a stainless steel component.

Table 10.1 Nominal effective length for a compression member.

<table>
<tr><td colspan="2">Conditions of restraint at ends</td><td>Effective Length</td></tr>
<tr><td rowspan="4">Effectively held in position at both ends</td><td>Restrained in direction at both ends</td><td>K = 0.7</td></tr>
<tr><td>Partially restrained in direction at both ends</td><td>K = 0.85</td></tr>
<tr><td>Restrained in direction at one end</td><td>K = 0.85</td></tr>
<tr><td>NOT restrained in direction at either end</td><td>K = 1.0</td></tr>
</table>

```
Restraint factor z-z axis       +Kz=????
Restraint factor y-y axis       +Ky=????
```

Table 10.1. was taken from table 22 in proforma calculation sc385.pro, see appendix C. The table of data is typical of the discontinuous behaviour of models for the structural design of components. The data required for the analysis of a given structural framework subjected to a given set of loads, has *exact* values for coordinates, connectivity, material constants and loading, where *exact* is used in the sense that two engineers working independently from the same data, would be expected to compute the same member forces. The data required for Kz & Ky is open to debate about what constitutes *partial restraint*, or more importantly what the engineer understands the code means by *partial restraint*.

Models for the structural analysis of frameworks. Each model:

- typical length from 2 pages for vm122.ndf to 9 pages for vm112.ndf
- straightforward using the NL-STRESS language
- small amount of logic
- time to develop each model is 1 to 6 days
- parameters are generally continuous over their range
- for a given frame and loading, values are right or wrong.

Models for the structural design of components. Each model:

- typical lengths, 42 pages for sc385.pro, 33 pages for sc075.pro
- more difficult because of length of model
- large amount of logic with considerable nesting
- time to develop each model is 1 to 6 weeks
- parameters are often discontinuous, see table 10.1 above
- values for data are sometimes dependent on judgment.

Summarising, the nature of models for the structural analysis of frameworks is that for a given model, variation of any parameter is usually accompanied by continuous behaviour of the model, whereas models for the structural design of components are more complicated and often associated with discontinuous behaviour of the parameters and uncertainty. Incorporation of the parameter table into a proforma calculation allows for the automation of the testing of many different combinations of the parameters, saving time, increasing the *coverage* of the model tested and thereby increasing the correctness of the model.

10.6 Typical structural steelwork component

Proforma calculation sc385.pro, for the design of stainless steel square and rectangular hollow section members, is used to describe the process of building the parameter table for a steelwork component. For ease of reference, it is recommended that flowcharts be printed for both sc385.pro and for sc3800.pro which is invoked by sc385.pro. To do this, run SCALE and in response to the prompt for *Option number*, type *385/P*, press Enter and click the Print button; repeat but type *sc3800.pro/P*, press Enter and click the Print button.

Inspection of the printout shows that there are 47 prompts for parameters: $92 Mz Fv F L D' B' T' t' My stype Lz Ly Kz Ky Ae moment restz Mz betaMz mz2 mz3 mz4 M24 resty My betaMy my2 my3 my4 My24 LT refno udl betaM m2 m3 m4 grade E $27569 sd(ssd1) sb(ssd1) st(ssd1) sd(srd1) sb(srd1) st(srd1). Each of the 47 parameters has its data input interactively *e.g.*

```
Factored bending moment axis zz   +Mz=???? kNm
```

which causes the engineer to be offered his/her previous response for Mz, which may be accepted by pressing Enter, or replaced by typing a new value. As an alternative to providing the data interactively, the proforma calculation may be run in batch mode for which all values are extracted from a *stack* of values. Thus the requirement is to automate the production of hundreds of files, each containing the same set of parameters for the model but with a unique set of values for each run, providing a self check at the end of each model with automatic reporting of results of the check for each run. It will be apparent from inspection of prompts for proforma calculation sc385.pro, listed in Appendix C, that the task is more difficult than that for the analysis of a structural framework.

Section 3.3 advocates a simple flowchart for understanding the structure of a proforma calculation; inspection of the simplified flowchart for sc385/sc3800 will show that *stype* takes the value 1 or 2, dependent on a square or rectangular hollow section being required. The model also allows a rectangular hollow section to be used in portrait or landscape orientation, this complication is mentioned as an example for demonstrating that an engineer is required for designing engineering models. Although there are firms specialising in verification and self-checking software, verifying software for the structural analysis of frameworks and the structural design of components, requires engineering expertise. To write a program which generates values for 47 parameters would be trivial if all parameters were continuous and each was independent from all others. To write a program that generates sensible values for 47 parameters

- with a large proportion being dependent on one or more other parameters
- with some parameters obtaining their values from tables and other indirect methods
- with the full set of parameters passing all error checks in the model, would be difficult but straightforward if each model had its own program for generating data.

To write one program for generating sets of values for interdependent parameters for 780 models having the complications described above, is essential for software sustainability.

Dummy parameters are parameters which are not input, but which are required in the parameter table so that they may be used in the self checking logic and/or the dependency conditions. An example of a table which requires dummy parameters is given in table 10.2. If the permissible design strength of the structural material is known, then the design strength is a parameter and may be used for both the input data and for the material strength used in the self-check. Occasionally, the authors of codes refer to types of materials rather than strengths of materials *e.g.* stainless steel to BS EN 10088-2 is described by table 10.2.

Table 10.2 Strengths of stainless steel to BS EN 10088-2.

Grade	Number	Type
Basic chromium-nickel austenitic	1.4301	1
Molybdenum-chromium-nickel austenitic	1.4401	2
Molybdenum-chromium-nickel austenitic	1.4404	3
Duplex Steel	1.4362	4
Duplex Steel	1.4462	5

```
Steel type (1,2,3,4 or 5)                  +grade=????
```

In table 10.2, the design strength will be dependent on the Type chosen. For Types = 1 to 5, the design strengths in BS EN 10088-2 are: 210,220,220,400,460 N/mm² respectively. If a model were to input the design strength directly *e.g.* 220 N/mm², then the molybdenum-chromium-nickel austenitic grade would be ambiguous, as its *Number* could be either 1.4401 or 1.4404. Accordingly the steel is best referred to by its type, with the model consulting a short table to give the design strength and Number.

It will be evident that for structural calculation models, parameters have to be set which may or may not be required *e.g.* for sc385.pro, properties for both square hollow sections and rectangular hollow sections are required dependent on *stype* taking the values 1 or 2. To carry out a thorough check on the logic of a model to detect the presence of unassigned variables *etc.*, two runs are necessary; the first including all parameters which may or may not be required for the run, the second including only data actually used in the first run. Thus for the first run with *stype=1* which specifies a square hollow section, section dimensions for depth, breadth & thickness will be provided for both square & rectangular hollow sections; for *stype=1* section dimensions for rectangular hollow sections are one type of dummy parameter. For the second run with *stype=1* which specifies a square hollow section, section dimensions for depth, breadth & thickness will only be provided for square hollow sections.

Another type of dummy parameter is needed when a parameter, such as design strength, is required for the self-check and that parameter has not been input directly but looked-up or computed during the first run. For such a situation, the parameter will not normally be passed to the second run. To force a pass from the first run to the second run, a dummy prompt is needed, preceded by a fast forward (>>) and followed by a stop fast forward (><) *e.g.* for the design strength:

```
>>
!Design strength                 +py=????
><
```

The three lines above are not seen in the calculations, but cause *py* to be written to the stack of values, just as though its value had been input by the engineer.

To look-up a table and extract the design strength would normally require a procedure. When the table is short, as for the design strengths in BS EN 10088-2 (210,220,220,400,460 N/mm²) corresponding to the five types of steel given in table 10.2, then the table may be incorporated into the parameter table *e.g.*

Table 10.3 Storage of short tables in the parameter table.

```
PARAMETER   Start   End     Type    Dependency conditions
No. name    zst()   zen()   zty()   and notes.
 ...
13  grade   1       5       5
14  py1     210     210     2
15  py2     220     220     2
16  py3     220     220     2
17  py4     400     400     2
18  py5     460     460     2
19  py      210     460     0       =zva(13+grade)
 ...
```

In the program for generating the sets of data, zva(n) is the current value of parameter *n*, thus if grade=1 then py is set to the current value of zva(14) *i.e.* 210; if grade=2 then py is set to the current value of zva(15) *i.e.* 220 and so on. See section 10.10 for an alternative method of saving short tables for which the table values may be included on one line.

Storage of long tables. Although the provision of structural properties or structural geometry is sufficient to define the cross-sectional area, moment/s of inertia *etc.* for the members of a framework to be analysed, the data required for the design of structural components such as: I-sections, structural hollow sections *etc.* presents another type of data. As an example, consider the following table 10.4.

Table 10.4 Stainless steel square hollow section sizes.

Size	Available thickness t mm									
40 x 40	2	3								Stainless
50 x 50	2	3	4							Square
60 x 60	2	3	4	5						Hollow
80 x 80	2	3	4	5						Sections
100 x 100		3	4	5	6					
125 x 125		3	4	5	6	8				
150 x 150		3	4	5	6	8				
175 x 175			4	5	6	8	10			
200 x 200			4	5	6	8	10			
250 x 250				5	6	8	10	12		
300 x 300				5	6	8	10	12		
350 x 350					6	8	10	12	15	
400 x 400					6	8	10	12	15	

```
Serial depth                       +sd(ssd1)=???? mm
Serial breadth                     +sb(ssd1)=???? mm
Thickness                          +st(ssd1)=???? mm
```

Although it is a trivial exercise to generate sensible integer values for the sizes, responding to the three prompts with 250,250,7 would not be acceptable data, as such a section does not exist. The checking in the model will detect that a section does not exist and advise the engineer accordingly, but generating a thousand triads for sd(),sb(),st() would not be helpful as the great majority will fail unless one of the following is provided:

- logic is written to describe the table
- the table is provided directly
- a procedure is provided to give a best fix for a generated triad.

For the simple table above, many lines of logic would be needed. Although logic could be devised; as from time to time manufacturers have to change section sizes in response to changes in codes of practice and/or changes in market conditions, maintaining the logic would be a long term commitment. Look up tables are already present in the model itself, it would be better to avoid duplication. A best fix for the sd(),sb(),st() triad, would be easier to maintain than storing the table. If the sd(),sb(),st() triad is combined into a single number: sd()*10^6+sb()*10^3+st(), as described in section 5.12, the triad may be used to find the nearest available section size. Although the table will need to be maintained, the maintenance will be much simpler than maintaining the many lines of logic needed to describe the table. The parameter table for proforma calculation sc385.pro follows, and in turn is followed by a detailed description of parameters 1-24, using the parameter number as a reference.

Table 10.5 Parameter table for stainless steel hollow section design.

```
PARAMETER  Start   End     Type    Dependency conditions
No. name   zst()   zen()   zty()   and notes.
 1  L      1       6       0       Length of member in m.
 2  stype  1       2       2       stype=1 is SHS, stype=2 is RHS
 3  ssd1   11      11      0       Constant for SHS table number
 4  sd11   40      400     1       >L*1000/24          <L*1000/8
 5  sb11   40      400     1       =sd11 for SHS
 6  st11   2       15      1       >sd11/66.667+1      <sd11/12+1
 7  tri    3       58      1E40
 8  srd1   12      12      0       Constant for RHS table number
 9  sd12   40      400     1       >L*1000/24          <L*1000/8
10  sb12   20      200     1       =sd12/2 say for RHS
11  st12   2       15      1       >sd12/66.667+1  <sd12/12+1
12  tri    3       72      1E40    Calls procedure tri
13  grade  1       5       5
14  py1    210     210     2
15  py2    220     220     2
16  py3    220     220     2
17  py4    400     400     2
18  py5    460     460     2
19  py     210     460     0       =zva(13+grade) N/mmý
20  sd     40      400     0       =sd11*(2-stype)+sd12*(stype-1)
```

21	sb	20	400	0	=sb11*(2-stype)+sb12*(stype-1)	
22	st	1.5	15	0	=st11*(2-stype)+st12*(stype-1)	
23	zrn	4	0.7	1E40	Proc. creates zrn(1:4) d=0.7.	
24	Mz	1	1000	0	=sb*st*py*sd/1E6*zrn1 kNm	
25	My	1	1000	0	=sd*st*py*sb/1E6*zrn2 kNm	
26	Fv	1	1000	0	=sd*2*st*py/SQR(3)/1E3*zrn3 kN	
27	F	1	1000	0	=(sd+sb)*2*st*py/1E3*zrn4 kN	
28	Lz	1000	6000	0	<L*1000 mm	
29	Ly	1000	6000	0	<L*1000 mm	
30	Kz	0.7	1	3		
31	Ky	0.7	1	3		
32	Ae	5	221	0	<(sd+sb)*2*st/100 cm2	
33	moment	1	0	2		
34	restz	1	0	2		
35	betaMz	1	1000	0	<Mz kNm	
36	mz2	1	1000	0	<Mz*zrn1	Must write zrn1 etc.
37	mz3	1	1000	0	<Mz*zrn2	here and not write
38	mz4	1	1000	0	<Mz*zrn3	zrn(1) etc.
39	M24	1	1000	0	<Mz*zrn4	
40	resty	1	0	2		
41	betaMy	1	1000	0	<My	
42	my2	1	1000	0	<My*zrn1	
43	my3	1	1000	0	<My*zrn2	
44	my4	1	1000	0	<My*zrn3	
45	My24	1	1000	0	<My*zrn4	
46	LT	1	6	0	<L	
47	refno	1	12	1		
48	udl	1	2	2		
49	betaM	1	1000	0	<Mz*zrn1	
50	m2	1	1000	0	<Mz*zrn2	
51	m3	1	1000	0	<Mz*zrn3	
52	m4	1	1000	0	<Mz*zrn4	
53	E	200	200	0	Young's modulus N/mmý	
54	NRESP	0	0	0	Avoids importing from NL-STRESS	
55	ans	0	0	0	ans=0 refuses default values	

Parameter 1. The length L is allowed to vary from 1 to 6 m. As zty(1)=0 an integer or real value will be generated. Dependency conditions may not be given for the first parameter; dependency conditions may only be specified in terms of parameters listed previously in the table.

Parameter 2. As zty(2)=2, the section type *stype* takes only 2 values *i.e. stype* =1 or =2. If the number of increments *zni=5* the values generated for *stype* will be: 1,2,1,2,1.

Parameter 3. Proforma sc385.pro prompts for sd(ssd1)=???? for which ssd1 is a constant =11, which must be set before the serial sizes are input.

Parameter 4. The serial depth *sd11* is specified to be an integer value by zty(4)=1, lying in the range 40 mm to 400 mm, with dependency condition >L*1000/24 which for L=1 m gives a minimum size of 42 mm and dependency condition <L*1000/8 which for L=6 m gives a maximum size of 750 mm which would be reduced to zen(4)=400 mm.

Parameter 5. The serial breadth *sb11* for a square hollow section lies in the range zst(5)=40 mm to zen(5)=400 mm but constrained by the dependency condition '=sd11' to be the same size as the serial depth, as necessary for a square hollow section.

Parameter 6. The serial thickness *st11* is specified to be an integer value by zty(6)=1, lying in the range 2 mm to 15 mm, with dependency condition >sd11/66.667+1 which for sd11=40 mm gives a minimum thickness 1 mm and dependency condition <sd11/12+1 which for sd11=400 mm gives a maximum thickness of 34 mm which would be reduced to zen(6)=15 mm.

Parameter 7. The setting of zty(7)=1E40 tells the program to invoke a procedure named *tri*, passing the values zst(7)=3 and zen(7)=58 to the procedure as *arguments*. The procedure *tri* is given in section 5.12 with an explanation; briefly the serial depth, breadth & thickness *i.e.* sd11, sb11 & st11 are fixed to be those of an available square hollow section for stainless steel.

Parameters 8-12 are similar to Parameters 3-7 except that they apply to a rectangular hollow section rather than a square hollow section.

Parameters 13-19 have been discussed earlier in this section.

Parameters 20-22, respectively set the current serial: depth, breadth and thickness, regardless of whether the section is square or rectangular. None of these parameters appear in proforma calculation sc385.pro, all are needed to be able to do engineers' arithmetic to compute sensible forces *e.g.* bending moments *etc.* The dependency conditions carry out the necessary assignments viz.

```
Expression for evaluation          stype=1        stype=2
=sd11*(2-stype)+sd12*(stype-1)     =sd11          =sd12
=sb11*(2-stype)+sb12*(stype-1)     =sb11          =sb12
=st11*(2-stype)+st12*(stype-1)     =st11          =st12
```

Parameter 23. The setting of zty(23)=1E40 tells the program to invoke a procedure named *zrn*, passing the values zst(23)=4 and zen(7)=0.7 to the procedure as *arguments*. The procedure creates 4 random numbers: zrn1 to zrn4, such than Σ(zrn1,zrn2,zrn3,zrn4)=0.7. The values of zrn1 to zrn4 are then used as multipliers for dependencies in parameters 24 to 27, respectively Mz,My,Fv,F so that when combined by a less than or equal to *unity formula,* the majority of the sets of data generated from the parameter table will be less than unity.

Parameter 24. The product of the serial breadth *sb*, thickness *st* and yield stress *py* gives the ultimate force in the flange for bending about the *z* axis, using engineers' arithmetic *=sb*st*py* where *py* is the yield strength, thus the ultimate bending moment *=sb*st*py*sd* Nmm =sb*st*py*sd/1E6 kNm. To reduce this value so that when it is combined with *My, Fv* & *F* it sums to 0.7, it is multiplied by *zrn1*.

Parameters 25 to 27 have their component values multiplied by *zrn2* to *zrn4* respectively. The /SQR(3) in the dependency conditions for parameter 26 is to reduce the yield strength to the shear yield strength. As can be seen by inspection of table 10.5, other parameters are more straightforward and are omitted for reason of space.

Summarising, for square hollow sections the serial breadth is made equal to the serial depth. For rectangular hollow sections sb() is set to serial depth/2. For both square and rectangular hollow sections, the serial thickness is constrained by the depth:thickness ratio. From inspection of the table for square hollow sections, the depth:thickness ratio has a maximum value of 66.66 (for 400:6) and a minimum value of 12 (for 60:5). These values are used for the dependency conditions in table 10.5. The problem of nominating a section has been reduced to one of taking the sd(),sb(),st() triad defined in the parameter table, combining all three component values into a single number and picking the nearest value in vector tri() and then separating the combined number back to the sd(),sb(),st() components for use by the model. To invoke a procedure for doing this, add a pseudo parameter called *tri i.e.* the name of the vector containing the table of values. The start value 3 in the table under the column headed zst(), tells the system that the previous 3 parameters must be fixed. The values 58 or 72 for parameters 7 & 12 respectively in the table under zen() tells the system that there are 58 or 72 values in the vectors giving sizes for square hollow sections and rectangular hollow sections respectively, the values 1E40 in the table under zty(), say that the line in the table is that for a pseudo parameter and therefore the line must be treated differently to other parameters.

Following the incorporation of the parameter table into sc385.pro, the model was run for various sets of automatically generated data to test for the presence/absence of bugs in the model. One bug was found identified by the message UNASSIGNED VARIABLE:

- the variable *stype* was erroneously named as *rtype* in the section of the model dealing with slender sections.

Typical reporting by the self check follows. For proforma calculation sc385.pro, the self check was provided by a finite element model of the stainless steel hollow section subjected to working load, with various types of restraint. Unexpected results are highlighted by asterisks.

```
Run 1              ┌ Max. comb. axial & bend. stress 149579.388 kN/m2
Self check at      │ Dist. from start to max. stress 6 m
working load.      │ Maximum combined displacement   .006 m
                   └ Dist. from start to max. displ. 3.0303 m
Run 2              ┌ Max. comb. axial & bend. stress 370185.0609 kN/m2
Self check at      │ Dist. from start to max. stress 3.15 m
working load.      │ Maximum combined displacement   .0026 m
                   └ Dist. from start to max. displ. 1.5597 m
```

```
Run 3             ┌ Max. comb. axial & bend. stress 100916.0675 kN/m2
Self check at     │ Dist. from start to max. stress .3 m
working load.     │ Maximum combined displacement   .57154E-04 m
                  └ Dist. from start to max. displ. .15 m
Run 4             ┌ Max. comb. axial & bend. stress 84074.4648 kN/m2
Self check at     │ Dist. from start to max. stress .3 m
working load.     │ Maximum combined displacement   .61737E-04 m
                  └ Dist. from start to max. displ. .15 m
Run 5             ┌ Max. comb. axial & bend. stress 131110.9227 kN/m2
Self check at     │ Dist. from start to max. stress 3.15 m
working load.     │ Maximum combined displacement   .0016 m
                  └ Dist. from start to max. displ. 1.5903 m
Run 6             ┌ Max. comb. axial & bend. stress 60091.3528 kN/m2
Self check at     │ Dist. from start to max. stress 6 m
working load.     │ Maximum combined displacement   .0025 m
                  └ Dist. from start to max. displ. 2.9508 m
Run 7             ┌ Max. comb. axial & bend. stress 289783.9797 kN/m2
Self check at     │ Dist. from start to max. stress 6 m
working load.     │ Maximum combined displacement   .0114 m
                  └ Dist. from start to max. displ. 2.9508 m
Run 8             ┌ Max. comb. axial & bend. stress .10092E+07 kN/m2
Self check at     │ Dist. from start to max. stress 3.15 m
working load.     │ Maximum combined displacement   .007 m
                  └ Dist. from start to max. displ. 1.5597 m
*** Combined stress= .10092E+07 exceeds yield stress= 400000
Run 9             ┌ Max. comb. axial & bend. stress 71071.0896 kN/m2
Self check at     │ Dist. from start to max. stress .3 m
working load.     │ Maximum combined displacement   .59195E-04 m
                  └ Dist. from start to max. displ. .15 m
```

It is desirable to have just one program for the automatic verification of a thousand engineering models, this is provided by sc924.pro. It is also desirable to have one table containing all the information needed to build a thousand stack files for each model, containing the parameters and their values, which provide the sets of data for verifying a model. This is achievable for 95% of structural models, for the remainder, procedure names such as *tri* and *zrn* together with their *arguments*, *i.e.* values to be passed to the procedure, are included in the parameter table, the procedures being included in proforma sc924.pro, which is used for generating and running the sets of data for both the structural analysis of frameworks and the structural design of components.

10.7 Typical reinforced concrete component

In February 1965 a revision of CP 114 (1957) *The structural use of reinforced concrete in buildings* was published as Amendment No.1 (PD 5463), bringing CP 114 up to date with the then new CP 116 (1965) *The structural use of precast concrete*. These two codes provided for the strength of members to be assessed by the then commonly employed elastic or modular ratio theory. The elastic theory is concerned with the equilibrium at working stresses of the forces and moments due to actual loads, the working stresses being the ultimate stresses reduced by a factor of safety. Concrete structures had been designed using elastic theory from 1932 (First edition of Reynolds' Reinforced Concrete Designer's Handbook) until 1972 when CP 110 (1972) *The structural use of concrete* was published. Whereas CP 114 was withdrawn, its steel counterpart BS 449 (1968) remains as an approved method for the elastic design of steel structures. The four decades commencing in 1932, included most of the replacement of buildings demolished during the London blitz. Although no figures are available, it is likely, that in 2005, more structures in Britain have been designed using elastic methods of design rather than limit state design. Limit state design was not in general use until the late seventies. From the late seventies to the mid nineties, the construction industry was the victim of economic experimentation. Thus, until the introduction of limit state design, tried and tested elastic methods of design were used. Following the introduction of CP 110 Part 1 November 1972, *the first limit state British Standard*, a few firms received contracts from central government to carry out so called *calibration tests* to compare calculations produced in accordance with CP 114, with those produced in accordance with CP 110. As computers were unavailable, comparisons between the codes were made for typically half a dozen beam or column designs. Experience in this research, is that half a dozen comparisons between modern and classical methods are inadequate, at least hundreds and preferably thousands of comparisons are needed in order that meaningful conclusions may be drawn. Increasing the number of sets of data to be generated and run, throws up structural anomalies due to the interplay between the rules such as: material strengths, section shape and sizes, exposure condition, concrete cover, fire rating, flange depth, neutral axis depth, percentage of compression and tension steel required, minimum steel percentage allowable, steel added to control deflection...

An inspection of the parameter tables for proforma calculations sc385.pro & sc075.pro given in tables 10.5 & 10.6 respectively, give an idea of just how complicated modern codes of practice have become, with a high dependency of parameters among themselves. A full listing of both proforma calculations will be found in appendix C. Comparison of the parameter tables for sc385.pro for stainless steel and sc075.pro for reinforced concrete, reveals that calculations for steelwork have a higher interdependency of parameters than for reinforced concrete, 35 dependencies for sc385.pro *cf.* 23 dependencies for sc075.pro.

Over the past two decades, code writers have increasingly devised expressions and formulae for the ultimate limit state, rather than the serviceability limit states. As an example, clause 3.4.6 in BS 8110, tells us that deflections will be OK if the basic span/effective depth ratios given in table 3.9 are used. In the same chapter, formulae are given for the section design of rectangular and flanged beams at the ultimate limit state, the implication is that chapter 3 of BS 8110 covers both the serviceability & ultimate limit states. Kong & Evans (1987) write "Lateley the serviceability of concrete structures has become a much more important design consideration than in the past, mainly because more efficient design procedures have enabled engineers to satisfy the ultimate limit state requirements with lighter but more highly stressed structural members. For example, during the past few decades, successive British codes have allowed the maximum service stress in the reinforcement to be approximately doubled in design.... Serviceability is concerned with structural behaviour under service loading, and service loading is sufficiently low for the results of an elastic analysis to be relevant."

Before the introduction of the limit state code CP 110 (1972), reinforced concrete design was based on elastic principles in accordance with CP 114, which resulted in structures which had a factor of safety of typically 2.5. Designs in accordance with BS 8110-1:1997 have a factor of safety of typically 1.5. *Typically* is the best we can do to describe the factor of safety. Rigour is not possible *e.g.* site operatives in concrete gangs, pat the top of their head meaning *toppings i.e.* they want more water in the mix. Changing the water/cement ratio from 0.4 to 0.6 reduces the seven day mean compressive strength of concrete from 33 N/mm² to 18 N/mm² *i.e.* almost halving the concrete strength, DSIR (1950). Shear reinforcement provided in accordance with CP 114 ignored the strength of the concrete, on the assumption that at least half of the concrete was cracked. The authors of BS 8110-1:1997 took a different view and assumed that cracked concrete has a strength and in consequence the amount of shear reinforcement provided can be less than half of that required by CP 114.

A practical structural concrete cannot exceed the strength of the aggregate used in making it, thus the use of aggregate made from recycled crushed concrete should be avoided for *structural concrete*. Age on loading has a major significance on deflection. BS 8110-2:1985, figure 7.1 shows the effects of relative humidity, age of loading and section thickness upon the creep factor. Concrete made with sandstone aggregate has a very much larger creep than that made with granite aggregate, Orchard (1958). Kong & Evans (1987), in figure 2.5-4 give shrinkages of specimen mixes. Illston (1994) in figure 15.28 gives a relationship between the modulus of elasticity of the aggregate and the relative creep. For a relative creep factor of 1 for basalt, sandstone has a creep factor of 4. Thus choice of aggregate is of major significance for concrete beams and slabs. The writer recalls a library he designed in the seventies, for which monitored deflections of the *waffle* floor spanning 12 m were three times those predicted based on C&CA published data from tests using Thames gravel aggregates and not the sandstone aggregate actually used in the library. It follows that some structures designed to BS 8110-1 alone, are likely to fail some of the serviceability requirements *hidden away* in BS 8110:2.

Whereas models for the design of structural steelwork components have to be able to recognise *serial sizes* for the many types of steelwork sections, necessitating the need to invoke procedures from the parameter table, models for the design of reinforced concrete components are more straightforward. Table 10.6 shows the parameter table developed as part of this research for proforma calculation sc075.pro for the design of flanged beams in bending with optional: shear, bar curtailment, lap length and span/effective depth checks. A complete listing for this proforma calculation will be found in appendix C. Proforma calculation sc075.pro was developed by Professor Bill Cranston (C&CA & Paisley University) and checked by Jim Steedman, the author of the *Reinforced Concrete Designer's Handbook* since the death of Chas Reynolds.

The first four parameters have no *dependencies*, the fifth parameter *M* dictates the size of the section, using engineers' arithmetic for a beam carrying bending moment *M* kNm and effective depth *d* mm and breadth of rib *bw=d/2*. For 2% reinforcement of characteristic strength *fy* N/mm², we may write:

M*1E6=d*d/2*0.02*d*fy, rearranging we get d=(M*1E6*2/0.02/fy)^(1/3) which gives a sensible effective depth for the section. Parameter sizes *h, b, bw & hf* respectively overall depth, breadths of flange and rib and thickness of flange, commence at parameter 17. The overall depth and breadth and thickness of the flange are limited by expressions involving *d*, the breadth of rib is in turn limited by expressions involving *b*. The remaining parameters have straightforward dependency conditions.

Table 10.6 Parameter table for reinforced concrete flanged beam design.

PARAMETER No.	name	Start zst()	End zen()	Type zty()	Dependency conditions and notes.
1	ans	0	0	0	Default values (1=Yes,0=No).
2	user	1	0	2	More detailed description.
3	Mbef	50	5000	0	Moment before redistribution.
4	cont	0	0	2	Continuous or not.
5	M	50	5000	0	=Mbef as beam not continuous.
6	fcu	30	60	4	Char. concrete strength.
7	hagg	10	60	6	Max aggregate size.
8	fy	250	460	3	Char. strength of longitudinal.
9	permn1	0.2	0.8	0	Minimum % when bw/b>=0.4.
10	permn2	0.2	0.8	0	Minimum % when bw/b<0.4.
11	dia	25	40	-3	Diameter of tension bars.
12	fyv	250	460	3	Char. strength of link steel.
13	dial	8	12	3	Diameter of link legs.
14	ccheck	1	0	2	Find cover (1=Yes,0=No).
15	d	250	2950	1	=(M*1E6*2/0.02/fy)^(1/3)
16	cover	20	80	1	>d*0.05 <d*0.1 Nominal cover.
17	h	300	3000	1	>d*1.1 <d*1.2 Overall depth.
18	b	300	3000	1	>d <2*d Breadth of flange.
19	bw	200	600	1	>0.3*b <0.7*b Breadth of rib.
20	hf	150	2950	1	>0.4*d <0.6*d Thick. of flange.
21	diac	16	25	-3	Diameter of compression bars.
22	d'	40	100	7	>d/10 <d/6 Depth to compr.
23	nbart	2	20	19	=4*M/(0.75*d*fy*PI*dia^2)
24	ans0	1	1	0	Check span/eff.depth ratio.
25	btyp	2	2	1	Cant./ss./con-one-end/both-con.
26	ans5	1	0	2	Comp. bars to contrl. defln.
27	nbarc	2	10	1	No. of compr. bars provided.
28	span	1	6	0	>8*d/1E3 <15*d/1E3 Span of beam.
29	ans1	1	0	2	Should perm. be x 10/span.
30	ans3	1	0	2	Find BM at bar curtail. points.
31	ans4	1	0	2	Data on anchorage & lap length.
32	Type	0	2	3	Plain, type-1&2 deformed round.
33	ans2	1	1	0	Undertake shear calculations.
34	V	1	10000	0	=4*M/span Ultimate shear force.
35	av	1E-6	20000	0	=2*d Distance from support.
36	re1	0	0	2	Respecify main tension bars.
37	dias	16	25	-3	Diameter of tension bars.
38	nbars	2	20	19	=nbart No. of bars effective.

```
39  re2    1       3       3      Options for comprn. bars.
40  diacs  16      25     -3      Diameter of comprn. bars.
41  nlegs  4       20      1     >nbars/2 <nbars   No. of legs.
42  flag1  1       2       2      Reduce spacing or links option.
43  dialr  8       12      3      Reduced dia. of link legs.
44  flag2  1       2       2      Adopt spacing or redesign optn.
45  sv'    50      3000    0     >d*.2 <d*.9 Chosen link spacing.
46  flag3  1       2       2      Options for incr. No. of legs.
47  ans6   0       0       2      Undertake another shear calc.
48  expos  1       5       5      Exposure condtn mild to severe.
49  mod    1       0       2      Systematic checking regime.
50  fire   0.5     2       4      Chosen fire resistance period.
51  expo   1       1       2      Indoor=1, outdoor=2.
52  aol    3       7       1      Age on loading 1 to 365 days.
53  Es     200E3   200E3   0      Young's modulus for steel.
54  Ec     28E3    28E3    0      Young's modulus for concrete.
```

BS 8110-1:1997 Clause 2.5.2 states that when linear elastic analysis is used, the relative stiffness of members may be based on:

- the concrete section
- the gross section on the basis of modular ratio
- the transformed section on the basis of modular ratio.

A modular ratio of 15 may be assumed, a consistent approach should be used for all elements of a structure. Thus BS 8110 permits linear elastic methods to be used for both the structural design of concrete frameworks and the design of reinforced concrete sections. This would permit a *self-check* to be included to compare the reinforcement required for elastically designed sections with that required by classical elastic methods.

NCE, 8 December 2005 reports *A £200M shopping complex in Bournemouth is being closed indefinitely* due to shear cracking and diagonal spalling at the ends of long span beams. Almost certainly the design complied with many of the clauses in BS 8110 but as the foreword to BS 8110 states in bold text **Compliance with a British Standard does not of itself confer immunity from legal obligations**. One such legal obligation is that a structure should be suitable for the purpose for which it was built. As mentioned in section 10.4, when limit state design was first introduced, with the exception of shear reinforcement for reinforced concrete beams, the equations were adjusted to give similar answers to designs produced in accordance with elastic theory. Prior to the introduction of limit state design, shear reinforcement design using stirrups, was based on the simple equation discussed by Pippard & Baker (1957) who derive the formula used in the code of practice *i.e.*

Stirrup load = S.p/a where *S* is the shear force at the section being considered, *p* is the distance between centres of stirrups along the length of the beam, *a* is the *moment arm* or *lever arm*. The strength of concrete was ignored as typically two thirds of the concrete is always in tension and thus cracked. Bray (1960) in figures 15 to 18, and Reynolds (1957) in figure 25 give typical examples for the provision of shear reinforcement prior to the introduction of limit state design. On 18.4.06 the writer and James Steedman, reminisced about the sixties, when we had say eight 1" tension bars at a simple support. Typically half of the bars were bent up to carry typically 50% of the shear force, the remaining 50% being carried by stirrups, now referred to as links. Today eight 25 mm tension bars would be curtailed to four at the support and just two 10 mm diameter link arms could be required. As James Steedman says "all of concrete failures are shear failures". For this reason, in the self check, shear reinforcement for the serviceability limit state for shear cracking will be calculated using the basic equation for stirrup load, as given above.

The self check, developed as part of this research, will be found in appendix C at the end of proforma calculation sc075.pro. Following the incorporation of the parameter table into sc075.pro, the model was run for various sets of automatically generated data to test for the presence/absence of bugs in the model. One bug was found:

- the variable *diacs* was erroneously set to zero causing problems when compression reinforcement was required.

More importantly, the automatic generation and running of up to 996 sets of data, highlighted strange behaviour with the shear check when the distance from the support was given as a low value. This matter is discussed in section 10.9.

10.8 Run time reporting

Even quite trivial programs with less than 10 programming structures, can have thousands of different paths through their logic. The table which gives the parameter specification will not guarantee that every single path through a *proforma* will be tested, but it does provide a reasoned approach to testing and it is straightforward to add the parameter table near to the start of a proforma calculation. The engineer is told when an error is found in an assignment or Boolean expression *etc.* and the line number at which the error occurred is reported. Errors are rare but when one does occur it is likely due to a parameter not being set. To trap errors in looping, a count is made of the number of times each and every line in the proforma calculation is accessed, and if the number exceeds 128,000 then an error in looping is reported.

SCALE proforma calculation sc924.pro is the program which generates the sets of data for verifying models for both the structural analysis of a framework and the structural design of components. Within sc924.pro is a procedure called *allcom*, short for *all combinations*, containing three nested loops, from outside to inside:

- the number of patterns (1, 3 or 6)
- the number of increments specified at run time (1 to 166 for 6 patterns)
- the number of parameters, typically 10 to 50.

Routine *allcom* currently assumes a limit of 996 files will be produced (scrtch.001 to scrtch.996) containing the sets of parameters. For 6 patterns then the number of increments must be <= 999/6 =166; *allcom* warns when the number of files exceeds 999. The maximum number of times that any line will be read, assuming 50 parameters, will be 6 x 166 x 50 = 49800 but will exceed the 128000 limit if the number of parameters increases to 150. Patterns are more applicable to sets of data generated for the structural analysis of frameworks, models for the design of structural components have a high degree of dependency among the parameters making patterns less important, accordingly just 3 patterns are suggested for testing models for the design of structural components.

10.9 Non intuitive design

In the sixties, all design equations were based on elastic section behaviour which was taught, understood, fully grasped by and intuitive to engineers. The position today is that few engineers, including the writer, understand the full implication of the rules, as the rules are no longer intuitive.

Code writers devise the rules, they receive comments from other engineers and add further rules which in turn generate more comments and yet more rules. Those who write computer software will immediately recognise that code writers enter the uncertain world colloquially referred to as *patches on patches*. Dijkstra (1972), the father of structured programming, writes "*As a slow witted human being I have a very small head and I had better learn to live with it and to respect my limitations, and give them full credit, rather than to try to ignore them, for the latter vain effort will be punished by failure*". Carl Sagan in "Cosmos" warns us that Ptolemy, astronomer and geographer, whose Earth-centred universe held sway for 1500 years, is a reminder that *intellectual capacity is no guarantee against being dead wrong*. Those who codify, should heed Dijkstra's & Sagan's warnings. They should also take heed of the *one:ten:hundred* rule, *i.e.* software written in one hour for the author's use, takes ten hours of work for use by colleagues and a hundred hours work for use generally. Ignorance of this rule results in codes being issued and then withdrawn *e.g.* BS 5950:1 2000 for the structural use of steelwork in building, also BS 8666:2000 for the scheduling of reinforcing bars.

The problem with codes of practice which have *patches on patches* is that very few who apply the rules and lamentably, those who codify the rules, can grasp the full implications of the application of any and every mix of the set of rules. As Diskstra tells us, we must respect our limitations. If the rules are intuitive then there is likely to be little confusion in their application; if the rules are non-intuitive then there is likely to be considerable confusion in their application.

As an example of intuitive design for structural steelwork, let us consider the design of a mild steel Universal Beam in accordance with BS 449:1959 and *Handbook for constructional engineers, 1964* by Dorman Long (1964). For a beam carrying a distributed *working load* of 20 tons, we read the size of beam directly from a *pink* (which denotes mild steel) table on page 380. To the right of the table, the maximum spacing between lateral supports is shown. The foot of the table tells us that tabular loads printed in italic type are within the web buckling capacity of the unstiffened web and produce a total deflection not exceeding 1/360th of the span, so we select from these a 16x7" x40lb/ft UB which will carry a safe distributed load of 22.5 tons, or a 16x6" x 40lb/ft UB which will carry a safe distributed load of 21.6 tons.

As an example of non-intuitive design for structural steelwork, we must follow BS 5950-1:2000, for a Universal Beam we need to:

- compute the factored bending moment, shear force (and axial load if any)
- establish distances between restraints about major & minor axes
- assess effective lengths
- classify the section
- compute the shear capacity and compare with the shear force
- compute the moment capacity and compare with the bending moment
- apply the interaction formula for the local capacity check
- apply the interaction formula for the overall buckling check if required.

As an example of intuitive design for reinforced concrete, let us consider the design of shear reinforcement for a concrete beam as enacted in the fifties. As discussed in section 10.7, Pippard & Baker (1957) derive the formula used in the code of practice *i.e.*

`Stirrup load = S.p/a`

where *S* is the shear force at the section being considered, *p* is the distance between centres of stirrups along the length of the beam, *a* is the *moment arm* or *lever arm*. In the formula for stirrup load, the strength of concrete is ignored as typically two thirds of the concrete is in tension and thus cracked; cracking is also caused by shrinkage. Increasing the pitch *p* or the shear force *S* increases the stirrup load, increasing the moment arm *a* reduces the stirrup load. The formula is both intuitive and sublime.

As an example of non-intuitive design for reinforced concrete, let us consider the design of shear reinforcement for a reinforced concrete beam in accordance with BS 8110-1:1997. As an introduction to the subject of shear in reinforced concrete beams designed in accordance with BS 8110, Allen (1988) writes:

Shear is to be considered at ultimate limit state only. No requirements are made for serviceability limit states*. When the bending moment is changing and a shear force is introduced, the equilibrium equations are complicated by the presence of the shear force and a new vertical equilibrium equation is required. The compatibility conditions must be altered to include shear displacements and the failure criterion must allow for concrete in states of biaxial, and in some cases triaxial, stress. A satisfactory design method which fulfils all these requirements has not so far been achieved, and Codes of Practice have therefore concentrated on producing reliable empirical methods of adding shear reinforcement to a structure to ensure that it has an adequate factor of safety at all points. Many people have carried out tests and put forward theories, more than in any other field.*

The above paragraph gives the background to the subject; BS 8110-1 section 3.4.5 includes equations, tables and a hundred requirements, conditions, provisos or qualifications. Equation 3 gives the value of **v** *i.e. the design shear stress at a cross-section*, and table 3.8 gives values for **vc** *i.e. the design concrete shear stress*. Note 2 in table 3.8 tells us that the table has been derived from the expression: `0.79*(100*As/(bv.d))^(1/3)*(400/d)^(1/4)/gammam` which is subject to six qualifications. Let us imagine that engineers trust this formula for all limit states, then for 0.25% tension reinforcement and an effective depth of 400 mm the design concrete shear stress from table 3.8 =0.40 N/mm². Clause 3.4.5.8 says that for sections near the supports, the design concrete shear strength `vc` may be increased to `2*d*vc/av` where `av` is the distance from face of support. Engineers know intuitively that shear forces are at a maximum at the support, so instinctively will substitute a small value for **av** for the design of shear reinforcement at supports, thus initiating structural collapse. In view of the foregoing, it is proposed that a check be made for the serviceability limit state for shear cracking, based on the concrete carrying a shear stress of the same magnitude as the long term concrete tension stress of 0.55 N/mm² given in BS 8110-2:1985 Clause 3.6, with the shear force over and above that carried by the concrete, being carried by stirrups designed by the classical formula `Stirrup load = S.p/a` where S is the shear force at the section being considered, p is the distance between centres of stirrups along the length of the beam, a is the *moment arm* or *lever arm*.

As Dijkstra (1972) says *we must respect our limitations, and give them full credit, rather than to try to ignore them, for the latter vain effort will be punished by failure*". To add a patch to the patches on patches, it is proposed that rules which are non-intuitive or which are counter intuitive as in the case of the enhancement of shear strength, should be explained.

This research is concerned with the correctness of structural engineering calculations, the foregoing is not a diatribe against modern codes of practice, it is included as it is important to note that modern codes of practice, by their increased complexity, are more prone to error than our classical codes of practice.

10.10 Some parametric dependency devices

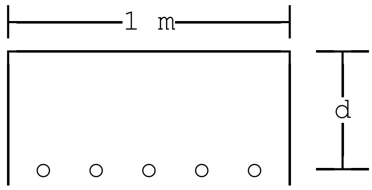
Selecting a suitable slab depth

The following extract is taken from the parameter table for sc080.pro. Given a design bending moment M, the effective depth d has to ensure that the strength of the concrete is not exceeded and that the amount of reinforcement is within a practical percentage e.g. from 0.4% to 2%. The extract achieves both requirements, an explanation follows the extract.

Table 10.7 Extract from a parameter table for a reinforced concrete slab.

```
PARAMETER   Start    End      Type     Dependency conditions
No. name    zst()    zen()    zty()    and notes.
 ...

 2  ans0    1        5        5        Location of section.
 3  sdr     1        5        126      20 7 20 23 26 Span:depth.
 4  M       5        400      0        Ultimate  moment of resistance.
 5  fcu     30       60       4        Char. concrete strength.
 6  fy      250      460      3        Characteristic steel strength.
 ...
10  dconc   75       375      1       =SQR(M*1E6/(0.1*1000*fcu))
11  d       1E20     1E20     1       =dconc > <
12  dia     3        8        100     =INT(d/12) Tens bar diameter.
 ...
21  span    2        20       0       =0.7*d*sdr/1000 Effective span
 ...
23  expos   1        5        5       >1 <INT(fcu/10)-1 Exp. condn.
 ...
(SQR(M*1E3/(0.75*fy*0.02)))
(SQR(M*1E3/(0.75*fy*0.004)))
```

These two expressions relate to the > & < for the 11'th parameter.



M kNm known, lever arm =0.75*d.
Let steel fraction of area =p,
then Ast mm² = d*1000*p and

$$M = \frac{Ast*fy*0.75*d}{10^6} = \frac{d*p*fy*0.75*d}{10^3}$$

$$\text{Rearranging} \quad d = SQR\left[\frac{M*10^3}{0.75*fy*p}\right]$$

Obviously an increase in steel fractional area *p* causes a reduction in *d* and vice versa. Taking a minimum fractional area of steel reinforcement of 0.004 i.e. 0.4% and a maximum fractional area of 0.02 i.e. 2.0%, to give a maximum & minimum slab thickness respectively, then:

`d >= SQR(M*1E3/(0.75*fy*0.02))` and `d <= SQR(M*1E3/(0.75*fy*0.004))`

as used in the PARAMETER table for sc080.pro. Expressions which follow the parameter table are associated with any isolated > = < found in the parameter table in order; this association is necessary when there is insufficient space to include the dependency condition/s on the line in the parameter table to which they refer.

For a minimum slab depth based on the concrete alone, supply dummy parameter, the 10th, dconc=SQR(M*1E6/(0.1*1000*fcu)) assuming a maximum of 30% redistribution, BS8110-1:1997 clause 3.4.4.4. Then *d*, the slab depth and 11th parameter, is set to *dconc*, the 1E20 for the start & end values means dynamically set the start and end values according to the > and < expressions which follow, as mentioned previously these dependency conditions follow the table because of insufficient space on the 11th line for their inclusion. When these conditions are applied, the assignment of *d=dconc* ensures that the concrete capacity is sufficient to support the bending moment *M* and the two dependencies ensure that the steel reinforcement is within normal percentages.

Selecting an exposure condition

Selecting an exposure condition is straightforward as the maximum exposure condition is dependent on just the concrete cube strength. Inspection of the parameter *expos*, the 23'rd parameter in the above extract from the parameter table for sc080.pro, shows that when fcu=30, *expos* is an integer number varying between 1 and INT(30/10)-1=2; when fcu=60, *expos* is an integer number varying between 1 and INT(60/10)-1=5. Thus *expos* is constrained by the concrete strength as appropriate.

Selecting a practical bar diameter

Selecting a practical bar diameter has to meet two requirements:

- the bar must be selected from the set: 6 8 10 12 16 20 25 40 50
- the bar must be a practical size for the section depth *e.g.* if section depth =100 mm, then a bar diameter of 50 mm would be wrong, if section depth =400 mm, then a bar diameter of 6 mm would be wrong, unless it was used to control surface cracking.

Specifying the parameter *Type* in the range 100 to 125 *i.e.* zty()=100 to zty()=125 accesses data stored in vectors za() to zz() respectively. Currently reinforcing bar diameters are stored in za() thus to access the bar diameters, Type=100 *i.e.* in the above extract from the parameter table for sc080.pro, zty(12)=100.

```
                ┌3'rd            ┌8'th
za(1)=VEC(6,8,10,12,16,20,25,40,50)
                └start value     └end value
```

The line from the parameter table below, tells the program to pick a bar diameter from the range 10 mm to 40 mm. If we divide section depth by 12 say and use as a trial diameter, for section depth =100, then bar diameter of 100/12 =8.33 mm
=400, then bar diameter of 400/12 =33.33 mm.

Obviously we need to use the trial diameter to select an available size from the set za(1)=VEC(6,8,10,12,16,20,25,40,50). This is done automatically when a dependency condition is provided e.g.

```
zva(3)=10mm┐      ┌zva(8)=40mm
12  dia    3      8      100     =INT(d/12) Tens bar diameter.
```

which is interpreted as: evaluate d/12 and find the nearest bar diameter within the range za(3) to za(8).

Selecting a suitable span

Selecting a sensible span has to meet two requirements:

- the span must give a sensible span:depth ratio for the continuity specified e.g. fixed, pinned ends or cantilever
- the span:depth ratio must be found from the location reference given in response to the prompt for the parameter *ans0* with values 1 to 5 corresponding to five location descriptions for the section.

From Table 10.7 it can be seen that parameter 2, *ans0,* offers the user options 1 to 5 corresponding to beam span:depth ratios respectively: 20 7 20 23 & 26, given in parameter 3, to accord with the order required by proforma sc080.pro. When Type=126, subsequent numbers are stored, in this case parameter 3 stores five span:depth ratios. When ans0=1 sdr=20, when ans0=2 sdr=7, when ans0=3 sdr=20, when ans0=4 sdr=23, when ans0=5 sdr=26. For Type=126, the pattern *i.e.* increasing or decreasing from zst to zen is always made the same as the previous parameter, which in this case is that of parameter 2. Parameter 3 is a dummy parameter *i.e.* it is not used in the model, but it is used in the expression given in parameter 21 to compute a sensible span for the current set of data which is being built.

Subscripted parameters

Occasionally it is advantageous to use subscripted parameters e.g. for a continuous beam having 10 spans stored in s(1) to s(10), then assuming s(1) was the 4th parameter, the follow section from a parameter table would suffice and describe parameters s(1) to s(10) in one line of the parameter table instead of 10 lines.

```
PARAMETER   Start    End     Type     Dependency conditions
No. name    zst()    zen()   zty()    and notes.
 ...
4 13 s      1.5      6       0        Spans s(1) to s(10).
14 ...
```

Optionally the 13, corresponding to s(10) may be omitted from the table as it may be computed to be one less than 14, the parameter number at the start of the next line. When a subscripted parameter is the last in the parameter table, both parameter numbers must be shown.

Restricting distributed loading to be within varying spans

An extract from a parameter table follows, which in turn is followed by a description.

```
PARAMETER   Start    End     Type     Dependency conditions
No. name    zst()    zen()   zty()    and notes.
 ...
2  ns       7        7       0        Number of spans.
3  sp       1        6       0       =RAN(29)*5+1  Spans l. to r.
10 uj       1        7       1       =INT(RAN(47)*ns+0.5) Span No.
17 23 splu 0         20      0       =+sn=zva(zp'-7),ztm=sp(sn)
```

Parameter 2 sets the number of spans *ns* to 7. Parameter 3 sets seven spans i.e. sp(1) to sp(7) to random spans in the range 1 to 6 m; the number in brackets following the RAN function is a *seed* for the random number generator. As mentioned previously, it is permissible to include the end parameter number on the line as in:

```
3  9 sp     1        6       0       =RAN(29)*5+1  Spans l. to r.
```

where parameter 3 refers to sp(1) and parameter 9 refers to sp(7), or omit the 9 as in the extract. When the 9 is omitted, it is deduced by the program from the first parameter number on the next line. Parameters 10 to 16 select a random set of seven span numbers, where parameter 10 refers to uj(1) and parameter 16 refers to uj(7). The problem is to find the span lengths corresponding to the random span numbers uj(1) to uj(7). This requires two stages:

- selecting the span numbers stored in uj(1) to uj(7) by sn=zva(zp'-7), where zp' is always the current parameter number
- assigning the current value of the parameter, which is always *ztm*, to the current span number *sp(sn)*.

The two (or more) assignments are concatenated, with comma/s as separator/s. A plus sign follows the first equals sign to let the program know that multiple assignments follow.

It has been found that with a little bit of thought, it is possible to engineer dependency conditions for most situations which arise, only having to resort to writing procedures for tables of steel section properties.

Chapter 11

Discussion

Although a unified treatment has been developed for both the verification of models for the structural analysis of frameworks and models for the structural design of components, it is convenient to report on each as a separate entity. Verifying the correctness of structural engineering calculations means establishing the truth of software models by examination or demonstration. Verification is defined as the satisfactory completion of a batch of 996 runs of engineered data from the parameter table. To obtain satisfactory completion of all 996 runs at the first attempt is rare, generally half a dozen, or more, batch runs are necessary to engineer out the blips, each time learning something about the nature of the data and modifying the parameter table accordingly, typically adding extra dependencies to prevent unpractical sets of data being generated.

11.1 Models for structural analysis

Discussions are given in order of verified model *number* as listed in section 7.8. The discussions are distilled versions of those given with each model. For reason of space, only the kernel of each model is included in appendix A but the unabridged set of verified models with notes on both theory and practical matters will be found in Appendix A. Before embarking on a thousand runs for full verification, it is prudent to vary just one parameter at a time keeping all others constant. The easiest way to do this, is to assign the required parameter to be varied, on a new line following the line commencing #cc924.stk, which imports sets of data for verification, leaving all the default values as originally set.

vm110 Deflection of beams including shear *cf.* Chebyshev polynomials

Verification is by comparison with both classical theory & the Chebyshev polynomials theory (Rolfe, 2004) for the shear force, bending moment, rotation & deflection at the centre & quarter points on a simply supported beam. Deflection computed using Chebyshev polynomials ignored shear deformation as shear deformation was not taken into account in the derivation of the theory.

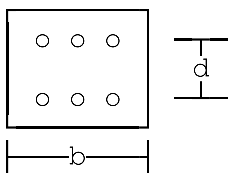
vm112 Cantilevered beam *cf.* equilibrium, compatibility & energy

Verification for the cantilevered/propped cantilevered beam model is by comparison with the general theory developed in chapter 8. By comparison of the propped cantilever results with those for the unpropped cantilever, it is clear that, for accuracy, a much higher number of segments is needed for the propped case than the unpropped case. The significant percentage differences are due to the audit of internal strain energy and external work done. For the unpropped cantilever the bending moment diagram does not have a point of contraflexure, for the propped case it does; the much *curvier* bending moment diagram needs a closer spacing of nodes to give the same accuracy for the energy audit as for the unpropped case. Following the first set of runs for verification, the default number of segments was increased from 16 to 32 to give more accurate results.

vm113 Cantilevered beam with many loads *cf.* unit load method

Grassie (1957) derives the unit load method from first principles and subsequently notes that the working formula for the determination of the deflection at any section of a straight beam is the same form as that derived by Castigliano's First Theorem Method. From preliminary runs of the model it was found that for a reinforced concrete beam, if the span:depth ratio for the cantilever is not less than 7.5, then shear deformation will not exceed 2% of bending deformation and may be ignored as 2% is small in comparison to the percentage variability of the concrete.

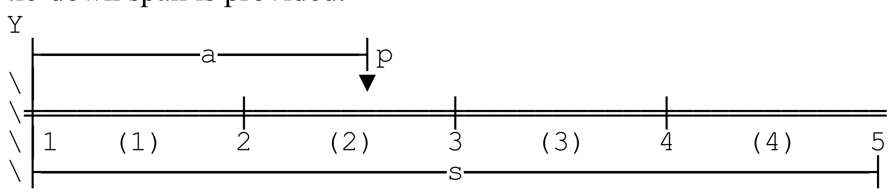
Most cantilevered beams are made from reinforced concrete. The deflection of a reinforced concrete beam depends on the amount of reinforcement contained in the beam and the creep factor of the concrete. When, as is usually the case for a cantilever, it is important that the deflection be controlled, *steel-beam theory* is the traditional method of design. In this method, the tension and compression reinforcement are made equal and the moment of inertia is computed from the reinforcement acting alone. When the amount of tension and compression reinforcement are equal, and the area of concrete is ignored, then the creep deformation of the concrete may be ignored.




```
EcIc concrete, for creep factor =2,
=(28E6/(2+1))*bd^3/12 =0.7777E6*bd^3
EsIs steel, for 1% reinforcement t&b
=205E6*(bd/100)*d^2 =2.05E6*bd^3.
Then EcIc/EsIs =0.7777/2.05 =0.379,
thus EcIc ≈0.38% reinforcement t&b.
```

vm114 Tapered cantilevered beam *cf.* unit load method

See vm113 above, for notes on the deflection of a reinforced concrete beam. The cantilever is tapered, this means it has to be segmented. The beam of span *s,* has *nsg* segments; thus for 4 segments the joint and member numbers will be as shown, members in brackets. The cantilever is rigidly supported at joint 1, this implies that a tie-down span is provided.



Many loads may be applied at various distances from the left support, thus which member they come within and the distance from the start of that member is required. Simple logic will suffice, for a point load *p* at distance *a* from joint 1:

Load contained within member	mn=INT(nsg*a/s)+1 and distance
from the start of that member	l=a-(mn-1)*s/nsg.

Beam Deflection by The Unit-load Method:
Grassie (1957) derives the Unit-load Method from first principles, and subsequently notes that the working formula for the determination of the deflection at any section of a straight beam *i.e.*

$$del = \int_0^L \frac{M.m.dx}{EI}$$

is the same form as that derived by Castigliano's first theorem method. For the application of the unit load method for a tapered cantilever, see vm114.ndf in Appendix A.

vm115 Cantilevered beam with tie down span *cf.* Roark

```
═══════════════════════════
▲                   ▲              Span l is the tie-down span
├────────l──────────┼──c─┤         for the cantilever of span c.

        Point load │p          ┌──────────────────────┐                          ┌────┐
                   ▼           │w load/unit leng.│                          │ cw │
══════════════════════        ══════════════════════        ══════════════════════
▲       (a)        ▲          ▲      (b)          ▲         ▲     (c)       ▲
```

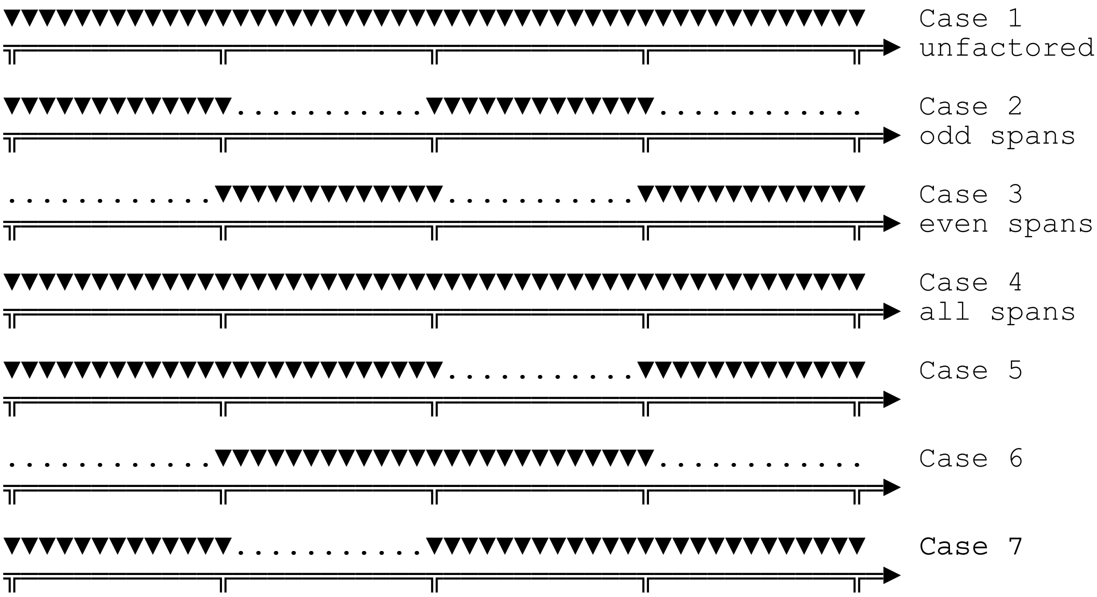
Three types of load are considered (a-c) as above. To ignore any type, set its magnitude to zero *i.e.* p=0, w=0, cw=0. Both w & cw are given as load per unit length. Downward loads are negative. By combining (b) & (c) with a sign change, loading on the span but not on the cantilever may be modelled. See vm115.ndf in Appendix A for formulae used.

vm117 Subframe, continuous beam + columns *cf.* equilibrium, compatibility & energy

Subframes (as with continuous beams) are subjected to loading patterns of live load; BS 8110 specifies adjacent bays and alternate bays. Forty years ago great emphasis was placed on saving material at the expense of design office time. Today the emphasis is on simplicity in the design with generous imposed loads sufficient to accommodate change of use of the building over its working life; thus several point loads on a span are often lumped into the distributed load.

The first load case is for serviceability (unfactored dead plus imposed); the second for factored dead+imposed on odd spans (left to right); the third for factored dead+imposed on even spans (left to right); the fourth for factored dead+imposed on all spans;

thereafter for factored dead+imposed on adjacent spans, thus for a general continuous beam, diagrams follow where . . . . . denotes unfactored dead load only.



For compatibility, energy, local & overall equilibrium check see chapter 8. For a full discussion see the notes in vm117.ndf in Appendix A.

vm120 Continuous beam *cf.* Hardy Cross

Verification is by comparison of NL-STRESS with the Moment Distribution method of Prof Hardy Cross (Cross, 1929 & 1932), for a continuous beam subjected to a mix of UDL's, point loads & linearly varying loads, with moment enveloping for the various combinations of DL & LL, with factoring in accordance with BS 8110. The moment distribution was simple to program. To envelope the various loading combinations specified in BS 8110, the number of loadings (load cases) is four for one or two span beams, else seven. Although a continuous beam of more than one span is expected, a single span may be analysed. Forty years ago great emphasis was placed on saving material at the expense of design office time. Today the emphasis is on simplicity in the design with generous imposed loads, sufficient to accommodate change of use of the building over its working life; thus several concentrated or partial distributed loads on a span are often lumped into the distributed load rather than being treated separately as was done forty years ago.

vm122 Two member lean-to or Mansard beam *cf.* equilibrium, compatibility & energy

For compatibility, energy, local & overall equilibrium, and Clerk Maxwell, Betti, Southwell check see the notes in vm112.ndf. A rectangular section is assumed, so that the model may be used with steel, concrete & timber sections. When the section thickness is given as zero, a solid section is assumed. To allow for the considerable shear deformation associated with timber, BS 5268 states that the modulus of rigidity should be taken as Young's Modulus divided by 16. The modulus of rigidity g=e/(2*(1+nu)), where nu=Poisson's ratio. For g=e/16 then e/16=e/(2*(1+nu)), equating the denominators 8=1+nu, thus nu=7. This may seem strange, but it is a BS 5268 requirement.

vm123 Three member lean-to/Mansard beam *cf.* equilibrium, compatibility & energy

vm124 Three member cranked beam *cf.* equilibrium, compatibility & energy

Both models listed above *i.e.* vm123 & vm124 have similar discussions to that for vm122, so are omitted for reason of space.

vm130 Ground beam on an elastic foundation *cf.* Hetényi

Verification is by comparison of NL-STRESS with the classical solution (Hetényi, 1948). An engineers' arithmetic check is also included, in which it is assumed that the beam is infinitely stiff; thus from the centroid of the loads a linear pressure beneath the beam is given by P/A±MY/I and from this linear pressure the bending moment at each load position may be calculated.

From Terzaghi (1955), the engineer assesses *k'*, the modulus of subgrade reaction (units kN/m3 *i.e.* pressure to give the soil unit deflection) by means of charts and tables, taking due account of the foundation size and the distribution of loads. The coefficient of subgrade reaction is then multiplied by the area (assumed lumped at a spring support) and the resulting spring stiffness used in the data. If the soil is of poor quality, the value of k' can be increased by: compacting soil; stabilizing the soil with cement or lime; applying a well compacted subbase of sufficient thickness; removing the poor quality layer and replacing it with well compacted sand or crushed stone, stabilised sand or lean concrete. The k' value cannot be used as a measure of settlement. The settlement must be calculated on the basis of the results of a geotechnical study. A.A.Alexandrou, formerly of the University of Greenwich has provided a table of moduli of subgrade reactions (k') which is contained in Appendix A. Soils do not behave in a linear elastic manner in the long term, they settle due to pore water dissipation and other effects which compact the soil, such as vibrations. Engineers measure the void ratio of the soil to estimate the amount of consolidation expected in the long term. It is normal to assume that the self weight of the ground beam is supported directly by the supporting soil, therefore the self weight is omitted from the model.

From preliminary runs, varying the centres of joints *i.e.* centres of springs, a close comparison between the two methods was obtained when the beam depth was greater than twice the joint centres. From an engineering viewpoint, a point load may be assumed to be spread at 45° from the top of the beam to the neutral axis. The vertical distance from the top of the beam to the neutral axis is half the beam depth, *i.e.* equal to the suggested centres for the joints. From preliminary runs varying the modulus of subgrade reaction, when the soil stiffness is increased from humus soil to crushed stone with sand, the average percentage difference between NL-STRESS & Hetényi (1948), increases from 0.21% to 3.19%.

Following the comparison between Hetényi (1948) and the stiffness method, comes the traditional method of analysis, entitled Engineers' Arithmetic. For this method, the centre of loading is first found, then pressures at each end of the ground beam are computed from P/A±M.y/I, assuming a linear pressure distribution beneath the ground beam, then moments & shears at load positions are calculated. The bending moments and shears computed by engineers' arithmetic do not agree with those computed by NL-STRESS or Hetényi. NL-STRESS/Hetényi take the soil stiffness into account, engineers' arithmetic does not; however, engineers' arithmetic and NL-STRESS/Hetényi can be reconciled by reducing the modulus of subgrade reaction to a very low value, thereby making the beam so stiff by comparison with the soil, that the pressure distribution beneath the beam is linear. Exact agreement between NL-STRESS & engineers' arithmetic can be seen from the table below, when the modulus of subgrade reaction =1E-3 kN/m3, *i.e.* 1 kN/m² (weight of a 16 stone man) spread over an area of 1 m² and resulting in a deflection of 1000 m =1 km.

Modulus of subgrade reaction kN/m3	Bending moment at the first load, Case 1. Nl-STRESS	Engineers' Arithmetic
10000	158.346	153.409
100	154.039	153.409
1	153.416	153.409
1E-3	153.409	153.409

Further runs, again varying just one parameter at a time, showed that the number of springs (nj) modelling the subgrade reaction, was insufficient. The dependency conditions were adjusted in the light of the above, such that the number of joints =INT(10*l/d), where *l* is the length of the beam and *d* is the depth. Using this new dependency condition, a study was made of the results of 996 runs with shear deformation suppressed, for which the average percentage difference was found to be 0.399%. Inspection of the averages for each run showed that most of the runs had an average percentage difference near zero, whereas one or two runs had appreciable differences, the largest being in run 832 which had a difference of 34.55%, accordingly this set of data was studied. The first obvious item of data to be considered in run 832, was the high number of joints =183, the previous dependency condition was =INT(2*l/d) which had been changed to INT(10*l/d), for the beam length=5.5758, breadth=4.9758, depth=0.30424, thus: number of joints =INT(10*5.5758/0.30424) =183.

Changing the:

Number of joints	366	183	91	45
Average %age diff.	0.001	34.55	0.049	0.50

The mystery deepens, closing in:

Number of joints	181	183	185	187	189
Average %age diff.	0.022	34.55	34.54	34.55	0.040

Inspection of the structure with the loading superimposed, gives a clue to what is happening. There are two sets of loads, each set having 5 loads, when the percentage difference is very low both sets come on the beam, when the percentage difference is high, only 4 out of the five loads in the second set come on the beam; thus the problem is due to roundoff. Placing load on the member was controlled by the following which is copied from the data.

```
IF x>som AND x<=som+crs THEN m FORCE Y CONCENTR. P p(lc) L x-som
```

The logic for this is robust, for all spans except the last one. Any load which occurs almost at a joint position will either be considered on the current span, or left for the next one, but for the last span there is a possibility that sometimes a load will be fluked onto the end of the beam, and sometimes will not. The problem could be solved by a special case for the last span.

```
IF x>som AND x<=l THEN m FORCE Y CONCENTRATED P p(lc) L l-som
```

More elegantly both cases are covered by:

```
eom=som+crs l'=x-som ;IF m=nm THEN eom=l l'=l-som
IF x>som AND x<=eom THEN m FORCE Y CONCENTRATED P p(lc) L l'
```

The dependency conditions were adjusted in the light of the above to be `nj=INT(4*l/d)`.

vm131 Ground beam on elastic piles *cf.* flexibility

Piles supporting a ground beam rarely behave as rigid supports. Loads applied to the beam cause the piles beneath to shorten and the beam to spread the load to adjacent piles. The amount of spreading is a function of the ratio of beam flexibility to pile flexibility. Verification of the NL-STRESS analysis is by the flexibility method to compute the shears, forces and settlements for a ground beam subjected to a train of loads such as those from a crane. The load train position may be stepped automatically.

For a continuous beam on elastic supports, if the internal support reactions are chosen as releases, expressions may be derived for:

a) settlement due to loading
b) flexure due to loading on released structure
c) settlement (-ve) due to unit bi-actions at and corresponding to the releases
d) flexure due to unit bi-actions at and corresponding to the releases.

Expressions a,b,c and d are continuous functions between end supports. Separate expressions are required when the loading comes on any cantilever. This choice of releases has the advantage that it is unnecessary to check in which internal span any particular load occurs and is therefore particularly convenient as the program is designed to step the loading any chosen increment to the right. The resulting flexibility matrix is not very well conditioned, but the degree of ill-conditioning has been checked and has been found to be quite acceptable when using double-precision arithmetic (15+ decimal digits). The terms of the flexibility matrix are evaluated using Simpson's rule. The matrix is inverted using Fox's method. In 1968, the writer, then a Civil design engineer with George Wimpey, developed the necessary theory and used it for the design of the ground beams supporting the Goliath crane at the Harland & Woolf shipyard. Since then, whenever there is an outside broadcast from Belfast, the Goliath crane is used as a backdrop, alas its days and the shipyard's days are numbered. The theory is contained in the file vm131.NDF in Appendix A.

It is normal to connect the piles to the ground beam: for concrete piles by taking the pile reinforcement into the ground beam; for steel bearing piles, casting the pile heads into the ground beam. Thus, although lift off can occur for ground beams supported by soil, lift off will not normally occur for ground beams supported by piles as the friction between the pile shaft and the soil will normally prevent lift off. The connections between pile heads and the ground beam are normally assumed to be pinned, thus the rotational stiffnesses are set to zero in the data but they may be changed if required. As the flexibility theory ignores rotational stiffness, the structural effect of any moment connection between pile heads and ground beam can be seen by inspection of the percentage differences given in the summary.

Before launching a thousand runs - to simulate the mixture of parameter variation likely in general usage - it is prudent to carry out several single runs varying just one parameter at a time. These preliminary runs are reported in the file vm131.NDF in Appendix A. As in all numerical studies, odd values crop up which are not as expected; the maximum percentage differences can be greatly influenced by one or two low values, of course one way to avoid this is to relate the difference between the NL-STRESS values and the flexibility approach to the average bending moment, but such a device complicates the matter.

vm140 Influence lines *cf.* Müller-Breslau

Verification is by comparison with the classical methods of Müller-Breslau for creating influence lines for: reaction, shear, and moment. The procedure used was originally derived by Donald Alcock in the seventies, but never published. The values for shear are those due to a unit load applied just to the right of the position to which the value is referenced. The procedure sets up and reduces a stiffness matrix in which each beam element contributes the submatrices. For each influence line, forces are applied to the appropriate element. The deflected form of the beam is the influence line by the Müller-Breslau principle. Areas in each span are computed by Simpson's rule.

vm150 Pratt through truss *cf.* method of joints

Lattice girders and portals offer a lightweight and architecturally interesting alternative to heavy long span beams; the latticing permits building services to be incorporated within the depth of the lattice. The analysis assumes that all members are pinned at their joints *i.e.* PLANE TRUSS, and that the truss is statically determinate. By reference to the structure plots, it is straightforward to edit the data to change the structure shape and/or take into account more general loadings.

The *Method of Joints*, as used for the verification of the *Pratt through truss*, is a traditional method for the analysis of pin-jointed trusses in which the engineer first computes the reactions by equilibrium *i.e.* applying ΣX=0 ΣY=0 ΣMZ=0, and then proceeding from the left support such that only two unknown member forces occur at each joint *i.e.* the same sequence that an engineer would follow in the manual solution of a truss. A good description of the 'Method of Joints' is given by Gennaro, chapter 6 (1965). Applying high loads can give central deflections in excess of 1 km, and although the axial forces in the members computed by NL-STRESS and the method of joints agreed precisely, obviously some engineers' arithmetic is needed to control the range of loads. A uniformly distributed load of 10 kN/m² is a sensible maximum for a truss; the spacing of trusses is unlikely to exceed the span *a* of the trusses; it follows that a uniformly distributed load of 10*a kN/m is a sensible maximum and from this value the maximum bending moment may be found; and thereby the maximum force in the chords, and using a sensible permissible stress the cross-sectional area of the chords may be found. In the verification, the range of cross-sectional areas is limited from a single 152x152x23 Universal Column section up to a maximum of 10 No. 356x406x634 Universal Column sections.

vm153 Pratt deck truss *cf.* method of joints
vm156 Howe through truss *cf.* method of joints
vm159 Howe deck truss *cf.* method of joints
vm162 Warren through truss *cf.* method of joints
vm164 Warren through truss with verticals *cf.* method of joints
vm165 Warren deck truss *cf.* method of joints
vm168 Warren deck with verticals *cf.* method of joints

All seven models listed above *i.e.* vm153 thru vm168 have similar discussions to that for vm150, so are omitted for reason of space. With the exception of vm162, the top chord for the models may either be flat or may form a ridge at the centre.

vm171 Two rafters with tie *cf.* method of joints

Morley (1948) describes two rafters with a single tie forming a roof as being: suitable for small spans, typically 4.5 m. There is always confusion with the use of the word truss. STRESS (1964), implicitly defined a TRUSS as a collection of members, all with pin jointed ends; a FRAME as a collection of members assumed with fixed ends unless MEMBER RELEASES were provided at either or both ends. Steel and timber truss manufacturers use the term TRUSS to mean a frame with a mixture of pinned & fixed ends. Statically determinate roof trusses could be analysed as TYPE PLANE TRUSS, but this would not reveal bending stresses in the middle of members due to loading distributed along the length of members, accordingly this model is declared as TYPE PLANE FRAME and member releases are provided to make the structural behaviour similar to that of TYPE PLANE TRUSS. To show member forces and stresses in the middle of each member, the command: NUMBER OF SEGMENTS 2 TRACE is used. TRACE tells NL-STRESS to include the forces and stresses at the end of each segment in the results.

vm172 Two rafters, post & tie *cf.* method of joints

Morley (1948) describes two rafters with a post and a single tie forming a roof as being: suitable for small spans, typically 6m. When it is required that a steel suspension rod be provided instead of the centre post, then work in equivalent timber units by multiplying the cross-sectional area of the rod by: Young's modulus steel/Young's modulus timber, then take the square root of the equivalent area to give the side of a square, and assign depth & breadth equal to the equivalent side of the square, finally the computed stresses in the rod need to be multiplied by Young's modulus steel/Young's modulus timber. Assuming that units are kN & m throughout, then stresses will be in kN/m^2 units, so divide by 1000 to get to N/mm^2. For a statically determinate plane truss, assuming deflections are small, as all members are pinned at their ends, the forces in the members are independent of the sectional properties.

vm173 King post roof truss *cf.* method of joints

Morley (1948) describes a King Post roof truss as being suitable for frames with larger spans, typically 9 m. Optionally the joint in the centre of the tie, may be raised to increase the headroom beneath, or set to zero for a horizontal tie. For a statically determinate plane truss, assuming small deflection theory with all members pinned at their ends, the forces in the members are independent of the sectional properties. Vertical joint loads from: purlins, ridge & fascia boards, supports for water tanks, walkways, storage, out of plane struts and ties, loft conversion *etc.* may be input directly as joint loads.

vm174 Three segment rafters, Pratt internals roof truss *cf.* method of joints

Morley (1948) describes this roof truss as being: suitable for larger spans. Optionally the joint/s in the centre of the tie, may be raised to increase the headroom beneath, or set to zero for a horizontal tie. For a statically determinate plane truss, assuming small deflection theory with all members pinned at their ends, the forces in the members are independent of the sectional properties. Vertical joint loads from: purlins, ridge & fascia boards, supports for water tanks, walkways, storage, out of plane struts and ties, loft conversion *etc.* may be input directly as joint loads.

vm175 Three segment rafters, Howe internals roof truss *cf.* method of joints

vm177 Trussed rafter, or Fink roof truss *cf.* method of joints

vm178 Three segment trussed rafter, Warren internals roof truss *cf.* method of joints

vm179 Three segment rafters, Warren internals roof truss *cf.* method of joints

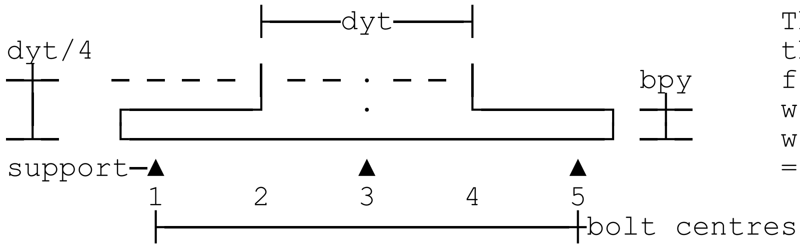
All four models listed above *i.e.* vm175 thru vm179 have similar discussions to those for vm174, so are omitted for reason of space, but all are contained in Appendix A.

vm181 Mansard truss *cf.* method of joints

Morley (1948) describes this roof truss as being: suitable when roof space is to be utilised for rooms, otherwise the discussion is as for vm175.

vm202 Pipe tree having two branches *cf.* equilibrium, compatibility & energy

For compatibility, energy, local & overall equilibrium, and Clerk Maxwell, Betti, Southwell check see the notes in vm112.ndf.



The model assumes that the baseplate can not flex within trunk width and its depth within trunk width =(trunk width)/4.

A pipe tree comprising centre *trunk* and horizontal *branches* for supporting pipes is used in the petrochemical and other industries. Pipe anchors stop movement of the pipes along the length of the pipes, except for expansion, and the pipe tree is usually designed for vertical loading from the weight of the pipes and lateral loading caused by wind on the pipework and the pipe support structure. Frequently pipe trees have severely unbalanced loads when pipes are supported on just one side, either permanently or during maintenance or replacement of the pipes; thus the tree should be designed for the worst unbalanced case expected. When expansion along the length of the pipes is considerable *e.g.* when the pipes are carrying liquid hydrogen, the lagged pipes are usually supported on a V set of rollers so that the expansion does not cause out-of-plane forces. If the rollers are kept lubricated the assumption that there are no out-of-plane forces is valid; if the structural engineer asks the mechanical services engineer for the highest out-of-plane forces that could arise - and designs for these - the resulting design will be uneconomic. In extreme cases very high out-of-plane structural stiffness can lead to bucking of pipes. Much of the country's pipe runs and their supporting structures are routinely wrapped in Densotape - a petroleum based product similar to Sylglass - both products being supplied by Wynne & Coales. The effort required to wrap a structural hollow section, rather than an I-section, with Densotape, is greatly reduced; furthermore the integrity of the wrapping is greatly improved. Thirty years ago, structural hollow sections were considerably more expensive than open sections such as UBs, UCs and angles; today there is little difference in the cost, so structural hollow sections are a good choice. If RHS (rectangular hollow sections) are used and the out of plane dimension of both the *trunk* and *branches* is made the same, then the curvature on the corner of the trunk will form a natural *weld pool* for approximately half of the weld length required to weld the branches to the trunk.

vm203 Pipe tree having four branches *cf.* equilibrium, compatibility & energy
vm204 Pipe tree having six branches *cf.* equilibrium, compatibility & energy

Both models listed above *i.e.* vm203 & vm204 have similar discussions to that for vm202, so are omitted for reason of space.

vm207 One storey bent, vertical/raking columns *cf.* equilib., compatibility & energy

For compatibility, energy, local & overall equilibrium, and Clerk Maxwell, Betti, Southwell check see the notes in vm112.ndf. Bents are used for supporting access roads to jetties and other offshore structures such as mooring and berthing dolphins.

vm208 Two storey bent, vertical/raking columns *cf.* equilib., compatibility & energy

vm209 Three storey bent, vertical/raking columns *cf.* equilib., compatibility & energy

Both models listed above *i.e.* vm203 & vm204 have similar discussions to that for vm207, so are omitted for reason of space.

vm210 Bent *cf.* column analogy

Verification is by comparison of the matrix stiffness method with column analogy. Column analogy was devised by Professor Hardy Cross (Cross, 1929,1932) more limited than his moment distribution, as column analogy only applies to singly connected structures. On the lines following the SOLVE command, come the calculations for the column analogy. Calculations may be displayed by starting each line with an asterisk and including a plus sign in front of each assignment *e.g.* * +a=h/(e*iz1)+l/(e*iz2)+h/(e*iz1) will cause the formula for the analogous area column to be displayed followed by the value assigned to *a*. This formula for area, which starts with h/(e*iz1) looks strange to engineers who are unfamiliar with the analogy, this area is the product of length of the member and 1/EI for the member.

vm211 Rigid pile cap *cf.* Reinforced Concrete Designers' Handbook

A classical treatment of the *loads on piles in groups* problem is given in table 19, Reynolds (1957). The post-processing calculations at the end of the NL-STRESS results are for Reynolds' treatment of the data. Reynolds is wrong for the axial loads caused by the horizontal force, he takes moments about the position of full fixity but ignores the fixing moments. To avoid the need to consider the fixing moments, take moments about O at the level of contraflexure.

```
           ┌──────────────────────────────┐
           │        Rigid pile cap        │    hl
───        └┬──────┬─────────────┬──────┬─┘  ◄──  ───
 │h/2       ║      ║      Y      ║      ║          │   h is depth to full
 │          ║      ║      │      ║      ║          │   fixity, thus pile
───         ╫      ╫      O──X   ╫      ╫        h │   bends in double
            ║      ║             ║      ║          │   curvature due to
1 pile──►   ║      ║◄─2      2─► ║      ║◄─1 pile  │   horizontal load.
into     __ ║______║_____________║______║__into    │   ╫ denotes point of
media //    ║\\//\\║//\\//\\//\\/║\\//\\║//media───     contraflexure.
           f1     f2            f3     f4
            ├──1───┼──1───┼──1───┼──1───┤
```

Equilibrium check: anticlockwise moments about O, acm=hl*h/2. Clockwise moments cm=2*(2*f1+2*f2), as pilecap is rigid f2=f1/2 & cm=2*(2*f1+f1); equating moments: hl*h=12*f1 thus f1=hl*h/12. Using the default data with the vertical loads suppressed, f1=40*9.8/12=32.67 f2=f1/2=16.33 for each of the two piles, f3=-16.33 for each of the two piles,f4=-32.67, where negative denotes tension. A second mistake in Reynolds' treatment is noted, his Km=x/(NI) should not have the N (number of piles into media) as Km has units m^-1, the confusion is caused by the fact that Reynolds' number of piles has an implied area, consequently its units are m^2.

One assumption in the classical treatment is that the pile cap is rigid, the diagram given by Reynolds says *Rigid pile cap* but the span:depth ratio of 9 shown in the diagram can lead to the impression that *rigid* means a sturdy beam rather than something which is more or less infinitely rigid *e.g.* a moment of inertia =1E12. Engineers know that the displacement of a cantilever due to an end load applied to cause bending, far exceeds the displacement of a load applied to cause axial shortening, thus a beam of span:depth ratio of 9 with a central point load of 800 kN, sitting at the centre of say 4 equally spaced piles along the beam's length, will not have a uniform pile load of 200 kN per pile as computed by Reynolds' simple method. In the default data, a square beam of side 2 m is assumed, which for a 9 m long ground beam gives a span:depth ratio of 9. If the piles are assumed driven to bedrock *e.g.* support spring stiffness =1E12 kN/m, then for a symmetrical centre load of 800 kN, one pile per row, no sway load, but other data (as supplied) the pile loads for the 4 piles are: 14, 386, 386, 14 kN. If the spring stiffness of the support is assumed as say 100 tonnes to give 10 mm displacement, *i.e.* 1000/.01 = 100000 kN/m, then the pile loads for the four piles are: 124, 276, 276, 124 kN.

It will be clear from the above that the engineer can get any pile loads required by changing the beam and support spring stiffness. The inclusion of the classical results as a comparison to NL-STRESS shows the importance of getting the right support stiffness. As with so much of structural engineering, the hard part is making a reasonable assessment of the support conditions. To model for the soil stiffness beneath each pile, springs are used, the stiffness of each spring depends on the modulus of subgrade reaction of the area of soil *lumped* to the pile, and the cohesion down the length of the pile. The definitive work on subgrade reaction was by Terzaghi (1955). From Terzaghi the engineer assesses a soil stiffness (units kN/m3 *i.e.* pressure to give the soil unit deflection) by means of charts and tables, taking due account of the foundation size and the distribution of loads. The coefficient of subgrade reaction is then multiplied by the area assumed lumped at a spring support, and the resulting spring stiffness used in the data.

vm215 Portal frame *cf.* equilibrium, compatibility & energy

Classical solutions provide bedrock beneath the matrix stiffness method of analysis; for frames containing: optional pins, vertical & horizontal joint loading, loading along & transverse to the members, *etc.* for which no classical solution exist, chapter 8 provides a self checking solution. The method developed can be applied to all types of structural frameworks and is thus more powerful than any classical method.

vm216 Mansard portal *cf.* equilibrium, compatibility & energy
vm217 Gable frame with inclined legs *cf.* equilibrium, compatibility & energy
vm218 Portal with skew corners *cf.* equilibrium, compatibility & energy
vm219 Trapezoidal frame *cf.* equilibrium, compatibility & energy
All four models listed above *i.e.* vm216 thru vm219 use the same self checking method to that of vm215 and thus have similar discussions to those for vm215, so are omitted for reason of space, but all are contained in Appendix A.

vm220 Two bay ridged portal *cf.* Kleinlogel
Kleinlogel's formulae are straightforward to use and easily included between the SOLVE & FINISH keywords. Kleinlogel's formulae omit axial and shear deformation.

vm223 Multi bay ridged portal, pinned/fixed feet *cf.* equil., compatibility & energy
Care is needed when abstracting section properties from handbooks where they are usually tabulated in *cm* or *in* units. An easy way to convert SI units is to use exponent form; a value of 2345 (units of cm4) may be entered in the data as 2345E-8 (units of m4) because the E says *...times ten to the power of.* An easy way to convert US customary units is to use an expression; a value of 2345 (units of in4) may be entered in the data as 2345/12^4 (units of ft4).

vm225 Couple roof frame *cf.* equilibrium, compatibility & energy
vm226 Couple close roof frame *cf.* equilibrium, compatibility & energy
vm227 Collar-tie roof frame *cf.* equilibrium, compatibility & energy
vm228 Collar-and-tie roof frame *cf.* equilibrium, compatibility & energy
vm230 Attic room roof frame *cf.* equilibrium, compatibility & energy
vm232 Fink room roof frame *cf.* equilibrium, compatibility & energy
vm233 King post roof frame *cf.* equilibrium, compatibility & energy
vm234 Queen post roof frame *cf.* equilibrium, compatibility & energy
vm235 Tied Mansard roof frame *cf.* equilibrium, compatibility & energy
All nine models listed above *i.e.* vm225 thru vm235 use the same self checking method to that of vm215 and thus have similar discussions to those for vm215, so are omitted for reason of space, but all are contained in Appendix A.

vm241 Vierendeel girder *cf.* equilibrium, compatibility & energy
A rectangular section is assumed, so that the model may be used with steel, concrete & timber sections. When the section thickness is given as zero, a solid section is assumed. To allow for the considerable shear deformation associated with timber, BS 5268 states that the modulus of rigidity is to be taken as Young's Modulus divided by 16. When the model is made from timber sections, it is frequently referred to as a *roof truss*, but the truss is analysed as a PLANE FRAME rather than as a PLANE TRUSS. Even fully triangulated roof trusses, need to be analysed as a PLANE FRAME when, as is normal, the engineer wishes to consider the rafters as continuous, but the internals as pin ended. By reference to the structure plot/s, it is straightforward to edit the data to change the structure shape and/or take into account more general loadings.

vm242 Vierendeel roof frame *cf.* equilibrium, compatibility & energy
vm244 N/Pratt lattice portal/girder *cf.* equilibrium, compatibility & energy
vm245 Howe lattice portal/girder *cf.* equilibrium, compatibility & energy
vm246 Warren portal/girder end diags in tension *cf.* equilib., compatibility & energy
vm247 Warren portal/girder end diags in compr. *cf.* equilib., compatibility & energy

All five models listed above *i.e.* vm242 thru vm247 use the same self checking method to that of vm241 and have similar discussions to those for vm215, so are omitted for reason of space, but all are contained in Appendix A.

vm260 Multi-storey frame *cf.* Hardy Cross

Verification is by comparison of the matrix stiffness method with moment distribution. On the 80 lines following the SOLVE command, come the calculations for the moment distribution method, these are followed by calculations for tables comparing results between the two methods. The procedure for moment distribution for multi-storey frames follows Gennaro (1965). If axial and shear deformation are suppressed, by setting Poisson's ratio =1E-12 which in turn sets shear areas to zero and multiplies cross-sectional areas by 1E6, then results agree precisely for all sets of data generated from the parameter table. When neither shear nor axial deformation is suppressed and Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for all sets of data generated from the parameter table =6.20%; the largest individual result being for run 664 when the average difference was 47.29%. Investigation of run 664 showed that the set of data had large differences in the storey heights, which were: 2,10,10,2,10 and large differences in the spans, which were: 25,25,2,25,25. Such spans and storey heights are quite possible, the writer recalls a standard hospital system called *Harness* which had pairs of columns closely spaced as corridors for people & mechanical services, with large spans between; similarly irregular storey heights are commonplace in industrial buildings, especially in the petrochemical industry where framing is defined entirely by vessel sizes and their best positions from a process viewpoint; Tate Modern has large spans.

```
Poisson's ratio        0.3      0.2      0.1     1E-6     1E-12
Average %age diff.    47.29    47.20    47.12    47.07     0.0
```

The above shows that when Poisson's ratio is changed from 0.3 to 1E-6 shear deformation makes little difference to the average percentage difference between the classical and modern methods. It is only when Poisson's ratio =1E-12, for which cross-sectional areas are multiplied by 1E6, see data within MEMBER PROPERTIES, that the average percentage difference drops to zero. Axial shortening does affect bending moments, particularly those in the outside columns of multi-storey frames. Axial effects must always be included, as structural members do change in length when axial loads are applied, the change in length of columns due to axial loads can significantly affect the bending moments in connected beams. The long term creep behaviour under compressive load for concrete in columns, which increases the compression stresses in the column reinforcement over design limits, is a concern.

vm262 Multi storey frame *cf.* equilibrium, compatibility & energy

Multi-storey frames subjected to vertical loads and sway loads are frequently analysed to compare wind moments with gravity load moments, and thereby assess whether wind loading is covered within the permitted stress increase for wind.

vm270 Pierced shear walls *cf.* Magnus

This analysis models coupled shear walls taking into account the shear deformation of the stiff storey-deep members which join the coupling beams to the centre lines of the shear walls. A shear wall may be modelled as a member running along the centre line of the wall, connected to short stiff members parallel to the end faces as described in section 3.8. The technique assumes symmetry in plan, not a building with its lift shaft and stair wells tucked away at one end, Schwaighofer & Microys (1969), also MacLeod (1973). NL-STRESS automatically considers shear deformation for all members for which the shear area is given, so the special calculations given by Schwaighofer and Microys need not be carried out. The shear area of rectangular sections is taken as 5/6 of the cross sectional area, Roark (1965). Before launching a thousand runs - to simulate the mixture of parameter variation likely in general usage - it is prudent to carry out several single runs varying just one parameter at a time. From these single runs, which engineers call *getting a feel for the problem*, the results show that Magnus gives higher forces in the lintels than NL-STRESS; this will be due to the fact that shear deformation reduces the stiffness of the lintel. The prudent engineer would design for the maximum result from either method.

vm280 Two pinned circular arch *cf.* Pippard & Baker

Verification is by comparison with the classical solution given by Pippard & Baker (1957). When a curved beam is bent about the plane of initial curvature, plane sections remain plane, but because of the different lengths of fibres on the inner and outer sides of the beam, the distribution of strain and stress is not linear; the neutral axis therefore does not pass through the centroid of the section and the relation M/I=E/R=f/y does not apply. The error involved in using this relation is slight as long as the radius of curvature is more than 10 times the depth of the beam, but becomes large for sharp curvatures. Pippard and Baker (1957) give a more exact treatment due to Winkler. Rib shortening is taken into account. It is suggested that the number of segments be an even number which will ensure a joint is provided at the centre of the arch, although the model will run correctly for an odd number of segments.

vm281 Encastré circular arch *cf.* Pippard & Baker

Verification is by comparison with the classical solution given by Pippard & Baker (1957), see the discussion for vm280.

vm282 Two pinned parabolic arch *cf.* Pippard & Baker

Verification is by comparison with the classical solution given by Pippard & Baker (1957). The parameter names have been kept close to those used by Pippard & Baker, so engineers may follow the logic by reference to Pippard & Baker. When a curved beam is bent about the plane of initial curvature, plane sections remain plane, but because of the different lengths of fibres on the inner and outer sides of the beam, the distribution of strain and stress is not linear; the neutral axis therefore does not pass through the centroid of the section and the relation M/I=E/R=f/y does not apply. The error involved in using this relation is slight as long as the radius of curvature is more than 10 times the depth of the beam, but becomes large for sharp curvatures. Pippard and Baker give a more exact treatment due to Winkler. The number of segments must be an even number so that a joint is provided at the crown. Pippard & Baker's thorough treatment of the circular arch, taking rib shortening into account is replaced by a cursory treatment for the parabolic arch, rib shortening is not mentioned by Pippard & Baker. Grassie (1957) prefaces his treatment with 'Neglecting rib shortening'. Grassie gives the same result as Pippard & Baker. Roark (2002) by Young & Budynas, omits parabolic arches, presumably because some readers reported shortcomings with the formula in the Roark (1965) which does not give the same result as Pippard & Baker (1957) and Grassie (1957). Morley (1948), which gives the same result as Pippard & Baker and Grassie, states: *The case of a parabolic rib is much simplified if we make the reasonable supposition that the value of the I varies proportionally to the secant of the slope of the rib, being unity at the crown, E being constant*. Obviously Morley's assumption has also been made by Pippard & Baker, and Grassie, though not stated by them. The matrix stiffness method distinguishes between local and global axes, the section properties are specified in the local axes, thus the secant adjustment is not appropriate for the matrix stiffness method, nevertheless it is included in the model for comparison but may be *commented out i.e.* suppressed. From preliminary testing, it is clear that the change in height of the arch has a major effect on the horizontal support reaction. Obviously as the arch becomes flatter, so the axial load increases and in consequence, so does the axial shortening. Although the rib shortening effect is taken into account for circular arches, it is ignored in the classical treatment for parabolic arches. To compare like-with-like, rib shortening (axial deformation) can be ignored in NL-STRESS by multiplying the cross-sectional area by 1E6. When this is done, close agreement is obtained between NL-STRESS and the classical method for the horizontal support reaction.

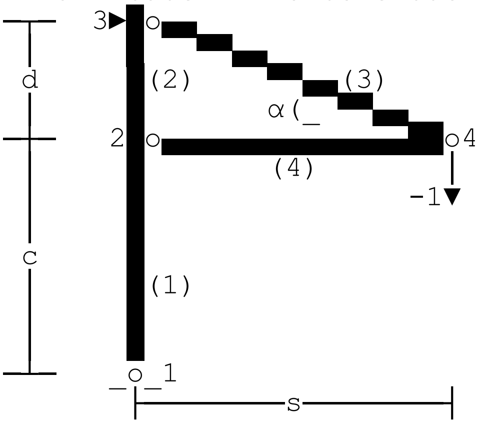
vm283 Encastré parabolic arch *cf.* Pippard & Baker

An abbreviated discussion is as for vm282, a full discussion including *getting a feel for the problem* will be found in Appendix A.

vm290 Outrigged frame *cf.* Castigliano

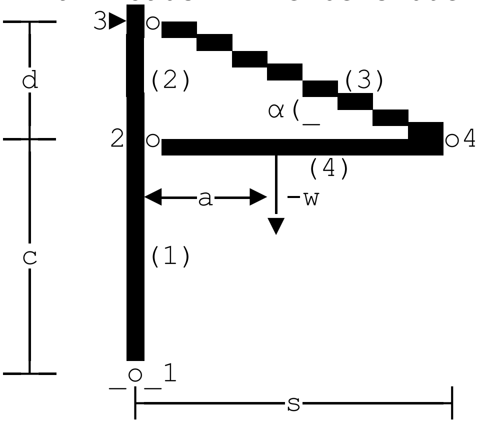
This braced outrigged frame is taken from Example 1.12 by Grassie (1957). By inspection of the outrigged frame and the moment releases, members 1 to 4 carry axial loading; members 1, 2 & 4 carry bending moments. Because the bending moments in members 1, 2 & 4 vary along the length of each member, then each needs an integration. The axial loads in members 1 to 4 are constant throughout the length of the member and therefore the integration is trivial. As the downward displacement at joint 4 is required, a unit load is applied in the downward direction. Firstly the strain energy due to axial loads is found, the subscript denotes the member number, axial loads are found by the method of joints.

```
Axial loads in members due to downward unit load at joint 4:
```




```
sinα=d/SQR(s^2+d^2)
cosα=s/SQR(s^2+d^2)

-1=n3*sinα therefore
n3=-SQR(s^2+d^2)/d=-l/d

n1=n2=1

n4=-n3.cosα
  =SQR(s^2+d^2)          s
   ───────────── . ───────────── =s/d
         d         SQR(s^2+d^2)
```

```
Axial loads in members due to load 'w' distance 'a' from support:
```




```
Component of -w at 4 =-w*a/s
                at 2 =-w(s-a)/s

-w*a/s=N3.sinα therefore
N3=-w*a*l/(s*d)

N4=-N3.cosα
     w*a*SQR(s^2+d^2)*s
  = ──────────────────── = w*a/d
     s*d*SQR(s^2+d^2)

N2=w*a/s  N1=w
```

```
The deflection component at joint 4 due to axial strain energy,
writing l=SQR(s^2+d^2) for column ax2=ax1, is:
```

$$da = \int_0^c \frac{N1.n1.dx}{e.ax1} + \int_0^d \frac{N2.n2.dx}{e.ax2} + \int_0^l \frac{N3.n3.dx}{e.ax3} + \int_0^s \frac{N4.n4.dx}{e.ax4}$$

$$da = \frac{w.1.c}{e.ax1} + \frac{w.a.1.d}{s.e.ax2} + \frac{l.w.a.l.l}{s.d.d.e.ax3} + \frac{s.w.a.s}{d.d.e.ax4}$$

```
da=(w*c/ax1+w*a*d/(s*ax1)+w*a*l^3/(s*d^2*ax3)+w*a*s^2/(d^2*ax4))/e

Bending strain energy is required for members 1,2 & 4.
The deflection component at joint 4 due to bending strain energy
```

$$db = \int_0^c \frac{M1.m1.dx}{e.iz1} + \int_0^d \frac{M2.m2.dx}{e.iz2} + \int_0^a \frac{M4.m4.dx}{e.iz4} + \int_a^s \frac{M4'.m4'.dx}{e.iz4}$$

```
For column, iz2=iz1.
M1 at x in member 1 = (w.a/d)*d/(c+d).x = w*a*x/(c+d)
m1 at x in member 1 = (s/d)*d/(c+d).x   = s*x/(c+d)
M2 at x in member 2 = (w.a/d)*c/(c+d).x = w*a*c*x/(d*(c+d))
m2 at x in member 2 = (s/d)*c/(c+d).x   = s*c*x/(d*(c+d))


As m4 & m4' are both zero because unit load is applied at joint 4,
then 3rd & 4th integrals are both zero.
```

$$\int_0^c M1.m1.dx = \int_0^c \frac{w.a.x\verb|^|2*s.dx}{(c+d)\verb|^|2} = \left[\frac{w.a.s.x\verb|^|3}{3*(c+d)\verb|^|2}\right]_0^c = \frac{w*a*s*c\verb|^|3}{3*(c+d)\verb|^|2}$$

$$\int_0^d M2.m2.dx = \int_0^d \frac{w.a.c\verb|^|2.s.x\verb|^|2.dx}{(d*(c+d))\verb|^|2} = \left[\frac{w.a.c\verb|^|2.s.x\verb|^|3}{3*(d*(c+d))\verb|^|2}\right]_0^d = \frac{w*a*c\verb|^|2.s.*d\verb|^|3}{3*(d*(c+d))\verb|^|2}$$

```
The deflection component at joint 4 due to bending strain energy:
db=w*a*s*c^3/(3*(c+d)^2*e*iz1)+w*a*c^2*s*d^3/(3*(d*(c+d))^2*e*iz1)
Total deflection at joint 4 in Y direction: dy4=da+db.
```

vm291 Braced outrigged frame *cf.* Castigliano

This braced outrigged frame is taken from Example 1.13 by Grassie (1957) and uses Castigliano's First Theorem method as also used in vm290. It would have been a similar approach to the above, that Adolf Kleinlogel and Arthur Haselbach (civil engineer and Klienlogel's co-worker of many years) used when devising their Rahmenformeln, Kleinlogel (1956). Whenever possible, Kleinlogel made use of intermediate levels which he called *Coefficients* when they referred to the geometric & material properties of the frame, and *Constants* when they referred to the loading.

vm300 Cantilever or propped cantilever *cf.* equilibrium, compatibility & energy

The normal length:minimum-width ratio for a cantilever is 12:1 thus for a width of 0.3 m the maximum cantilever length would be 3.6 m. The zero percentage difference throughout the range of the cantilever length, contrasts with the percentages given for the cantilever when modelled as a plane frame member rather than as a plane grid member. The reason for the difference is that for a plane frame member the checks include axial displacement effects, whereas for a plane grid member axial effects are assumed to be negligible. By comparison of the propped cantilever results with those for the unpropped cantilever, it is clear that, for accuracy, a much higher number of segments is needed for the propped case than the unpropped case. The significant percentage differences are due to the audit of internal strain energy and external work done. For the unpropped cantilever the bending moment diagram does not have a point of contraflexure, for the propped case it does; the much *curvier* bending moment diagram needs a much closer spacing of nodes to give the same accuracy for the energy audit. Following the first set of runs for full verification, the number of segments was increased from 16 to 32 to give more accurate results.

vm301 Circular arc cantilever *cf.* Pippard & Baker

Classical methods for the analysis of bow girders and cantilevers are covered in few textbooks. Bhatt (1999) gives an example of a quarter circle bow girder subjected to a uniformly distributed load in his section 1.18.1. and an example of a quarter circle cantilever bow girder subjected to end moments in section 10.8.2. Young & Budynas (Roark, 2002) in their table 9.4 give 29 pages of complicated formulas for various out-of-plane loads on bow girders with various support condition. Coefficients for many of the formulas have to be looked-up in tables, derivation of the formulas is not given. Verification is by comparison with the classical solution given by Pippard & Baker (1957). The parameter names have been kept close to those used by Pippard & Baker, so engineers should be able to follow the logic and theory by reference to Pippard & Baker.

vm302 Circular arc bow girder *cf.* Pippard & Baker

Discussion as for vm301, omitted for reason of space.

vm310 Grillage of beams *cf.* Pilkey & Chang

The verification is with the classical solution by Pilkey & Chang (1978) who use a Navier approach with a double trigonometric expression to provide a relationship between displacements and loading for regularly spaced beams forming a grillage. They define their model as a *uniform grillage*, for which the beams parallel to the X-axis are denoted *girders*, those parallel to the Y-axis denoted *stiffeners*. The stiffeners may be larger/smaller or more/less numerous that the girders, but the distributed loads are always distributed to the stiffeners. The girders are identical in size, end conditions, and spacing. The same holds true for stiffeners, although the stiffeners and girders may differ from each other. Pilkey & Chang warn "The beams may be open or closed cross section, although for closed sections there may be an error of up to 5 percent because the torsional rigidity of the beams is not taken into account". As with all infinite series, a finite number of terms must be chosen. Pilkey & Chang suggest *ns* terms where *ns* is the number of stiffeners, then subsequently say if the loading is not uniform, more terms of the series must be employed. From several test runs, *ns***ng* terms were chosen, where *ng* is the number of girders.

vm311 Grillage of beams *cf.* equilibrium, compatibility & energy

For compatibility, energy, local & overall equilibrium, and Clerk Maxwell, Betti, Southwell check see chapter 8.

vm410 Plastic analysis of cantilever *cf.* equilibrium, compatibility & energy
When the plastic capacity at the support is exceeded, then rotation will take place at the support. If, when a plastic hinge formed, the hinge were modelled by a pin (hinge of zero stiffness), then the cantilever would collapse completely. In real structures, strain hardening ensures that for a slowly increasing load, complete collapse of a cantilever would not occur. NL-STRESS initially models each plastic hinge by a weak spring of stiffness 2/(No of loading increments), thus for 100 loading increments, stiffness=0.02 kNm/rad, *i.e.* 20N (2 lb) on a lever arm of 1m to cause a rotation of 1 radian (57.3°). For this weak spring, NL-STRESS computes the rotation, and for the computed rotation, uses a spring stiffness of Mp/rotation, where Mp is the plastic moment at the support, cycling until equilibrium and compatibility are obtained. The procedure for the above is contained in chapter 8.

vm411 Plastic analysis of propped cantilever *cf.* equilibrium, compatibility & energy
Discussion is similar to that for vm410, omitted for reason of space.

vm420 Plastic analysis of continuous beam *cf.* equilibrium, compatibility & energy
For notes on strain hardening & plastic hinges *etc.* see vm411.ndf in Appendix A. Point loads on the beam may be applied at any position; for point loads applied very close to supports, the shear strain energy is predominant over bending strain energy. To give good agreement between the NL-STRESS results and its check, for a span of length *s*, it was found necessary to limit point loads to be within the range 10% to 90% of *s*. Preliminary testing used sets of data extracted from various publications as quoted in Appendix A. Each set of data may be copied to file cc924.stk so that it is imported *via* the #cc924.stk in the data.

vm430 Plastic analysis of rectangular portal *cf.* equilibrium, compatibility & energy
When any continuous process is modelled by a number of discrete steps *e.g.*

- the loading is applied in 100 increments
- cycling at each load increment is carried out to introduce or unload one plastic hinge at a time
- each member is segmented so that plastic hinges may be located within members

then the model needs rigorous testing to find and explain any rogue results brought about by any combination of the discrete steps. Such rigorous testing must be carried out before release of the software, the following is an actual example.

The plastic moment capacity Mp in the presence of a shear force is reduced to Mpr. Let the shear force be denoted F, then the shear stress on the web of an I section is F/(D*tw) where D is the depth of the Isection & tw is the web thickness. Let the yield stress be denoted py, and the yield stress in shear be denoted pv, then pv=py/SQR(3) according to von Mises. Then equation 3.14 by Horne and Morris (1981) may be written: Mp-Mpr =D^2.tw.py/4*(1-SQR(1-(F/(D*tw)/pv)^2)). Let the right hand side be denoted *rhs* then Mp-Mpr=rhs, dividing by Mp then 1-Mpr/Mp=rhs/Mp, rearranging then Mpr/Mp=1-rhs/Mp, where the left hand side is a factor by which Mp must be multiplied to take account of shear. The ratio Mpr/Mp was computed by NL-STRESS for values of shear stress which exceed 0.5*pv.

No problems occurred with the above treatment until verification of vm430.ndf. On the first run using 996 sets of data generated, the analysis failed in runs 146 & 160, which reported a failure in the square root routine, which on investigation was traced to the above equation. Obviously when F/(D*tw*pv) exceeds 1 then the SQR will not have a real root, or in engineering terms the equation will fail when the shear stress exceeds the permissible.

Such a bug could go undetected for many years, it is of paramount importance that verification should be carried out by the author/s of software; locating such a bug by an engineer who was unfamiliar with programming, or by an IT person who was unfamiliar with the engineering, would be difficult. For the author of the software, the bug was located and fixed in no longer than the time taken to write the above four paragraphs; thus verification has a sustainability dimension.

vm435 Plastic analysis of ridged portal *cf.* equilibrium, compatibility & energy

When a symmetrical loading is applied to a symmetrical sway frame such as a ridged portal, the resulting deflections are symmetrical to an accuracy of 15+ decimal digits. For method *plastic* the left and right eaves intersection points move outwards and downwards as the loading is increased. When the loading is sufficiently high, plastic hinges will form at the eaves. Because only one plastic hinge is permitted to form at a time (there are many cycles in each loading increment) then the first plastic hinge is replaced by a pin and equal and opposite compensating plastic moments about the hinge. In the next cycle at the same loading level, an identical plastic hinge will form at the opposite eave, and will be replaced by a pin and equal and opposite compensating plastic moments about the hinge. In a real structure, because of rolling tolerances, there will not be exact symmetry and only one plastic hinge will form at any loading level. Three strategies are available to prevent both plastic hinges forming at the same time, these three strategies follow.

(1) Multiply Young's modulus by 10. This reduces displacements tenfold, and removes *stability* from the analysis and therefore gives the collapse mechanism for the classical plastic theorem assumptions of: small deflections and no stability problems *i.e.* linear elastic plastic analysis.

(2) Apply a member distortion to one of the columns. Such a member distortion would make the model more representative of the real frame. The distortion should model typical rolling tolerances *e.g.* 0.2% of length.

(3) Apply a bow to all members and reverse the start & ends of the right hand column.

Strategy (1) is the simplest but ignores stability. Strategy (2) is straightforward to apply but needs the stain energy due to the distortion to be included in the strain energy check. Strategy (3) is easy to apply (add '0.2' after the number of segments command), makes the model closer to the real behaviour of a frame, and is therefore used in this model.

For notes on strain hardening & plastic hinges *etc.* see vm411.ndf in Appendix A. Point loads on the rafters may be applied at any position; for point loads applied very close to columns, the shear strain energy is predominant over bending strain energy, accordingly the position of point loads is controlled to be within the range 10% to 90% of the member carrying the point load. NL-STRESS permits the engineer to model the hinge stiffness remaining after a plastic hinge has formed by specifying a percentage of the plastic moment following the METHOD command *e.g.* METHOD PLASTIC 5 which would specify that 5% of the plastic moment be used as the hinge stiffness. If the percentage is omitted NL-STRESS assumes a percentage of 100/(number of loading increments) *i.e.* 1% for a loading applied in 100 increments.

Traditionally, the *Collapse Method* factored the combined dead & imposed load with a load factor of 1.75, to compare published worked examples with NL-STRESS results, the dead+imposed load is doubled and applied in typically 200 increments, if collapse occurs *after* loading increment 185 has been carried, then the load factor =185*2/200 =1.85.

For verification of a model for plastic analysis, it is necessary to ensure that sets of data are designed such that collapse will occur at a sensible number of loading increments. This means that the loading must be related to Mp. Engineer's arithmetic is given in model vm411 in Appendix A, to relate the applied loading to Mp.

NL-STRESS uses interaction equations from Horne & Morris (1981), extended for the effects of shear forces. NL-STRESS takes stability into account as well as plastic behaviour. It has been found from extensive testing that occasionally the combination of non-linear behaviour including shear forces with that of stability, flukes a stable condition with a large deflection. Such a situation can be detected by plotting the deflected shape with TRACE added to the number of increments, or detected by inspection of the strain energy and change in slope checks. Usually, just changing *nli* the NUMBER OF INCREMENTS slightly, is sufficient to exclude the rogue case. As an example, for one run which had *nli=200* and reported an average percentage difference of 19.625% between NL-STRESS and the checks, when *nli* was changed to 199, the percentage difference was reduced to 0.123%. Respectively the load factor was reduced from 1.22 to 1.216.

Very many runs were required to engineer the verification data to sensible ranges of parameters *e.g.* the depth of the sections was originally based on a portal-span:rafter-depth ratio in the range 30 to 50. This range gave accuracies of the percentage difference between NL-STRESS and its checks in the range 10-20%. When the span:depth ratio was increased to the range 60 to 75, the percentage differences were reduced to below 10%.

vm436 Plastic analysis of multi bay ridged portal *cf.* equilib., compatibility & energy

vm440 Plastic analysis of multi storey frame *cf.* equilibrium, compatibility & energy

An abbreviated discussion for vm436 & vm440 is as for vm436, a full discussion will be found in Appendix A.

vm501 Cantilever beam in space *cf.* equilibrium, compatibility & energy

For a 3D structure for which partial varying distributed loads may be applied to any member in or about any axis for BETA (the angle of rotation of the member about its centroidal axis) set to any value, the formulation of the data, although complicated, is treated rigorously by NL-STRESS. For a self check, it would be simple to pick up the loads after they have been distributed by NL-STRESS to the joints but this would impinge on the independence of the self check. After some deliberation, it was decided for the self check to convert all loading on the members to the joints at the end of each segment, in much the same way as finite element analysis does, thus avoiding all the complications of partial distributed loads which vary in or about the three axes. As with the finite element method, the accuracy will be compromised by a coarse mesh, a suggested minimum number of segments is 16; the parametric formulation of the model makes it easy to experiment with variation of the number of segments.

vm510 Four legged stool space frame *cf.* equilibrium, compatibility & energy

The model allows for the provision of fixed or pinned feet. For releases about the X, Y & Z axes for all feet, it was possible to achieve a linear elastic analysis without excessive cycles to achieve satisfaction for equilibrium & compatibility. For non-linear elastic analysis it was not possible to achieve satisfaction for equilibrium & compatibility as the structure oscillated about the Y axis. Accordingly for the pinned feet case, feet were only released to permit rotation about the X & Z axes. Although engineers refer to pinned feet, it is general practice to fix the feet of a structure *e.g.* by bolting, to prevent rotation of the columns about their centroidal axis.

vm520 Spiral stairs space frame *cf.* equilibrium, compatibility & energy

Modern spiral, formerly *corkscrew* stairs are formed by a single curved reinforced concrete plate. (The term *staircase* refers to *the stairs and the part of the building containing the stairs*). For spiral stairs, made from concrete, the flight is usually modelled as a single member. Each change in *going*, *rise* or diameter of the spiral, means a great deal of numerical work computing joint coordinates; parametric data reduces the workload dramatically, less than ten parameters define the geometry. Changing these takes just a minute, rather than hours renumbering and recomputing. The data assumes a stair commencing with an optional straight flight, then a curved flight, then an optional final straight flight. To omit the initial and/or final straight flights, set the number of treads *nti* & *ntf* respectively, to zero. The risers for the three flights are set independently thus the flights may be flat or pitched in any combination *e.g.* the stair parameters may be set to model a flat bow girder by setting *ri*, *rc* and *rf* to zero. The first flight always starts at the origin with the flight pointing along the X axis, a positive moment about the Y axis is clockwise when looking along the Y axis *i.e.* up, therefore anticlockwise in plan when looking down. Set the angle turned through to a positive value to curve the stair anti-clockwise when viewed on plan, or negative to curve the stair clockwise when viewed on plan.

How the stair is supported, can have a considerable bearing on the forces and deflections. It is up to the engineer to make an assessment of the conditions available on site and to supply continuity reinforcement as required for the assumed support conditions. If the lower support is at ground level, then stiffness can be cheaply provided by a mass concrete block to resist coincident forces in the X, Y and Z directions and moments about the X, Y and Z axes. Small movements in the foundation can make a large difference in the bending and torsional moments and displacements of the stair. As the file is text, it may be edited before running the analysis. The support at the start of the stair is shown as: 1 FORCE X -1 Y -1 Z -1 MOMENT X -1 Y -1 Z -1 which (as -1 means fixed) means that joint 1 is fixed in the X, Y & Z directions, and fixed against rotation about the X, Y & Z axes. If the engineer wants to release say, the fixity about the Y axis, then the above command should be amended to: 1 FORCE X -1 Y -1 Z -1 MOMENT X -1 Z -1.

The short term value of Young's modulus is given in BS 8110, and is generally used in structural analysis. The creep factor for concrete varies depending on the amount of reinforcement and the nature of the loading: bending, compression, torque *etc.* A typical value for the creep factor is 2, *i.e.* the long term deflection is twice the short term deflection; thus the overall deflection - being the sum of the short term and the long term deflection - is three times the short term deflection.

Because of high torsional stresses occurring in reinforced concrete spiral stairs, engineers often cast an insitu spine beam typically 250 x 250mm and use precast concrete treads with toughened glass risers. Such a compact section is more torsionally efficient and more structurally predictable than a wide shallow waist, which has *tearing* forces on the inside radius. Being able to see through the risers, provides more visual/architectural interest. When a spine beam is not acceptable, all stirrups should be code 33 or 63 to BS 8666:2005, arranged so that all longitudinal bars are held in 2 directions, just as in a column. To control cracking, a larger number of smaller diameter bars is preferable to a smaller number of large diameter bars. For typical spiral stairs, if there is such a case, a maximum bar diameter of 12mm is traditional; older engineers mutter "half inch at 6" centres both directions, top and bottom".

The first thing that happens when a firm gets a spiral stair is that the whole firm jams the stair for a photograph. Because the stairs are stepped, crowd packing density can be higher than for people at the same level. As the reinforcement and concrete cost is typically 15% of the overall cost, being generous with the loading provides cheap insurance for both engineer and client. The loading is generally assumed to be applied along the centroid of the flight, as the treads are wider at the outside of the flight, then an increased weight on the outside causes a torque about the centre of the flight. The natural frequency of the stair when carrying dead and factored imposed loads, to allow for crushing, should be checked.

vm601 Plate with point loads *cf.* Navier double trigonometric series

Verification is by comparison with the Navier Solution for which the deflection of a simply supported rectangular plate can always be represented in the form of a double trigonometric series, Timoshenko & Woinowsky-Krieger (1959). This model is for a slab which is simply supported; inspection of the results will show that near the corners there are tie down forces. These forces are not imaginary they exist in real life. Cut a stiff rectangular piece of cardboard 150x150mm, and arrange four plastic CD cases placed flat on a desk to form a square with the hinged sides, leaving a hole in the middle. Place the cardboard over the hole and press down in the middle of it, noting that the corners rise. John Rolfe, an email colleague, reports a real life situation where a simply supported slab was constructed, supported on compression-only bridge bearings. When heavy traffic came onto the slab, the corners lifted-off then thumped down with the passage of each set of wheel loads.

The torsional stiffness of reinforced concrete beams under long term loading has been investigated by Goode (1975), quoting from his conclusions:

- The initial torsional stiffness was accurately predicted by the elastic theory based on the concrete section only.
- The increase in rotation with time under a sustained torsional load was considerable; after about three weeks the rotation was double its initial value and after a year was five times its initial value.
- The amount of reinforcement had little influence on the behaviour under sustained load except when the torsional moment was sufficiently large to cause considerable cracking of the beam.

Dr Swan - formerly with the C&CA - told the writer that the figure of 0.5 x St Venant value first given in CP 110 (1972), was the highest value that he found in the C&CA tests, most values were very much lower. Unless a simply supported floor slab is prestressed in both directions, then lateral strain is unlikely to be significant. For this reason West and other researchers at the old C&CA advocated that reinforced concrete slabs be designed as a grillage of strips in two directions. The behaviour of reinforced concrete slabs has not changed, but their analytical treatment has changed.

The writer met Dr Randal Wood at a conference (Morris, 1983) and asked him why we were letting torsional moments arise in our analysis and then adding them back into the bending moments using the Wood-Armer equations. His reply was that when he first published his paper he did not think anyone would take it seriously, and yes torsional stiffnesses could be neglected (thereby not giving rise to torsional moments in the tabulated results). He added the qualification that to avoid serviceability problems, design moments should not depart from elastic bending moments by more than 30%.

The above is only a historical note, over the past two decades, most UK bridge engineers have used grillage analysis allowing for the bending and torsional stiffness of the members forming the grillage. Finite elements are now displacing the grillage method in many firms, yield-line analysis is popular in Europe, but Randal Wood's advice is still as valid today as it was in 1983; *to avoid serviceability problems, design moments should not depart from elastic bending moments by more than 30%.*

The long term deflection of a reinforced concrete slab is of the order of twice the short term deflection, thus the modulus of elasticity used as data for this FE checking model, should be that given in BS 8110, divided by typically: 1(short)+2(long)=3. Although BS 8110 suggests a Poisson's ratio of 0.2, for the reasons given above a value of 1E-12 may be considered more appropriate, unless the slab is prestressed in both directions.

For an elastic material, the modulus of rigidity (also known as the shear modulus) is G=E/(2*(1+nu)) where nu=Poisson's ratio. Thus engineers usually assume that the modulus of rigidity to be used in the design of reinforced concrete is G=E/2.4; this value may be compared to G=E/16 specified in BS 5268 to cater for the long term shear deflection of timber beams. Not all structural analysis programs cater for shear deformation, and those which do usually state that shear deformation is ignored if a shear area is not provided by the engineer. A shear area for a plane grid, may be specified directly by specifying AZ or indirectly by specifying the section properties by geometry *e.g.* 1 THRU 2 ISECTION DZ .85 DY .45 TY 0.15 TZ 0.2. From preliminary studies, the deflection computed by NL-STRESS exceeded that computed by Navier. Taking more terms in the double trigonometric series of Navier increased the value of the calculated deflection. Taking more elements in the NL-STRESS FEM analysis, reduces the computed deflection. A full discussion, including tables, may be found in Appendix A, omitted here for reason of space.

vm602 Flat plate in flexure with area loading *cf.* Navier double trigonometric series

Discussion similar to that for vm601, omitted for reason of space.

vm605 Floor panel with hole *cf.* equilibrium, compatibility & energy

For compatibility, energy, local & overall equilibrium, and Clerk Maxwell, Betti, Southwell check, see the notes in vm300.ndf.

vm610 Plate with free edge *cf.* finite differences & exact formulae

Verification is by comparison with the finite difference method for the case of three simply supported edges and one free edge. Verification is by comparison with the simple beam theory including shear deformation for the case of two simply supported edges and two free edges *i.e.* one way spanning. The so called 'Exact' formulae given by Ghali & Neville (1997) which are based on a Poisson's ratio of 0.3 are not suitable for the design of reinforced concrete slabs. Poisson's ratio is the ratio of lateral unit strain to longitudinal strain. When a concrete slab is simply supported on two opposite edges parallel to the Y axis, and free on the other two, a uniformly distributed load will not only cause curvature about the Y axis, but also about the X axis. The curvature about the Y axis is Poisson's ratio times the curvature about the X axis and is of opposite sign. A uniformly distributed load causes the bottom fibres of the slab to be stretched in the X direction, and this stretching causes a shortening of the bottom fibres in the Y direction *i.e.* the Poisson's ratio effect; hence sagging in the X direction is accompanied by hogging in the Y direction on the centre line of the slab. Unless a floor slab is prestressed in both directions, then lateral strain is unlikely to be significant. Reinforcement parallel to the Y axis will tend to prevent - rather than aid - lateral shortening. For this reason West and other researchers and the old C&CA advocated that reinforced concrete slabs be designed as a grillage of strips in two directions. This model may be converted to that for a simply supported slab, by changing the *nss* parameter from 3 simply supported edges to 2 simply supported edges. For 2 simply supported edges and Poisson's ratio of 1E-12 (effectively zero but avoiding it), the deflection along the centre line is exactly that for a simply supported beam when shear deformation is taken into account. Poisson's ratio has a major effect on the moment about the X axis at the centre of the plate. This moment is the lower of the two principal moments. Classical methods of analysis refer to moments being in directions X & Y, modern structural analysis refers to moments being about axes. The free edge of the model runs along the X axis, therefore the moment about X at the free edge must be zero. The moment about X at the centre of the plate varies greatly with Poisson's ratio, the *Exact solution* is only exact when Poisson's ratio =0.3. Both the Exact solution and the Finite difference solution were for a fixed value of Poisson's ratio =0.3, which is inappropriate for reinforced concrete, but is used in the verification for consistency. This discussion is an abbreviated version of that given in vm610, which may be found in Appendix A.

vm618 Plate/wall in extension with hole *cf.* equilibrium, compatibility & energy

For compatibility, energy, local & overall equilibrium, and Clerk Maxwell, Betti, Southwell check, see the notes in vm112.ndf.

This finite element model is for assessing the racking strength of an infill masonry panel with an opening in any position, within a steel frame. Although this model is for linear elastic behaviour, it is straightforward to convert the behaviour to non-linear behaviour. The keyword DIRECTION permits members to be tension or compression members only. DIRECTION should be followed by +1 for compression-only members,

-1 for tension-only members thus: CONSTANTS DIRECTION -1 13 14 15 sets members 13, 14 and 15 as tension only members. The implementation of DIRECTION facility is rigorous; the method of analysis should be given as METHOD SWAY and the NUMBER OF INCREMENTS should be set. In some structures it may be that due to non-linear effects, a member specified as a tension only member goes into compression (and thus carries no axial load) and at a higher loading level, once again becomes a tension only member, NL-STRESS will handle such cases.

For values of DIRECTION >0 and <1, NL-STRESS prevents the nominated member carrying tension if a positive fraction is given (or compression if negative fraction) and multiplies remaining member stiffness by the fraction given, thus: DIRECTION 0.1 ALL would cause all members which go into tension to carry no tension, and have their various stiffnesses reduced to 10% of that given in the member properties table, leaving all members which do not go into tension unchanged. Similarly: DIRECTION -0.2 ALL would cause all members which go into compression to carry no compression, and have their various stiffnesses reduced to 20% of that given in the member properties table, leaving all members which do not go into compression unchanged. As with all such modelling devices, it is up to the engineer to satisfy him/herself that the device is appropriate for the structure being analysed. In both cases the constants E & G are multiplied by the absolute value of the fraction given for the current loading.

vm620 Circular balcony *cf.* classical analysis & Roark

Verification is by comparison with a longhand classical elastic solution based on that given by Jaeger (1964) but with corrected C4, and Roark (1965). Some circular balconies may be modelled as a circular plate with a hole; for such modelling, exact elastic methods exist using classical plate theory. This option considers a uniformly distributed load on the balcony and compares the results with exact elastic methods. Of course few circular balconies will conform to the model of a circular plate with a circular hole; obviously they will have the circular hole but it is unlikely that a circular line of supports will be present at the outside of the plate model. Engineers *engineer* the problem and will look at a plan of the balcony and make an engineering judgment as to the diameter of the ring of supports even though the supports may be just four beams forming a square. For the case of a square (or squarish support) it is likely that engineers will include torsional reinforcement in the slab corners or add support beams to make the support ring octagonal.

vm630 Spherical shell *cf.* Roark's Formulas for Stress & Strain

Spherical shells often have a ring beam at the pole. This option considers a concentrated load *p* distributed on the ring beam and compares the deflection with that given by Roark (1965). By reference to the structure plots, it is straightforward to add further loadings to take into account arbitrary line and point loads for which only a solution by finite elements is possible. For accuracy, elements should be as square as possible, so rings of joints widen as they approach the outside. Experience with the finite element method is that better results are obtained by increasing the number of elements, it would be illogical to conclude that because the results for 8 elements in a ring, agree to within 3% with Roark's formula, that NL-STRESS and Roark agree. NL-STRESS takes shear deformation into account, unless Poisson's ratio is set to 1E-12. Although Roark's formula uses Poisson's ratio, it is clear that it is only used in the calculation of the stiffness of the shell and that Roark's formula does not take shear deformation into account. The foregoing is a brief extract from the notes given in Verified Model vm630, the full version will be found in Appendix A.

vm640 Torque on I-section *cf.* analysis by Roark & Timoshenko

Verification is by comparison with the classical analysis given by Roark (1965) based on Timoshenko (1930). An extensive discussion is contained in the Verified Model vm640 in Appendix A. I sections are weak in twisting; this checking model is for the analysis of an I section in space. The initial setting of parameters is for an I section supported at one end with loading applied at the other end, for which the results of analysis are compared with those by Roark (1965). Of course such a simple analysis will have few practical applications but by reference to a plot showing the joint numbers, it is easy to add general loading, intermediate supports and stiffeners as required. For simplicity, it is suggested that stiffeners be modelled by stiff bars; the free end has a stiffener already modelled, this can be rendered effective or ineffective by setting the parameter *warp* to 1 or 0 respectively. Such changes as those described above, take minutes rather than hours when starting anew.

For values of Poisson's ratio varying between 0.1 and 0.3, the average difference in the rotation from 996 runs was 2.478%, with the largest difference being 8% in run 541, when shear deformation was suppressed the average difference in the rotation for 996 runs was 4.299%, with the largest difference being 11% in run 173. Accordingly the number of elements was increased to six across the flange & nine up the web. With shear deformation suppressed, the average difference in the rotation for 996 runs was 2.423%, with the largest difference being 7% in run 531. On investigation of run 531 it was found that the width & depth of the I-section were approximately the same, yet the number of elements across the flange was 6, whereas the number of elements down the web was 9; which would have contributed to the difference; accordingly a further dependency condition was introduced to compute the number of elements modelling the web depth so that their size was similar to those used in the flange.

vm641 Biaxial bending and/or torque on rectangular hollow section *cf.* Roark

vm642 Bending and/or torque on T-section *cf.* Roark

vm643 Bending and/or torque on channel section *cf.* Roark

vm644 Torque on angle section *cf.* Roark

All four models listed above *i.e.* vm641 thru vm644 use the same self checking method to that of vm640 and thus have similar discussions to those for vm640, so are omitted for reason of space, but all are contained in Appendix A.

vm650 Circular tank *cf.* analysis by Timoshenko & Woinowski-Krieger

For joint coordinate generation for the tank, joints are numbered clockwise on plan starting from 1 o'clock and proceeding from the lowermost ring at the base, up to the top of the tank. Best results for FE methods are when rectangular elements are squarish; the logic in the data file ensures this, unless the tank height is made very small. As the fluid pressure increases with depth, the pressure on an element is lower at the top of the element than at the bottom. The pressure may be distributed to horizontal members only or vertical members only or to both horizontal & vertical members. In this model, the weight per unit volume is divided by 2 and the resulting pressure applied to both horizontal and vertical members.

At the top of the wall the hydraulic pressure in zero and the displacement is considerably lower than in the element below. Timoshenko & Woinowsky-Krieger (1959) state "In most practical cases the wall thickness *h* is small in comparison with both the radius *a* and the depth *d* of the tank, and we may consider the shell as infinitely long." Deflections computed by NL-STRESS take into account the fact that the shell is not infinitely long *i.e.* the boundary conditions are different for an NL-STRESS analysis to that assumed by Timoshenko (1959), in consequence there cannot be agreement. For this reason the displacement at the top of the tank is ignored in the table comparing the two methods of analysis. From preliminary studies, it was found that the height:thickness ratio had a major effect on the percentage difference for as stated above, Timoshenko's solution assumes the shell to be infinitely long. Accordingly, the dependency conditions in the parameter table, was amended to restrict the wall thickness to be in the range height/60 to height/12.

vm710 Natural frequency of beam or frame *cf.* flexibility & latent root

Verification is by comparison with the classical method viz: form flexibility matrix, then find largest latent root (lambda) using power iteration method, chapters 9 & 11 McMinn (1962). Warburton (1964) tells us Rayleigh's method can be used with the vibration form of a uniform beam clamped at both ends. The deflected form is dependent on the ratio of shear strain energy to bending strain energy which in turn is dependent on the span:depth ratio. NL-STRESS uses Rayleigh's method for frames in which the mass can be assumed to be lumped at floor levels, and where compression of columns and rotation of column heads may be considered negligible in comparison to sway effects. A description of the method will be found in vm710 in Appendix A.

vm718 Natural frequency of built-in plate *cf.* Roark & Warburton

Verification is by comparison of the natural frequency computed by NL-STRESS and

Rayleigh's method, with formulae given by Roark (1965) & Warburton (1964). The dynamic modulus is higher than Young's modulus. BS 8110 gives the dynamic modulus as 31+0.16*fcu, thus for a cube strength of 40, the dynamic modulus is 31+0.16*40=37.4E6 kN/m² *cf.* the static modulus 20+0.20*40=28.0E6 kN/m²; the value 30E6 used in this model is therefore conservative.

Warburton (1964) derives the strain energy U for a plate subjected to lateral (or normal) loads, and equates this to the kinetic energy T. Warburton states "Although there is no closed form solution to this problem, the value of 36 for the non-dimensional frequency factor for the fundamental mode agrees to within 0.06% with values obtained by the Rayleigh-Ritz and other methods...". Accordingly the frequency factor of 36 is used. Agreement within 2% was obtained between Warburton's formula based on classical theory, and NL-STRESS & Rayleigh's method, when span:depth ratios are in excess of 15. For span:depth ratios lower than 15, percentages differences rise, *e.g.* to 7% when span:depth=5 due to the fact that NL-STRESS takes shear deformation into account; confirmed by setting Poisson's ratio =1E-12 which tells NL-STRESS to ignore shear deformation. Warburton's formula is derived on the basis that the plate is thin, span:depth=5 is certainly not thin, furthermore the natural frequency of such thick plates will be appreciably above 10 Hertz and therefore not be of interest to structural engineers.

The majority of structural engineers only become involved with vibrations when the natural frequency of a suspended floor slab or beam is less than 10 Hertz. Damping, especially with reinforced concrete floors, ensures that vibration is not normally a problem for structures having a natural frequency greater than 10 Hertz. Accordingly sets of test data can be engineered to avoid analysing members for which the natural frequency exceeds say 36 Hertz (the non-dimensional frequency factor). Rearranging Warburton's formula (1964), for the natural frequency of a square slab with built-in edges:

```
d=e*t^3/(12*(1-nu^2)) om=36/SQR((w*lx^4)/(d*g)) nfr=om/(2*PI)
```

we can evaluate the natural frequency from:

```
nfr=36/(2*PI*SQR((w*lx^4*12*(1-nu^2))/(e*t^3*g)))
```

Substituting 36 for nfr and rearranging again, we can engineer the thickness for limiting the natural frequency to 36 Hertz thus：

```
t'=(48*PI^2*w*lx^4*(1-nu^2)/(e*g))^(1/3)
```

Although this limit for *t'* is based on a square plate of side *lx* with all sides built-in, as the limit is only for controlling the range of values used for thickness and does not affect accuracy, it may be applied to a simply supported rectangular plate taking *lx* as the average length of the sides. Roark's formula is included for reference, but omitted from the calculation of percentage difference, as Roark's formula omits Poisson's ratio. A full explanation of the above will be found in vm718 in Appendix A.

vm720 Natural freq. of simply supported plate *cf.* Navier, flexibility & latent root

Verification is by comparison of the natural frequency computed by NL-STRESS and

Rayleigh's method, with that computed thus:

- form flexibility matrix for inside joints using Navier's solution in the form of a double trigonometric series giving the deflection at any position on a simply supported rectangular plate for a load applied at this position or any other position
- find the largest latent root lambda, using the power iteration method, McMinn (1962)
- find the period from T=2π√(lambda/g) and hence natural frequency from 1/T.

See discussion for vm718 above and vm718 in Appendix A.

vm802 Cantilever beam with large displacements *cf.* equilib., compatibility & energy

Non-linear analysis may be used for a structure for which the loading is well within linear elastic limits, and used for the same structure to determine critical loads. In the elastic analysis of a cantilever, the longitudinal displacement of an end loaded cantilever is zero because elastic analysis assumes all displacements are very small, and a very small lateral displacement at the end of a cantilever causes no longitudinal displacement. When the end lateral displacement is significant, it will be necessary to carry out a non-linear analysis as used in this verified model. Professor Horne gives the horizontal movement of the tip of a horizontal cantilever with a vertical end point load V as:

```
V^2.L^5 + 2.V^2.L^4 + V^2.L^3
delta = ------- ¦ 1 - --------- ¦ - ---------
15(EI)^2+ 63 (EI)^2 + 3.E^2.I.A
```

where: V=vertical end load & L=length of cantilever, A=cross-sectional area, I=moment of inertia & E=Young's modulus.

For an axial load approaching the Euler buckling load, very non-linear behaviour happens, with small increases of axial compressive load causing a large lateral displacement at the tip of the cantilever. The strain energy summations are based on classical integration of the axial, shear and bending stress components which do not take into account the reduction in bending stiffness due to axial load. The summation of the external work done is inviolable, thus the NL-STRESS strain energy summation will be greater than the check. The difference between the strain energy and external work done is negligible for structures having normal member displacements *e.g.* span:deflection ratios of 100 or more and axial loads well below the buckling load, but becomes significant when axial loads approach the buckling load and with span:deflection ratios of 10 and less.

An extensive discussion on non-linear aspects of the cantilever beam will be found in Appendix A, omitted here for reason of space.

vm810 Stability of columns with various supports *cf.* classical formulae by Euler

Verification is by comparison of the buckling load for various support conditions with Roark (1965). This verified model is a parametrically written data file for the elastic stability analysis of columns (or bars or plate stiffeners) subjected to axial load for various support conditions. The engineer may use formulae for standard loadings and support conditions; the model is easily modified to cater for non-standard loading and conditions, such as those occurring in real structures. Structural analysis is based on assumptions *e.g.* plane sections remain plane; putting in impossible data for the parameters can produce incorrect results *e.g.* a section having a cross sectional area of 1E6 in SI units means the section size is 1 kilometre by 1 kilometre, this may be analysed if the member length is a few kilometres long and the moment of inertia is bd^3/12 =1000^4/12, but significant errors can occur if impossible figures are used especially when carrying out non-linear analysis. Even with linear elastic analysis, incorrect results can be produced if Young's modulus is given as 28E-6 instead of 28E6, such an error causes shear strain energy to swamp the bending strain energy and in consequence, a plot of the bending moments looks as though all members have pinned ends.

For columns which are free to rotate at either or both ends, the segments of each member are arranged in a bow such that maximum displacement from the chord is *bow%* times the length of each member. Whereas MEMBER DISTORTIONS are used for studying lack of fit problems (the member is distorted in the directions specified, then *clamped* into the structure and let go); the bow specified in the NUMBER OF SEGMENTS command only tells NL-STRESS that each member has a parabolic bow which does not give rise to stresses due to *lack of fit*. If the keyword TRACE is added to the NUMBER OF SEGMENTS command, the set of results includes the additional nodes in any table giving displacements; also the forces at the end of each segment in any table of member forces or stresses. A bow of 1% is typical of average quality control, 8 segments give good results. A lateral force of one millionth of the axial load is applied to initiate buckling for columns for which the top end is free to sway. For columns which are not free to sway, this lateral force will have no effect. Modern *light design* in which stability determines strength rather than yield, is finding increased use in box-girders and other structural components, especially those fabricated from corrosion resistant high strength alloys. Text books say that one advantage of light design is that since buckling can occur without damage, the resistance afforded by the buckled component is known; in practice support conditions and component straightness has a major effect on buckling capacity. One disadvantage of light design is that it is susceptible to vibrations caused by mechanical and electrical equipment, thus dynamical behaviour may be the controlling limit state rather than elastic stability.

Initial results from *ad-hoc* changing of the parameters was that differences in the computed results *cf.* classical formulae were up to 25%. After some frustration, the lesson learnt was *never vary more than one parameter at a time*. It was only after changing the section type, that it became clear where the difference lay, for the Rectangle & H-section have in common that their form factor F is 1.2, whereas that for thin walled hollow sections F=2. Chapter 1 of The Stability of Frames (Horne & Merchant, 1965) states: *In this treatment, it has been assumed that deflections due to shear deformation and direct axial compression may be ignored. If these are allowed* *for, the deflections are slightly increased, but are still linearly related to the applied load.*

Taking shear deformation into account reduces the computed buckling load, especially for I-sections and structural hollow sections. For an I-section of slenderness (length:depth) 8/0.6 ≈13, the reduction in buckling load is ≈25%.

In section 1.8, Horne & Merchant (1965) state: "*When a pin ended strut buckles laterally, the applied loads Pe will have components transverse to the bent longitudinal axis, thus introducing into the member, shear forces F. These forces F will in turn produce additional deformation due to shear and when these deformations are allowed for in the analysis, the buckling load is reduced below the Euler load Pe. It is found that, for practical purposes, the effect is unimportant, amounting to a reduction of a fraction of 1 percent.*" The *fraction of 1 percent* is referenced to Bleich (1952). There is little point in considering the buckling load of columns if the column material crushes before the column buckles. All studies in stability should limit the range so that the crushing load of the column is not exceeded; a full discussion on this matter is contained in vm810 in Appendix A.

From an initial run of 996 sets of data generated from the parameter table, the average difference between Euler's formulae and NL-STRESS, was 1.979% with run 582 having a difference of 14%. Inspection of the data shows that the section was a RHS having a wall length:thickness ratio of 0.19253/0.03=6.418; when the wall thickness was reduced from 0.03 to 0.02 and the data again run, the difference was reduced from 14% to 1%. Non-linear analysis has complicated procedures to satisfy both equilibrium and compatibility. When members are at the point of buckling, very non-linear behaviour is present. In the light of the above, the length to thickness ratio was specified to be a minimum of 10:1 as for normally rolled steel sections.

From a second run of 996 sets of data generated from the parameter table, the average difference between Euler's formulae and NL-STRESS, was 2.052% with run 54 having a difference of 13%. Inspection of the data shows that the section was an H-section having both flange & web length:thickness ratio of 10:1 as limited above; running the set of data but changing the section from H-section to I-section reduced the difference from 13% to 1%. It is evident that better results are obtained with an I-section than with an H-section. Of course the difference between the two sections is that for an I-section typically 25% of the cross-sectional area is carrying the shear, whereas for an H-section typically 75% of the cross-sectional area is carrying the shear. This difference means the ratio of bending:shear strain energy will be very different. Accordingly run 54 was repeated but suppressing shear strain deformation; for which the difference was reduced from 13% to 0%.

vm830 Stability of circular ring/pipe *cf.* classical formulae by Roark

Verification is by comparison of the buckling load for various pipe and ring sizes with

Roark (1965). In order to carry out a non-linear analysis to find the buckling load on a ring subjected to a uniform radial pressure, supports have to be provided in the X & Y direction and about the Z direction. If all three supports are rigid, then buckling of a ring will be resisted by the supports themselves. A solution is to restrain the joint at 6 o'clock in the X & Y direction and prop the joint at 12 o'clock with a spring in the X direction. The value of the spring is taken as the trial elastic critical pressure *p* (force/length) as it has the same dimensions. Just as in many classical methods where a particular solution is substituted into a general differential equation to obtain a complete solution, it is convenient to use Roark's formula for the buckling capacity of a ring to define a loading regime for the non-linear analysis of a ring to find its buckling capacity. For simplicity, Roark's buckling pressure is doubled and applied in 200 increments, thus expecting collapse when approximately 100 increments have been applied. The radial force *rf* at each joint is computed from: rf=p*2*PI*r/nsg, where *p* is the radial pressure.

From single runs to *get a feel for the problem* , of particular interest is that for 256 segments, NL-STRESS results exactly match Roark, for the default value of nsg=32 the percentage difference is 5%. Unfortunately, things are not cut and dried, for there is a parameter which is not declared in the Parameter table in the Verification data and that is the horizontal spring stiffness at the top of the ring. This stiffness was arbitrarily chosen as *p*, *i.e.* Roark's radial distributed load to cause buckling, units *force per unit length*. A spring stiffness was provided for if a fixity was given, then the radial load will carry on increasing past the buckling load as the three reactions and the uniform radial loading to an accuracy of 15+ decimal digits, ensures that the circular form is preserved. It was a happy coincidence that a spring stiffness of *p* ensured equality between NL-STRESS and Roark's formula over a wide range of data. For spring values less than p, NL-STRESS gave buckling loads less than Roark, for spring values greater than p, NL-STRESS gave buckling loads in excess of Roark. The horizontal spring at the top of the ring allowed a free body rotation of the system leading to collapse. Without the spring at the top of the ring the whole structural system was in equilibrium for every loading increment and Euler's small displacement assumption to initiate buckling was absent and in consequence buckling did not occur until the 15+ arithmetic decimal accuracy eventually became unstable.

vm850 Stability of cantilever with udl & end load *cf.* equilib., compatibility & energy
Compatibility, energy, local & overall equilibrium checks are discussed in vm112.ndf and chapter 8.

vm852 Non-linear elastic analysis of multi storey frame *cf.* equilib., compat. & energy
Compatibility, energy, local & overall equilibrium checks are discussed in vm112.ndf and chapter 8.

vm950 Hanging cable with flexible platform *cf.* Pippard & Baker

The so called elastic theory of suspension bridges, used here, is given by Pippard & Baker (1957), who give several examples of suspension bridges in their chapter 13. Suspension bridges should always be checked for dynamic effects, see vm301 for the inclusion of a check in a linear elastic analysis of a circular-arc cantilever with out-of-plane loading. Cables have little stiffness, a 0.5 m dia cable spanning 100m has a span:diameter ratio of 200:1. For the hanging cable supporting a loaded flexible platform, moment releases at the ends of the hangers and segments of the platform ensures that the only stiffness available is that of the cable. Just as structural members in compression have a reduction in their bending stiffness, so structural members in tension, such as cables, have an increase in their bending stiffness. It has been found that cables having a span:diameter ratio of 200:1 have sufficient stiffness to enable a stiffness analysis to be carried out without having to resort to techniques such as *guy diameter modelling factors*. Accordingly this ratio is used in the dependency conditions in the parameter table.

vm951 Suspension bridge with three pinned stiffening girder *cf.* Pippard & Baker

See the discussion for vm950 above. In the model, moment releases are given at the start and end of each hanger, thus the deck has no restraint in the X direction. To prevent failure as a mechanism, a horizontal restraint is provided at joint 1 (the left end of the deck) by the statement 1 FORCE X - 1, where the -1 indicates fixity. If required, a horizontal spring may be provided, *e.g.* 1 FORCE X 100, where the 100 represents force/unit length which for SI units means 100 kN/m or 1 kN/cm. From single runs to *get a feel for the problem*, although the percentage differences between NL-STRESS & classical analysis are small, the large concentrated force in the middle of the cable due to the central pin in the stiffening girder, does produce a significant bending moment; the smaller the diameter of the cable, the smaller the bending moment. The lower the stress, the better the fatigue resistance; to achieve this:

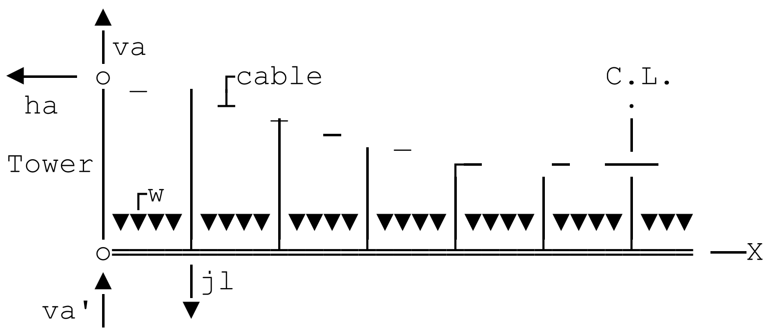
- increase the cable dip:span ratio
- reduce the cable diameter.

Ideally, eliminate the central pin in the girder. The stiffness of the cable may be reduced by replacing the cable with links such as those used in Hammersmith Bridge. Each side of Hammersmith Bridge has two chains, one above the other but not connected, alternate chains carry alternate hangers. The span of Hammersmith Bridge is approximately 128 m, the dip of the chains is 8 m *i.e.* span:depth ratio = 16:1; its stiffening girders are two pinned.

vm952 Suspension bridge with two pinned stiffening girder *cf.* Pippard & Baker

The so called elastic theory of suspension bridges, used here, is given by Pippard & Baker (1957) who give several examples of suspension bridges in their chapter 13. Suspension bridges should always be checked for dynamic effects, see vm301 for an example for the inclusion of a check in a linear elastic analysis. Pippard & Baker qualify their analysis thus: *The analysis of this type of bridge is complicated by the fact that the structure is redundant and also because the flexible cable does not follow a* *linear load-deflection relation. The principle of superposition cannot be applied and the effects of live load are not calculable separately from those due to the dead load as would be the case if the structure obeyed a linear law. The methods of strain energy are not, therefore, really applicable to the problem but are used to obtain an approximate solution. The initial shape of the cable is assumed to be a parabola and under all subsequent loads is assumed to retain this shape. This theory of the suspension bridge is sometimes known as the elastic theory to distinguish it from the more general and accurate treatment which takes into account the deflection of the girder and cable under live loading.*

In the model, moment releases are given at the start and end of each hanger, thus the deck has no restraint in the X direction. To prevent failure as a mechanism, a horizontal restraint is provided at joint 1 (the left end of the deck) by the statement 1 FORCE X -1, where the -1 indicates fixity. It would be unwise not to provide a horizontal restraint at deck level. If required, a horizontal spring may be provided, *e.g.* 1 FORCE X 100, where the 100 represents force/unit length which for SI units means 100 kN/m or 1 kN/cm. Just as structural members in compression have a reduction in their bending stiffness, so structural members in tension, such as cables, have an increase in their bending stiffness. Stability considerations are not taken into account in Pippard & Baker's treatment of suspension bridges, thus to compare like-with-like the moment of inertia of the cable will be assumed =1E-6 *i.e.* effectively zero. Pippard & Baker assume that all loads on the deck are carried entirely by the cable. In practice the ends of the deck will be supported vertically, thus reducing the total load on the cable, so the cable tension can not be found from √(va^2+ha^2) where va is computed by statics for all loading & ha from Pippard & Baker. Joint load jl and the distributed load w are not completely carried by the cable. The reaction va' at the base of the left tower can not be found from statics, so the cable tension is found from ha*SQR(1+16*d^2/l^2) for the hanging cable, where *d* is the dip and *l* is the span.



11.2 Models for structural design

The automatic generation of sets of data from a parameter specification held in a table entitled *PARAMETER table*, was successful in picking up bugs in the logic of models for the design of concrete and steelwork components.

BS 8110-1:1997 Clause 2.5.2 states when linear elastic analysis is used, the relative stiffnesses of a member may be based on:

- the concrete section
- the gross section on the basis of modular ratio
- the transformed section on the basis of modular ratio.

A modular ratio of 15 may be assumed, a consistent approach should be used for all elements of a structure. Thus BS 8110 permits linear elastic methods to be used for both the structural design of concrete frameworks and the design of reinforced concrete sections. Accordingly a *check* to calculate the concrete and steel stresses at *working loads* was incorporated into proforma calculation sc075.pro, which is a model for the reinforced concrete design of T sections.

A similar position exists with structural steelwork *e.g.* proforma calculation sc385.pro which is a model for the design of a stainless steel rectangular hollow section, was extended to incorporate a linear elastic finite element analysis for the section using data for loading and support conditions, partial restraints *etc.* to calculate steel stresses at *working loads*. This additional information for the engineer was made optional.

Summarising, the automatic generation of sets of data from a parameter specification held in a table entitled *PARAMETER table*, was successful in picking up bugs in the logic of models for the design of concrete and steelwork components. Models for the structural design of steelwork and reinforced concrete components, which have been devised using British Standard limit state codes, can be self checked using classical elastic design principles.

11.3 Limit state design

The writer first learnt about limit state design, in particular the ultimate limit state for reinforced concrete from Professor A.L.L. Baker at Imperial in 1966, at the same time Professor J.F. Baker at Cambridge was propounding plastic design principles for structural steelwork, as also was Professor M.R. Horne at Manchester. Early methods for the ultimate load design of concrete beams and slabs and portal frames were simple to apply, they had to be, as 99% of calculations were prepared using slide rules. The first *limit state* code was CP 110:Part 1:1972 entitled *The structural use of concrete*. Nowhere did it state that the main design limit state to be considered should be the ultimate limit state, indeed equal weighting was given to both ultimate & serviceability limit states. That position has changed in 2006 for both BS 8110-1:1997 incorporating Amendments 1,2 & 3 published November 2005, and BS 5950-1:2000 incorporating Corrigendum No. 1 published May 2001, both give pre-eminence to the ultimate limit state. The outcome of this shift in emphasis has caused:

- complicated design procedures especially for structural steelwork
- the anomaly of analysing structures elastically and sizing sections for the ultimate limit state inadvertently creating mechanisms
- loss of the intuitive elastic section design principles formerly taught.

Complication in design procedures has been due largely to the incorporation of serviceability limit state problems *e.g.* stability, into ultimate limit state procedures. Perusal of the parameter specification given in table 10.5 for the design of stainless steel rectangular hollow sections, shows that there are 55 parameters, 32 of which are dependent on other parameters. The reason for this complexity is that it is not possible to design a cross-section for a rectangular hollow section using just the bending moment/s, shear force/s, axial load and torque, in isolation. The length of the member, the nature of vertical, lateral and rotational supports, together with applied loading and its eccentricity, all need to be taken into account. This necessitates the engineer having to make assumptions, providing data read from tables such as table 10.1 which gives effective length factors about the principal axes of the section. Undoubtedly a set of such assumptions will result in calculations which vary from precise to imprecise. As the time taken to run a finite element model of a rectangular hollow section member is less than a minute, plus 5 minutes to consider and select the loading and supports, it is most likely that results providing elastic stresses at working loads would be more acceptable to the engineer for a self check, particularly if the model used had been verified, than would the results obtained by substituting numbers into a family of equations given in a code of practice.

11.4 Eurocodes

When the subject is *verifying the correctness of structural engineering calculations,* it would be foolhardy to ignore what is happening with Eurocodes. Unfortunately, at the time of writing, the position is that instead of having one Eurocode for steel and one for concrete *etc.*, Eurocodes are being split into several parts and issued/sold separately. For example, Eurocode 3 is available for Parts: 1-1, 1-2, 1-8, 1-9 & 1-10, some of the parts have a *National annexe*, but the remainder does not. At the commencement of this research, it had been hoped to incorporate a full Eurocode check into models for the design of structural components such as beams and slabs, but this will not be sensible until all parts of the structural Eurocodes are published and have *National annexes*. In consequence, consideration was given to devising self checks based on classical elastic section behaviour, as used before *limit state* design was introduced. It was found possible to get reasonable agreement between the classical and modern methods for some sets of data; getting agreement to within two or three percent for a wide range of sets of data, as was achieved for the set of models for the structural analysis of frameworks, was not possible. The writer considers that self checks by Eurocodes for models for the structural design of components, could send out confusing messages to engineers, but nevertheless would provide useful information.

BS 8110-1:1997 issued 30 November 2005, clause 3.4.5.8 gives formulae for the enhancement of shear strength of sections close to supports. The rules given permit concrete to carry a shear stress of 5 N/mm² at sections which are predominantly *cracked*. This is in marked contrast to Eurocode 2 which ignores the shear strength of concrete, only utilising the concrete as a compressive strut in association with longitudinal tension reinforcement and vertical shear reinforcement. New Civil Engineer 22 June 2006 announces that *the British Standards Institution is to withdraw BS 8110 from March 2008, two years before the deadline for implementing the Eurocodes.*

It is right and proper that matters which affect the correctness of structural engineering calculations should be considered even if such matters have a political dimension. The British Standards Institution has announced that it will withdraw structural British Standards starting in March 2008. IStructE members have expressed concern about the cost of the introduction of Eurocodes, not just the cost of purchase but the far higher hidden cost of staff time in absorbing the new material. All *learned societies* should be responsible for their own design guides. The BMA (British Medical Association) would never accept Euro procedures for a craniotomy being made mandatory. Eurocodes should be concerned only with the standardisation of materials and performance. Including equations in Eurocodes, for the design of buildings in Ecclefechan or Ochtermochty, is not appropriate.

The Concrete Centre publication entitled *How to design concrete structures using Eurocode 2, 2. Getting started* states: "The recommendations for durability in Eurocode 2 are based on BS EN 206-1. In the UK the requirements of BS EN 206-1 are applied through the complementary standard BS 8500. The UK National Annex (table 4.3 (N) (BS)) gives durability requirements that comply with BS 8500, but which significantly modify the approach taken in Eurocode 2. To determine the minimum cover for durability (and also the strength class and minimum water cement ratio) either the UK National Annex or BS 8500 can be used." It is abundantly clear from the above that:

- the aim to have a common set of structural codes for all the nation states is good but unachievable
- little/no *calibration* has been done to reconcile the new Eurocodes with proven codes.

Currently, the writer's set of Eurocode 0 to 8, contains approximately 4000 A4 sheets with close type. With frequent changes to codes of practice, it is difficult even for experts to keep up to date, and more difficult for sole practitioner engineers who design and detail using all the main structural materials. Alistair Beal, a frequent contributor to both The Structural Engineer and the New Civil Engineer, in a letter published in NCE 16.02.06 concludes: "Unless the government is prepared to provide sufficient resources for a safe, efficient changeover to Eurocodes, the only responsible course of action is to withdraw from the project. Has ICE got the guts to tell them this?" It is common sense that the adoption of a code of practice which contains:

- an increase in complexity
- the introduction of new terminology

will increase the number of structural failures rather than reduce it. In the seventies, the writer served on Comité Européen Béton, commission 14. One of the recommendations of that commission was that Greek symbols should be avoided in codes of practice as European engineers who use the Roman alphabet, do not know how to pronounce or read, Greek symbols. Just as it is obvious that changing the side of the road on which a country drives will cause many accidents, so the introduction of an unfamiliar nomenclature *e.g.*

- factor for frequent value of a variable action
- factor for quasi-permanent value of a variable action *etc.*

which are represented by Greek symbols in the Eurocode, will cause misunderstandings. It is forty years since the writer became a chartered structural engineer, the cause of structural failures today is the same as it was then, *i.e.* cock-ups, singly or in combination. Cock-ups have nothing to do with partial safety factors, cock-ups are to do with:

- complexity and confusion
- misunderstanding rules
- arithmetic error
- not appreciating the seriousness of a situation
- thinking that someone else is responsible but not informing them
- not checking dimensions cf. measure twice, cut once
- bamboozlement & mischief
- schism between design and construction
- poor quality control
- not monitoring in-service performance.

When national annexes are available and have been calibrated against British Standards, Eurocodes will provide sensible self checks for inclusion in the design of structural components. Section 11.5 discusses preliminary calibration tests for Eurocode 2 *versus* BS 8110 for flexural reinforcement, column reinforcement and shear reinforcement.

11.5 Further models

As mentioned in section 10.5, there is a considerable difference between the lengths of models for the structural analysis of frameworks and models for the structural design of components *e.g.* for the former the length of time to develop each model is from 1 to 6 days, for the latter the length of time to develop each model is from 1 to 6 weeks. An A4 printout of the full set of proforma calculations which occupy 723,000 lines for the structural design of components at 60 lines per page occupies 12,050 pages *i.e.* a pile of paper 4 ft high when printed on 80gsm paper. Sections 10.6 and 10.7, respectively typical structural steelwork and reinforced concrete components, give formal descriptions of each model in terms of its PARAMETER table. A modest amount of study of each parameter table will reveal a great deal about each model. The self check for the structural steelwork component, developed as part of this research, was provided by invoking NL-STRESS to carry out a linear elastic finite element analysis of the section subjected to characteristic dead plus imposed loading and finding the maximum combined stresses and deflections and their locations and presenting the information for the engineer to consider with respect to the stress combination factors. The self check for the structural concrete component, developed as part of this research, was provided by including serviceability checks for long term deflection, flexural cracking and cracking due to overstress of the shear reinforcement based on the truss model and shear cracks at 45°, and presenting the information for the engineer to consider with respect to the reinforcement provided in accordance with BS 8110.

Papers are circulating which compare Eurocode 2 with BS 8110 *e.g* **EC2 and BS8110 compared** by Dr Moss of BRE & Rod Webster of *Concrete Innovation & Design*, with funding provided by ODPM and BCA. The writer has not found any papers comparing these two codes numerically. Accordingly three further models were extended as part of this research to include PARAMETER tables enabling the automatic generation of sets of data, and comparisons of reinforcement produced by BS 8110 with Eurocode 2. Each of tables 11.1 to 11.3 are in accordance with BS 8110-1:1997, last amendment 16016, 30 November 2005, and Eurocode 2: BS EN 1992-1-1:2004. To print out 1000 runs for the table requires 16 pages, for reason of space, just the first two pages of tables 11.1 to 11.3 are shown.

Table 11.1 shows the start of a table for numerically comparing the rules for the flexural reinforcement of beams which are contained in BS8110 clauses 3.4.4.3 to 3.4.4.5, with those of Eurocode 2 clauses 3.1.6 to 3.1.9 using the design procedure given by The Concrete Centre, based on the simplified rectangular stress block. Occasionally, because of the differences between the stress blocks, BS 8110 requires compression steel but Eurocode 2 does not.

Table 11.2 shows the start of a table for numerically comparing the rules for column reinforcement contained within BS 8110 clauses 3.8.1 to 3.8.6 with Eurocode 2 rules contained within clauses 5.8.1 to 5.8.9 with those of Eurocode 2 contained within clauses 6.2.1 to 6.2.5. See also the IStructE *Manual for the design of concrete building structures to Eurocode 2* clauses 5.5.1 to 5.5.5. The percentage differences in table 11.2 are due to the different rules between the codes *e.g.* the minimum reinforcement percentage for columns designed in accordance with BS 8110 is 0.4% *cf.* 0.2% to Eurocode 2 *cf.* 1.0% to CP110. *You only need one bucket of bad concrete in a column to be in real trouble*, see *The collapse of the Hotel New World, Singapore: a technical inquiry*, The Structural Engineer, 16 March 1993.

Table 11.3 shows the start of a table for numerically comparing the rules for shear reinforcement contained within BS 8110 clauses 3.4.5.1 to 3.4.5.13 with Eurocode 2 rules contained within clauses 6.2.1 to 6.2.5. Allen (1988) tells us that earlier British Codes such as CP114 were in need of revision for two main reasons:

- the permissible shear stresses for concrete were too high
- the truss model for designing shear reinforcement had consistently shown that actual shear strengths were much higher than those calculated by this approach.

Most engineers would have taken the view that the permissible shear stresses for the concrete should have been increased and the truss model retained. In actual fact, the truss model as enunciated by Pippard & Baker (1957) and incorporated in CP114, had been abandoned in CP 110:1972 *i.e.* sixteen years prior to the publication of Allen's book. Eurocode 2 uses the truss model, confusingly referred to as the *strut inclination method* by *The Concrete Centre*. Design for vertical shear reinforcement is generally straightforward and avoids the need to look up tables. One uncertainty is the value to be taken for *cotΘ* given with reference (6.7N) *where N does not mean Newtons,* and for which Eurocode 2 refers the engineer to the *National Annex*, which repeats the advice given in Eurocode 2 *i.e.* $1 \leq \cot\Theta \leq 2.5$, and adds *except in elements in which shear co-exists with externally applied tension (i.e. tension caused by restraint is not considered here). In these elements, cotΘ should be taken as 1.0.* Generally the shear failure plane for a reinforced concrete beam subjected to uniformly distributed loading is at 45° *i.e.* $\cot\Theta=1$. Assuming $\cot\Theta=2.5$ with vertical stirrups will cause shear cracking. Although homogeneous materials can give values of $\cot\Theta > 1$, reinforced concrete is not homogeneous. Kong & Evans in *Reinforced and Prestressed Concrete, Table 2.5-4* give drying shrinkage increasing considerably with the amount of cement in the mix. For a nominal 1:1:2 mix with w/c=0.4, they give the long term shrinkage as 600 microstrain, qualified by their *Table 2.5-5* which shows that after 6 months, 60% of the long term shrinkage will have taken place. Thus at the time a beam is loaded 360 microstrain may be expected, or considerably higher if the concrete is not properly cured. The presence of reinforcement in concrete does not prevent drying shrinkage, it merely controls the size of cracks and distance between. The writer contends that the assumption of $\cot\Theta=2.5$ with vertical stirrups will cause severe cracking.

In the comparisons between shear resistances computed by BS 8110 & Eurocode 2, the shear resistances tabulated are for the area of links at the centres given. The percentages tabulated are the differences between the BS 8110 & EC2 shear resistances provided for the areas of links at the link centres. For the reasons stated above, tables for $\cot\Theta>1$ are not given. The structural cost of a reinforced concrete building is typically 20% of the total cost. The reinforcement cost is typically 25% of the structural cost, *i.e.* 5% of the total cost, the shear reinforcement cost is typically 20% of the of the total reinforcement cost, *i.e.* 1.0% of the total cost. Today the emphasis is on simplicity in the design, with generous imposed loads sufficient to accommodate change of use of the building over its working life; in such an environment, skimping on the amount of shear reinforcement is false economy.

Table 11.1 Flexural reinforcement: BS 8110 *cf*. Eurocode 2

Run No.	Beam dpth & breadth mm	mm	Design moment kNm	BS 8110 comprn. mm²	EC2 comprn. mm²	BS 8110 tension area mm²	EC2 mm² tension area	Perc. diff. %
1	200	362	50	0	0	612.72	670.69	9
2	202	1090	100	0	0	2188.6	2390.4	9
3	204	967	150	0	0	1846.7	2020.9	9
4	229	1072	200	0	0	4063.6	4446.7	9
5	218	1063	250	0	0	3133.2	3425.6	9
6	264	1054	300	0	0	5635.4	6162.3	9
7	236	1045	350	0	0	3880.6	4245	9
8	294	1036	400	0	0	6332.5	6929.4	9
9	248	1027	450	0	0	5159.3	5637.5	9
10	320	1018	500	0	0	7606.7	8319.9	9
11	262	1009	550	1244.9	978.71	6091.7	6866.5	7
12	343	1000	600	0	0	9220.4	10074	9
13	280	990	650	0	0	6557.4	7165.9	9
14	364	981	700	0	0	9222.7	10089	9
15	296	972	750	782.76	346.33	7376.4	8356.2	7
16	383	963	800	0	0	10545	11529	9
17	310	954	850	2403	2215.1	7825.1	8801.8	8
18	401	951	900	857.8	10.627	12026	13685	6
19	325	936	950	702	0	8567.3	9480.6	2
20	417	984	1000	0	0	11668	12762	9
21	337	918	1050	1857.5	1497	8947.4	10110	7
22	431	1019	1100	0	0	12972	14180	9
23	349	900	1150	3510.3	3410.5	9269	10406	8
24	446	1048	1200	1039.6	0	14446	16245	5
25	360	881	1250	1317.9	788.93	10040	11385	7
26	459	1078	1300	0	0	13727	15015	9
27	371	889	1350	2778.2	2478.1	10337	11660	8
28	472	1100	1400	0	0	15008	16407	9
29	381	911	1450	4257.3	4189.1	10621	11919	8
30	484	1100	1500	1174.8	0	16608	18809	6
31	391	931	1550	1440.8	835.44	11452	12997	7
32	496	1100	1600	0	0	15730	17204	9
33	399	953	1650	3017.2	2659.8	11731	13246	8
34	507	1100	1700	0	0	17202	18802	9
35	409	967	1750	4657.1	4565.9	11908	13375	8
36	518	1100	1800	1676.9	628.41	18496	21060	8
37	418	989	1850	1427.1	726.85	12796	14538	7
38	528	1100	1900	0	0	17712	19370	9
39	430	1004	1950	2984.1	2543.1	12878	14562	8
40	538	1100	2000	0	0	19322	21115	9
41	442	1013	2050	4698.3	4546	12907	14518	8
42	548	1100	2100	2479.6	1444.5	20273	23060	8
43	454	1007	2150	1159.5	354.21	13732	15628	7
44	557	1100	2200	0	0	19607	21440	9
45	466	1001	2250	3013.5	2524.2	13704	15511	8
46	567	1100	2300	0	0	21322	23297	9
47	478	993	2350	4936.4	4776.2	13639	15348	8
48	575	1100	2400	3289.1	2274.1	21959	24954	8
49	490	985	2450	1215.2	372.16	14489	16496	7
50	583	1100	2500	0	0	21467	23471	9
51	503	973	2550	3145.8	2637.3	14362	16263	8
52	592	1100	2600	0	0	23381	25541	9
53	516	965	2650	5083.5	4911.6	14213	16002	8
54	600	1100	2700	4077.5	3084.8	23559	26751	8
55	529	956	2750	1162.3	275.97	15070	17171	7
56	608	1100	2800	0	0	23220	25385	9
57	542	947	2850	3129.8	2586.5	14893	16876	8
58	616	1100	2900	0	0	25347	27683	9
59	556	938	2950	5104.9	4908.8	14667	16524	8
60	624	1100	3000	4806.1	3832.1	25067	28445	8
61	569	929	3050	1150.1	86.99	15562	17746	7
62	630	1100	3100	0	0	25007	27337	9
63	582	920	3150	3063.3	2482.9	15334	17390	8
64	638	1100	3200	1513.6	0	27037	29828	4
65	595	911	3250	5145.2	4930.3	15083	17003	8

Table 11.1 Continued

Run No.	Beam dpth mm	& breadth mm	Design moment kNm	BS 8110 comprn. mm²	EC2 comprn. mm²	BS 8110 tension area mm²	EC2 mm² tension area	Perc. diff. %
66	645	1100	3300	5618.8	4677.9	26562	30119	8
67	610	899	3350	1186.7	0	15967	18145	6
68	652	1100	3400	0	0	26673	29156	9
69	623	889	3450	3001.2	2390	15694	17809	8
70	659	1100	3500	1559.8	0	28515	31922	6
71	636	880	3550	5141	4906.7	15402	17372	8
72	666	1100	3600	6340.7	5424.1	27965	31694	8
73	649	871	3650	1217.7	0	16364	18524	5
74	672	1100	3700	0	0	28358	30993	9
75	662	862	3750	2977.1	2340.8	16055	18227	8
76	679	1100	3800	1603.8	0	29937	33984	8
77	676	853	3850	5136.4	4883.1	15708	17726	8
78	685	1100	3900	7110.7	6227.3	29352	33246	8
79	689	845	3950	1248.9	0	16705	18794	5
80	695	1100	4000	0	0	29736	32498	9
81	702	836	4050	2910.8	2245.9	16360	18584	8
82	709	1100	4100	1669.8	0	30981	34945	7
83	715	827	4150	5158.3	4891.2	15999	18061	8
84	722	1100	4200	6610.9	5600.7	30070	34099	8
85	729	816	4250	1271.3	0	16998	19071	4
86	735	1100	4300	0	0	29853	32632	9
87	742	808	4350	2869.1	2183	16618	18885	8
88	748	1100	4400	1757.8	0	31688	34944	4
89	756	800	4450	5133.9	4850.1	16222	18320	8
90	761	1100	4500	6002.2	4852.5	30712	34869	8
91	768	792	4550	1297.3	0	17289	19328	4
92	775	1100	4600	0	0	30011	32809	9
93	782	784	4650	2823.5	2115.3	16874	19184	8
94	788	1100	4700	0	0	31993	34944	9
95	793	776	4750	5194.6	4905	16484	18619	9
96	798	1100	4800	5417.1	4132.3	31344	35628	8
97	800	768	4850	1305.6	0	17652	19893	5
98	798	1100	4900	0	0	31054	33949	9
99	798	760	4950	3549.9	2922.1	17458	19817	8
100	798	1100	5000	1870	0	33775	37243	4
101	202	666	50	0	0	1108.7	1213.6	9
102	200	683	100	0	0	1289.4	1410.7	9
103	214	1081	150	0	0	3205	3507.7	9
104	211	1072	200	0	0	2414.5	2642	9
105	245	1063	250	0	0	4692.4	5135.4	9
106	228	1054	300	0	0	3391.8	3710.9	9
107	280	1045	350	0	0	6263.4	6848.2	9
108	242	1036	400	605.02	0	4845.8	5429.2	0
109	307	1027	450	0	0	7095	7760.7	9
110	254	1018	500	0	0	5729.1	6257.8	9
111	332	1009	550	0	0	7840.5	8578.3	9
112	271	1000	600	0	0	6207.2	6784	9
113	354	990	650	0	0	9771	10674	9
114	287	981	700	1854.3	1628.6	7016.2	7899.2	7
115	374	972	750	0	0	10059	10998	9
116	304	963	800	936.28	508.03	7645.4	8658.2	7
117	392	954	850	0	0	10535	11522	9
118	317	945	900	693.63	0	8332.3	9155.7	1
119	408	971	950	889.44	0	12477	14196	6
120	330	927	1000	2974.4	2831.1	8582.9	9644.3	8
121	425	1000	1050	0	0	12558	13728	9
122	343	909	1100	2043.2	1697.6	9188.6	10378	8
123	439	1032	1150	0	0	12720	13913	9
124	354	890	1200	1156.6	614.98	9821.1	11140	7
125	452	1064	1250	1068.3	0	14849	16690	5
126	365	879	1300	4073.4	4020.8	9955.6	11166	8
127	465	1091	1350	0	0	14696	16066	9
128	376	898	1400	2834.2	2527.1	10572	11927	8
129	479	1000	1450	0	0	14628	16059	9
130	368	919	1500	2516	2018.9	11817	13317	7

Table 11.2 Column reinforcement: BS 8110 *cf*. Eurocode 2

Run No.	Col dep & bread mm	mm	Factored axial kN	BM kNm	BS8110 No.bars & dia		EC2 No. of bars & dia		BS8110 area mm2	EC2 area mm2	Perc. diff. %
1	250	250	100	75	8	20	6	20	2513	1884	33
2	260	260	200	84	6	20	6	20	1884	1884	0
3	260	260	300	84	6	20	6	20	1884	1884	0
4	270	270	400	94	6	20	6	20	1884	1884	0
5	270	270	500	94	6	20	4	20	1884	1256	50
6	280	280	600	105	8	20	8	20	2513	2513	0
7	280	280	700	105	6	20	6	20	1884	1884	0
8	290	290	800	117	6	20	6	20	1884	1884	0
9	290	290	900	117	6	20	6	20	1884	1884	0
10	300	300	1000	129	6	20	6	20	1884	1884	0
11	310	310	1100	142	10	20	6	25	3141	2945	7
12	310	310	1153	142	8	20	6	25	2513	2945	17
13	320	320	1228	157	6	25	8	20	2945	2513	17
14	320	320	1228	157	4	25	8	20	1963	2513	28
15	330	330	1306	172	4	25	6	20	1963	1884	4
16	330	330	1306	172	8	25	12	20	3926	3769	4
17	340	340	1387	188	8	25	10	20	3926	3141	25
18	340	340	1387	188	6	25	6	25	2945	2945	0
19	350	350	1470	205	6	25	8	20	2945	2513	17
20	360	360	1555	223	4	25	8	20	1963	2513	28
21	360	360	1555	223	8	25	12	20	3926	3769	4
22	370	370	1642	243	8	25	12	20	3926	3769	4
23	370	370	1642	243	6	25	10	20	2945	3141	7
24	380	380	1732	263	6	25	6	25	2945	2945	0
25	380	380	1732	263	6	25	8	20	2945	2513	17
26	390	390	1825	284	10	25	6	32	4908	4825	2
27	390	390	1825	284	8	25	6	32	3926	4825	23
28	400	400	1920	307	6	32	12	20	4825	3769	28
29	410	410	2017	330	4	32	10	20	3216	3141	2
30	410	410	2017	330	4	32	8	20	3216	2513	28
31	420	420	2116	355	8	32	12	25	6433	5890	9
32	420	420	2116	355	6	32	6	32	4825	4825	0
33	430	430	2218	381	6	32	8	25	4825	3926	23
34	430	430	2218	381	4	32	12	20	3216	3769	17
35	440	440	2323	408	4	32	6	25	3216	2945	9
36	440	440	2323	408	8	32	12	25	6433	5890	9
37	450	450	2430	437	8	32	12	25	6433	5890	9
38	460	460	2539	467	6	32	6	32	4825	4825	0
39	460	460	2539	467	6	32	12	20	4825	3769	28
40	470	470	2650	498	4	32	10	20	3216	3141	2
41	470	470	2650	498	8	32	8	32	6433	6433	0
42	480	480	2764	530	8	32	12	25	6433	5890	9
43	480	480	2764	530	6	32	6	32	4825	4825	0
44	490	490	2881	564	6	32	6	32	4825	4825	0
45	490	490	2881	564	4	32	12	20	3216	3769	17
46	500	500	3000	600	6	40	6	40	7539	7539	0
47	510	510	3121	636	6	40	8	32	7539	6433	17
48	510	510	3121	636	6	40	12	25	7539	5890	28
49	520	520	3244	674	4	40	6	32	5026	4825	4
50	520	520	3244	674	4	40	12	20	5026	3769	33
51	530	520	3307	701	6	40	10	32	7539	8042	7
52	530	520	3307	701	6	40	6	40	7539	7539	0
53	540	510	3304	713	6	40	12	25	7539	5890	28
54	540	510	3304	713	4	40	10	25	5026	4908	2
55	550	500	3300	726	4	40	12	20	5026	3769	33
56	560	490	3292	737	10	32	10	32	8042	8042	0
57	560	490	3292	737	8	32	6	40	6433	7539	17
58	570	480	3283	748	8	32	12	25	6433	5890	9
59	570	480	3283	748	6	32	6	32	4825	4825	0
60	580	470	3271	758	6	32	12	20	4825	3769	28
61	580	470	3271	758	10	32	6	40	8042	7539	7
62	590	460	3256	768	8	32	8	32	6433	6433	0
63	590	460	3256	768	8	32	12	25	6433	5890	9
64	600	450	3240	777	6	32	6	32	4825	4825	0
65	610	440	3220	785	6	32	12	20	4825	3769	28

Table 11.2 Continued

Run No.	Col dep mm	& bread mm	Factored axial kN	BM kNm	BS8110 No.bars	& dia	EC2 No. of bars	& dia	BS8110 area mm2	EC2 area mm2	Perc. diff. %
66	610	440	3220	785	10	32	6	40	8042	7539	7
67	620	430	3199	793	8	32	8	32	6433	6433	0
68	620	430	3199	793	8	32	12	25	6433	5890	9
69	630	420	3175	800	6	32	6	32	4825	4825	0
70	630	420	3175	800	6	32	12	20	4825	3769	28
71	640	410	3148	806	10	32	6	40	8042	7539	7
72	640	410	3148	806	8	32	8	32	6433	6433	0
73	650	400	3120	811	6	32	12	25	4825	5890	22
74	660	390	3088	815	8	25	6	32	3926	4825	23
75	660	390	3088	815	8	25	12	20	3926	3769	4
76	670	380	3055	818	14	25	6	40	6872	7539	10
77	670	380	3055	818	12	25	12	25	5890	5890	0
78	680	370	3019	821	10	25	6	32	4908	4825	2
79	680	370	3019	821	8	25	6	32	3926	4825	23
80	690	360	2980	822	6	25	10	20	2945	3141	7
81	690	360	2980	822	14	25	8	32	6872	6433	7
82	700	350	2940	823	12	25	12	25	5890	5890	0
83	710	360	3067	871	10	25	10	25	4908	4908	0
84	710	360	3067	871	8	25	6	32	3926	4825	23
85	720	360	3110	895	8	25	12	20	3926	3769	4
86	720	360	3110	895	14	25	6	40	6872	7539	10
87	730	370	3241	946	12	25	8	32	5890	6433	9
88	730	370	3241	946	10	25	12	25	4908	5890	20
89	740	370	3285	972	8	25	6	32	3926	4825	23
90	740	370	3285	972	8	25	12	20	3926	3769	4
91	750	380	3420	1000	16	25	6	40	7853	7539	4
92	760	380	3465	1000	12	25	8	32	5890	6433	9
93	760	380	3465	1000	10	25	12	25	4908	5890	20
94	770	390	3603	1000	8	25	8	25	3926	3926	0
95	770	390	3603	1000	6	25	6	25	2945	2945	0
96	780	390	3650	1000	14	25	6	40	6872	7539	10
97	780	390	3650	1000	12	25	12	25	5890	5890	0
98	790	400	3792	1000	6	32	6	32	4825	4825	0
99	790	400	3792	1000	4	32	12	20	3216	3769	17
100	800	400	3840	1000	4	32	8	20	3216	2513	28
101	250	250	750	75	4	20	4	20	1256	1256	0
102	260	260	811	84	6	20	6	20	1884	1884	0
103	260	260	811	84	6	20	6	20	1884	1884	0
104	270	270	874	94	8	20	8	20	2513	2513	0
105	270	270	874	94	8	20	8	20	2513	2513	0
106	280	280	940	105	6	20	6	20	1884	1884	0
107	280	280	940	105	6	20	6	20	1884	1884	0
108	290	290	1009	117	8	20	8	20	2513	2513	0
109	290	290	1009	117	8	20	8	20	2513	2513	0
110	300	300	1080	129	10	20	6	25	3141	2945	7
111	310	310	1153	142	6	20	6	20	1884	1884	0
112	310	310	1153	142	6	20	8	20	1884	2513	33
113	320	320	1228	157	6	25	8	20	2945	2513	17
114	320	320	1228	157	6	25	6	25	2945	2945	0
115	330	330	1306	172	8	25	12	20	3926	3769	4
116	330	330	1306	172	4	25	6	20	1963	1884	4
117	340	340	1387	188	6	25	8	20	2945	2513	17
118	340	340	1387	188	6	25	6	25	2945	2945	0
119	350	350	1470	205	8	25	12	20	3926	3769	4
120	360	360	1555	223	8	25	12	20	3926	3769	4
121	360	360	1555	223	4	25	8	20	1963	2513	28
122	370	370	1642	243	6	25	6	25	2945	2945	0
123	370	370	1642	243	6	25	10	20	2945	3141	7
124	380	380	1732	263	8	25	12	20	3926	3769	4
125	380	380	1732	263	10	25	6	32	4908	4825	2
126	390	390	1825	284	6	25	8	20	2945	2513	17
127	390	390	1825	284	6	25	6	25	2945	2945	0
128	400	400	1920	307	6	32	12	20	4825	3769	28
129	410	410	2017	330	6	32	6	32	4825	4825	0
130	410	410	2017	330	8	32	10	25	6433	4908	31

Table 11.3 Shear reinforcement: BS 8110 *cf*. Eurocode 2

Run No.	Beam depth mm	& breadth mm	Area of links mm^2	Link crs. mm	Design shear kN	BS 8110 shear resist.	EC2 shear resist. prov. kN	Perc. diff. %
1	200	150	100.53	75	80	81.659	52.451	56
2	235	174	157.08	110	163.56	173.82	120.81	44
3	271	198	226.19	80	214.63	227.89	149.92	52
4	307	222	150.8	90	272.62	285.69	185.18	54
5	342	247	235.62	60	337.9	366.55	262.77	39
6	378	271	339.29	160	409.75	414.32	288.57	44
7	414	295	301.59	70	488.52	510.42	348.99	46
8	450	320	314.16	130	576	592.35	391.49	51
9	485	344	452.39	90	667.36	669.28	476.98	40
10	521	368	201.06	70	766.91	767.61	538.73	42
11	557	392	392.7	70	873.38	894.64	611.37	46
12	592	417	565.49	180	987.46	1004.7	669.54	50
13	628	441	402.12	60	1107.8	1143.2	823.48	39
14	664	465	392.7	100	1235	1322.2	938.71	41
15	700	490	678.58	90	1372	1480.2	1032.6	43
16	735	514	301.59	75	1511.2	1583.5	1064	49
17	771	538	549.78	70	1659.2	1663.1	1184.8	40
18	807	562	678.58	150	1814.1	1848.6	1314.3	41
19	842	587	552.92	60	1977	2140.3	1518.1	41
20	878	611	549.78	120	2145.8	2153.9	1448.1	49
21	914	635	791.68	80	2321.6	2457.6	1769.7	39
22	950	660	402.12	75	2508	2568.2	1833.7	40
23	985	684	628.32	60	2695	2856.9	2018.1	42
24	1021	708	904.78	160	2891.5	3031.9	2078.5	46
25	1057	732	603.19	50	3094.9	3389.3	2494.8	36
26	1092	757	706.86	110	3306.6	3516.5	2526.2	39
27	1128	781	1017.9	90	3523.9	3555.1	2496	42
28	1164	805	452.39	70	3748.1	3921.5	2708.1	45
29	1200	830	785.4	60	3984	4180.6	3073.3	36
30	1235	854	1131	160	4218.8	4373.1	3142.7	39
31	1271	878	603.19	45	4463.8	4659.7	3333.3	40
32	1307	902	785.4	110	4715.7	4847.7	3359.5	44
33	1342	927	1244.1	80	4976.1	5478.9	4083.1	34
34	1378	951	552.92	70	5241.9	5404.5	3918.5	38
35	1414	975	942.48	60	5514.6	6038.1	4345.6	39
36	1450	1000	1357.2	175	5800	5846.2	4048.2	44
37	1450	1000	904.78	300	1450	2493.8	1574.3	58
38	1414	975	863.94	240	1378.7	1880.8	995.88	89
39	1378	951	552.92	190	1310.5	2462.1	1443.6	71
40	1342	927	1244.1	140	1244	3498.4	2333.2	50
41	1307	902	785.4	300	1178.9	1998.7	1231.8	62
42	1271	878	502.65	240	1115.9	1186.7	520.82	>99
43	1235	854	791.68	190	1266.3	2809.9	1852.5	52
44	1200	830	785.4	140	1464	2132	1317.1	62
45	1164	805	452.39	180	1661.7	1711.1	1053.2	62
46	1128	781	1017.9	190	1859.4	1913.6	1182.3	62
47	1092	757	706.86	190	2057.1	2227.1	1462.5	52
48	1057	732	502.65	60	2254.9	2494.5	1732.5	44
49	1021	708	904.78	190	2452.6	2484.2	1750.3	42
50	985	684	628.32	60	2650.3	2817.9	2018.1	40
51	950	660	402.12	75	2508	2563.5	1833.7	40
52	914	635	791.68	80	2321.6	2536.9	1769.7	43
53	878	611	549.78	110	2145.8	2211.7	1579.8	40
54	842	587	552.92	60	1977	2098.7	1518.1	38
55	807	562	678.58	150	1814.1	1889.2	1314.3	44
56	771	538	549.78	75	1659.2	1674.3	1105.8	51
57	735	514	301.59	70	1511.2	1590.5	1140	40
58	700	490	678.58	90	1372	1476.9	1032.6	43
59	664	465	392.7	110	1235	1277.8	853.37	50
60	628	441	402.12	70	1107.8	1115	705.84	58
61	592	417	565.49	175	987.46	991.76	688.67	44
62	557	392	392.7	70	873.38	916.31	611.37	50
63	521	368	201.06	75	766.91	801.05	502.82	59
64	485	344	452.39	100	667.36	698.67	429.28	63
65	450	320	314.16	130	576	589.11	391.49	50

Table 11.3 Continued

Run No.	Beam depth mm	& breadth mm	Area of links mm²	Link crs. mm	Design shear kN	BS 8110 shear resist.	EC2 shear resist. prov. kN	Perc. diff. %
66	414	295	301.59	80	488.52	489.51	305.36	60
67	378	271	339.29	180	409.75	422.24	256.5	65
68	342	247	235.62	75	337.9	349.1	210.21	66
69	307	222	150.8	100	272.62	279.91	166.66	68
70	271	198	226.19	90	214.63	224.72	133.26	69
71	235	174	157.08	140	163.56	164.87	94.921	74
72	200	150	150.8	80	120	125.78	73.759	71
73	200	150	150.8	80	120	125.78	73.759	71
74	235	174	157.08	140	163.56	164.87	94.921	74
75	271	198	226.19	90	214.63	224.72	133.26	69
76	307	222	150.8	100	272.62	279.91	166.66	68
77	342	247	235.62	75	337.9	349.1	210.21	66
78	378	271	339.29	180	409.75	422.24	256.5	65
79	414	295	301.59	80	488.52	489.51	305.36	60
80	450	320	314.16	130	576	589.11	391.49	50
81	485	344	452.39	100	667.36	698.67	429.28	63
82	521	368	201.06	75	766.91	801.05	502.82	59
83	557	392	392.7	70	873.38	916.31	611.37	50
84	592	417	565.49	175	987.46	991.76	688.67	44
85	628	441	402.12	70	1107.8	1115	705.84	58
86	664	465	392.7	110	1235	1277.8	853.37	50
87	700	490	678.58	90	1372	1476.9	1032.6	43
88	735	514	301.59	70	1511.2	1590.5	1140	40
89	771	538	549.78	75	1659.2	1674.3	1105.8	51
90	807	562	678.58	150	1814.1	1889.2	1314.3	44
91	842	587	552.92	60	1977	2098.7	1518.1	38
92	878	611	549.78	110	2145.8	2211.7	1579.8	40
93	914	635	791.68	80	2321.6	2536.9	1769.7	43
94	950	660	402.12	75	2508	2563.5	1833.7	40
95	985	684	628.32	60	2650.3	2817.9	2018.1	40
96	1021	708	904.78	190	2452.6	2484.2	1750.3	42
97	1057	732	502.65	60	2254.9	2494.5	1732.5	44
98	1092	757	706.86	190	2057.1	2227.1	1462.5	52
99	1128	781	1017.9	190	1859.4	1913.6	1182.3	62
100	1164	805	452.39	180	1661.7	1711.1	1053.2	62
101	1200	830	785.4	140	1464	2132	1317.1	62
102	1235	854	791.68	190	1266.3	2809.9	1852.5	52
103	1271	878	502.65	240	1115.9	1186.7	520.82	>99
104	1307	902	785.4	300	1178.9	1998.7	1231.8	62
105	1342	927	1244.1	140	1244	3498.4	2333.2	50
106	1378	951	552.92	190	1310.5	2462.1	1443.6	71
107	1414	975	863.94	240	1378.7	1880.8	995.88	89
108	1450	1000	904.78	300	1450	2493.8	1574.3	58
109	200	150	100.53	140	80	93.796	51.702	81
110	235	174	157.08	60	163.56	177.72	120.37	48
111	271	198	226.19	150	214.63	218.57	147.12	49
112	307	222	201.06	60	272.62	281.3	201.28	40
113	342	247	235.62	130	337.9	349.3	223.15	57
114	378	271	339.29	90	409.75	409.7	278.81	47
115	414	295	150.8	60	488.52	532.46	374.58	42
116	450	320	314.16	60	576	636.58	460.99	38
117	485	344	452.39	140	667.36	824.35	564.19	46
118	521	368	351.86	60	766.91	857.18	597.78	43
119	557	392	314.16	100	873.38	896.13	629.95	42
120	592	417	565.49	90	987.46	1011.8	727.76	39
121	628	441	251.33	75	1107.8	1137.7	757.6	50
122	664	465	471.24	70	1235	1273	874.57	46
123	700	490	678.58	175	1372	1381.8	977.16	41
124	735	514	502.65	60	1511.2	1669	1204.7	39
125	771	538	471.24	110	1659.2	1754.7	1189.1	48
126	807	562	678.58	80	1814.1	1906.8	1339.3	42
127	842	587	351.86	75	1977	1993.7	1422.1	40
128	878	611	549.78	60	2145.8	2204.8	1574	40
129	914	635	791.68	140	2321.6	2675	1860.7	44
130	950	660	603.19	60	2508	2655.7	1868.6	42

Chapter 12

Conclusions

A comprehensive library search has not revealed literature on the subject of verification of engineering calculations, the failure of structures reported in the NCE (New Civil Engineer) over recent years makes the development of self-checking software and the verification of engineering models, compelling and urgent. Although a unified treatment has been developed for both the verification of models for the structural analysis of frameworks and models for the structural design of components, it is convenient to report on each as a separate entity.

The following **objectives**, described in section 1.4, have been met:

- a **self-check** has been included in over a hundred structural models ensuring that the calculations given are correct or highlighted when not
- a unified method for dealing with calculations for the structural analysis of a framework or for the design of a structural component has been achieved
- the nature of the structural data has been classified enabling a PARAMETER table to be produced from which a thousand sets of engineered data may be produced automatically
- the software developed is predominantly plain text rather than *procedural*, which will ensure that the software may be read in a hundred years
- the tools to aid verification, given in chapter 3, have been developed
- by comparing the results from running a model with those given by the *self check* the engineer can get a feel for the problem, the kernel for each model is given in appendix A
- the system devised is capable of verifying the correctness of all the calculations in the *IStructE wishlist*, Seifert *et al.* (2000).

12.1 Models for structural analysis

Conclusions are given in order of verified model *number* listed in section 7.8. The conclusions are distilled versions of those presented at the end of each model. For reason of space, only the kernel is included in appendix A. Average percentage differences reported in this chapter are computed from summing the absolute values of the differences to yield a total, then dividing the total by the number of differences sampled.

vm110 Deflection of beams including shear *cf.* Chebyshev polynomials

All three load cases, each of which is equivalent to a uniformly distributed loading, gave identical results, as expected. NL-STRESS & classical theory agree exactly for zero shear deformation and for any sensible value of Poisson's ratio which included shear deformation effects, for all 996 sets of test data. The percentage differences between classical beam theory and the Chebyshev polynomials method varies according to the span:depth ratio of the beam, differences being in excess of 10% for short spans.

vm112 Cantilevered beam *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.02%, the maximum percentage difference being 0.3% in run 1.

vm113 Cantilevered beam with many loads *cf.* unit load method

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the unit load method was 1.234%, the maximum percentage difference being 7% in run 499. When the data for run 499 was extracted and run with shear deformation suppressed then the percentage difference was 0% as expected for shear deformation was not included in the unit load method.

vm114 Tapered cantilevered beam *cf.* unit load method

The principle of using logic to avoid difficult integrations is profound. The entire procedure for the unit load method is contained within 5 lines. For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the unit load method was 1.40%, the maximum percentage difference being 12% in run 501; when the data for run 501 was extracted and run with shear deformation suppressed then the percentage difference was 0%.

vm115 Cantilevered beam with tie down span *cf.* Roark

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Roark's formulae with shear deformation suppressed was 0.0%. When shear deformation was taken into account for 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Roark's formulae was 3.89%, the maximum percentage difference being 50% in run 73. Investigation of run 73 showed that the difference was due to shear deformation having a significant effect on a very low end displacement ≈0.2mm.

vm117 Subframe, continuous beam + columns *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.2869%, the maximum percentage difference being 1.25% in run 499.

vm120 Continuous beam *cf.* Hardy Cross
For 996 runs generated from the parameter table, for continuous beams of various spans, with a mixture of loads, exact agreement was obtained when >=32 cycles were taken for the moment distribution and shear deformation was excluded. When shear deformation is included, over 100% differences in bending moments occur within end spans having spans:depth ratios of 4:1.

vm122 Two member lean-to or Mansard beam *cf.* equilibrium, compatibility & energy
For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.072%, the maximum percentage difference being 0.778% in run 169.

vm123 Three member lean-to/Mansard beam *cf.* equilibrium, compatibility & energy
For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.061%, the maximum percentage difference being 4.0% in run 550.

vm124 Three member cranked beam *cf.* equilibrium, compatibility & energy
For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.03%, the maximum percentage difference being 0.56% in run 173.

vm130 Ground beam on an elastic foundation *cf.* Hetényi
The engineer who devises models such as this, in addition to devising robust logic, must also be aware of roundoff, especially when a set of load data is incremented. The problem of roundoff only came to light when 996 runs in a batch was completed and the results studied. If shear deformation is suppressed, results from NL-STRESS and Hetényi (1948), agree to an average accuracy of 0.04302% for all 996 sets of data generated from the parameter table. When shear deformation is not suppressed but Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for 996 sets of data generated from the parameter table =0.0425%, with the largest individual result being for run 502 when the average difference was 2.037%. It is remarkable that such close agreement was achieved, for Hetényi (1948) solved the governing differential equations and provided a solution which includes many trigonometric and hyperbolic functions, as will be evident by perusal of the expressions between the SOLVE and FINISH commands in the data; whereas NL-STRESS uses the stiffness method, modelling the soil by lumped stiffness springs at the joints.

vm131 Ground beam on elastic piles *cf.* flexibility
The bending moments computed by the stiffness method were compared with those computed by flexibility. When shear deformation was suppressed, results from NL-STRESS and the flexibility method agree to an average accuracy of 0.000% for all 996

sets of data generated from the parameter table. When shear deformation was not suppressed but Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for 996 sets of data generated from the parameter table =2.1515%, with the largest individual result being in run 498 for which the average difference was 19.944%. Investigation of this run showed that Poisson's ratio was 0.3 and that the loading was concentrated in the first tenth of the beam, with the remainder of the beam being used to tie down the high bending moments concentrated in the first tenth of the beam; using the set of data for run 498 but suppressing shear deformation, again gave exact agreement.

vm140 Influence lines *cf.* Müller-Breslau

If shear deformation is suppressed, results from NL-STRESS and Müller-Breslau agree precisely for all 996 sets of data generated from the parameter table. When shear deformation in not suppressed but Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for 996 sets of data generated from the parameter table =8.489%; the largest individual result being for run 671 when the average difference was 50.947%. Investigation of this run showed that the set of data had a combination of very large differences in the span lengths combined with loading concentrated in the first half of the beam.

vm150 Pratt through truss *cf.* method of joints

For 996 runs generated from the parameter table, the average percentage difference between the forces in the members of the truss computed by NL-STRESS and those found by the method of joints was zero percent as expected.

vm153 Pratt deck truss *cf.* method of joints
vm156 Howe through truss *cf.* method of joints
vm159 Howe deck truss *cf.* method of joints
vm162 Warren through truss *cf.* method of joints
vm164 Warren through truss with verticals *cf.* method of joints
vm165 Warren deck truss *cf.* method of joints
vm168 Warren deck with verticals *cf.* method of joints

All seven models listed above *i.e.* vm153 thru vm168 have similar conclusions to that for vm150, their conclusions are omitted for reason of space, but all are contained in Appendix A.

vm171 Two rafters with tie *cf.* method of joints

The stresses due to bending, greatly exceed the axial stresses found *via* the method of joints. Bending stresses due to loads applied between the joints, must be computed for both the design loads and possible future loads due to change of usage. Generally the method of joints, applied alone, is insufficient to predict the maximum stresses in roof trusses throughout their life. For 996 runs generated from the parameter table, the average percentage difference between the forces in the members computed by NL-STRESS and those computed by the method of joints was zero percent as expected.

vm172 Two rafters, post & tie *cf.* method of joints
vm173 King post roof truss *cf.* method of joints
vm174 Three segment rafters, Pratt internals roof truss *cf.* method of joints
vm175 Three segment rafters, Howe internals roof truss *cf.* method of joints
vm177 Trussed rafter, or Fink roof truss *cf.* method of joints
vm178 Three segment trussed rafter, Warren internals roof truss *cf.* method of joints
vm179 Three segment rafters, Warren internals roof truss *cf.* method of joints
vm181 Mansard truss *cf.* method of joints

All eight models listed above *i.e.* vm172 thru vm181 have similar conclusions to those for vm171, their conclusions are omitted for reason of space, but all are contained in Appendix A.

vm202 Pipe tree having two branches *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.015%, the maximum percentage difference being 0.889% in run 653.

vm203 Pipe tree having four branches *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.013%, the maximum percentage difference being 0.33% in run 166.

vm204 Pipe tree having six branches *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.151%, the maximum percentage difference being 0.667% in run 161.

vm207 One storey bent, vertical/raking columns *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.026%, the maximum percentage difference being 0.778% in run 155.

vm208 Two storey bent, vertical/raking columns *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.055%, the maximum percentage difference being 0.333% in run 99.

vm209 Three storey bent, vertical/raking columns *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.059%, the maximum percentage difference being 0.222% in run 57.

vm210 Bent *cf.* column analogy

If sensible section properties are given to the transom & columns, then the matrix stiffness method and column analogy agree to within a few percent; or agree precisely if axial and shear deformation are suppressed. NOT OK will be reported when silly section properties are provided *e.g.* when columns are small *e.g.* 203 x 133 x 25 UB and the transom is large *e.g.* 914 x 419 x 388 UB and axial and shear deformation effects are taken into account. This is because axial strain effects swamp the bending strain to the extent of reversing the sign of the moment at joint 2. If axial and shear deformation are suppressed (by multiplying areas by 1000 & setting Poisson's ratio to 1E-12) then even silly section properties show exact agreement between the matrix stiffness method and column analogy. When this verified model is tested against 996 sets of data (imported *via* #cc924.stk), then exact agreement can be achieved by adding the line: nu=1E-12 ax1=ax1*1000 ax2=ax2*1000 following the import. As this extra line comes after the import of data, it will reset the parameters: nu, ax1 & ax2 to suppress both axial and shear deformation. The above raises the question: should axial and shear deformation be ignored, the answer must be an emphatic *no*. Axial shortening does effect bending moments (particularly those in the outside columns of a multi-storey frame). It is worrying to see that the IStructE Guidelines for the use of computers in engineering calculations, in a highlighted section entitled *Validate the model*, quote the case of an engineer who had *inadvertently included the effect of axial shortening in the members*. Axial effects must always be included; as structural members do change in length when axial loads are applied, the change in length due to axial loads on columns can significantly effect the bending moments in connected beams.

When axial and shear deformation are taken into account, for 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the column analogy method was 0.1674%, the maximum percentage difference being 28.75% in run 498.

vm211 Rigid pile cap *cf.* Reinforced Concrete Designers' Handbook

If sensible section properties are given to the transom & columns, and the ratio of vertical load to horizontal load is not greater than 20 then the matrix stiffness method and Reynolds' corrected method agree to within a few percent. When this verified model is tested against 996 sets of data, and mcr=1 to make the pile cap rigid, then the average percentage difference between Reynolds and NL-STRESS is 0.7959% with the maximum percentage difference of 15.583% in run 656. When tested against 996 sets of data and mcr=0, then the average percentage difference between Reynolds and NL-

STRESS is reduced to 0.1123% with the maximum percentage difference being 15.583% again in run 656.

vm215 Portal frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.1340%, the maximum percentage difference being 5.0% in run 663.

vm216 Mansard portal *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.00680% the maximum percentage difference being 0.2222% in run 326.

vm217 Gable frame with inclined legs *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.00156%, the maximum percentage difference being 0.1111% in run 2.

vm218 Portal with skew corners *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.0126%, the maximum percentage difference being 0.2222% in run 4.

vm219 Trapezoidal frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.01138%, the maximum percentage difference being 2.0% in run 664.

vm220 Two bay ridged portal *cf.* Kleinlogel

For 996 runs, with shear both shear and axial deformation suppressed, NL-STRESS and Kleinlogel agree exactly. For 996 runs, with shear both shear and axial deformation taken into consideration, the average difference was 0.1609% with a maximum difference of 15.5% in run 829. Results generally agreed to within 1% save for three rogue runs. Investigation of these rogue runs, showed they were caused by axial deformation; axial deformation is ignored by Kleinlogel.

vm225 Couple roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.000111%, the maximum percentage difference being 0.111% in run 157.

vm226 Couple close roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.000223%, the maximum percentage difference being 0.1111% in run 797.

vm227 Collar-tie roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.02711%, the maximum percentage difference being 0.2222% in run 332.

vm228 Collar-and-tie roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.01110, the maximum percentage difference being 1.4444% in run 957.

vm230 Attic room roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.0087%, the maximum percentage difference being 0.1111% in run 43.

vm232 Fink room roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.006024%, the maximum percentage difference being 0.2222% in run 172.

vm233 King post roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.007251%, the maximum percentage difference being 0.2222% in run 503.

vm234 Queen post roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.000892%, the maximum percentage difference being 0.1111% in run 658.

vm235 Tied Mansard roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.010262%, the maximum percentage difference being 0.2222% in run 678.

vm241 Vierendeel girder *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.221%, the maximum percentage difference being 0.667% in run 790.

vm242 Vierendeel roof frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.000446%, the maximum percentage difference being 0.1111% in run 316.

vm244 N/Pratt lattice portal/girder *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.006804%, the maximum percentage difference being 0.3333% in run 825.

vm245 Howe lattice portal/girder *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.00625%, the maximum percentage difference being 0.3333% in run 825.

vm246 Warren portal/girder end diags in tension *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.0068%, the maximum percentage difference being 0.3333% in run 825.

vm247 Warren portal/girder end diags in compr. *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.0429%, the maximum percentage difference being 0.7778% in run 777, the triple sevens being a coincidence.

vm260 Multi-storey frame *cf.* Hardy Cross

When axial deformation is included in the design of multi-storey frames, over 100% differences in bending moments can occur, with the biggest differences in the outside columns & roof beams. BS 8110 produces substantially less reinforcement than both CP 110 (the code it has replaced) and the Eurocode. The design stress for high yield deformed round bars was 312 N/mm² (425/1.36) to CP 110 but this has been increased to 460/gammaS N/mm² to BS 8110; furthermore the minimum percentage of reinforcement in columns has been reduced from 1% required by CP 110 to 0.4% required by BS 8110. You only need one bucket of bad concrete in a column to be in real trouble, Hume *et al.* (1993). If axial and shear deformation are suppressed, by setting Poisson's ratio =1E-12 which in turn sets shear areas to zero and multiplies cross-sectional areas by 1E6, then results agree precisely for all 996 sets of data generated from the parameter table. When neither shear nor axial deformation is suppressed and Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for 996 sets of data generated from the parameter table =6.2034%; the largest individual result being for run 664 when the average difference was 47.291%.

vm262 Multi storey frame *cf.* equilib., compatibility & energy
For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.1095%, the maximum percentage difference being 0.6667% in run 332.

vm270 Pierced shear walls *cf.* Magnus
Magnus (1968) ignores axial deformation due to applied vertical loading and only considers axial deformation in the lintel; thus for a comparison, shear deformation was suppressed in NL-STRESS. Results comparing NL-STRESS with Magnus agree to an average accuracy of 0.4689% for all 996 sets of data generated from the parameter table with a maximum difference of 4% in run 501. When Poisson's ratio is varied between 0.1 and 0.3, results comparing NL-STRESS with Magnus agree to an average accuracy of 4.1717% for all 996 sets of data generated, with a maximum difference of 23% in run 830.

vm280 Two pinned circular arch *cf.* Pippard & Baker
From 996 runs, NL-STRESS agrees with the classical analysis of Pippard & Baker (1957) to an average accuracy of 0.001406% with a maximum percent difference of 0.6% in run 831.

vm281 Encastré circular arch *cf.* Pippard & Baker
From 996 runs, NL-STRESS agrees with the classical analysis of Pippard & Baker (1957) to an average accuracy of 0.6863% with a maximum difference of 15.571% in run 340. This difference was due a sign change in the very low bending moment at the right support. Increasing the number of segments to 96 for this set of data reduced the average percentage difference from 15.571% to 0.1429%. Only systematic testing can find such blips.

vm282 Two pinned parabolic arch *cf.* Pippard & Baker
Classical methods for the analysis of parabolic arches are covered in many textbooks, though all seem to have been derived from a common parent as they all ignore rib shortening which has a major effect on the structural performance of arches for which the height:span is less than 1/3.

When rib-shortening is excluded and the moment of inertia varies as the secant of the arch slope, then for 996 runs, NL-STRESS agrees with the classical analysis of Pippard & Baker to an average percentage difference of 0.52229%, the maximum percentage difference being 20% in run 260. When the data for run 260 was run with an increase in the number of loads to 10, then the average percentage difference for run 260 was reduced from 20% to 4.6%.

When rib-shortening is excluded and the moment of inertia does not vary, then for 996 runs, NL-STRESS agrees with the classical analysis of Pippard & Baker to an average percentage difference of 4.0594%, the maximum percentage difference being 20.4% in run 297. When the data for run 297 was run and the moment of inertia varied as the secant of the arch slope, then the average percentage difference for run 297 was reduced from 20.4% to 2.6%.

When rib-shortening is included then large percentage differences arise dependent on the cross-sectional area of the arch. When rib-shortening is included and the moment of inertia varies as the secant of the arch slope, then for 996 runs the average percentage difference between NL-STRESS and the classical analysis of Pippard & Baker is 24.0%, the maximum percentage difference being 30.4% in run 618.

Of course rib-shortening is real and should be included especially when the height:span is less than 1/3; when the arch is flat, stability also needs to be considered. The exclusion of rib-shortening by setting the parameter ax=1E6 was only done to compare like-with-like for verification purposes. Likewise, making the moment of inertia of the beam vary as the secant of the slope of the arch was only done to compare like-with-like for verification purposes. This can be switched on by changing the member properties from:
`i AX ax IZ iz !*ds/dx` to: `i AX ax IZ iz*ds/dx`
but normally should be switched off.

vm283 Encastré parabolic arch *cf.* Pippard & Baker
Although the change in height of an arch has a considerable effect on the percentage difference between NL-STRESS and Pippard & Baker when rib shortening is taken into account, it has little effect on the horizontal support reaction when it is excluded and the height of the arch varied. As in the finite element method, better results are obtained by using more segments to model the curvature. On a modern computer (Pentium-4 2.5 GHz), analysis using 256 segments takes about 3 seconds.

When rib-shortening is suppressed (to match the classical method) and the moment of inertia of the cross-section of the arch made to vary as the secant of the slope (to match the classical method), for 996 runs NL-STRESS agrees with the classical analysis of Pippard & Baker (1957) to an average percentage difference of 0.06968%, with a maximum of 19.571% in run 154. Investigation of this run showed that the difference was due to a very low bending moment at joint 1 for which the sign changed.

When rib-shortening is included and the moment of inertia of the cross-section of the arch is kept constant, for 996 runs NL-STRESS agrees with the classical analysis of Pippard & Baker to an average percentage difference of 65.334%, with a maximum of

72.286% in run 31. When the data for run 31 was re-run and rib-shortening was suppressed (to match the classical method) and the moment of inertia of the cross-section of the arch made to vary as the secant of the slope (to match the classical method), then the percentage difference was reduced from 72.286% to 0.1429%.

vm290 Outrigged frame *cf.* Castigliano

Using Castigliano's First Theorem for the structural analysis of frameworks is straightforward for simple structures such as outrigged frames, but difficult for more complicated frameworks. For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Castigliano's First Theorem with shear deformation suppressed was 0.0%. When shear deformation is taken into account for 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Castigliano's First Theorem is 1.4297%, the maximum percentage difference being 6% in run 665.

vm291 Braced outrigged frame *cf.* Castigliano

Using Castigliano's First Theorem for the structural analysis of frameworks is straightforward for simple structures such as outrigged frames, but difficult for more complicated frameworks. For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Castigliano's First Theorem with shear deformation suppressed was 0.0%. When shear deformation is taken into account for 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Castigliano's First Theorem is 2.1376%, the maximum percentage difference being 7.5% in run 829.

vm300 Cantilever or propped cantilever *cf.* equilibrium, compatibility & energy

In many of the verified models, checks of NL-STRESS results are by using a classical method of structural analysis; most of the classical methods ignore shear deformation and/or axial deformation. In the checks for compatibility, local & overall equilibrium and strain energy, the checks include shear and axial deformation. If the shear area is not provided in the data, or Poisson's ratio (E/2G-1) is evaluated to 1E-12, then shear deformation is excluded by NL-STRESS and excluded by the checks for compatibility, local & overall equilibrium and strain energy. If Poisson's ratio is not equal to 1E-12 and a shear area is provided, then shear deformation is included in both the analysis and the checks. For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was zero percent.

vm301 Circular arc cantilever *cf.* Pippard & Baker

For 996 runs, NL-STRESS agrees with the classical analysis of Pippard & Baker (1957) to an average percentage difference of 0.01877%, the maximum percentage difference being 2.6667% in run 996; thus vm301 becomes a Verified Model for the parameter ranges and dependencies given in the parameter table.

vm302 Circular arc bow girder *cf.* Pippard & Baker

Verification is by comparison with the classical solution given by Pippard & Baker (1957). For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Pippard & Baker was 0.1% when shear deformation was suppressed, the maximum percentage difference being 9% in run 355 due to a very small moment about X at joint 1. When the data for run 355 was run with an increase in the number of segments to 256, then the average percentage difference for run 355 was reduced from 9% to 0.5%. When shear deformation is not suppressed, for 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Pippard & Baker was 0.9767%, the maximum percentage difference being 39.5% in run 494 which was for a short girder. When the data for run 494 was run with shear deformation suppressed, the average percentage difference for run 494 was reduced from 39.5% to 0.5%.

vm310 Grillage of beams *cf.* Pilkey & Chang

The maximum deflection computed by the stiffness method generally agrees with that computed by Pilkey & Chang's Navier approach. The average percentage difference for 996 runs using data generated from the parameter table, was 0.0%. It was found that Pilkey & Chang's expression for loads given at the girder-stiffener intersection points only applies to a constant force at all such points; this was not made clear in the text, where they state *"If the concentrated forces differ for the different intersections, then more terms of the series must be employed. For such problems, it may well be easier to use a computer program"*. The single example Pilkey & Chang provided was for a constant load at all intersection points.

vm311 Grillage of beams *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.007474%, with the maximum percentage difference being 0.2222% in run 831.

vm410 Plastic analysis of cantilever *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.002259% the maximum percentage difference being 0.125% in run 3.

vm411 Plastic analysis of propped cantilever *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.0498%, the maximum percentage difference being 0.125% in run 59.

vm420 Plastic analysis of continuous beam *cf.* equilibrium, compatibility & energy
For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.0254%, the maximum percentage difference being 0.63% in run 503.

vm430 Plastic analysis of rectangular portal *cf.* equilibrium, compatibility & energy
For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.1686%, the maximum percentage difference being 15.5% in run 180. When the data for run 180 was rerun but with the number of segments/member reduced from 32 to 16, the percentage difference was reduced to 0.0%.

vm435 Plastic analysis of ridged portal *cf.* equilibrium, compatibility & energy
Morris & Plum (1988) in sub-section 12.6.3 state: "*Those programs that analyse the final collapse state directly,* i.e. *linear programming (optimisation) methods or predetermined patterns of hinges solved by equilibrium equations, will give a correct solution for the conditions analysed.*"; after many thousands of runs, it was found that this statement needs the qualification "... assuming that changes in geometry are negligible and that the framework including any plastic hinges is well within its elastic critical load".

When plastic hinges form, there are significant changes in geometry, ignoring the changes in geometry can give rise to false solutions. For a single bay ridged portal with pinned feet, symmetrically loaded on plan, when the first hinge forms at one eave, then due to elastic deformation prior to the hinge formation, both eaves have moved outwards and downwards. As further loading is added, because of strain hardening at the eave which has the plastic hinge, the opposite eave will be at its plastic limit, thus the portal will be at its critical load, having a critical load of typically 1.5 times working load, *i.e.* below the traditional load factor of 1.75.

Although suppressing changes in geometry is not recommended, to study the effect of suppressing changes in geometry, increase Young's modulus by a factor of 10 thereby ensuring that the framework is well within its elastic critical load at the formation of the first plastic hinge.

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.465%, the maximum percentage difference being 13.75% in run 813. When the data for run 813 was rerun but with the number of increments increased from 100 to 102, the percentage difference was reduced to 1.25%.

Stability analysis is non-linear, plastic analysis which incorporates stability analysis is markedly non-linear. The formation of one new plastic hinge has an effect on all the members and existing plastic hinges of a framework, it is remarkable that the equilibrium and compatibility checks give such good agreement.

vm436 Plastic analysis of multi bay ridged portal *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.065%, the maximum percentage difference being 1.125% in run 678.

vm440 Plastic analysis of multi storey frame *cf.* equilibrium, compatibility & energy

For 988 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 1.159%, the maximum percentage difference being 25.5% in run 826. When the data for run 826 was used but with the number of loading increments increased from 100 to 101, the percentage difference between NL-STRESS and the various checks was 0%. Stability analysis is non-linear, plastic analysis which incorporates stability analysis is markedly non-linear. The formation of one new plastic hinge has an effect on all the members and existing plastic hinges of a framework. Inspection of the results of run 826 showed that when the number of loading increments was 100, there were 11 unloaded plastic hinges *i.e.* plastic hinges that had ceased to be plastic due to reversal of direction of the hinge, whereas when the number of loading increments was 101, there were only 2 unloaded plastic hinges. It is recommended that any plastic analysis is carried out with the loading applied for several different increments.

vm501 Cantilever beam in space *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.2131%, the maximum percentage difference being 0.5625% in run 597.

vm510 Four legged stool space frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.0465%, the maximum percentage difference being 0.4375% in run 1.

vm520 Spiral stairs space frame *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.21611%, the maximum percentage difference being 1.1875% in run 328.

vm601 Plate with point loads *cf.* Navier double trigonometric series

From 996 runs with 32x32 elements, and a mix of element sizes and thicknesses, loads and load positions, the average percentage difference in the results was 1.578%, the maximum percentage difference being 5% for run 501 which had a single concentrated load in one corner. A more even distribution of loading reduces the percentage difference between the matrix stiffness method and Navier's solution.

vm602 Flat plate in flexure with area loading *cf.* Navier double trigonometric series

From 996 runs with shear deformation included, the average percentage difference in the results was 0.1345%, the maximum percentage difference being 6% for run 493. From 996 runs with shear deformation excluded, the average percentage difference in the results was 0.200%, the maximum percentage difference again being 6% for run 493. For the parameter ranges considered, shear strain energy is negligible. When the data for run 493 was rerun but with the number of elements increased from 16x16 to 32x32, the percentage difference was reduced from 6% to 0.4% (central deflection by NL-STRESS =-0.0940 *cf.* -0.0936 by Navier).

vm605 Floor panel with hole *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.928%, the maximum percentage difference being 5.333% in run 831. When the data for run 831 was examined, it was seen to have a central hole of dimension 8x8 elements with just 2 elements surrounding the hole. Changing the 2 element surround to a 4 element surround reduced the percentage difference to 2.778%.

vm610 Plate with free edge *cf.* finite differences & exact formulae

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.4315%, the maximum percentage difference being 1.5% in run 193.

vm618 Plate/wall in extension with hole *cf.* equilibrium, compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.670%, the maximum percentage difference being 1.333% in run 685.

vm620 Circular balcony *cf.* classical analysis & Roark

NL-STRESS gives excellent agreement with a longhand classical elastic solution based on that given by Jaeger (1964), but with corrected C4, and the formula given by Roark (1965). This problem has been treated by other authors including: Timoshenko & Woinowsky-Krieger (1959), omitted as their solution ignores Poisson's ratio. Experience with using complicated formulae taken from technical books, is that the results of using formulae from one publication should be compared with those from another. From 996 runs, for a mixture of Poisson's ratio, and PLANE GRID and SPACE FRAME, the maximum percentage difference did not exceed 2%. For structure types PLANE GRID & SPACE FRAME, results show that the use of a nearly square rectangular element gives excellent agreement with classical theory.

vm630 Spherical shell *cf.* Roark's Formulas for Stress & Strain

When shear deformation is suppressed, the NL-STRESS deflection at the pole is just 2% higher than Roark. This small difference will be due to the beam-stitching to fill in the shell at the pole. The beam elements are taken as rectangular of depth *t* and width equal to half the element size at the pole. As the beam stitches do not form a continuous plate then the NL-STRESS deflection is larger than the theoretical deflection by a few percent. If shear deformation is suppressed, results from NL-STRESS and Roark (1965) agree to an average accuracy of 0.431% for all 996 sets of data generated from the parameter table with a maximum difference of 2% in run 1. When shear deformation is not suppressed but Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for 996 sets of data generated from the parameter table =2.9317%, with the maximum difference being for run 501 when the percentage difference was 8%.

vm640 Torque on I section *cf.* analysis by Roark & Timoshenko

The end rotation for an I-section, built-in at one end with a torque applied at the other without warping restraint, computed by the stiffness method was compared with that computed by Roark (1965). When shear deformation is suppressed, the average difference in the rotation for 996 runs is 1.051%, with the largest difference being 6% in run 504. When shear deformation is not suppressed the average difference in the rotation for 996 runs is 2.024%, with the largest difference being 12% in run 532. From inspection of the results it is clear that close results are obtained when the I section has a depth 50% greater than its width. It is remarkable that a formula from Timoshenko, first published in 1930, and the modern matrix stiffness method, should give average results within 2.0% of each other.

vm641 Biaxial bending and/or torque on rectangular hollow section *cf.* Roark

Roark's formulas ignore Poisson's ratio. Generally, when Poisson's ratio is ignored, it is assumed to be equal to 1/3 as appropriate for steel, accordingly this value was used in the data. End displacements and rotation for the rectangular structural hollow section analysed using NL-STRESS & computed using Roark's formulas agree to an average percentage difference of 0.2440% with a maximum percentage difference of 2.3333% in run 633, when tested for 996 runs generated from the parameter table.

vm642 Bending and/or torque on T section *cf.* Roark

End displacements and rotation for T sections analysed using NL-STRESS & computed using Roark's formulas, agree to an average percentage difference of 2.372% with a maximum percentage difference of 5% in run 648, when tested for 996 runs using the data given in the parameter table. Roark's formula for the torsional constant of a T section ignores Poisson's ratio. Generally, when Poisson's ratio is ignored, it is assumed to be equal to 1/3 as appropriate for steel, accordingly this value was used in the data.

vm643 Bending and/or torque on channel section *cf.* Roark

End displacements and rotation for the channel section analysed using NL-STRESS & computed using Roark's formulas agree to an average percentage difference of 2.365% with a maximum percentage difference of 7.5% in run 229, when tested for 996 runs using the data given in the parameter table. Roark's formula for the torsional constant of a channel section ignores Poisson's ratio. Generally, when Poisson's ratio is ignored, it is assumed to be equal to 1/3 as appropriate for steel, accordingly this value was used in the data.

vm644 Torque on angle section *cf.* Roark

End rotation for an angle section analysed using NL-STRESS & computed using Roark's formulas agree to an average percentage difference of 1.262% with a maximum percentage difference of 3% in run 499, when tested for 996 runs using the data given in the parameter table. Roark's formula for the torsional constant of an angle section ignores Poisson's ratio. Generally, when Poisson's ratio is ignored, it is assumed to be equal to 1/3 as appropriate for steel, accordingly this value was used in the data.

vm650 Circular tank *cf.* analysis by Timoshenko & Woinowsky-Krieger

Results comparing NL-STRESS with Timoshenko & Woinowsky-Krieger agree to an average accuracy of 1.769% for all 996 sets of data generated from the parameter table with a maximum difference of 12% in run 308. Investigation of this run showed that the wall height was low with just 2 elements, when the wall height was tripled, the difference was reduced to 3.4%.

vm710 Natural frequency of beam or *cf.* flexibility & latent root

If shear deformation is suppressed, natural frequencies by NL-STRESS and Rayleigh *cf.* flexibility and latent root, agree to an average accuracy of 0.093% for all 996 sets of data generated from the parameter table. When shear deformation in not suppressed but Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for 996 sets of data generated from the parameter table is 0.29217%, with the largest individual result being for run 5 when the average difference was 7%. Investigation of this run showed that the set of data had only two joints. Even for this worst case out of 996, the natural frequency computed by NL-STRESS was within 7% of that computed by flexibility and largest latent root.

vm718 Natural frequency of built-in plate *cf.* Roark & Warburton

Natural frequencies not exceeding 36 Hertz, for square built-in plates subjected to uniformly distributed loading, computed by the matrix stiffness method using finite elements and Rayleigh's method, for a minimum of 6 x 6 elements in the slab, generally agree to within 2% with those given by Warburton's classical solution. When shear deformation is suppressed, the average percentage difference between the matrix stiffness method and Rayleigh's method and that from Warburton's formula, for 996 runs, is 0.335% with the largest percentage difference in any run being 1%. When shear deformation is not suppressed, the average percentage difference between the matrix stiffness method and Rayleigh's method and that from Warburton's formula, for 996 runs, is 1.0924% with the largest percentage difference in any run being 4% in run 784.

vm720 Natural frequency of simply supported plate *cf.* Navier, flexibility & latent root

The natural frequencies of simply supported rectangular plates subjected to uniformly distributed loading, computed by the matrix stiffness method using finite elements and Rayleigh's method, agree with the natural frequencies given by Navier, flexibility & latent root with an average percentage difference of 0.6165% for 996 sets of data generated from the parameter table, giving a maximum percentage difference of 2% in run 169. For the self check, the flexibility matrix was formed using Navier's method, the largest latent root *lambda* was found using the power iteration method, and the period found from $T=2\pi\sqrt{(lambda/g)}$ and hence the natural frequency found from 1/T.

vm802 Cantilever beam with large displacements *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, with the axial compressive load on the cantilever limited to 25% of Pe, the average percentage difference between NL-STRESS and the various checks was 0.124%, the maximum percentage difference being 1.625% in run 1.

vm810 Stability of columns with various supports *cf.* classical formulae by Euler

In verifying the correctness of structural engineering calculations, it is essential that the sets of data are practical; each set of data must be engineered. The average difference between Euler's formulae and NL-STRESS, was 2.0522% for 996 sets of data generated from the parameter table, with a maximum percentage difference of 13% in run 54. When shear deformation is taken into account, I-sections give a lower percentage difference between NL-STRESS and Euler than H-sections.

vm830 Stability of circular ring/pipe *cf.* classical formulae by Roark

In verifying the correctness of structural engineering calculations, it is essential that the sets of data are practical; each set of data must be engineered. The average percentage difference between Roark's formula and NL-STRESS, was 3.1606% for 996 sets of data generated from the parameter table with a maximum percentage difference of 10% in run 283. Halving the thickness, reduced the percentage difference to 2%.

vm850 Stability of cantilever with udl & end load *cf.* equilib., compatibility & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.040%, the maximum percentage difference being 1.1111% in run 167.

vm852 Non-linear elastic analysis of multi storey frame *cf.* equilib., compat. & energy

For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and the various checks was 0.475%, the maximum percentage difference being 0.625% in run 332.

vm950 Hanging cable with flexible platform *cf.* Pippard & Baker

Verification is by comparison of NL-STRESS with the classical solution given by Pippard & Baker (1957). For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Pippard & Baker was 0.71586%, the maximum percentage difference being 2.333% in run 167. Non-linear structural analysis involving a mixture of cables, which by their nature have little stiffness, and other members which have stiffness, is not simple; the small percentage difference between Pippard & Baker's classical analysis and NL-STRESS was expected.

vm951 Suspension bridge with three pinned stiffening girder *cf.* Pippard & Baker

Verification is by comparison of NL-STRESS with the classical solution given by Pippard & Baker (1957). For 996 runs generated from the parameter table, using 100% of the cable bending stiffness, the average percentage difference between NL-STRESS and Pippard & Baker was 1.9%, the maximum percentage difference being 10.7% in run 173. For 996 runs generated from the parameter table, using 50% of the cable bending stiffness, the average percentage difference between NL-STRESS and Pippard & Baker was 1.759%, the maximum percentage difference being 9.333% in run 173. For 996 runs generated from the parameter table, using 25% of the cable bending stiffness, the average percentage difference between NL-STRESS and Pippard & Baker =1.6%, the maximum percentage difference being 8.0% in run 173.

Non-linear structural analysis involving a mixture of cables, which by their nature have little stiffness, and girders which have considerable stiffness but behave like a mechanism due to the three pins, is not simple; the percentage difference between Pippard & Baker's classical analysis and NL-STRESS was expected. High bending moments at the centre of the cable due to the passage of heavy loads, will flex the cable and thus cause fatigue. The interplay between the various structural parameters is not simple, many hours are needed to get to grips with the structural behaviour of three-pinned suspension bridges. It is not the purpose of this research to provide a definitive work on the behaviour of three-pinned suspension bridges, but it is the purpose of this research to compare the matrix stiffness method of analysis with the classical treatment and this has been done; furthermore attention has been drawn to the high bending moments present in the middle of the cable caused by the structural discontinuity produced by having a pin at the middle of the stiffening girder. Accordingly, it is recommended that suspension bridges with three-pinned stiffening girders should use chains and not cables, for the reasons stated.

vm952 Suspension bridge with two pinned stiffening girder *cf.* Pippard & Baker
Verification is by comparison of NL-STRESS with the classical solution given by Pippard & Baker (1957). For 996 runs generated from the parameter table, the average percentage difference between NL-STRESS and Pippard & Baker was 1.2788%, the maximum percentage difference being 7.667% in run 173; when the data for run 173 was inspected, it was seen to contain very lopsided loading. The avoidance of the central pin, present in the three-pinned stiffening girder, considerably reduces the bending moment in the cable at the centre of the bridge below that of the equivalent three-pinned stiffening girder. The two-pinned stiffening girder is suitable for use with cables.

12.2 Models for structural design

The automatic generation of sets of data from the parameter table was successful in locating minor bugs in the logic of models for the design of concrete and steelwork components, see section 12.4. Models for the design of structural components, which have been devised using British Standard limit state codes, may be self checked using:

- the serviceability limit state e.g. checking that for a flanged beam designed for the ultimate limit state also satisfies deflection and crack width requirements specified in part two of BS 8110
- structural analysis e.g. running a finite element model for a structural hollow section member subjected to a variety of restraints
- Eurocodes and their national annexes, when both are available and robust.

Whereas the self checks devised as part of this research for the structural analysis of frameworks gave agreement to within a few percent, self checks for the design of structural components such as reinforced concrete flanged beams and steel rectangular hollow section members, did not give good agreement. For this reason, the results of the self check for the design of structural components were expressed by information rather than numerically.

12.3 General conclusions

A comprehensive library search did not reveal literature on the subject of *verification of engineering calculations*. Consequently the search was widened to include testing and self checking software, the literature for which was reviewed in chapter 2, did not contain any information pertinent to engineering models. It follows that the methods, procedures and theory developed and discussed in this work, are original. On a personal level, the writer has found this work: absorbing, fascinating and all consuming, good moments being:

- classifying the types of data
- avoiding verbosity by expressing the characteristics of any parameter by a number
- devising the parameter table
- using patterns to get each parameter varying with respect to every other
- developing a simple method to express dependencies between parameters
- using logic to check logic

- combining the self check with each model
- providing an escape path from and back to the parameter table for specials
- discovering that the serviceability limit state provides a good self check
- getting one procedure to work for verifying both structural analysis and design models.

It is expected that over the next decade, the methodology contained herein will be developed by others and will reveal several disparities between modern codes of practice and classical design *e.g.* shear reinforcement design for concrete beams as discussed in section 10.9.

Considerable difficulty was experienced in reconciling the various formulae and rules for the design of a structural steel section with the stresses obtained from a finite element model of the section, setion 10.6. As an example, for a beam bending about its principal axis subjected to vertical loading, BS 5950-1:2000 table 13 gives the effective length factor for no restraint on plan for the destabilising condition as 1.2 *cf.* 0.85 for both flanges fully restrained on plan. When the same member carries a horizontal load the restraints on plan, provided to assist stability, change the behaviour of the model from simply supported ends to built-in ends, giving maximum stresses at the ends rather than at mid span.

12.4 How the objectives have been met

The paramount objective of this research was **that structural calculations produced by computer should be correct**. Errors arise from several sources *e.g.* incorrect data, bugs in the logic of the software employed, inappropriate structural modelling, not understanding the design assumptions *etc.* One hundred and eight models for the structural analysis of frameworks have been developed as part of this research, all are shown pictorially in Figure 7.1 in section 7.8. Each has been demonstrated to give correct results, section 11.1 gives discussions for all, section 12.1 gives conclusions for all. In a similar manner, typical models for the design of structural steel and reinforced concrete components are presented in sections 10.6 & 10.7, section 11.2 gives their discussions, section 12.2 gives their conclusions. Ensuring that the output calculations are correct involved several subordinate objectives being met, these follow.

The second objective was to provide **a unified method** for dealing with calculations for the structural analysis of a framework and for the structural design of a component such as: beam, slab, column, foundation, pile cap, retaining wall *etc.* This has been achieved and enables engineers to have a single procedure for verifying both types of structural model.

The third objective was to **get to grips with the nature of data**. Some items of data are integer values *e.g.* a joint number, some items are real values *e.g.* coordinates, some items are dependent on other items *e.g.* Young's modulus and the modulus of rigidity are related by Poisson's ratio, some items belong to sets *e.g.* reinforcing bars of 9mm diameter are not manufactured neither are universal beams of 185 mm serial depth, some items can only vary within a fixed range *e.g.* the distance to the start of a partial UDL on a beam cannot be negative and the distance to the end must be greater than the start and not exceed the beam length, and so on. The classification of the various types of data required and the dependency of any item of data on any other has been achieved, see sections 4.1 to 4.7. The classification of the data takes the form of a PARAMETER table.

The fourth objective was to **ensure sustainability** of software and systems developed, this means that the software should be easy to maintain, which in turn means that the software should be plain text rather than computer code. This objective has been achieved for 85% of the software, the remaining 15% being computer code.

The fifth objective was **identifying and applying tools** to models to increase the robustness of the results, described in sections 3.2 to 3.15. Tools include: flow charts, comments in the data *e.g.* how floor loading is shared to a lintel, checking aids, help, worked examples, checks against known solutions, use of symmetry. Tools developed and applied as part of this research included: self checks, engineered sets of test data *via.* PARAMETER tables, avoiding information overload, benchmarking, file conversion from parametric to numeric and the cross referencing of variables. Testing all the nooks and crannies of each structural model was achieved by combining the self check for each and every model with its PARAMETER table and using a single procedure to generate a thousand sets of data automatically and run to produce percentage differences between the stiffness matrix and an independent check using a classical method of analysis *e.g.* moment distribution, or a modern method *e.g.* reciprocity, equilibrium and compatibility.

The sixth objective was to provide **a simple system** to satisfy engineers that the results of running any structural model were as expected. This objective was achieved by including a self check in each run of each model, engineers do not have to add any additional data to the computer runs, or carry out any additional work other than inspect the check.

The seventh objective was that the system developed should be capable of verifying the correctness of the structural models in the IStructE wishlist "Computer toolkit for small consultancies", The Structural Engineer, 1 February 2000, which follows: **Analysis**, 2D frame, continuous beam, subframe, foundations; **Concrete**, RC slab beam and columns to BS 8110, retaining walls; **Steel**, beams and stanchions to BS 5950, composite construction, section properties; **Masonry**, walls, pier, *etc.* to BS 5628, vertical load and wind; **Timber**, floor joists, beams *etc.*; **General**, geometric properties, loading data, material weights, construction weights, imposed loads, ability to customise/simplify calculation sheets. The types of structural analysis objectives on the IStructE wishlist have been achieved, discussion is given in section 11.1 and conclusions in section 12.1. Verifying the correctness of models for the design of structural components has been achieved for the design of typical structural steel and reinforced concrete components. The tools developed and described herein are being applied to all 780 proforma calculations, this work will take several man years.

The eighth objective, which was finding bugs in existing and new software written as part of this research was successful, examples follow:

Bug 1. Proforma calculation 370 in section 3.3 gives the first part of a flow chart which identifies by > > > a bug caused by a missing ENDIF before the start of the first procedure.

Bug 2. As described in section 10.6, following the incorporation of the parameter table into proforma sc385, the model was run for various sets of automatically generated data to test for the presence/absence of bugs in the model. One bug was found, the variable *stype* was erroneously named as *rtype* in the section of the model dealing with slender sections. This bug had remained undetected for over a year.

Bug 3. Section 11.1 describes a bug in vm430.ndf for the plastic analysis of rectangular portal *cf.* equilibrium, compatibility & energy. Such a bug could go undetected for many years, it is of paramount importance that verification should be carried out by the author/s of software; locating such a bug by an engineer who was unfamiliar with programming, or by an IT person who was unfamiliar with the engineering, would be difficult.

Bug 4. Following the incorporation of the PARAMETER table into proforma calculation sc085.pro, when panel lengths lx & ly are input such that lx/ly=2, the proforma stops reporting that lx/ly is greater than 2, now fixed.

Bug 5. Following the incorporation of the PARAMETER table into proforma calculation sc102.pro, the proforma faulted fy=500 N/mm^2 which is the new yield stress for reinforcement and says it must be 460 N/mm^2 which was the old yield stress, now fixed.

Bug 6. Following the incorporation of the PARAMETER table into proforma calculation sc118.pro, the setting +cover=15 which followed the START caused the cover to be reset without the engineer knowing, now fixed by moving +cover=15 to before the START.

Bug 7. Following the incorporation of the PARAMETER table into proforma calculation sc162.pro settings to initialise variables placed after the START caused previous data to be overwritten, moving the settings to before the start cured the bug.

Bug 8. Section 3.15 describes a bug in a proforma calculation which may not have been found by self checking, but which is likely to have been found by inspection of the cross-referencing table developed in this research.

Chapter 13

Recommendations

The general recommendations below, refer to chapters 1 to 10 in order. Specific recommendations follow the general recommendations.

- That the growing concern regarding the appropriate use of computers be actioned.
- That traditional methods used for engineering calculations be bound into modern methods to provide bedrock beneath the modern methods.
- That authors of engineering software use the tools developed in this research for verification, adapting as necessary, to their own discipline.
- That as a prerequisite to the automatic production of sets of engineered test data, the essence of each model be distilled into tabular form so that just one program may generate the test data for any model, according to the discipline.
- That a thousand sets of engineered data, providing extensive coverage, be used to test engineering models for the robustness of their logic.
- That the warnings of history be heeded and that the design of large engineering software systems take into account the need for the structure of the systems to be made modular and the software written in English rather than computer code so that the next generation of engineers is able to become expert and thereby able to carry out maintenance on the systems.
- That verified models, which are self checking, be developed as appropriate to the engineering discipline.
- That reverse engineering strategies, such as those described in chapter 8, be developed according to the engineering discipline, to highlight computed results which are in error.
- That for each engineering discipline, hundreds of benchmarks be collected and tested automatically before each new revision of the software is released.
- That models for the design of structural components incorporate self checks based on the serviceability limit states or Eurocodes.

The tools developed in this research, enable a thousand sets of data to be generated and run and the results compared with self-checks produced using alternative methods, these tools are now available. During this research, as described in chapters 11 and 12, several anomalies were found when comparing models based on current codes of practice with calculations for the serviceability limit states using elastic analysis.

13.1 Robustness of conceptual/computational models

A fictional firm entitled N&M, has software for the production of structural engineering calculations. N&M are satisfied that their validation procedures are capable of satisfying the requirements. Their concern now is with **is the process correct?** *i.e.* verification. One aspect of correctness is robustness. Bolton (2004) has given us the metaphor: in some cases there are two or more bad dancers at a party, normally everything is all right but if one bad dancer encounters the other while in a certain state, they trip over each other's feet. N&M have had experience of such situations with their software and wish to avoid the considerable losses in professional time and morale when they find bugs in their supplier's software. It is recommended that the suppliers of software to N&M should develop and use PARAMETER tables, as fully described herein, to generate a thousand sets of data for running in batch on their software systems to locate bugs hidden away in the nooks and crannies of the logic.

13.2 Self checking

MacLeod (1990) advocates checking models which are: simplified, more approximate versions of the main model, but which have adequate accuracy for checking purposes. The writer wholeheartedly supports checking models. Appendix A, includes 108 models for the structural analysis of frameworks, each incorporates a checking model together with percentage differences between the main model and the checking model at key positions, thereby providing a *self check*. Self checks are not limited to structural analysis, it is recommended that self checks be incorporated into models for the design of structural components, see sections 10.6 & 10.7, also section 11.5 which uses Eurocode 2 to provide self checks for the reinforcement for beams & columns designed in accordance with BS 8110.

13.3 Sustainability

To **ensure sustainability** of software and systems, the software should be easy to maintain, accordingly it is recommended that suppliers of software should provide their software in plain text rather than computer code.

13.4 Tools

It is recommended that the tools described in sections 3.2 to 3.15 should be developed and used by software suppliers as part of the armoury against bugs. Tools include: flow charts, comments in the data *e.g.* how floor loading is shared to a lintel, checking aids, help, worked examples, checks against known solutions, use of symmetry, avoiding information overload by concise summaries, benchmarking, file conversion from parametric to numeric, and the cross referencing of variables used in models.

13.5 Simple systems

Sole practitioners have a different perspective on structural design to that of the large consultancies. Few sole practitioners can afford the sophisticated modelling systems used in the large consultancies, furthermore they are too busy with carrying out structural steel, reinforced concrete, timber & masonry design, roads, sewerage, attic room conversions, structural surveys *etc.* to spare the time to learn how to use sophisticated modelling systems. It is recommended that software suppliers should provide **simple systems** which do not require a considerable amount of training and that means *interactive* systems *i.e.* question and answer. Question and answer with proactive help is recommended for the teaching of structural component design.

13.6 Computer toolkit for small consultancies

It is recommended that suppliers of structural software should provide systems for verifying the correctness of the structural models in the IStructE wishlist "Computer toolkit for small consultancies", The Structural Engineer, 1 February 2000.

13.7 The elastic analysis of plates and grillages

When a plate can be represented in the form of a double trigonometric series then a solution for the plate can be obtained using Navier, as described in section 2.21. Timoshenko & Woinowsky-Krieger (1959) in their section 29 entitled *Further Applications of the Navier Solution* derive their equation 133 for the deflection at any point on a simply supported rectangular plate due to a single point load anywhere on the plate. It is recommended that Navier's approach be developed for rectangular plates having any practical mixture of: free, simply supported, built-in edges and if tractable, edges of known rotational spring stiffness. Rolfe (2004) has recently applied Chebyshev polynomials to rectangular plates having edges of rotational spring stiffness. Rolfe, an email acquaintance, who has spent many years developing Chebyshev polynomials, lives in South Africa and is the same age as our Queen. Pilkey & Chang (1978) use a similar approach to Navier for a simply supported grillage of beams. It is recommended that further research be carried out to generalise Pilkey & Chang's formulae to give the deflection at any beam to beam intersection point on a grillage due to a unit load applied at any other beam to beam intersection point. This would enable a simple self check to be applied to regular grillages carrying non-uniform loading. A mathematically gifted student would be required for this research. The proposed research would enable elastic self checks to be applied to a wider range of plate problems than those presently covered by publications such as Roark (1965 & 2002).

13.8 Yield line analysis

As mentioned in the discussion in chapter 11 for verified model **vm601**, Dr Randal Wood told the writer that to avoid serviceability problems, design moments should not depart from elastic bending moments by more than 30%, if this is accepted, then it follows that a set of elastic bending moments is an important tool for checking the correctness of reinforcement chosen for concrete slabs. Jones and Wood (1967) extensively covered yield line analysis; at the time of Dr Wood's death both Dr Wood and Professor Jones were engaged on a final book on the application of yield line analysis to the design of reinforced concrete slabs *including the effects of columns*. Professor Jones, who is 75 and retired, has vast practical knowledge which should be tapped by a researcher so that it may be published academically for the benefit of other researchers. Both yield line analysis and elastic analysis are important tools for checking the correctness of reinforcement chosen for concrete slabs.

13.9 Calibration of Eurocode 2

This research has shown that comparison between two models which purport to do the same, cannot be done with a dozen sets of data, a proper comparison requires testing with a thousand sets of engineered data before meaningful conclusions may be drawn. Accordingly it is recommended that the tools and techniques developed in this research, be utilised immediately for the calibration of Eurocode 2, initially for the serviceability and ultimate limit states, eventually for: sustainability, durability, fatigue, fire, explosion, prolonged soaking etc.

13.10 Calibration of Eurocode 3

As mentioned in section 12.3, considerable difficulty was experienced in reconciling the various formulae and rules for the design of steel sections with the stresses obtained from a finite element model of the section. It is recommended that further research be carried out to compare the rules given in BS 5950-1:2000 with elastic stresses given by finite element models for equivalent structures. For the reasons given in section 11.3, it is recommended that the serviceability limit states should be given equal prominence with the ultimate limit state and that classical intuitive elastic methods of section design, should be included in teaching syllabuses. It is recommended that the tools and techniques developed in this research, should be utilised immediately for the calibration of Eurocode 3, initially for the serviceability and ultimate limit states.

13.11 Matters which affect the correctness of calculations

CIBSE (Chartered Institute of Building Services Engineers) has always produced its own *design guides* and *codes of practice* which its members use. The practising building services engineer reaches for his/her CIBSE publication first, then ASHRAE (American Society of Heating, Refrigerating & Air-conditioning Engineers) next, if still in doubt, BS and EN publications after that. Technical publications are the *backbone* of CIBSE and the service that members value the most. It is recommended that the Institution of Structural Engineers should give its members a postal ballot, choosing one from the following:

- IStructE to produce and sell its own design guides just as CIBSE does
- Eurocodes to be made mandatory
- British Standards to be made mandatory.

IStructE should support the members' decision, for it is the members who lie awake at night worrying about their calculations, it is the members who are held responsible for their designs. IStructE (2006) are to be applauded for the production of their manual for the design of concrete building structures.

13.12 Bring back the serviceability limit states

Structural failures, such as a roof collapse due to snow, which are shown on television news programmes or reported in the New Civil Engineer, are almost invariably caused at a loading level which is less than the serviceability load; overloading on scaffolding being an exception. It follows that the serviceability limit states for deflection and cracking should be considered and not shunted into a sideline as has been done with the *deflection of reinforced concrete beams and slabs*, which has been banished to part 2 of BS 8110. The serviceability limit states should include stability effects where appropriate; section design should be based on classical assumptions *e.g. plane sections remain plane*. Although it is not suggested that the complicated procedures required for the ultimate limit state should be abandoned, it is recommended that the serviceability limit states should be given equal prominence. Fortunately, Eurocode 2 states that the following may be used for structural analysis: linear elastic analysis, linear elastic analysis with limited distribution, plastic analysis. Linear elastic analysis may be carried out assuming that cross sections are uncracked using linear stress:strain relationships and assuming mean values of elastic modulus.

13.13 Structural Calculations' Centre

It will be clear from the forgoing that a considerable amount of basic research needs to be done to calibrate the national annexes to Eurocode (when these become available) against existing codes of practice, it is recommended that such work be coordinated by a Structural Calculations' Centre. Eurocode News November 2005, reports a disagreement between the British Constructional Steelwork Association (BCSA) and the British Standards Institution (BSI). BCSA wants to include its own additional information in *national annexes* against the advice of BSI which wants minimal content and full copyright. Just as claiming copyright of the structure of genes has been outlawed by international consent, copyright of design formulae should also be outlawed for such formulae are developed in the world's universities and verified in their laboratories. Basing the *structural calculations' centre* at a university would avoid the commercial pressures associated with trade organisations.

The writer's recommended provider of national annexes to Eurocodes and structural *design guides* would be the Institution of Structural Engineers in combination with trade bodies such as: the Steel Construction Institute, the British Constructional Steelwork Association, the Concrete Centre, the Brick Development Association, the Timber Research and Development Association. If the Institution of Structural Engineers is unwilling to take the lead role, then the Institution of Civil Engineers would be a good second choice.

It is recommended that the supply of an electronic service be offered by the *structural calculations' centre* (web site & email address but no office) for the verification of structural analysis software, using the methods developed in this research. Such a service would improve the correctness of calculations and at the same time forge links with commercial firms.

13.14 Neural networks

Rafiq *et al.* (2000) tell us that knowledge-based expert systems (KBESs) were used to model some of the activities of conceptual design, but owing to their restricted scope, the success of these systems was very limited. Rafiq *et al.* say that because artificial neural networks (ANNs) are capable of learning and generalising from examples and experience, they are able to produce meaningful solutions to problems, even when input data contain errors or are incomplete. This makes ANNs a powerful tool for modelling some of the activities of the conceptual stage of the design process. Jenkins (2001) identifies a number of desirable developments provided by a neural network, he tells us that the genetic algorithm has now *evolved* into a more practical, engineered orientated, style and it is worth having another look at it.

13.15 Mathematical assistants

The proof of the Four Colour Theorem, section 2.5, employed a mathematical assistant which is a new kind of computer program that a human mathematician uses in an interactive fashion, with the human providing ideas and proof steps and the computer carrying out the computation and verification.

The PARAMETER table, invented during this research, contains a complete numerical description of a set of parameters and their inter-dependencies. From the parameter table, a short program *e.g.* proforma calculation sc924.pro developed in this research, builds multiple sets of engineered data which are run in batch mode to test for bugs in a model written in a structured language *e.g.* Praxis (1990). In this research the sets of data, generated from the parameter table, are used to activate *flip-flops* in the model, the error reporting in the model and/or Praxis identifying any bugs. It is conceivable that a mathematical assistant, working from the parameter table, program syntax and a given model, called *e.g.* SMART (Structured Model Algorithmic Research Tool), could locate any bugs and report them.

Chapter 14

References

ALLEN A.H., "Reinforced Concrete Design to BS 8110 Simply Explained", 1988, Spon, London.

ALLWOOD R.J. & COOPER C.N., "The Use of an Expert System in Selecting Paint Schemes to Protect Structural Steelwork", 1990, The Structural Engineer, Volume 68, Number 7, The Institution of Structural Engineers, London.

ANDREWS A.A., OHLSSON M.C., WOHLIN S., "Deriving fault architectures from defect history", 2000, Computer Science Department, Colorado State University, Fort Collins.

ARIANE 5 "Flight 501 Failure", 1966, Report by the Inquiry Board, homepages.inf.ed.ac.uk/perdita/Book/ariane5rep.html

AutoTester, 2002 "Best Practices Guide to Automated Testing", www.autotester.com Dallas Texas.

BAKER A.R., Discussion on paper by SAWKO F. & WILLCOCK B.K., The Structural Engineer, June 1968.

BAKER E.H., KOVALEVSKY L., RISH F.L., "Structural Analysis of Shells", 1972, McGraw Hill, New York.

BASU A.H. & CHAPMAN J.C., "Large deflexion behaviour of transversely loaded rectangular orthotropic plates", Paper 6927, 1966, The Institution of Civil Engineers, London.

British Computer Society, Computer Bulletin - March 2002.
British Computer Society, "What is the SIGiST? Special Interest Group in Software Testing", September 2002 .

BCS SIGIST, 2001 "Standard for Software Component Testing", British Computer Society, www.testingstandards.co.uk

BS 7925-1:1998, Software testing - Part 1: Vocabulary.

BS 7925-2:1998, Software testing - Part 2: Software component testing.

BS ISO/IEC 12207:1995 Information technology - Software life cycle processes.

BCSA Plastic & Elastic Design of Portal Frames, Steel Frames for Multi-storey Buildings, Composite Construction and many other publications covering steelwork 1963-1971, colloquially known as the Black Books; Buildings in Composite Construction 1965, Manual on Connections for Beam and Column Construction 1969, Handbook on Structural Steelwork 1971 & 1978, Metric Practice for Structural Steelwork 1972, Manual on Connections 1985, The British Constructional Steelwork Association, London.

BENNETT J.M. PhD Thesis, Cambridge University, 1952.

BHATT P., "Structures", Longman, 1999, Harlow.

BLEICH F., "The Buckling Strength of Metal Structures", 1952, McGraw-Hill.

BOLTON M., "Pairwise Testing", 2004, www.developsense.com/reprints.html

BOOLE G., "The Mathematical Analysis of Logic", 1847, London.

BRAY T.J., "Reinforced Concrete Design", Third Edition, 1960, Chapman & Hall, London.

BROWN D.W., "Stanchion design charts", 1974, published by Constrado, Croydon.
BROWN D.W., "Reinforced concrete design charts", 1975, Fitzroy Computer Systems, Worcester Park.

BRUNDTLAND G.H. *et al.*, "Our Common Future, the final report of the World Commission on Environment and Development", 1987, which led to the United Nations Earth Summit in Rio de Janeiro, 1992.

BS 449:Part 2:1969 Deals with the use in buildings of hot rolled sections, plates and normalised tubular shapes, the steel of which of which complies with the requirements of BS 4360, The British Standards Institution.

The following British Standards comprise several parts, BSI Standards Catalogue, 2005, gives publication dates for the various parts.
BS5400 Steel, concrete and composite bridges, The British Standards Institution, London.
BS5268 Structural use of timber, The British Standards Institution, London.
BS5628 Code of practice for use of masonry, The British Standards Institution, London.
BS5750 Quality systems, Part 0 to Part 6, The British Standards Institution, London.
BS5950 Structural use of steelwork in building, The British Standards Institution, London.
BS6399 Deals with loading on buildings, The British Standards Institution, London.
BS8002 Retaining walls, The British Standards Institution, London.
BS8007 Concrete structures for retaining aqueous liquids, The British Stand. Inst., London.
BS8110 Structural use of concrete, The British Standards Institution, London.
BS8666 Specification for scheduling, dimensioning, bending and cutting of steel reinforcement for concrete, The British Standards Institution, London.

The Building Regulations 1972, Her Majesty's Stationery Office, London.

C&CA "Handbook on the Unified Code for structural concrete (CP 110:1972)", Cement and Concrete Association, London.
C&CA "Program specification for the elastic analysis of plate grids for bridge decks", 1972, Wexham Springs.

COATES R.C., COUTIE M.G. & KONG F.K., "Structural Analysis", Third Edition, 1988, Van Nostrand Reinhold, Wokingham.

COE T., "Inside the Pentium FDIV Bug", Dr. Dobb's Journal #229 April 1995. http://www.ddj.com/documents/s=992/ddj95041/

COHEN D.M., DALAL S.R., PARELIUS J., PATTON G.C., "The AETG System: An approach to Testing Based on Combinatorial Design", IEEE Transactions on Software Engineering, July 1997, Volume 23, No. 7.
www.argreenhouse.com/papers/gcp/AETGieee97.shtml

COMPUTER AIDED VERIFICATION, "Proceedings of the 17th International Conference, CAV 2005, Edinburgh, July 2005. Springer

COULL A. & STAFFORD-SMITH B., "Structural analysis of tall concrete buildings", Paper 7560, 1973, The Institution of Civil Engineers, London.

CP 3:Chapter V (1952) Code of Functional requirements of Buildings, Chapter V, Loading, The Council for Codes of Practice for Buildings, British Standards Institution, London.
CP 110:1972 Code of practice for the structural use of concrete, The British Standards Institution, London.
CP 114:Part 2:1969 The structural use of reinforced concrete in buildings, The British Standards Institution.

CROSS H., "The Method of Distribution of Fixed End Moments", Journal American Concrete Institute December 1929.
CROSS H., "Analysis of Continuous Frames by Distributing Fixed End Moments", Proceedings American Society of Civil Engineers, 1930; and Transcript American Society of Civil Engineers, 1932, volume 96 (Paper 1793).

DIJKSTRA E.W., DAHL 0.J., HOARE C.A.R., "Structured Programming", 1972, Academic Press, London and New York.

DORMAN, LONG & Company, "Handbook for Constructional Engineers containing tables relating to Steel", Middlesbrough, 1924.

DSIR (1950), "Road Research Laboratory, Road Note 4: Design of Concrete Mixes", HMSO, London.

DUSTIN E., RASHKA J., PAUL J., "Automated Software Testing: Introduction, Management, Performance", 2005, Addison-Wesley Professional, ISBN 0201432870.

EMKIN L.Z., WILL K.M., KECK D.W., THURMOND M.W., "GTICES STRUDL User's Manual Update" School of Civil Engineering, Georgia Institute of Technology, 1977.

FANTA R. & RAJLICH V., "Removing Clones from the Code", 1999, Department of Computer Science, Wayne State University, Detriot.

GASHNIG J., *et al.* "Evaluation of expert systems: issues and case studies", 1983, in "Building expert systems", Addison-Wesley, Reading, Mass.

GENNARO J.J., "Computer Methods in Solid Mechanics", 1965, Macmillan, London & New York.

GERE J.M., "Moment Distribution", 1963, Van Nostrand, New York.

GHALI A. & NEVILLE A.M., "Structural Analysis - A unified classical and matrix approach", Fourth Edition 1997, E & FN Spon, London & New York.

GOLDBERG J.E. & LEVE H.L., "Theory of Prismatic Folded Plate Structures", International Association for Bridge and Structural Engineering, Volume 17, 1957.

GOODE C.D., "Reinforced concrete beams subjected to a sustained torque", The Structural Engineer, May 1975, London.

GRASSIE J.C., "Analysis of Indeterminate Structures", First Edition, 1957, Longmans, Green & Co, London & New York.

GURP J., BRINKKEMPER S., BOSCH J., "Design preservation over subsequent releases of a software product: a case study of Baan ERP", 2005, Creative Online Development B.V., Nijmegen, The Netherlands.

HANNA P.B., HANNA A.S., PAPAGIANAKIS T.A., "A knowledge-based advisory system for flexible pavement routine maintenance", Canadian Journal of Civil Engineering, Volume 20, pp154-163, 1993.

HARRIS P.R. *et al.*, "Guidelines for the use of computers for engineering calculations", 2002, The Institution of Structural Engineers, London.

HARRISON H.B., "Computer Methods in Structural Analysis", 1973, Prentice-Hall, New Jersey.

HAWKEN P., LOVINS A.B, LOVINS L.H., "Natural Capitalism - The next Industrial Revolution", 1999, Earthscan Publications Ltd, London. (This excellent book gives 627 references - not listed here for reason of space.)

HETENYI M., "Beams on elastic foundations", 1948, University of Michigan Press.

HEYMAN J., "The development of plastic theory 1936:48: some notes for a historical sketch", in "Instability and Plastic Collapse of Steel Structures", 1983, Granada, London.

HORNE M.R. & MERCHANT W., "The Stability of Frames", 1965, Pergamon Press, Oxford & London.

HORNE M.R. & MORRIS L.J., "Plastic Design of Low-Rise Frames", 1981, Constrado.

HRENNIKOFF A., "Solution of Problems of Elasticity by the Framework Method", Journal of Applied Mechanics, December 1941, Vancouver.

HUME W., PARMAR H.S., HON K.H., SCIPATHY P., "The collapse of the Hotel New World, Singapore: a technical inquiry", The Structural Engineer, 16 March 1993.

IEEE Standard for Software Verification and Validation, IEEE Computer Society 2005, New York.

ILLSTON J.M., "Construction Materials" edited by J.M. Illston, Second Edition, 1994, Spon, London.

IStructE, "Manual for the design of concrete building structures to Eurocode 2", 2006, The Institution of Structural Engineers, London.

JAEGER L.G., "Elementary Theory of Elastic Plates", 1964, Pergamon Press, Oxford & London.

JENKINS W.M., "An introduction to neural computing for the structural engineer", 1997, The Structural Engineer, Volume 75, Number 3, The Institution of Structural Engineers, London.

JENKINS W.M., "A neural-network-based reanalysis method for integration within structural design", 2001, The Structural Engineer, Volume 79, Number 13, The Institution of Structural Engineers, London.

JENKINS W.M., "Structural design optimisation by evolution", 2004, The Structural Engineer, Volume 82, Number 13, The Institution of Structural Engineers, London.

KASPER C.J. & GODFREY M.W., "Supporting the analysis of clones in software systems: a case study", 2006, Cheriton School of Computer Science, University of Waterloo, Ontario.

KHOSHGOFTAAR T.M., ALLEN E.B., JONES W.D., HUDEPOHL J.P., "Which Software Modules have Faults which will be Discovered by Customers?", 1999, Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, Florida.

KONG F.K. & EVANS R.H., "Reinforced and Prestressed Concrete", Third Edition, 1987, Chapman & Hall, London.

KLEINLOGEL A., "Mehrstielige Ramen", Band I+II, 1956, pub. Wilhelm Ernst.

KLEINLOGEL A., "Rigid Frame Formulas", Second American Edition, Translated from the German Rahmenformeln, 1952, Frederick Ungar, New York.

KRON G., "Tensorial analysis and equivalent circuits of elastic structures", J. Franklin Institute, 283, p400, 1944.

LEE D. at al. "Manual for the design of reinforced concrete building structures", 1985, The Institution of Structural Engineers & The Institution of Civil Engineers, London.

LEVESON N., "Safeware: System Safety and Computers", Addison-Wesley, 1995.
htlp//sunnyday.mit.edu/papers/therac.pdf
Software engineering - Guidelines for the application of ISO 9001:2000.
- Software testing - Part 1: Vocabulary

LIN T.Y., "Load-balancing method for Design & Analysis of Prestressed Concrete Structures", Journal of the American Concrete Institute, June 1963.

LIVESLEY R.K., "The pattern of structural computing: 1946-66", The Structural Engineer, June 1968, London.
LIVESLEY R.K., "Some aspects of structural computing: 1943-1983. Instability and Plastic Collapse of Steel Structures", 1983, Granada, London.

MACLEOD I.A., "Symposium on tall buildings", University of Southampton, April 1966.
MACLEOD I.A. & Green D. R., "Frame idealisation for shear wall support systems", The Structural Engineer, February 1973, The Institution of Structural Engineers, London.
MACLEOD I.A., RAFIQ M.Y., "Integrated computer aided design of buildings", 1988, The Structural Engineer, Volume 66, Number 20, The Institution of Structural Engineers, London.
MACLEOD I.A., "Analytical modelling of structural systems", 1990, Ellis Horwood, New York.
MACLEOD I.A., "A strategy for the use of computers in structural engineering", The Structural Engineer, 7 Nov 1995, The Institution of Structural Engineers, London.
MACLEOD I.A., CAIRNS R., KUMAR B., "Information technology for the structural engineer", The Structural Engineer, Volume 77, Number 3, The Institution of Structural Engineers, London.
MACLEOD I.A., "Modern structural analysis - modelling process and guidance", 2005, Thomas Telford Ltd, London.
MACLEOD I.A., "Taming the finite element tiger", 2006, The Structural Engineer, Volume 84, Number 3, The Institution of Structural Engineers, London.

MAGNUS D., "Pierced Shear-walls", 1968, Concrete Publications, London.

MAHER M.L. (editor), "Expert systems for civil engineers: technology and applications", 1987, American Society of Civil Engineers, 1987.

MAHMOOD O.K.D., "An expert system for checking the design of reinforced concrete elements to BS 8110", PhD thesis, University of Leeds, 1989.

MAMDANI E.H. & GAINES B.R., "Fuzzy Reasoning and its Applications", 1981, Academic Press, London.

MARICK B., "The Craft of Software Testing", 1995, Prentice Hall, (marick@testing.com).

MAYRHAUSER A. & ZHANG N., "Automated Regression Testing Using DBT and Sleuth", 1999, Department of Computer Science, Colorado State University, Fort Collins.

McMINN S.J., "Matrices for Structural Analysis", 1962, Spon, London.

MCO,MIB Mars Climate Orbiter, Mishap Investigation Board, 1999.
ftp://ftp.hq.nasa.gov/pub/pao/reports/1999/MCO_report.pdf

Microsoft Visual C++ .NET 2003, includes Microsoft C.

MORLEY A. & INCHLEY W., "Elementary Applied Mechanics", Third Edition, 1915, Longmans, Green & Co., London.
MORLEY A., "Theory of Structures", Fifth Edition, 1948, Longmans, Green & Co., London.

MORRIS L.J. & PLUM D.R., "Structural Steelwork Design to BS 5950", Longman Scientific & Technical, 1988, Harlow, Essex.
MORRIS L.J. & RANDALL A.L., "Plastic Design", Constrado, 1975.
MORRIS L.J., "Instability and Plastic Collapse of Steel Structures", University of Manchester, 1983, Granada, London.

NCE 1995, Consultants File, published as a supplement to the New Civil Engineer, March 1995.
NCE 2005, Consultants File, published as a supplement to the New Civil Engineer, March 2005.

NIESSINK F. & VLIET H., "Software maintenance from a service perspective", 2000, Faculty of Sciences, Division of mathematics and Computer Science, Vrije Universiteit, Amsterdam, The Netherlands.

NOBLE P.W. & LEECH L.V., "Design tables for composite steel and concrete beams", 1978, Constrado, Croydon.

NOOSHIN H., "Formex Configuration Processing" Applied Science Publishers, 1984.
NOOSHIN H., "Third International Conference on Space Structures", 1984, University of Surrey.

ORCHARD D.F., "Concrete Technology, Volume 1, Properties of Materials", 1958, Contractors Record Ltd.

PILKEY W.D. & CHANG P.Y., "Modern Formulas for Statics and Dynamics, a Stress-and-Strain Approach, 1978, McGraw-Hill, New York & London.

PIPPARD A.J.S. & BAKER J.F., "The Analysis of Engineering Structures", Third Edition, 1957, Edward Arnold, London.

PRAXIS: A Computer Program for Reproducing Proforma design calculations", ALCOCK D.G. & BROWN D.W., The Computer Journal, Vol 33, No.4, 1990.

PRUKL R.P. & LOPES P.M., "Finite element analysis (FEA) tests on a simple beam - Important information for users of FEA software", SEMC Symposium, University of Cape Town, 2001.

RAFIQ M.Y., BUGMANN G., EASTERBROOK D.J., "Artificial neural networks to aid conceptual design", 2000, The Structural Engineer, Volume 78, Number 3, The Institution of Structural Engineers, London.

REDPATH, BROWN & Company, "Handbook of Structural Steelwork", Third Edition, 1924, London, Edinburgh, Manchester, Glasgow.
REDPATH DORMAN LONG Ltd. "Autofab beams, girders & columns", 1978, Middlesbrough.

REYNOLDS C.E., "Reinforced Concrete Designer's Handbook", Fifth Edition, 1957, Concrete Publications, London.
REYNOLDS C.E. & STEEDMAN J.C. "Reinforced Concrete Designer's Handbook", Tenth Edition, 1988, E & FN Spon, London & New York.

ROARK R.J., "Formulas for Stress and Strain", Fourth Edition, McGraw-Hill, New York & London, 1965.
"ROARK'S Formulas for Stress and Strain", Seventh Edition, by Young W.C. & Budynas R. G., McGraw-Hill, 2002.

ROCKEY K.C., EVANS H.R., GRIFFITHS D.W., NETHERCOT D.A., "The Finite Element Method", Second edition, 1983, Granada, London.

ROLFE J., "An alternative mathematical framework for structural theory", The Structural Engineer, 7 December 2004.

RYDER G. H., "Strength of Materials", Second Edition, 1957, Cleaver-Hume, London.

SAWKO F., "A note on the computer analysis of foundation rafts resting on plastic soils", The Structural Engineer, April 1972, London.

SCHACH S.R. & TOMER A., "A maintenance-orientated approach to software construction", 2000, Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville.

SCHWAIGHOFER J. & MICROYS H.F., "Analysis of Shear Walls Using Standard Computer Programs", Title 66-89, American Concrete Institute, 1969.

SCORDELIS A.C. & LO K.S., "Computer Analysis of Cylindrical Shells", Journal of the American Concrete Institute, May 1964.

SCOSS, Tenth report of the Standing Committee on Structural Safety, 1994, Structural Engineers Trading Organisation, London.

SEARLE J., "The Great Philosophers", in conversation with Bryan Magee about Wittgenstein's philosophy, 1987, BBC Books, London.

SEIFERT J., GARDNER P., GAY J., "Computer toolkit for small consultancies", 1 February 2000, The Structural Engineer, London.

SCI (Steel Construction Institute), "Curved beams, BCC 842, Example 6", 2001, Ascot.

"STEEL DESIGNERS' MANUAL", Third Edition, British Iron and Steel Federation & British Steel Producers' Conference, July, 1966.
"STEEL DESIGNERS' MANUAL", Fifth Edition, Steel Construction Institute, Blackwell Scientific Publications, 1992.

SOUTHWELL R.V., 1923, Philosophy Magazine, 45,193; also "Introduction to the Theory of Elasticity", 1936, Clarendon Press.

STEPHENSON J., "The Building Regulations Explained", 2000 revision, Spon, London.

"STRESS: A User's Manual, A Problem-Orientated Computer Language for Structural Engineering", S.J Fenves, R.D. Logcher, S.P. Mauch, K.F. Reinschmidt, The Department of Civil Engineering MIT, 1964.
"STRESS: A Reference Manual, A Problem-Orientated Computer Language for Structural Engineering", S.J Fenves, R.D. Logcher, S.P. Mauch, The Department of Civil Engineering MIT, 1965.

SUTHERLAND I.E., "Sketchpad, a man-machine graphical communication system", The Department of Electrical Engineering, MIT, 1963.

TERZAGHI K., "The Modulus of Subgrade Reaction", Geotechnique, Dec. 1955.

THOMSON A.I., KUMAR B., MACLEOD I.A., "Computer based structural design and conformance checking", 2000, The Structural Engineer, Volume 78, Number 3, The Institution of Structural Engineers, London.

THOMPSON S. & CUTLER R.W., "Moment distribution method for the determination of secondary stresses", Transcript, American Society of Civil Engineers, Volume 96, 1932.

TIMOSHENKO S.P., "Strength of Materials Part II", 1930, D. Van Nostrand.
TIMOSHENKO S.P. & WOINOWSKY-KRIEGER S., "Theory of Plates and Shells", Second Edition, 1959, McGraw-Hill, New York & London.

TIZANI W.M.K., "A knowledge-based system for the diagnosis of cracking in buildings", PhD thesis, 1990, Department of Civil Engineering, University of Bradford.

TOLL D.G., "A knowledge based system for interpreting geotechnical data for foundation design", in "Knowledge-based systems; current developments and future trends", Editor ANUMBA C.J. September 1996, The Institution of Structural Engineers, London.

WAGER D.M., "Expert systems and the construction industry", 1984, Construction Industry Computing Association, Cambridge.

WARBURTON G.B., "The Dynamical Behaviour of Structures", 1964, Pergamon Press, Oxford & London.

WEAVER W., "Computer Programs for Structural Analysis", 1967, Van Nostrand, Princeton, New Jersey.

WEIZSACKER E.V., LOVINS A.B, LOVINS L.H., "Factor Four - Doubling wealth, halving resource use - the new report to the Club of Rome", 1997, Earthscan Publications Ltd, London. (This book gives 155 references - not listed here for reason of space.)

WILBY C.B. & NAQVI M. N., "Deflexions of rectangular plates supported elastically", Paper 7411, Institution of Civil Engineers, 1971, London.

WOMAK J.P., JONES D.T., "Lean Thinking: Banish Waste and Create Wealth in Your Corporation", 1996, Simon and Schuster, London.

YASSAEE M.H., "Formex formulation of the space structures of S. Du Chateau", Third International Conference on Space Structures, Elsevier Applied Science Publishers, 1984.

YETTRAM A.L. & HUSAIN H.M., "Grid-Framework Method for Plates in Flexure" , Engineering Mechanics Division of the American Society of Civil Engineers, June 1965.

YETTRAM A.L. & HUSAIN H.M., "Plane-Framework Methods for Plates in Extension" , Engineering Mechanics Division of the American Society of Civil Engineers, February 1966.

ZADEH L.A., "PRUF - a meaning representational language for natural languages", 1977, Computer Science Division, University of California, CA 94720.

ZIENKIEWICZ O.C., "The Finite Element Method", Third Edition, 1977, McGraw-Hill, London & New York.

APPENDIX A - VERIFIED MODELS

Appendix A lists the kernel i.e. data followed by self check, for each of 108 models developed as part of this research to test the methods devised for verifying the correctness of each model for the structural analysis of frameworks. It is not expected that the reader will follow the algebra comprising the self check between the SOLVE and FINISH commands, but it is hoped that the engineer will:

- get the gist of what is going on from the figure and the associated parameters
- compare the volume of NL-STRESS data before the SOLVE with the self check after.

	PLANE FRAME & TRUSS VERIFIED MODELS
vm110.ndf	Simply supported beam including shear deflection.
vm112.ndf	Cantilevered beam.
vm113.ndf	Cantilevered beam subjected to many point loads.
vm114.ndf	Tapered cantilevered beam subjected to many point loads.
vm115.ndf	Cantilever with tie-down beam.
vm117.ndf	Subframe, continuous beam with columns below & above.
vm120.ndf	Continuous beam with pattern loadings.
vm122.ndf	Two member lean-to or Mansard beam.
vm123.ndf	Three member lean-to or Mansard beam.
vm124.ndf	Three member cranked beam.
vm130.ndf	Ground beam on an elastic foundation.
vm131.ndf	Ground beam on elastic piles.
vm140.ndf	Influence lines for continuous beams.
vm150.ndf	Pratt through truss.
vm153.ndf	Pratt deck truss.
vm156.ndf	Howe through truss.
vm159.ndf	Howe deck truss.
vm162.ndf	Warren through truss.
vm164.ndf	Warren through truss with verticals.
vm165.ndf	Warren deck truss.
vm168.ndf	Warren deck with verticals.
vm171.ndf	Two rafters with tie.
vm172.ndf	Two rafters, post & tie.
vm173.ndf	King post roof truss.
vm174.ndf	Roof truss, three segment rafters, Pratt internals.
vm175.ndf	Roof truss, three segment rafters, Howe internals.
vm177.ndf	Trussed rafter, or Fink roof truss.
vm178.ndf	Roof truss, three segment trussed rafter, Warren internals.
vm179.ndf	Three segment rafters, Warren internals roof truss.
vm181.ndf	Mansard truss.
vm202.ndf	Pipe tree having two branches.
vm203.ndf	Pipe tree having four branches.
vm204.ndf	Pipe tree having six branches.
vm207.ndf	One storey bent having vertical/raking piles.
vm208.ndf	Two storey bent having vertical/raking piles.
vm209.ndf	Three storey bent having vertical/raking piles.
vm210.ndf	Bent, or rectangular portal frame.
vm211.ndf	Rigid pile cap with several piles.
vm215.ndf	Ridged portal frame with pinned or fixed feet.
vm216.ndf	Nissen or Mansard portal.
vm217.ndf	Gable frame with inclined legs.
vm218.ndf	Portal frame with skew corners.
vm219.ndf	Trapezoidal frame.
vm220.ndf	Two bay ridged portal with pinned feet.
vm223.ndf	Multi bay haunched ridged portal with pinned/fixed feet.
vm225.ndf	Couple roof frame.

vm226.ndf   Couple close roof frame.
vm227.ndf   Collar-tie roof frame.
vm228.ndf   Collar-and-tie roof frame.
vm230.ndf   Attic room roof frame.
vm232.ndf   Fink roof frame.
vm233.ndf   King post roof frame.
vm234.ndf   Queen post roof frame.
vm235.ndf   Tied Mansard roof frame.
vm241.ndf   Vierendeel girder.
vm242.ndf   Vierendeel roof frame.
vm244.ndf   N/Pratt lattice portal/girder.
vm245.ndf   Howe lattice portal/girder.
vm246.ndf   Warren lattice portal/girder, end diagonals in tension.
vm247.ndf   Warren lattice portal/girder, end diagonals in compression.
vm260.ndf   Multi-storey frame, moment distribution self-check.
vm262.ndf   Multi-storey frame, equilibrium & compatibility self-check.
vm270.ndf   Pierced shear walls.
vm280.ndf   Two pinned circular arch.
vm281.ndf   Encastre circular arch.
vm282.ndf   Two pinned parabolic arch.
vm283.ndf   Encastre parabolic arch.
vm290.ndf   Outrigged frame.
vm291.ndf   Braced outrigged frame.

PLANE GRID VERIFIED MODELS

vm300.ndf   Cantilever or propped-cantilever on plan.
vm301.ndf   Circular arc cantilever on plan.
vm302.ndf   Circular arc bow girder on plan.
vm310.ndf   Grillage of beams, classical check.
vm311.ndf   Grillage of beams, modern check.

PLASTIC ANALYSIS

vm410.ndf   Plastic analysis of cantilever.
vm411.ndf   Plastic analysis of propped cantilever.
vm420.ndf   Plastic analysis of continuous beam.
vm430.ndf   Plastic analysis of rectangular portal.
vm435.ndf   Plastic analysis of ridged portal.
vm436.ndf   Plastic analysis of multi-bay ridged portal.
vm440.ndf   Plastic analysis of multi-storey frame.

FINITE ELEMENT VERIFIED MODELS

vm601.ndf   Plate with out-of-plane loading.
vm602.ndf   Flat plate in flexure with area loading.
vm605.ndf   Floor panel with hole.
vm610.ndf   Plate with free edge.
vm618.ndf   Plate/wall in extension with hole.
vm620.ndf   Circular balcony plate.
vm630.ndf   Spherical shell.
vm640.ndf   I-section subjected to torque.
vm641.ndf   RHS subjected to torque & biaxial bending.
vm642.ndf   T-section subjected to torque and bending.
vm643.ndf   Channel section subjected to torque and bending.
vm644.ndf   Angle section subjected to torque.
vm650.ndf   Circular tank.

VERIFIED MODELS FOR DYNAMICS

vm710.ndf   Natural frequency of beam.
vm718.ndf   Natural frequency of built-in plate.
vm720.ndf   Natural frequency of simply supported plate.

VERIFIED MODELS FOR STABILITY

vm802.ndf	Cantilever beam with large displacements.
vm810.ndf	Stability of columns with various supports.
vm830.ndf	Stability of circular ring/pipe.
vm850.ndf	Stability of cantilever with udl & end load.
vm852.ndf	Multi-storey frame using non-linear elastic analysis.
vm950.ndf	Hanging cable with flexible platform.
vm951.ndf	Suspension bridge with three pinned stiffening girder.
vm952.ndf	Suspension bridge with two pinned stiffening girder.

For reason of space, the verified models which follow contain only the kernel, the PARAMETER table, help notes, theory, references to the self-check method used, and conclusions from the testing have been omitted, but all are contained on the accompanying CD in SCALE accessed using the filename in the above list.

The first method devised for the parametric description of a model follows, briefly:

- znp=24 says there are 24 parameters in the model
- strings $27001 to $27024 hold the names of the 24 parameters
- strings $28001 to $28024, if present, hold expressions for minimum limiting values
- strings $29001 to $29024, if present, hold expressions for maximum limiting values
- strings $30001 to $30024, if present, hold expressions for values to be used
- zst(1) to zst(24) hold start values for the range
- zen(1) to zen(24) hold end values for the range
- zty(1) to zty(24) hold a code for the type of parameter e.g. 0=real, 1=integer.

```
znp=24
$27001=ans $27002=Mbefor $27003=V $27004=N $27005=len
$27006=h $27007=b $27008=cover $27009=d $27010=d'
$27011=M $27012=fcu $27013=fyv $27014=dia $27015=dial
$27016=supp $27017=diac $27018=ans1 $27019=ans2 $27020=dias
$27021=ans3 $27022=ans5 $27023=nbarc $27024=ans4
$29009=zva(6)-70 $29010=INT(.25*zva(9)) $30011=zva(2)
zst(1)=VEC(0,1,1,0,2,220,200,30,150,45,1,30,250,4,2,1,4,0,1,4)
zst(21)=VEC(1,1,2,1)
zen(1)=VEC(0,1000,500,50,10,2500,2500,35,2500,50,1000,60,460,10,4)
zen(16)=VEC(0,10,0,0,10,0,0,10,4) zty(1)=VEC(0)*5
zty(6)=VEC(1,1,2,1,1,0,4,2,100,100,2,100,0,2,100,2,2,1,4)
```

After coding many models using the above method in table A.1, it became tiresome maintaining the parametric models. When a model is run to generate automatically hundreds of sets of data, some of the sets contain invalid values e.g. when the depth to compression steel exceeds a sensible limit for shallow beams. Including restrictions on parameters which are dependent on other parameters frequently requires that the order of the data in the parametric description needs to be changed, and this can take considerable time and care. Another disadvantage to the first method devised for the parametric description of a model is that it is difficult to read the parametric description at a glance. Accordingly the first method of coding shown above was abandoned in favour of a tabular format. As the research progressed, several refinements were made to the tabular format which resulted in going back to the first table and revising all previously completed tables. This resulted in over 300,000 runs of NL-STRESS over an eighteen month period.

The self check for each model is contained between the SOLVE and FINISH commands, sometimes the check contains just a few lines, sometimes it is longer than the data, as in vm110.ndf, which contains two self checks. All self checks include a table comparing the results of analysis by NL-STRESS with those obtained by the self check, listing percentage differences at each point of interest, and averaging percentage differences for all points of interest.

A contrived verified model for a simply supported beam carrying a uniformly distributed load is shown in the figure below. All the verified models follow the same structure as that shown. The purpose of the figure is to show how the two main components of verification i.e. the self check and the parameter table, are included in the model. For this trivial model, only two values are included in self check.

- midspan bending moment bc=w*l^2/8
- midspan deflection dc=5*w*l^4/(384*e*b*d^3/12)

both of which will be recognised by structural engineers. The self-check compares bc & dc with the corresponding values bn & dn computed by NL-STRESS and read from the arrays file using the ARR() function e.g. bn=ARR(13,1,6) where bn is assigned the value read from array 13, row 1, column 6. For engineers who are interested in devising their own self checks, a formal description of the arrays file will be found in appendix B.

Appendix B gives a summary of the NL-STRESS language for those engineers who are unfamiliar with languages for describing problems for structural analysis. Engineers who are familiar with STRESS (1964), will be able to follow the data up to the SOLVE command. Editing the parameter values to those required, should be intuitive. When using the NL-STRESS editor, when the cursor is in a line containing parameters, the text in that line changes from white to yellow. It is not expected that an engineer will follow the self check by reference to the kernel alone, but the words of the table in the self-check will help the engineer to understand what is being checked.

```
TITLE SIMPLY SUPPORTED TIMBER BEAM
TITLE SUBJECTED TO UDL, WITH SELF
TITLE CHECK OF CENTRAL MOMENT &
TITLE DEFLECTION. ;TABULATE ALL
PRINT DATA FROM 1 ;TYPE PLANE FRAME
MADEBY DWB ;DATE 22.03.06 ;REFNO VM108
*       ┌w load/unit length     ├─b─┤
*   Y│▼▼▼▼▼┌A▼▼▼▼▼▼▼           ┌───┐ ─┬─
*    ═══════════════  ───X       │   │  d
*   _▲_1    └A     _▲_2        └───┘ ─┴─
*   ///             ooo          A-A
*    ├──────l──────┤
l=6    w=-20   ! Beam length, udl -ve.
d=0.40 b=0.25  ! Beam depth & breadth.
e=7.9E6        ! Young's modulus.
g=e/16         ! Modulus of rigidity.
#cc924.stk     ! Used for verification.
NUMBER OF JOINTS 2 ;NUMB OF MEMBERS 1
NUMB OF SUPPORTS 2 ;NUMB OF LOADINGS 1
NUMB OF SEGMENTS 2 TRACE ;JOINT COORDS
1 0 0 SUPP ;2 l 0 SUPP ;JOINT RELEASES
1 MOMENT Z ;2 MOMENT Z FORCE X
MEMBER INCIDENCES ;1 1 2
CONSTANTS E e 1 G g 1 ;MEMB PROPERTIES
1 AX b*d AY b*d*5/6 IZ b*d^3/12
LOADING CASE 1 ;MEMBER LOADS
1 FORCE Y UNIFORM W w ;SOLVE
status=1 gtot=0 nur=0 bc=-w*l^2/8
dc=5*w*l^4/(384*e*b*d^3/12)
bn=ARR(13,1,6) n=ARR(8,3,2) r=3*n-1
dn=ARR(6,r,1) d1=bn d2=bc ;#vmper.ndf
* Location   NL-STRESS  Classical %age
*            analysis   formulae  diff
* Centre BM  +bn        +bc       $ok
d1=dn d2=dc ;#vmper.ndf
* Cent defln +dn        +dc       $ok
fnm=$(vm108.stk) ;#vmres.ndf
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
FINISH

VERIFICATION DATA explaining table.

PARAMETER                   Dependency
No.Name Start End   Type   conditions
1  l    1.2   4.8   0
2  d    0.1   0.6   0      >l/20 <l/5
3  b    0.05  0.18  0      >d/2  <d*3
4  e    2E6   9E6   0
5  g   0.1E6 0.6E6  0      =e/16
6  w   -0.5   -50   0      >     <
(1000*b*d^2*1.33/l^2)
(12000*b*d^2*1.33/l^2)

NOTES ON THIS VERIFIED MODEL ...
CONCLUSIONS from the tests ...
```

Firstly comes page heading information.

A semi-colon with a space before starts a new statement.

An asterisk starts a comment line for help or a figure.

Six parameters set all the data. The engineer types new values over old.

#cc924.stk imports data from file cc924.stk for use in verification.

bc=check bend. mmt.
dc=check deflection
bn=NL-STRESS moment
dn=NL-STRESS defln.

vmper.ndf computes %age differences. vmres.ndf computes average percentages. vmtes.ndf is used for benchmarking the verified models.

From the PARAMETER table, a maximum of 996 sets of data is generated. The 6'th parameter says w must be within the range -0.5 to -50, type=0 says w is real. The > & < values follow the table and limit w to keep stresses sensible, dependent on the units used.

```
TITLE SIMPLY SUPPORTED BEAM WITH VARIOUS LOADINGS TO
TITLE MODEL UDL'S.  RESULTS COMPARE THE MATRIX STIFFNESS
TITLE METHOD WITH THOSE USING CHEBYSHEV POLYNOMIALS (CPs)
TITLE VIA J. M. ROLFE, THE STRUCT ENG. 07.12.2004 ;MADEBY DWB
DATE 24.12.04 ;PRINT DATA RESULTS COLLECTION FROM 1 ;REFNO VM110
TABULATE DISPLACEMENTS FORCES ;TYPE PLANE FRAME run=0 ;*/6
*       ┌q load/unit length, also modelled by partial loads to be
*   Y│▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼         equivalent to the UDL q, as follows:
*    ═══════════════ ───X     loading case 1  l. & r. triangular
*   _▲_1          _▲_2                     2  n partial udl's
*   ///           ooo                      3  n triangular loads &
*    ├───────l───────┤                        n trapezoidal loads.
l=6 d=.4 b=.3 n=21  ! Beam length, depth & breadth; No. of loads.
e=28E6/3 nu=.2  ! Young's mod; nu=Poisson, 1E-12 ignore shear def.
q=-20           ! Uniformly distributed load.
#cc924.stk !Import verification data from sc924.stk if available.
NUMBER OF JOINTS 2 ;NUMBER OF MEMBERS 1 ;NUMBER OF SUPPORTS 2
NUMBER OF LOADINGS 3 ;NUMBER OF SEGMENTS nsg=4 nsg TRACE
JOINT COORDINATES ;1 0 0 SUPPORT ;2 l 0 SUPPORT ;JOINT RELEASES
1 MOMENT Z ;2 MOMENT Z FORCE X ;MEMBER INCIDENCES ;1 1 2
CONSTANTS E e 1 G g=e/(2*(1+nu)) g 1 ;MEMBER PROPERTIES ;ax=d*b
ay=ax*5/6 ;IF nu=0 THEN ay=0 ;1 AX ax AY ay IZ iz=b*d^3/12 iz
LOADING CASE 1: UDL MODELLED BY LEFT & RIGHT TRIANGULAR LOADS
MEMBER LOADS ;1 FORCE Y LINEAR WA q ;1 FORCE Y LINEAR WB q
LOADING CASE 2: UDL MODELLED BY n PARTIAL UDLs
MEMBER LOADS ;i=0 x=-l/n ;:20 ;i=i+1 x=x+l/n
1 FORCE Y UNIFORM W q LA x LB x+l/n ;IF i<n GOTO 20
LOADING CASE 3: UDL MODELLED BY n TRIANGULAR & n TRAPEZOIDAL LOADS
MEMBER LOADS ;i=0 x=-l/n ;:30 ;i=i+1 x=x+l/n
1 FORCE Y LINEAR WA q/2 WB 0 LA x LB x+l/n
1 FORCE Y LINEAR WA q/2 WB q LA x LB x+l/n ;IF i<n GOTO 30
SOLVE ;status=1 gtot=0 nur=0 ;*/10 ;! Chebychev Polynomials:
x=1/2 c0=x^0 c1=2*x-1 c2=8*x^2-8*x+1 c3=32*x^3-48*x^2+18*x-1
c4=c0+20*(c2-c0)+128*(x-2*x^3+x^4) vc=-0.5*q*l*c1
mc=q*l^2*(c2-c0)/16 thec=-q*l^3*(c3-9*c1)/(192*e*iz)
delc=q*l^4*((c4-c0)-20*(c2-c0))/(3072*e*iz) x=1/4 c0=x^0 c1=2*x-1
c2=8*x^2-8*x+1 c3=32*x^3-48*x^2+18*x-1
c4=c0+20*(c2-c0)+128*(x-2*x^3+x^4) vq=-0.5*q*l*c1
mq=q*l^2*(c2-c0)/16 theq=q*l^3*(c3-9*c1)/(192*e*iz)
delq=q*l^4*((c4-c0)-20*(c2-c0))/(3072*e*iz)
* Classical theory at centre:
* Shear force    +sfc=0 by symmetry
* Bending moment +bmc=-q*l^2/8
* Rotation       +thc=0 by symmetry
* Bending defln  +dcb=5*q*l^4/(384*e*iz)
* Shear defln    +dcs=(1/8)*(6/5)*q*l^2/(ax*g)
* Total defln    +dct=dcb+dcs
* Classical theory at quarter points: +lq=l/4 +lh=l/2
* Shear force    +sfq=q*l/2-q*l/4
* Bending moment +bmq=-(q*l/2)*(l/4)+(q*l/4)*l/8
* Rotation       +thq=q*lh^3/(24*e*iz)-bmq*lq/(e*iz)
* Bend. defln +dqb=q*l*lq*(l^3-2*l*lq^2+lq^3)/(24*e*iz*l)
* Shear defln    +dqs=dcs-(1/8)*q*lh^2/(ay*g)
* Total defln    +dqt=dqb+dqs
node=ARR(8,3,2) r=3*node-1 dyq=ARR(6,r,1) r=3*node dzq=ARR(6,r,1)
node=ARR(8,4,2) r=3*node-1 dym=ARR(6,r,1) r=3*node dzm=ARR(6,r,1)
fyq=ARR(13,1,5) mzq=ARR(13,1,6) fym=ARR(13,2,5) mzm=ARR(13,2,6)
```

```
*/10 ;* The following table compares the first NL-STRESS loading
* with the longhand calculations given above. ;d1=mzm d2=bmc
* Location          NL-STRESS     Classical     Chebyshev     %age
*                   analysis      theory        Polynomials   diff
* Midspan SF        +fym          +sfc          +vc
#vmper.ndf !Compute percentage difference as text message.
* Midspan BM        +mzm          +bmc          +mc           $ok
* Midspan rotn      +dzm          +thc          +thec
d1=dym d2=dct
#vmper.ndf
* Midspan defln     +dym          +dct          +delc         $ok
d1=fyq d2=sfq
#vmper.ndf
* Qrt point SF      +fyq          +sfq          +vq           $ok
d1=mzq d2=bmq
#vmper.ndf
* Qrt point BM      +mzq          +bmq          +mq           $ok
d1=dzq d2=thq
#vmper.ndf
* Qrt point rotn    +dzq          +thq          +theq         $ok
d1=dyq d2=dqt
#vmper.ndf
* Qrt point defln   +dyq          +dqt          +delq         $ok
fnm=$(vm110.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CANTILEVER OR PROPPED CANTILEVER BEAM SUBJECTED TO
TITLE UNIFORMLY DISTRIBUTED LOADING & END VERTICAL LOAD,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 24.12.04 ;TYPE PLANE FRAME run=0 ;REFNO VM112
PRINT DATA, RESULTS, FROM 1 ;NUMBER OF JOINTS 3
NUMBER OF MEMBERS 2 ;NUMBER OF SUPPORTS 1 ;NUMBER OF LOADINGS 3
*/8
sp=3.0          ! Span of cantilever.  Span is divided into 2*nsg
nsg=16          ! No. of segments.     segments of length sp/2/nsg.
dy=0.36         ! Depth of beam.              Y +ve             │v
dz=0.3          ! Breadth of beam.            │      ┌w         ▼
e=28E6/3        ! Young's modulus.           /│▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
nu=0.2          ! Poisson's ratio.           /╪═════════════════ ──X
w=-36           ! Load/unit length.          /│1       2        3
v=-80           ! End vertical load.          ├────────sp───────┤
prop=0          ! Prop at end of cantilever, 1=yes, 0=no.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 0 0 SUPPORT ;2 sp/2 0
3 sp 0 ;IF prop=1 ;JOINT RELEASES ;3 FORCE Y -1 ;ENDIF
MEMBER INCIDENCES ;1 1 2 ;2 2 3
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU 2 RECTANGLE DY dy DZ dz ay=dz*dy*5/6 iz=dz*dy^3/12
LOADING CASE 1 ;MEMBER LOADS ;1 THRU 2 FORCE Y UNIFORM W w
JOINT LOADS ;3 FORCE Y v
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=3 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19
SOLVE ;va1=VEC(0)*2 vc1=VEC(w)*2 hjl1=VEC(0)*3 vjl1=VEC(0,0,v)
ch9=1 nn=ARR(8,3,2) nr=(nn-1)*3+2 nl10=ARR(6,nr,1)
ch10=w*sp^4/(8*e*iz)+w*sp/(2*ay*g)+v*sp^3/(3*e*iz)+v*sp/(ay*g)
IF prop=1 THEN ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
IF prop=0
* Defln at joint 3 including shear +nl10       +ch10        $ok
ENDIF
fnm=$(vm112.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CANTILEVERED BEAM SUBJECTED TO 'n' POINT LOADS OF
TITLE MAGNITUDE w(1)-w(n) AT POSITIONS a(1)-a(n)
TITLE INCLUDING CHECK FOR END DEFLECTION COMPUTED
TITLE BY THE 'UNIT LOAD METHOD'.
MADEBY DWB ;DATE 15.08.05 ;TYPE PLANE FRAME run=0 ;REFNO VM113
PRINT DATA, RESULTS, FROM 1 ;NUMBER OF JOINTS 2
NUMBER OF MEMBERS 1 ;NUMBER OF SUPPORTS 1 ;NUMBER OF LOADINGS 1
*/6
s=3.0          ! Span of cantilever.
dy=0.36        ! Depth of beam.               Y
dz=0.3         ! Breadth of beam.             ├───a(n)─┤p(n)
e=28E6/3       ! Young's modulus.            /│        ▼
nu=0.2         ! Poisson's ratio.            /╞═══════════════ ──X
n=3            ! Number of point loads.      /├───────s───────┤
p1=VEC(-30,-20,-10)       ! Magnitude of vertical point loads.
a1=VEC(1.5,2,3.0)         ! Distances to vertical point loads.
#cc924.stk !Import verification data from cc924.stk if available.
JOINT COORDINATES ;1 0 0 SUPPORT ;2 s 0 ;MEMBER INCIDENCES ;1 1 2
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 AX dy*dz AY ay=dy*dz*5/6 ay IZ iz=dz*dy^3/12 iz
LOADING CASE 1 ;MEMBER LOADS ;i=0 ;:5 ;i=i+1
1 FORCE Y CONCENTRATED P p(i) L a(i) ;IF i<n GOTO 5
SOLVE ;status=1 gtot=0 nur=0 np=0 del=0 ;:10 ;np=np+1 p=p(np)
a=a(np) del=del+p*(a^2*s/2-a^3/6)/(e*iz) ;IF np<n GOTO 10
nn=ARR(8,2,2) nr=3*(nn-1)+2 ndel=ARR(6,nr,1) d1=ndel d2=del
#vmper.ndf !Compute percentage difference & any message.
*
*                               NL-STRESS    Unit Load      %age
*                                            Method         diff.
* Deflection at end of beam     +ndel        +del           $ok
fnm=$(vm113.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE TAPERED CANTILEVERED BEAM SUBJECTED TO 'n' POINT
TITLE LOADS OF MAGNITUDE w(1)-w(n) AT POSITIONS a(1)-a(n)
TITLE INCLUDING CHECK FOR DEFLECTION BY THE 'UNIT LOAD
TITLE METHOD' USING DIRECT SUMMATION. ;MADEBY DWB ;DATE 19.08.05
PRINT DATA, RESULTS, FROM 1 ;METHOD ELASTIC ;REFNO VM114
TYPE PLANE FRAME run=0 ;*/6
s=4.0    nsg=32    ! Span of cantilever; No. of segments.
dy=0.5   dyl=0.2   ! Depths of beam at start & end.
dz=0.5   dzl=0.2   ! Breadth at start & end.  ├───a(n)─┤p(n)
e=28E6/3           ! Young's modulus.        /│Y       ▼
nu=0.2             ! Poisson's ratio.        /╪═══════════════ ──X
n=3                ! Number of point loads.  /├───────s───────┤
p1=VEC(-30,-20,-10)           ! Magnitude of vertical point loads.
a1=VEC(1.7,2.0,4.0)           ! Distances to vertical point loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=nsg+1 nj ;NUMBER OF MEMBERS nsg
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;1 THRU nj X 0 Y 0 XL s
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nsg,nj
JOINT RELEASES ;1 FORCE X -1 Y -1 MOMENT Z -1
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nsg RECTANGLE DY dy DZ dz DYL dyl DZL dzl
LOADING CASE 1 ;MEMBER LOADS ;i=0 ;:2 ;i=i+1 p=p(i) a=a(i)
IF a=s THEN a=a-1E-12 ;mn=INT(nsg*a/s)+1 l=a-(mn-1)*s/nsg
mn FORCE Y CONCENTRATED P p L l ;IF i<n GOTO 2
SOLVE ;status=1 gtot=0 nur=0 np=0 del=0 ;:10 ;np=np+1 p=p(np)
a=a(np) dx=s/nsg x=-dx/2 i=0 ;:20 ;i=i+1 x=x+dx
dy'=dy+(dyl-dy)/(nsg-1)*(i-1) dz'=dz+(dzl-dz)/(nsg-1)*(i-1)
iz=dz'*dy'^3/12 ;IF a<x+dx/2 ;dx'=a-x+dx/2 x=x-dx/2+dx'/2 dx=dx'
ENDIF ;del=del+p*(a-x)*(s-x)*dx/(e*iz) ;IF i*s/nsg<a GOTO 20
IF np<n GOTO 10
nn=ARR(8,nj,2) nr=3*(nn-1)+2 ndel=ARR(6,nr,1) d1=ndel d2=del
#vmper.ndf !Compute percentage difference & any message.
*
*                              NL-STRESS    Unit Load      %age
*                                           Method         diff.
* Deflection at end of beam     +ndel        +del           $ok
fnm=$(vm114.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CANTILEVER WITH TIE-DOWN SPAN SUBJECTED TO UDL'S
TITLE ON SPAN & CANTILEVER & CONCENTRATED LOAD ON END OF
TITLE CANTILEVER, INCLUDING A CHECK OF RESULTS BY ROARK'S
TITLE FORMULAS FOR STRESS & STRAIN. ;MADEBY DWB ;DATE 26.08.05
METHOD ELASTIC ;PRINT DATA DIAGRAMS RESULTS FROM 1 ;REFNO VM115
TYPE PLANE FRAME run=0 ;NUMBER OF JOINTS 3 ;NUMBER OF MEMBERS 2
NUMBER OF SUPPORTS 2   ;NUMBER OF LOADINGS 1 ;*/5
*        Point load │p  Load/unit length w┐    Load/unit lng. cw─┐
*                   ▼   ▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼               ▼▼▼▼▼
* ══════════════════   ═══════════════════   ══════════════════
* ▲            ▲        ▲1            ▲2   3   ▲     (1)     ▲ (2)
* ├──────l─────┼──c─┤      Joint numbers.       Member numbers.
l=4.8       c=1.2         ! Lengths of tie-down span & cantilever.
d=0.4       b=0.3         ! Depth & breadth of beam.
e=28E6/3    nu=0.3        ! Young's modulus & Poisson's ratio.
p=-20   w=-12.7   cw=-2   ! Point load & udl's, down is negative.
#cc924.stk !Import set of parameters if available from cc924.stk.
JOINT COORDINATES ;1 0 0 SUPPORT ;2 l 0 SUPPORT ;3 l+c 0
JOINT RELEASES ;1 MOMENT Z ;2 FORCE X MOMENT Z ;MEMBER INCIDENCES
1 THRU 2 CHAIN 1,2,3 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU 2 RECTANGLE DY d DZ b iz=b*d^3/12
LOADING CASE 1 ;JOINT LOADS ;3 FORCE Y p
MEMBER LOADS ;1 THRU 2 FORCE Y UNIFORM W w ;2 FORCE Y UNIFORM W cw
SOLVE ;status=1 gtot=0 nur=0 n=c/l cbm=-(p*c+w*c^2/2+cw*c^2/2)
ya=p*c^2*(c+l)/(3*e*iz) yb=w*l^3*c*(3*n^3+4*n^2-1)/(24*e*iz)
dy3=ya+yb+cw*l*c^3*(4+3*c/l)/(24*e*iz) nn=ARR(8,3,2) nr=3*(nn-1)+2
ndy3=ARR(6,nr,1) ncbm=ARR(13,2,3) d1=ndy3 d2=dy3
#vmper.ndf !Compute percentage difference & any message.
*
*                               NL-STRESS    Roark          %age
*                                                           diff.
* Deflection Y at joint 3       +ndy3        +dy3           $ok
d1=ncbm d2=cbm
#vmper.ndf !Compute percentage difference & any message.
* Cantilever bending moment     +ncbm        +cbm
fnm=$(vm115.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf !Used for benchmarking.
< ;FINISH
```

```
TITLE SUBFRAMES - BEAMS WITH OPTIONAL COLUMNS BELOW AND
TITLE ABOVE HAVING 'nb' SPANS SUBJECTED TO LOAD PATTERNS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
TABULATE FORCES REACTIONS  ;TYPE PLANE FRAME ;REFNO VM117
PRINT DATA, RESULTS FROM 1 ;DATE 09.09.05 ;*/8
*                                             ___
*      ║             ║             ║            │
*   Y  ║   beam 1    ║  beam 2 etc.║          ha
*      ║═════════════║═════════════║     —X  ─┼─   s(1)=s1, s(2)=s2... i.e.
*      ║             ║             ║          hb   brackets may be omitted.
*      ║             ║             ║          _│_
*      ├──────s(1)───┼──────s(2)───┼───▶▶          ha=0 if columns above
*      1             2             3               beam are absent.
nb=3  nc=4  hb=3  ha=3 ! No. of spans & columns; storey heights.
s1=VEC(4.8,4.5,6.2)                  ! Spans in order left to right.
ax1=VEC(0.36)*nb                     ! Areas for 'nb' beams.
ay1=VEC(0.3)*nb                      ! Shear areas for 'nb' beams.
iz1=VEC(0.0243)*nb                   ! Moments of inertia for beams.
abx1=VEC(0.16)*nc                    ! Areas of columns below.
aby1=VEC(0.13333)*nc                 ! Shear area for columns below.
ibz1=VEC(0.0243)*nc                  ! Inertias for columns below.
fb1=VEC(1)*4       ! End fixities below (1=fixed, 0=pin, -1=free).
aax1=VEC(0.16)*nc                    ! Areas of columns above.
aay1=VEC(0.13333)*nc                 ! Shear area for columns above.
iaz1=VEC(0.0243)*nc                  ! Inertias for columns above.
fa1=VEC(1)*4       ! End fixities above (1=fixed, 0=pin, -1=free).
dlf=1.4  llf=1.6          ! Dead and imposed load factors.
d1=VEC(-12)*3             !  'nb' unfactored dead loads l. to r.
l1=VEC(-24)*3             !  'nb' unfactored imposed loads l. to r.
e=28E6/3  nu=0.2  nsg=10 ! Young's mod., Poisson's rat., No. segs.
#cc924.stk !Import set of parameters if available from cc924.stk.
nj=(nb+1)*2 ;IF ha<>0 THEN nj=nj+nb+1 ;nm=2*nb+1
IF ha<>0 THEN nm=nm+nb+1 ;ns=nb+1 ;IF ha<>0 THEN ns=ns+nb+1
NUMBER OF JOINTS nj ;NUMBER OF MEMBERS nm ;NUMBER OF SUPPORTS ns
nlc=4 ;IF nb>2 THEN nlc=7 ;NUMBER OF LOADINGS nlc
NUMBER OF SEGMENTS nsg TRACE
JOINT COORDINATES ;i=0 s(0)=0 x=0 ;REPEAT ;x=x+s(i) i=i+1 i x 0
i+nb+1 x -hb SUPPORT ;IF ha<>0 THEN i+2*(nb+1) x ha SUPPORT
UNTIL i=nb+1 ;ENDREPEAT ;JOINT RELEASES ;i=0 ;REPEAT ;i=i+1
IF fb(i)=0 THEN i+nb+1 MOMENT Z
IF fb(i)=-1 THEN i+nb+1 FORCE X Y MOMENT Z ;IF ha<>0
IF fa(i)=1 THEN i+2*(nb+1) FORCE Y
IF fa(i)=0 THEN i+2*(nb+1) FORCE Y MOMENT Z
IF fa(i)=-1 THEN i+2*(nb+1) FORCE X Y MOMENT Z ;ENDIF
UNTIL i=nb+1 ;ENDREPEAT
MEMBER INCIDENCES ;1 THRU nb RANGE 1,2 nb,nb+1
nb+1 THRU 2*nb+1 RANGE nb+2,1 2*nb+2,nb+1
IF ha<>0 THEN 2*nb+2 THRU nm RANGE 1,2*nb+3 nb+1,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;i=0 ;REPEAT ;i=i+1 i AX ax(i) AY ay(i) IZ iz(i)
UNTIL i=nb ;ENDREPEAT ;i=0 ;REPEAT ;i=i+1
i+nb AX abx(i) AY aby(i) IZ ibz(i) ;UNTIL i=nb+1 ;ENDREPEAT
IF ha<>0 ;i=0 ;REPEAT ;i=i+1
i+2*nb+1 AX aax(i) AY aay(i) IZ iaz(i) ;UNTIL i=nb+1 ;ENDREPEAT
ENDIF
LOADING CASE 1: UNFACTORED DL+IMPOSED ;MEMBER LOADS ;i=0 ;REPEAT
i=i+1 ;i FORCE Y UNIFORM W d(i)+l(i) ;UNTIL i=nb ;ENDREPEAT
LOADING CASE 2: IMPOSED PATTERN  1 0 1 0... ;MEMBER LOADS ;i=0
```

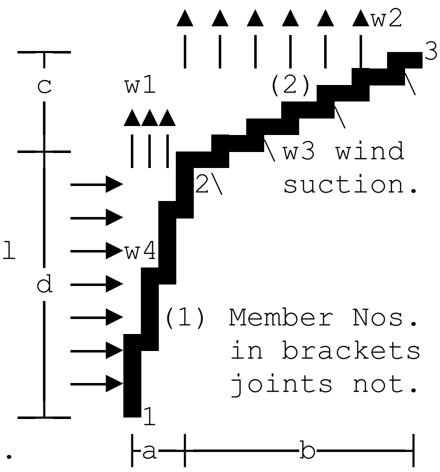
```
REPEAT ;i=i+1
IF i/2<>INT(i/2) THEN i FORCE Y UNIFORM W d(i)*dlf+l(i)*llf
IF i/2=INT(i/2) THEN i FORCE Y UNIFORM W d(i) ;UNTIL i=nb
ENDREPEAT
LOADING CASE 3: IMPOSED PATTERN  0 1 0 1... ;MEMBER LOADS ;i=0
REPEAT ;i=i+1
IF i/2=INT(i/2) THEN i FORCE Y UNIFORM W d(i)*dlf+l(i)*llf
IF i/2<>INT(i/2) THEN i FORCE Y UNIFORM W d(i) ;UNTIL i=nb
ENDREPEAT
LOADING CASE 4: IMPOSED PATTERN  1 1 1 1... ;MEMBER LOADS ;i=0
REPEAT ;i=i+1 ;i FORCE Y UNIFORM W d(i)*dlf+l(i)*llf ;UNTIL i=nb
ENDREPEAT ;IF nlc<=4 GOTO 450
LOADING CASE 5: IMPOSED PATTERN  1 1 0 1... ;MEMBER LOADS ;i=0
REPEAT ;i=i+1
IF i/3<>INT(i/3) THEN i FORCE Y UNIFORM W d(i)*dlf+l(i)*llf
IF i/3=INT(i/3) THEN i FORCE Y UNIFORM W d(i) ;UNTIL i=nb
ENDREPEAT
LOADING CASE 6: IMPOSED PATTERN  0 1 1 0... ;MEMBER LOADS ;i=0
REPEAT ;i=i+1 ;IF (i+1)/3<>INT((i+1)/3)
i FORCE Y UNIFORM W d(i)*dlf+l(i)*llf ;ENDIF
IF (i+1)/3=INT((i+1)/3) THEN i FORCE Y UNIFORM W d(i)
UNTIL i=nb ;ENDREPEAT
LOADING CASE 7: IMPOSED PATTERN  1 0 1 1... ;MEMBER LOADS ;i=0
REPEAT ;i=i+1 ;IF (i+2)/3<>INT((i+2)/3)
i FORCE Y UNIFORM W d(i)*dlf+l(i)*llf ;ENDIF
IF (i+2)/3=INT((i+2)/3) THEN i FORCE Y UNIFORM W d(i)
UNTIL i=nb ;ENDREPEAT ;:450 ;SOLVE ;va1=VEC(0)*nb i=0 ;:500 ;i=i+1
vc(i)=d(i)+l(i) ;IF i<nb GOTO 500
vjl1=VEC(0)*nj hjl1=VEC(0)*nj ch9=0 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm117.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CONTINUOUS BEAM OF nm SPANS, WITH CHARACTERISTIC
TITLE UNIFORM, CONCENTRATED, LINEAR, DEAD & IMPOSED LOADS.
TITLE RESULTS COMPARE SUPPORT BENDING MOMENTS BY STIFFNESS
TITLE METHOD WITH THOSE BY MOMENT DISTRIBUTION. ;MADEBY DWB
DATE 20.10.04 ;TYPE PLANE FRAME run=0 ;REFNO VM120
*/4
* Y│  member 1      member 2 etc.
*  ═══════════════════════════════ ──X     s(1)=s1, s(2)=s2... i.e.
*  ├─────s(1)─────┼─────s(2)─────┼──►►     brackets may be omitted.
*  1             2             3
nm=4                         ! No. of spans.
s1=VEC(4.8,4.8,4.8,4.8)      ! Spans left to right.
lsc=1 rsc=1                  ! Left/right suppts: 0=pin,1=fixed.
e=28E6/3 nu=.2   ! Young's; nu=Poisson, 1E-12 to ignore shear def.
ax1=VEC(.36)*nm              ! Areas for nm spans.
ay1=VEC(.3)*nm               ! Shear areas for nm spans.
iz1=VEC(.0243)*nm            ! Moments of inertia.
dlf=1.4 llf=1.6              ! Dead & live (imposed) load factors.
du1=VEC(-10,-10,-10,-10)     ! Unfactored Dead UDLs, down is -ve.
lu1=VEC(-12,-12,-12,-12)     ! Unfactored Live UDLs, down is -ve.
nac=4 nal=4                  ! nac,nal =No. of unfactored additional
IF nac>0                     ! concentrated & linear loads on spans.
nc(1)=VEC(1,2,3,4)           ! Spans for i=1—nac concentrated loads.
cs(1)=VEC(2.4)*nac           ! ·       ▲cd(i) DL     ┌i'th load occurs
cd(1)=VEC(-10)*nac           ! ├─cs(i)─┤cl(i) LL     │on span nc(i)
cl(1)=VEC(-12)*nac           ! ╪═══════╧═════════════╪
ENDIF ;IF nal>0
nl(1)=VEC(1,2,3,4)           ! Span numbers for i=1—nal linear loads.
ls(1)=VEC(0)*nal             ! ├───────le(i)───┼End dist. for load i.
le(1)=VEC(4.8)*nal           ! ├─ls(i)─┤     ▲▲┐
sd(1)=VEC(-10)*nal           ! ·  sd(i) ┌▲▲    │ed(i) End DL load i.
sl(1)=VEC(-12)*nal           ! ·  sl(i) │      │el(i) End LL .......
ed(1)=VEC(-10)*nal           ! ╪═════════════════════╪
el(1)=VEC(-12)*nal           !    └i'th load occurs on span nl(i)
ENDIF
#cc924.stk !Import verification data from cc924.stk if available.
TABULATE FORCES REACTIONS ;PRINT DATA, RESULTS, DIAGRAMS FROM 1
NUMBER OF JOINTS nj=nm+1 nj ;NUMBER OF MEMBERS nm nsg=1 ;nlc=4
NUMBER OF SUPPORTS nj ;IF nm>2 THEN nlc=7
NUMBER OF LOADINGS nlc ;JOINT COORDINATES ;1 0 0 SUPPORT i=0 d=0
REPEAT ;i=i+1 d=d+s(i) ;i+1 d 0 SUPPORT ;UNTIL i=nm ;ENDREPEAT
JOINT RELEASES ;2 THRU nj FORCE X
IF lsc=0 AND rsc=0 THEN 1 THRU nj MOMENT Z
IF lsc=1 AND rsc=0 THEN 2 THRU nj MOMENT Z
IF lsc=0 AND rsc=1 THEN 1 THRU nj-1 MOMENT Z
IF lsc=1 AND rsc=1 AND nm>1 THEN 2 THRU nj-1 MOMENT Z
MEMBER INCIDENCES ;1 THRU nm RANGE 1,2 nj-1,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES ;i=0
REPEAT ;i=i+1 ;IF nu=1E-12 THEN ay(i)=0
i AX ax(i) AY ay(i) IZ iz(i) ;UNTIL i=nm ;ENDREPEAT
LOADING 1: UNFACTORED DL+IMPOSED ;MEMBER LOADS ;i=0 ;:100 ;i=i+1
i FORCE Y UNIFORM W du(i)+lu(i) ;IF nac>0 ;j=0 ;:120 ;j=j+1
p=cd(j)+cl(j) ;IF nc(j)=i THEN i FORCE Y CONCENTRATED P p L cs(j)
IF j<nac GOTO 120 ;ENDIF ;IF nal>0 ;j=0
:140 ;j=j+1 wa=sd(j)+sl(j) wb=ed(j)+el(j)
IF nl(j)=i THEN i FORCE Y LINEAR WA wa WB wb LA ls(j) LB le(j)
IF j<nal GOTO 140 ;ENDIF ;IF i<nm GOTO 100
LOADING 2: IMPOSED PATTERN 1 0 1 0... ;MEMBER LOADS
i=0 ;:200 ;i=i+1 w=du(i)*dlf+lu(i)*llf
IF i/2<>INT(i/2) THEN i FORCE Y UNIFORM W w
IF i/2=INT(i/2) THEN i FORCE Y UNIFORM W du(i) ;IF nac>0 ;j=0
:220 ;j=j+1 p=cd(j)*dlf+cl(j)*llf
```

```
IF nc(j)=i AND i/2<>INT(i/2) THEN i FORCE Y CONC P p L cs(j)
IF nc(j)=i AND i/2=INT(i/2) THEN i FORCE Y CONC P cd(j) L cs(j)
IF j<nac GOTO 220 ;ENDIF ;IF nal<=0 GOTO 280 ;j=0
:240 ;j=j+1 wa=sd(j)*dlf+sl(j)*llf wb=ed(j)*dlf+el(j)*llf
IF nl(j)=i AND i/2<>INT(i/2)
i FORCE Y LINEAR WA wa WB wb LA ls(j) LB le(j) ;ENDIF
IF nl(j)=i AND i/2=INT(i/2)
i FORCE Y LINEAR WA sd(j) WB ed(j) LA ls(j) LB le(j)
ENDIF ;IF j<nal GOTO 240 ;:280 ;IF i<nm GOTO 200
LOADING 3: IMPOSED PATTERN 0 1 0 1... ;MEMBER LOADS ;i=0
:300 ;i=i+1 w=du(i)*dlf+lu(i)*llf
IF i/2=INT(i/2) THEN i FORCE Y UNIFORM W w
IF i/2<>INT(i/2) THEN i FORCE Y UNIFORM W du(i)
IF nac>0 ;j=0 ;:320 ;j=j+1 p=cd(j)*dlf+cl(j)*llf
IF nc(j)=i AND i/2=INT(i/2) THEN i FORCE Y CONC P p L cs(j)
IF nc(j)=i AND i/2<>INT(i/2) THEN i FORCE Y CONC P cd(j) L cs(j)
IF j<nac GOTO 320 ;ENDIF ;IF nal<=0 GOTO 380 ;j=0
:340 ;j=j+1 wa=sd(j)*dlf+sl(j)*llf wb=ed(j)*dlf+el(j)*llf
IF nl(j)=i AND i/2=INT(i/2)
i FORCE Y LINEAR WA wa WB wb LA ls(j) LB le(j)
ENDIF ;IF nl(j)=i AND i/2<>INT(i/2)
i FORCE Y LINEAR WA sd(j) WB ed(j) LA ls(j) LB le(j) ;ENDIF
IF j<nal GOTO 340 ;:380 ;IF i<nm GOTO 300
LOADING 4: IMPOSED PATTERN 1 1 1 1... ;MEMBER LOADS ;i=0
:400 ;i=i+1 w=du(i)*dlf+lu(i)*llf ;i FORCE Y UNIFORM W w
IF nac>0 ;j=0 ;:420 ;j=j+1 p=cd(j)*dlf+cl(j)*llf
IF nc(j)=i THEN i FORCE Y CONCENTRATED P p L cs(j)
IF j<nac GOTO 420 ;ENDIF ;IF nal<=0 GOTO 480 ;j=0
:440 ;j=j+1 wa=sd(j)*dlf+sl(j)*llf wb=ed(j)*dlf+el(j)*llf
IF nl(j)=i THEN i FORCE Y LINEAR WA wa WB wb LA ls(j) LB le(j)
IF j<nal GOTO 440 ;:480 ;IF i<nm GOTO 400 ;IF nlc<5 GOTO 800
LOADING 5: IMPOSED PATTERN 1 1 0 1... ;MEMBER LOADS ;i=0
:500 ;i=i+1 w=du(i)*dlf+lu(i)*llf
IF i/3<>INT(i/3) THEN i FORCE Y UNIFORM W w
IF i/3=INT(i/3) THEN i FORCE Y UNIFORM W du(i)
IF nac>0 ;j=0 ;:520 ;j=j+1 p=cd(j)*dlf+cl(j)*llf
IF nc(j)=i AND i/3<>INT(i/3) THEN i FORCE Y CONC P p L cs(j)
IF nc(j)=i AND i/3=INT(i/3) THEN i FORCE Y CONC P cd(j) L cs(j)
IF j<nac GOTO 520 ;ENDIF ;IF nal<=0 GOTO 580 ;j=0
:540 ;j=j+1 wa=sd(j)*dlf+sl(j)*llf wb=ed(j)*dlf+el(j)*llf
IF nl(j)=i AND i/3<>INT(i/3)
i FORCE Y LINEAR WA wa WB wb LA ls(j) LB le(j) ;ENDIF
IF nl(j)=i AND i/3=INT(i/3)
i FORCE Y LINEAR WA sd(j) WB ed(j) LA ls(j) LB le(j) ;ENDIF
IF j<nal GOTO 540 ;:580 ;IF i<nm GOTO 500
LOADING 6: IMPOSED PATTERN 0 1 1 0... ;MEMBER LOADS ;i=0
:600 ;i=i+1 w=du(i)*dlf+lu(i)*llf
IF (i+2)/3<>INT((i+2)/3) THEN i FORCE Y UNIFORM W w
IF (i+2)/3=INT((i+2)/3) THEN i FORCE Y UNIFORM W du(i)
IF nac<=0 GOTO 630 ;j=0 ;:620 ;j=j+1 p=cd(j)*dlf+cl(j)*llf
IF nc(j)=i AND (i+2)/3<>INT((i+2)/3) ;i FORCE Y CONC P p L cs(j)
ENDIF ;IF nc(j)=i AND (i+2)/3=INT((i+2)/3)
i FORCE Y CONC P cd(j) L cs(j) ;ENDIF ;IF j<nac GOTO 620 ;:630
IF nal<=0 GOTO 680 ;j=0
:640 ;j=j+1 wa=sd(j)*dlf+sl(j)*llf wb=ed(j)*dlf+el(j)*llf
IF nl(j)=i AND (i+2)/3<>INT((i+2)/3)
i FORCE Y LINEAR WA wa WB wb LA ls(j) LB le(j) ;ENDIF
IF nl(j)=i AND (i+2)/3=INT((i+2)/3)
i FORCE Y LINEAR WA sd(j) WB ed(j) LA ls(j) LB le(j) ;ENDIF
IF j<nal GOTO 640 ;:680 ;IF i<nm GOTO 600
LOADING 7: IMPOSED PATTERN 1 0 1 1... ;MEMBER LOADS ;i=0
:700 ;i=i+1 w=du(i)*dlf+lu(i)*llf
```

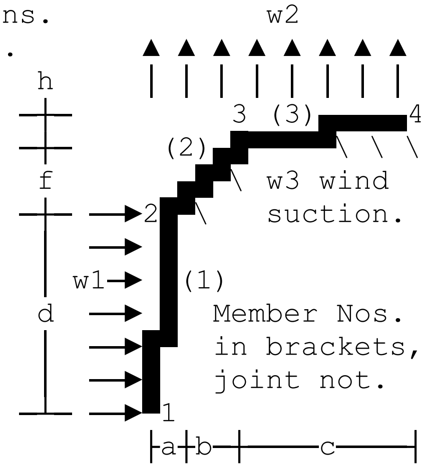
```
IF (i+1)/3<>INT((i+1)/3) THEN i FORCE Y UNIFORM W w
IF (i+1)/3=INT((i+1)/3) THEN i FORCE Y UNIFORM W du(i)
IF nac<=0 GOTO 730 ;j=0 ;:720 ;j=j+1 p=cd(j)*dlf+cl(j)*llf
IF nc(j)=i AND (i+1)/3<>INT((i+1)/3) ;i FORCE Y CONC P p L cs(j)
ENDIF ;IF nc(j)=i AND (i+1)/3=INT((i+1)/3)
i FORCE Y CONC P cd(j) L cs(j) ;ENDIF ;IF j<nac GOTO 720 ;:730
IF nal<=0 GOTO 780 ;j=0
:740 ;j=j+1 wa=sd(j)*dlf+sl(j)*llf wb=ed(j)*dlf+el(j)*llf
IF nl(j)=i AND (i+1)/3<>INT((i+1)/3)
i FORCE Y LINEAR WA wa WB wb LA ls(j) LB le(j) ;ENDIF
IF nl(j)=i AND (i+1)/3=INT((i+1)/3)
i FORCE Y LINEAR WA sd(j) WB ed(j) LA ls(j) LB le(j) ;ENDIF
IF j<nal GOTO 740 ;:780 ;IF i<nm GOTO 700 ;:800 ;SOLVE
status=1 gtot=0 nur=0 ;*/10
* Loading  Member No.   Bending moment  Bending moment   %age
* number   & location   by stiffness    by Hardy Cross   diff.
lcase=0 ;:820 ;lcase=lcase+1 n=INT((nm-1)/2)+1 i=INT((nm-1)/3)+1
IF lcase=1 THEN p1=VEC(0,0)*n ;IF lcase=2 THEN p1=VEC(1,0)*n
IF lcase=3 THEN p1=VEC(0,1)*n ;IF lcase=4 THEN p1=VEC(1,1)*n
IF lcase=5 THEN p1=VEC(1,1,0)*i ;IF lcase=6 THEN p1=VEC(0,1,1)*i
IF lcase=7 THEN p1=VEC(1,0,1)*i ;n=0 ;:830 ;n=n+1
gks(n)=-du(n)*s(n)^2/12 gke(n)=-gks(n)
qks(n)=-lu(n)*s(n)^2/12 qke(n)=-qks(n) i=0 ;:840 ;i=i+1
IF nac>0 AND nc(i)=n
gks(n)=gks(n)-cd(i)*cs(i)*(s(n)-cs(i))^2/s(n)^2
gke(n)=gke(n)+cd(i)*cs(i)^2*(s(n)-cs(i))/s(n)^2
qks(n)=qks(n)-cl(i)*cs(i)*(s(n)-cs(i))^2/s(n)^2
qke(n)=qke(n)+cl(i)*cs(i)^2*(s(n)-cs(i))/s(n)^2 ;ENDIF
IF i<nac GOTO 840 ;i=0 ;:850 ;i=i+1 ;IF nal>0 AND nl(i)=n
l=s(n) a=ls(i) b=le(i) c=b-a d=l-a/2-b/2 w=-1*c
k=24*d^3/l-6*b*c^2/l+3*c^3/l ma=w*(k+4*c^2-24*d^2)/(24*l)
mb=-w*(k+2*c^2-48*d^2+24*d*l)/(24*l) d=l-a/3-2*b/3
w=-1*c/2 k=d^3/l+51*c^3/(810*l)-c^2*b/(6*l)
ma'=w*(k+c^2/9-d^2)/l mb'=-w*(k+c^2/18-2*d^2+d*l)/l
gks(n)=gks(n)-ma*sd(i)-ma'*(ed(i)-sd(i))
gke(n)=gke(n)+mb*sd(i)+mb'*(ed(i)-sd(i))
qks(n)=qks(n)-ma*sl(i)-ma'*(el(i)-sl(i))
qke(n)=qke(n)+mb*sl(i)+mb'*(el(i)-sl(i)) ;ENDIF
IF i<nal GOTO 850 ;IF p(n)=1 THEN fdl=dlf fll=llf
IF p(n)<>1 THEN fdl=1 fll=0 ;IF lcase=1 THEN fdl=1 fll=1
ms(n)=fdl*gks(n)+fll*qks(n) me(n)=fdl*gke(n)+fll*qke(n)
IF n<nm GOTO 830 ;n=0 ;:860 ;n=n+1 ;IF n=1 THEN dfs(n)=1-lsc
IF n>1 THEN dfs(n)=1-dfe(n-1) ;IF n=nm THEN dfe(n)=1-rsc
IF n<nm THEN dfe(n)=(iz(n)/s(n))/(iz(n)/s(n)+iz(n+1)/s(n+1))
IF n<nm GOTO 860 ;ncyc=0 ;:880 ;ncyc=ncyc+1 n=0 ;:890 ;n=n+1
IF n=1 AND dfs(1)=1 THEN me(1)=me(1)-ms(1)/2 ms(1)=0 ;IF n>1
m=me(n-1)+ms(n) ml=-m*dfe(n-1) mr=-m*dfs(n) me(n-1)=me(n-1)+ml
ms(n-1)=ms(n-1)+ml/2 ms(n)=ms(n)+mr me(n)=me(n)+mr/2 ;ENDIF
IF n=nm AND dfe(nm)=1 THEN ms(nm)=ms(nm)-me(nm)/2 me(nm)=0
IF n<nm GOTO 890 ;IF ncyc<32 GOTO 880 ;n=0
end=$(start) ;:900 ;n=n+1 i=(lcase-1)*nm+n ml=ARR(13,i,3)
ms=ms(n) ok=$( ) d1=ml d2=ms
#vmper.ndf !Compute percentage difference & any message.
* +lcase   +n  start    +ml            +ms             $ok
IF n<nm GOTO 900 ;ml=ARR(13,i,6) ms=me(n) d1=ml d2=ms
#vmper.ndf
* +lcase   +n  end      +ml            +ms             $ok
IF lcase<nlc GOTO 820 ;fnm=$(vm120.stk) ;#vmres.ndf !Conclude.
mjn=1 lcn=1 tot=1 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE TWO MEMBER LEAN-TO OR MANSARD BEAM
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM122
DATE 19.09.05 ;TYPE PLANE FRAME run=0 ;NUMBER OF JOINTS 3
NUMBER OF MEMBERS 2 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/14
a=0.5     ! Horizontal dimension a.
b=3       ! Horizontal dimension b.
c=2       ! Vertical dimension c.
d=4       ! Vertical dimension d.
f2=1      ! Fixity at 2, 1=fixed, 0=pin.
f1=2      ! Fixity at base: 1=pin & roller
          ! 2=pin, 3=built-in.
f3=2      ! Fixity at top: 1=pin & vertical
          ! roller, 2=pin, 3=built-in.
dy=0.4       ! Depth of section.
dz=0.2       ! Breadth into media.
t=0.0        ! Thickness, zero if solid.
e=8E6        ! Young's modulus.
```




```
nu=7         ! Poisson's ratio, see notes.
nsg=12       ! No. of segments, typ. 8-12 for strain energy check.
w1=-6  w2=-8 ! Plan PROJECTED udl on members 1 & 2, positive up.
w3=2.75      ! Suction normal to member 2, positive up.
w4=3.4       ! Wind on side PROJECTED load, positive to right.
p2h=6.6 p2v=-9.7 ! Horiz. & vert. loads at 2, positive right & up.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 0 0 SUPPORT
2 a d ;3 a+b c+d SUPPORT ;JOINT RELEASES ;IF f1=2 THEN 1 MOMENT Z
IF f1=1 THEN 1 FORCE X MOMENT Z ;IF f3=2 THEN 3 MOMENT Z
IF f3=1 THEN 3 FORCE Y MOMENT Z ;MEMBER INCIDENCES
1 THRU 2 CHAIN 1,2,3 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU 2 RECTANGLE DY dy DZ dz T t
IF f2=0 ;MEMBER RELEASES ;1 END MOMENT Z ;ENDIF
LOADING CASE 1 ;MEMBER LOADS
IF w1<>0 AND a<>0 THEN 1 FORCE Y PROJECTED UNIFORM W w1
IF w2<>0 THEN 2 FORCE Y PROJECTED UNIFORM W w2
IF w3<>0 THEN 2 FORCE Y UNIFORM W w3
IF w4<>0 THEN 1 FORCE X PROJECTED UNIFORM W w4
IF p2h<>0 OR p2v<>0 ;JOINT LOADS ;2 FORCE X p2h FORCE Y p2v ;ENDIF
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=3 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
l=SQR(a^2+d^2) cx=a/l cy=d/l vay=w1*cx*cy vcy=w1*cx^2
vax=w4*cy*cx vcx=-w4*cy^2 va1=vax+vay vc1=vcx+vcy l=SQR(b^2+c^2)
cx=b/l cy=c/l vay=w2*cx*cy vcy=w2*cx^2 va2=vay vc2=w3+vcy
vjl1=VEC(0,p2v,0) hjl1=VEC(0,p2h,0) ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm122.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE THREE MEMBER LEAN-TO OR MANSARD BEAM
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM123
DATE 19.09.05 ;TYPE PLANE FRAME run=0 ;NUMBER OF JOINTS 4
NUMBER OF MEMBERS 3 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/14
a=0.5  b=1.5  c=2    ! Horizontal dimensions.
d=2.4  f=1.5  h=0.5  ! Vertical dimensions.
nsg=12   ! No. of segments, typ. 8-12.
f2=1     ! Fixity at 2, 1=fixed, 0=pin.
f3=1     ! Fixity at 3, 1=fixed, 0=pin.
f1=2     ! Fixity at base: 1=pin & roller
         ! 2=pin, 3=built-in.
f4=2     ! Fixity at top: 1=pin & vertical
         ! roller, 2=pin, 3=built-in.
dy=0.4       ! Depth of section.
dz=0.2       ! Breadth into media.
t=0.0        ! Thickness, zero if solid.
e=8E6        ! Young's modulus.
nu=7         ! Poisson's ratio, see notes.
```




```
w1=3.4       ! Wind on member 1 PROJECTED load, positive to right.
w2=-6        ! Plan PROJECTED udl on all members, positive up.
w3=2.75      ! Suction normal to members 2 & 3, positive up.
p2h=6.6 p2v=-9.7 ! Horiz. & vert. loads at 2, positive right & up.
p3h=6.6 p3v=-9.7 ! Horiz. & vert. loads at 3, positive right & up.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 0 0 SUPPORT ;2 a d
3 a+b d+f ;4 a+b+c d+f+h SUPPORT ;JOINT RELEASES
IF f1=1 THEN 1 FORCE X MOMENT Z ;IF f1=2 THEN 1 MOMENT Z
IF f4=1 THEN 4 FORCE Y MOMENT Z ;IF f4=2 THEN 4 MOMENT Z
MEMBER INCIDENCES ;1 THRU 3 CHAIN 1,2,3,4
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER RELEASES
IF f2=0 THEN 1 END MOMENT Z ;IF f3=0 THEN 2 END MOMENT Z
MEMBER PROPERTIES ;1 THRU 3 RECTANGLE DY dy DZ dz T t
LOADING CASE 1 ;MEMBER LOADS
IF w1<>0 THEN 1 FORCE X PROJECTED UNIFORM W w1
IF w2<>0 THEN 1 THRU 3 FORCE Y PROJECTED UNIFORM W w2
IF w3<>0 THEN 2 THRU 3  FORCE Y UNIFORM W w3 ;JOINT LOADS
IF p2h<>0 OR p2v<>0 THEN 2 FORCE X p2h FORCE Y p2v
IF p3h<>0 OR p3v<>0 THEN 3 FORCE X p3h FORCE Y p3v
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=4 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
l=SQR(a^2+d^2) cx=a/l cy=d/l vay=w2*cx*cy vcy=w2*cx^2
vax=w1*cy*cx vcx=-w1*cy^2 va1=vax+vay vc1=vcx+vcy
l=SQR(b^2+f^2) cx=b/l cy=f/l va2=w2*cx*cy vc2=w3+w2*cx^2
l=SQR(c^2+h^2) cx=c/l cy=h/l va3=w2*cx*cy vc3=w3+w2*cx^2
vjl1=VEC(0,p2v,p3v,0) hjl1=VEC(0,p2h,p3h,0) ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm123.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE THREE MEMBER CRANKED BEAM WITH VARIOUS LOADINGS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM124
DATE 20.09.05 ;TYPE PLANE FRAME run=0 ;NUMBER OF JOINTS 4
NUMBER OF MEMBERS 3 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/11
*           w1        w2         w3          Downward UDLs are -ve.
*       ▼▼▼▼▼▼▼▼▼▼ ▼▼▼▼▼▼▼▼▼▼ ▼▼▼▼▼▼▼▼▼▼▼▼
*                                 (3)
*   ─┬─                       ████████████4  Fixities at 1 & 4.
*    │                      ██               1=pin & roller, 2=pin,
*    │                (2) ██                 3=built-in.
*    d                  ██                   Member numbers are in
*    │                ██                     brackets joint numbers
*    │      (1)     ██                       are not.
*   ─┴─ 1███████████2
*       ├────a────┼────b────┼────c────┤
f1=1  f4=2   ! Fixity at 1 & 4: 1=pin & roller, 2=pin, 3=built-in.
a=1.2  b=1.5  c=1.5   ! Horizontal dimensions
d=1.4     nsg=12      ! Vertical dimension, number of segments.
dy=0.4    dz=0.2      ! Depth and breadth of section.
t=0.0                 ! Thickness of section, zero if solid.
e=205E6   nu=7        ! Young's mod., Poisson's ratio, see notes.
w1=-2  w2=-2.8  w3=-2 ! Plan PROJECTED loads per unit length
p2h=6.6 p2v=-9.7      ! Horiz. & vert. loads at 2, +ve right & up.
p3h=6.6 p3v=-9.7      ! Horiz. & vert. loads at 3, +ve right & up.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 0 0 SUPPORT ;2 a 0
3 a+b d ;4 a+b+c d SUPPORT ;JOINT RELEASES
IF f1=1 THEN 1 FORCE X MOMENT Z ;IF f1=2 THEN 1 MOMENT Z
IF f4=1 THEN 4 FORCE Y MOMENT Z ;IF f4=2 THEN 4 MOMENT Z
MEMBER INCIDENCES ;1 THRU 3 CHAIN 1,2,3,4
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU 3 RECTANGLE DY dy DZ dz T t
LOADING CASE 1 ;MEMBER LOADS ;1 FORCE Y PROJECTED UNIFORM W w1
2 FORCE Y PROJECTED UNIFORM W w2
3 FORCE Y PROJECTED UNIFORM W w3 ;JOINT LOADS
IF p2h<>0 OR p2v<>0 THEN 2 FORCE X p2h FORCE Y p2v
IF p3h<>0 OR p3v<>0 THEN 3 FORCE X p3h FORCE Y p3v
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=4 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=0 vc1=w1 l=SQR(b^2+d^2) cx=b/l cy=d/l va2=w2*cx*cy vc2=w2*cx^2
va3=0 vc3=w3 vjl1=VEC(0,p2v,p3v,0) hjl1=VEC(0,p2h,p3h,0) ch9=1
ch10=0 ;#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm124.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE GROUND BEAM ON ELASTIC FOUNDATION SUBJECTED
TITLE TO TRAIN OF MOVING LOADS; CHECKING OF RESULTS
TITLE AGAINST HETENYI'S CLASSICAL SOLUTION, UNIVERSITY
TITLE OF MICHIGAN PRESS; AND BY BUILDERS' ARITHMETIC.
MADEBY DWB ;DATE 08.02.05 ;REFNO VM130
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE FRAME run=0 ;*/7
* ├──────────────────────a(nl)──────────────────┤
* ├────────a(2)──────┤            nl=number     │p(nl)
* ├─a(1)─┤p(1)       │p(2)        of loads      │
*        ▼           ▼                          ▼
* ██████████████████████████████████████████████████████
* //\\//\\//\\//\\//\\//\\//\\//\\//\\//\\//\\//\\//\\/
* ├──────────────────────────l──────────────────────────┤
l=11 br=1 d=0.5  ! Length breadth & depth of ground beam.
nj=45            ! No. of joints each modelling lumped stiffness.
k'=1E3           ! Modulus of subgrade reaction kN/m3.
nl=3             ! Number of load points in train of loads.
nls=3   spc=1    ! nls=number of loadings; spc=step spacing.
e=28E6/3 nu=.2   ! Young's mod; nu=Poisson, 0 to ignore shear def.
a(1)=VEC(1.5,5.5,9.5)         ! Distances to loads.
p(1)=VEC(-500)*nl             ! Magnitude of loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj           ;NUMBER OF MEMBERS nm=nj-1 nm nsg=1
NUMBER OF SUPPORTS nj         ;NUMBER OF LOADINGS nls
JOINT COORDINATES ;crs=l/(nj-1) ;1 THRU nj X 0 Y 0 XL l SUPPORT
JOINT RELEASES ;1 FORCE Y k'*br*crs/2 MOMENT Z 0
nj FORCE X 0 FORCE Y k'*br*crs/2 MOMENT Z 0
IF nj>2 THEN 2 THRU nj-1 FORCE X 0 Y k'*br*crs MOMENT Z 0
MEMBER INCIDENCES ;1 THRU nm RANGE 1,2 nj-1,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
ay=br*d*5/6 ;IF nu=1E-12 THEN ay=0
1 THRU nm AX br*d AY ay IZ br*d^3/12
n=0 ;REPEAT ;n=n+1 m=0 som=-crs
LOADING TRAIN OF MOVING POINT LOADS (DOWN IS NEGATIVE)    POSITION
MEMBER LOADS ;:200 ;som=som+crs lc=0 m=m+1 !som is start of memb.
:300 ;lc=lc+1 x=a(lc)+(n-1)*spc
IF x=0 THEN x=1E-6  !Fix to get left end load on the span.
eom=som+crs l'=x-som ;IF m=nm THEN eom=l l'=l-som
IF x>som AND x<=eom THEN m FORCE Y CONCENTR P p(lc) L l'
IF lc<nl GOTO 300 ;IF m<nm GOTO 200 ;UNTIL n=nls ;ENDREPEAT ;SOLVE
* BUILDERS' ARITHMETIC.  Find: centre of loading, pressures at
* each end (P/A±M.y/I), then moments & shears at load positions.
* Load  Dist.   Pressure      Shear to      Shear to      Moment @
*  No.  m       kN/m2         left kN       right kN      load kNm
lcase=0 inc=-spc ;:310 ;lcase=lcase+1 inc=inc+spc
i=0 p=0 m=0 a(0)=0 ;:320 ;i=i+1 ai=a(i)+inc p=p-p(i) m=m-p(i)*ai
IF i<nl AND a(i+1)+inc<=l GOTO 320 ;IF m/p<l/3 OR m/p>2*l/3
* Centre of loading outside middle third; lift off will occur.
STOP ;ENDIF ;ec=l/2-m/p pa=p/(l*br) p(0)=pa+p*ec/(br*l^2/6)
pr=pa-p*ec/(br*l^2/6) pd=pr-p(0)
i=0 df(0)=0 mc(0)=0 ma(0)=0 sr(0)=0
:340 ;i=i+1 ai=a(i)+inc ps(i)=p(0)+pd*ai/l
u(i)=0.5*(p(0)+ps(i))*ai*br sl(i)=u(i)-df(i-1) df(i)=df(i-1)-p(i)
sr(i)=u(i)-df(i) mc(i)=p(0)*br*ai^2/2+(ps(i)-p(0))*br*ai^2/6
ma(i)=ma(i-1)+df(i-1)*(a(i)-a(i-1)) m(i)=mc(i)-ma(i)
* +i  +ai      +ps(i)        +sl(i)        +sr(i)        +m(i)
IF i<nl AND a(i+1)+inc<=l GOTO 340 ;IF lcase<nls GOTO 310
```

```
*/10 ;* M.HETENYI in 'Beams on elastic foundations', University
* of Michigan Press, gives formulae for deflection, slope,
* bending moment & shearing force for a concentrated force
* at an arbitrary point as used below.
* Position    Loading     Deflection      Deflection        %age
* ref. No.    case No.    by NL-STRESS    by Hetenyi        diff.
status=1 gtot=0 nur=0 com=a(1) ext=a(1)+(nls-1)*spc
st=e*br*d^3/12 k=k'*br s(1)=0 i=1 ;:400 ;i=i+1 s(i)=a(i)-a(1)
IF i<nl GOTO 400 ;i=0 ax(0)=-crs ;:500 ;i=i+1 ax(i)=ax(i-1)+crs
IF i<nj GOTO 500 ;lam=(k/(4.*st))^0.25 sinhLL=SNH(lam*l)
fp=1./(sinhLL^2-(SIN(lam*l))^2) c=0 ;:600 ;c=c+1 x=ax(c)
sinhlx=SNH(lam*x) coshlx=CSH(lam*x)
sp1=coshlx*SIN(lam*x)+sinhlx*COS(lam*x)
fip1=coshlx*SIN(lam*x)-sinhlx*COS(lam*x)
z=l-x sinhlz=SNH(lam*z) coshlz=CSH(lam*z)
sp2=coshlz*SIN(lam*z)+sinhlz*COS(lam*z)
fip2=coshlz*SIN(lam*z)-sinhlz*COS(lam*z) d'=com-spc lcase=0
:700 ;d'=d'+spc lcase=lcase+1 toty=0 i=0
:800 ;i=i+1 p=p(i) a=d'+s(i) ;IF a>l GOTO 799
b=l-a sinhla=SNH(lam*a) sinhlb=SNH(lam*b) coshla=CSH(lam*a)
coshlb=CSH(lam*b) tp1=SIN(lam*a)*coshlb-COS(lam*a)*sinhlb
fp1=sinhla*COS(lam*b)-coshla*SIN(lam*b)
sip1=sinhLL*COS(lam*a)*coshlb-SIN(lam*l)*coshla*COS(lam*b)
sip2=sinhLL*COS(lam*b)*coshla-SIN(lam*l)*coshlb*COS(lam*a)
IF x<a
y=(2.*coshlx*COS(lam*x)*sip1+sp1*(sinhLL*tp1+SIN(lam*l)*fp1))
y=(p*lam*fp/k)*y ;ENDIF
IF x=a ;sinh2a=SNH(2.*lam*a) sinh2b=SNH(2.*lam*b)
c1=(coshlb^2+(COS(lam*b))^2)*(sinhla*coshla-SIN(lam*a)*COS(lam*a))
c2=(coshla^2+(COS(lam*a))^2)*(sinhlb*coshlb-SIN(lam*b)*COS(lam*b))
y=(p*lam*fp/k)*(c1+c2) ;ENDIF ;IF x>a
c1=2.*coshlz*COS(lam*z)*sip2+sp2*(sinhLL*(-fp1)+SIN(lam*l)*(-tp1))
y=(p*lam*fp/k)*c1 ;ENDIF ;toty=toty+y ;:799
IF i<nl GOTO 800 ;row=3*c-1 dy=ARR(6,row,lcase) d1=dy d2=toty
#vmper.ndf !Compute percentage difference & any message.
*   +c          +lcase    +dy              +toty              $ok
IF lcase<nls GOTO 700 ;IF c<nj GOTO 600 ;fnm=$(vm130.stk)
#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE GROUND BEAM ON ELASTIC PILES SUBJECTED
TITLE TO TRAIN OF MOVING LOADS; CHECKING OF RESULTS
TITLE AGAINST FLEXIBILITY, USING SIMPSON'S RULE & FOX'S
TITLE METHOD FOR INVERSION; AND BY ENGINEERS' ARITHMETIC.
MADEBY DWB ;DATE 08.02.05 ;METHOD ELASTIC JOINTS ;REFNO VM131
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE FRAME run=0 ;*/7
*  ├──────────────────────a(nl)───────────────────┤
*  ├────────a(2)──────┤          nl=number     │p(nl)
*  ├─a(1)─┤p(1)       │p(2)      of loads      │
*         ▼           ▼                         ▼
*  █████████████████████████████████████████████████
*  ▲1         ▲2         ▲3                   nk▲   └joint
*  ├───b(1)───┤          │   nk=number of piles │     nk+1
*  ├─────────b(2)────────┤
*  ├────────────────────────b(nk-1)─────────────┼───┼can
br=1 d=0.5 can=1.524  ! Beam breadth, depth, cantilever len (m).
nk=13      nl=13      ! Number of piles & load points in train.
e=4709.856E6 nu=0.2   ! Young's modulus kN/m2 & Poisson's ratio.
nls=2      si=1       ! No. of load positions; spacing of steps.
* Distances from left end (m) to each pile/joint in order follow.
b0=VEC(0,1.8288,4.8768,7.9248,10.973,14.021,17.069,20.117) !
b8=VEC(23.165,26.213,29.261,32.309,35.357,38.405) !
* Distances from left end (m) to each load in order follow.
a1=VEC(0,2.6426,6.1874,8.8636,12.421,15.088,18.645,21.488) !
a9=VEC(25.045,27.712,31.266,33.924,37.49) !
* Load magnitudes (kN) from leftmost load in order follow.
p1=VEC(-619.76,-2146.2,-1607.2,-1534.5,-1534.5,-1474.7,-1381) !
p8=VEC(-1476.7,-1444.8,-1534.5,-1534.5,-1474.7,-2052.6) !
ks(1)=VEC(158689.85)*nk  ! Vertical stiffnesses, left-►right kN/m.
kr(1)=VEC(0)*nk     ! Rotational stiffnesses, left-►right kNm/rad.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=nk+1 nj   ;NUMBER OF MEMBERS nm=nj-1 nm nsg=1
NUMBER OF SUPPORTS nk         ;NUMBER OF LOADINGS nls
JOINT COORDINATES  ;i=0 b(nk)=b(nk-1)+can ;:10 ;i=i+1
i b(i-1) 0 SUPPORT ;IF i<nk GOTO 10 ;nj b(nk) 0
JOINT RELEASES ;i=0 ;:15 ;i=i+1 i FORCE Y ks(i) MOMENT Z kr(i)
IF i<nk GOTO 15 ;MEMBER INCIDENCES ;1 THRU nm RANGE 1,2 nk,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
ay=br*d*5/6 ;IF nu=1E-12 THEN ay=0
1 THRU nm AX br*d AY ay IZ iz=br*d^3/12 iz n=0 ;REPEAT ;n=n+1 m=0
LOADING TRAIN OF MOVING POINT LOADS (DOWN IS NEGATIVE)    POSITION
MEMBER LOADS ;:200 ;lc=0 m=m+1 ;:300 ;lc=lc+1 x=a(lc)+(n-1)*si
IF x=0 THEN x=1E-6  !Fix to get left end load on the span.
IF x>b(m-1) AND x<=b(m) THEN m FORCE Y CONCENTR P p(lc) L x-b(m-1)
IF lc<nl GOTO 300 ;IF m<nm GOTO 200 ;UNTIL n=nls ;ENDREPEAT ;SOLVE
l=b(nk-1) lo=l+can nl'=0 i=0 ;:197 ;i=i+1 ;IF a(i)<lo THEN nl'=i
IF i<nl GOTO 197 ;nl=nl' status=1 i=0 ;:198 ;i=i+1 k(i)=1/ks(i)
IF i<nk GOTO 198 ;st=e*iz ni=nls c=0 ns=nk-2 va=ns vb=ns vc=ns
vd=ns ;IF nl>ns THEN vd=nl ;! Dimension larger of ns or nl.
u=6 nl'=ns t(1)=VEC(1)*nl' i=0 ;:210 ;i=i+1 a'(i)=b(i)
IF i<ns GOTO 210 ;i=0 ;:211 ;i=i+1 p'(i)=t(i) ;IF i<nl' GOTO 211
! Compute terms of the flexibility matrix for the ground beam with
! all internal supports released, Simpson's rule. ;iret=2159 ;:212
div=6*st*l^2 i=0 ;:3200 ;i=i+1 j=0 ;:3210 ;j=j+1 vd(i,j)=0
imi=INT(a'(j)/l) ini=INT((a'(j)-l)/l) ;IF imi+ini<0 GOTO 3030
IF imi+ini=0 GOTO 3040 ;IF imi+ini>0 GOTO 3180
:3030 ;vd(i,j)=p'(j)*a'(j)*b(i)*(l-b(i))*(2.0*l-b(i))*l/div
GOTO 3190 ;:3040 ;IF a'(j)-b(i)<0 GOTO 3050
IF a'(j)-b(i)=0 GOTO 3100 ;IF a'(j)-b(i)>0 GOTO 3150 ;:3050
c1=p'(j)*a'(j)*(l-b(i))/div c2=2.0*(a'(j)^2)*(l-a'(j))
c3=2.0*b(i)*(l-b(i))^2 c4=a'(j)*(l-a'(j))*(b(i)-a'(j))
c5=b(i)+a'(j) c6=(b(i)-a'(j))*(2.0*l-a'(j)-b(i))
```

```
c7=b(i)*(l-b(i))*(b(i)-a'(j)) vd(i,j)=c1*(c2+c3+c4+c5*c6+c7)
GOTO 3190 ;:3100 ;vd(i,j)=p'(j)*(a'(j)^2)*((l-a'(j))^2)/(3.0*st*l)
GOTO 3190 ;:3150 ;c1=p'(j)*b(i)*(l-a'(j))/div
c2=2.0*(b(i)^2)*(l-b(i)) c3=2.0*a'(j)*(l-a'(j))^2
c4=b(i)*(l-b(i))*(a'(j)-b(i)) c5=a'(j)+b(i)
c6=(a'(j)-b(i))*(2.0*l-b(i)-a'(j)) c7=a'(j)*(l-a'(j))*(a'(j)-b(i))
vd(i,j)=c1*(c2+c3+c4+c5*c6+c7) ;GOTO 3190 ;:3180
vd(i,j)=p'(j)*b(i)*(l-a'(j))*(l-b(i))*(l+b(i))*l/div ;:3190
IF j<nl' GOTO 3210 ;IF i<ns GOTO 3200 !End Simpson's. ;GOTO iret
:2159 ;i=0 ;:2160 ;i=i+1 j=0 ;:2165 ;j=j+1 vb(i,j)=0 b'=b(i)*b(j)
vb(i,j)=(k(1)/l^2)*(l^2-b(i)*l-b(j)*l+b')+(k(nk)*b'/l^2)
IF j<ns GOTO 2165 ;IF i<ns GOTO 2160 ;i=0 ;:2170 ;i=i+1 i1=i+1
vb(i,i)=vb(i,i)+k(i1) ;IF i<ns GOTO 2170 ;i=0 ;:2180 ;i=i+1
j=0 ;:2185 ;j=j+1 vb(i,j)=vd(i,j)+vb(i,j)
IF j<ns GOTO 2185 ;IF i<ns GOTO 2180
! Invert flexibility matrix using Fox's method; inverts the square
! symmetric matrix vb(,), returning the inverted matrix in va(,).
n=ns va(1)=VEC(0)*2500 vc(1)=VEC(0)*2500 vc(1,1)=SQR(vb(1,1))
j=1 ;:4030 ;j=j+1 vc(1,j)=(vb(1,j))/(vc(1,1)) ;IF j<n GOTO 4030
i=1 ;:4200 ;i=i+1 j=i-1 ;:4205 ;j=j+1 z=i-1 ;IF i-j>=0 GOTO 4080
sig=0 k=0 ;:4060 ;k=k+1 sig=sig-vc(k,i)*vc(k,j) ;IF k<z GOTO 4060
vc(i,j)=(vb(i,j)+sig)/(vc(i,i)) ;GOTO 4210 ;:4080 ;sig=0
k=0 ;:4090 ;k=k+1 sig=sig-vc(k,i)*vc(k,i) ;IF k<z GOTO 4090
IF vb(i,j)+sig<0 GOTO 4530 ;vc(i,i)=SQR(vb(i,i)+sig)
:4210 ;IF j<n GOTO 4205 ;IF i<n GOTO 4200 ; q=n-1
v=-1 ;:4400 ;v=v+1 y=-1 ;:4410 ;y=y+1 i=n-v j=n-y
IF j-i=0 GOTO 4350 ;IF j-i>=0 GOTO 4390 ;z=j+1 sig=0
k=z-1 ;:4330 ;k=k+1 sig=sig-vc(j,k)*va(i,k) ;IF k<n GOTO 4330
va(i,j)=sig/(vc(j,j)) ;GOTO 4395 ;:4350 ;IF i-n<0 GOTO 4360
va(i,i)=1.0/(vc(i,i)*vc(i,i)) ;GOTO 4395 ;:4360 ;z=i+1 sig=0
k=z-1 ;:4370 ;k=k+1 sig=sig-vc(i,k)*va(i,k) ;IF k<n GOTO 4370
va(i,i)=(1.0/vc(i,i)+sig)/vc(i,i) ;GOTO 4395 ;:4390
va(i,j)=va(j,i) ;:4395 ;IF y<q GOTO 4410 ;IF v<q GOTO 4400
GOTO 4550 ;:4530 ;* Error in flexibility matrix. ;STOP
:4550 ;! End of matrix inversion. ;fl=0 ;*/8 ;* Flexibility method
*    Pile load    Deflection    Shear left   Shear right    Moment
*       kN              m           kN            kN            kNm
:2192 ;nl'=nl i=0 ;:2193 ;i=i+1 a'(i)=a(i) ;IF i<nl' GOTO 2193
i=0 ;:2194 ;i=i+1 p'(i)=p(i) ;IF i<nl' GOTO 2194 ;iret=2197
GOTO 212 ;:2197 ;c=c+1 i=0 ;:2020 ;i=i+1 v(i)=0 j=0 ;:2025 ;j=j+1
k'=(k(1)*p(j)/l^2)*(l^2-a(j)*l-b(i)*l+a(j)*b(i))
v(i)=v(i)+vd(i,j)+k'+(k(nk)*p(j)*a(j)*b(i)/l^2)
IF j<nl GOTO 2025 ;IF i<ns GOTO 2020 ;mac=0 mc=0 u(1,1)=0 ptot=0
utot=0 i=0 ;:2220 ;i=i+1 i1=i+1 u(i1,1)=0 ;j=0 ;:2210 ;j=j+1
u(i1,1)=u(i1,1)+va(i,j)*v(j) ;IF j<ns GOTO 2210
mac=mac+u(i1,1)*b(i) ;utot=utot+u(i1,1) ;IF i<ns GOTO 2220
i=0 ;:2230 ;i=i+1 mc=mc+a(i)*p(i) ptot=ptot+p(i)
IF i<nl GOTO 2230 ;u(nk,1)=(mc-mac)/l u(1,1)=ptot-utot-u(nk,1)
i=0 ;:2270 ;i=i+1 u(i,2)=u(i,1)*k(i) ;IF i<nk GOTO 2270
sigs=0 sigm=0 i=0 ;:2300 ;i=i+1 ;IF a(i)<0 GOTO 2310
IF a(i)=0 GOTO 2320 ;IF a(i)>0 GOTO 2330 ;:2310 ;sigs=sigs-p(i)
sigm=sigm+p(i)*a(i) ;GOTO 2300 ;:2320 ;IF i-nl>0 GOTO 2335
u(1,4)=sigs-p(i)+u(1,1) ;GOTO 2340 ;:2330 ;i=i-1 ;:2335
u(1,4)=sigs+u(1,1) ;:2340 ;u(1,3)=sigs u(1,5)=sigm u(1,6)=0
j=1 ;:2360 ;j=j+1 j1=j-1 u(j,6)=b(j1) ;IF j<nk-1 GOTO 2360
u(nk,6)=l j=1 ;:2490 ;j=j+1 j1=j-1 sigs=0 sigm=0 ;:2410 ;i=i+1
IF i-nl>0 GOTO 2445 ;IF a(i)-u(j,6)=0 GOTO 2430
IF a(i)-u(j,6)>0 GOTO 2440 ;sigs=sigs-p(i)
sigm=sigm-p(i)*(u(j,6)-a(i)) ;GOTO 2410 ;:2430
u(j,4)=sigs-p(i)+u(j1,4)+u(j,1) ;GOTO 2450 ;:2440 ;i=i-1 ;:2445
u(j,4)=sigs+u(j1,4)+u(j,1) ;:2450 ;u(j,3)=sigs+u(j1,4)
u(j,5)=sigm+u(j1,5)+u(j1,4)*(u(j,6)-u(j1,6)) ;IF j<nk GOTO 2490
```

```
*   Position +c ;i=0 ;:2545 ;i=i+1 fl=fl+1 fl(fl)=u(i,5)
* ++u(i,1)      ++u(i,2)      ++u(i,3)      ++u(i,4)      ++u(i,5)
IF i<nk GOTO 2545 ;IF c-ni>=0 GOTO 2960
! Increment load positions by si for next.
nsum=0 i=0 ;:2635 ;i=i+1 a(i)=a(i)+si ;IF a(i)-lo>0 GOTO 2630
nsum=nsum+1 ;GOTO 2634 ;:2630 ;p(i)=0 ;:2634 ;IF i<nl GOTO 2635
nl=nsum ;GOTO 2192 ;:2960 ;! Start of summary. ;gtot=0 nur=0 ;*/11
*    Loading      Pile and       Moment       Moment        %age
*    position     joint No.      NL-STRESS    flexibility   diff.
i=0 fl=0 ;:900 ;i=i+1 j=0 ;:905 ;j=j+1 row=(i-1)*nm+j
bnl=ARR(13,row,3) fl=fl+1 bfl=fl(fl) d1=bnl d2=bfl
#vmper.ndf !Compute percentage difference as text message.
*       +i            +j      ++bnl          ++bfl          $ok
IF j<nm GOTO 905 ;IF i<nls GOTO 900 ;fnm=$(vm131.stk) ;#vmres.ndf
mjn=1 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE INFLUENCE LINES FOR REACTION, SHEAR & MOMENT FOR
TITLE A CONTINUOUS BEAM HAVING 'nsp' SPANS.
TITLE RESULTS COMPARE THE MATRIX STIFFNESS METHOD WITH
TITLE THAT OF MULLER-BRESLAU. ;MADEBY DWB ;DATE 15.11.04
PRINT DATA RESULTS COLLECTION FROM 1 ;REFNO VM140
TABULATE FORCES ;TYPE PLANE FRAME run=0
*/5
* 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21..
* ├──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──┼──▶▶
*_▲_  └──┘joint numbering based _▲_                          _▲_
*\\\      on 10 segments/span   ooo                          ooo
* ├──────────────s(1)───────────┼──────────────s(2)──────────┼──▶▶
nsp=2  nseg=10            ! No. spans & segments.
s1=VEC(10,10)             ! Spans left to right for 'nsp' spans.
ax(1)=VEC(.36,.36)        ! Areas, left to right for 'nsp' spans.
ay(1)=VEC(.30,.30)        ! Shear areas, left to right.
iz(1)=VEC(.0243,.0243)    ! Moments of inertia, left to right.
e=28E6/3 nu=.2   ! Young's mod; nu=Poisson, 0 to ignore shear def.
nilc=4                    ! No. of influence line checks (1 to 4).
! For reaction & shear IL's, joint numbers must be at a support.
jno(1)=VEC(1,11,11,12)    ! Influence line (IL) joint positions.
opt(1)=VEC(1,2,3,3)       ! IL type: 1=reaction, 2=shear, 3=moment.
#cc924.stk !Import set of parameters if available from cc924.stk.
IF nsp>8 OR nilc<1 OR nilc>4 GOTO 9000 ;opt(nilc+1)=4 nsg=1
NUMBER OF JOINTS nj=nsp*nseg+1 nj ;NUMBER OF MEMBERS nm=nj-1 nm
NUMBER OF SUPPORTS 0     ;NUMBER OF LOADINGS nj
JOINT COORDINATES ;1 0 0 ;x=0 j=1 i=0 ;:10 ;i=i+1
inc=s(i)/nseg k=0 ;:20 ;k=k+1 j=j+1 x=x+inc ;j x 0
IF k<nseg GOTO 20 ;IF i<nsp GOTO 10 ;JOINT RELEASES
1 FORCE X -1 ;1 THRU nj STEP nseg FORCE Y -1 ;MEMBER INCIDENCES
1 THRU nj-1 RANGE 1,2 nj-1,nj ;g=e/(2*(1+nu))
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES ;i=0 ;:30 ;i=i+1
IF nu=1E-12 THEN ay(i)=0 ;! If nu=1E-12 shear def. ignored.
k=nseg*(i-1)+1 ;k THRU k+nseg-1 AX ax(i) AY ay(i) IZ iz(i)
IF i<nsp GOTO 30 ;n=0 ;REPEAT ;n=n+1
LOADING ;IF n<nj ;MEMBER LOADS ;n FORCE Y CONCENTRATED P -1 L 0
ENDIF ;IF n=nj ;JOINT LOADS ;n FORCE Y -1 ;ENDIF
UNTIL n=nj ;ENDREPEAT ;SOLVE ;status=1 gtot=0 nur=0
i=0 ;:5 ;i=i+1 ;IF opt(i)<3 THEN loc(i)=INT((jno(i)-1)/nseg)+1
IF opt(i)=3 THEN loc(i)=(jno(i)-1+nseg)/nseg ;IF i<nilc GOTO 5
r0=0.00001 p0=2*nseg i=10 il=10+nsp ;:40 ;i=i+1 ;s(i)=iz(i-10)
IF i<10+nsp GOTO 40 ;k=21 i=0 ;:50 ;i=i+1
IF k>=30 THEN opt(i)=4 ;IF opt(i)>3 GOTO 60 ;IF opt(i)=1
* Influence line for support at joint +jno(i) ;s(k)=1 ;ENDIF
IF opt(i)=2 ;* Influence line for shear to right of joint +jno(i)
s(k)=2 ;ENDIF ;IF opt(i)=3
* Influence line for moment at joint +jno(i) ;s(k)=3 ;ENDIF
ij=k+1 s(ij)=loc(i) ij=k+2 s(ij)=0 k=k+2 ;GOTO 50 ;:60
k=k-2 n=2*p0*nsp+2 e=n-1 h=n+3 e8=4*n+8 j0=n/2 b0=2
b1=2+nsp+1 b2=b1+nseg*nsp ;i=0 ;:70 ;i=i+1 a(i)=0
! Set up stiffness matrix for span s
IF i<e8 GOTO 70 ;s=0 ;:80 ;s=s+1 l9=s(s)/p0 ij=10+s
i9=s(ij) k2=2*i9/l9 k4=k2+k2 k6=(k2+k4)/l9
k0=(k6+k6)/l9 k'(1)=k0 k'(2)=k6 k'(3)=-k0 k'(4)=k6
k'(5)=k4 k'(6)=-k6 k'(7)=k2 k'(8)=0.0 k'(9)=k0 k'(10)=-k6
k'(11)=0.0 k'(12)=0.0 k'(13)=k4 j=0 ;:90 ;j=j+1
a=8*p0*(s-1)+8*(j-1) i=0 ;:100 ;i=i+1 ij=a+i a(ij)=a(ij)+k'(i)
IF i<13 GOTO 100 ;IF j<p0 GOTO 90 ;IF s<nsp GOTO 80 ;i=0 ;:110
! Deal with supports
i=i+1 a=8*p0*(i-1)+1 a(a)=1.0 ij=a+1 a(ij)=0 ij=a+2 a(ij)=0
ij=a+3 a(ij)=0
IF i<>1 THEN ij=a-3 a(ij)=0 ij=a-6 a(ij)=0 ij=a-9 a(ij)=0
```

```
! Reduce matrix to triangular form.
IF i<nsp+1 GOTO 110 ;i=0 ;:120 ;i=i+1 d=4*i-3 k=0 ;:130
k=k+1 ij=d+k m=a(ij) ;IF m<>0 ;m=m/a(d) l=k-1 ;:140
l=l+1 b=d+3*k+l ij=d+l a(b)=a(b)-m*a(ij) ;IF l<3 GOTO 140
ENDIF ;IF k<3 GOTO 130 ;IF i<e GOTO 120 ;j=b1 s=0 ;:150
s=s+1 p(j)=s j=j+1 i=0 ;:160 ;i=i+1 p(j)=s+i/nseg j=j+1
IF i<nseg-1 GOTO 160 ;IF s<nsp GOTO 150 ;p(j)=nsp+1 i=0 ;:170
i=i+1 ij=2+i p(ij)=i ;IF i<nsp GOTO 170 ;w=19 line=0 ;:180
w=w+2 line=line+1 ;IF s(w)<>0 OR b0=2 GOTO 4401 ;iret=4401
! Interpret one instruction.  Clear vector b() down to h.
GOTO 4400 ;:4401 ;IF s(w)=0 GOTO 230 ;i=0 ;:190 ;i=i+1 b'(i)=0
IF i<h GOTO 190 ;t=s(w) ij=w+1 v=s(ij) ;IF t=1 AND v<>1
! Apply forces to left of support selected.
a=2*p0*(v-1)-1 ij=v-1 l9=s(ij)/p0 ij=9+v i9=s(ij) k6=6*i9/(l9*l9)
k0=(k6+k6)/l9 b'(a)=k0 ij=a+1 b'(ij)=k6 ij=a+3 b'(ij)=k6
ENDIF ;IF t=1 AND v<>nsp+1
! Add forces to right of support selected.
a=2*p0*(v-1)-1 l9=s(v)/p0 ij=10+v i9=s(ij) k6=6*i9/(l9*l9)
k0=(k6+k6)/l9 ij=a+3 b'(ij)=b'(ij)-k6 ij=a+4 b'(ij)=k0 ij=a+5
b'(ij)=-k6 ;ENDIF ;IF t<>1 GOTO 191 ;iret=3501 ;GOTO 3500
:3501 ;ij=a+2 b'(ij)=1.0 ;:191 ;IF t<>2 GOTO 192
a=2*p0*(v-1)-1 l9=s(v)/p0 ij=10+v i9=s(ij) k6=6*i9/(l9*l9)
k0=(k6+k6)/l9 ij=a+3 b'(ij)=b'(ij)-k6 ij=a+4 b'(ij)=k0 ij=a+5
b'(ij)=-k6 iret=3502 ;GOTO 3500 ;:3502 ;ij=a+2 b'(ij)=1.0
:192 ;IF t<>3 GOTO 193 ;! Apply forces to right of selected point.
i7=INT(v) i8=INT(p0*(v-i7)+r0) a=INT(2*p0*(v-1)+1+r0) l9=s(i7)/p0
ij=i7+10 i9=s(ij) k2=2*i9/l9 k6=3*k2/l9
IF i8<>0 THEN b'(a)=-k6 ;ij=a+1 b'(ij)=-k2-k2
IF i8<>p0-1 THEN ij=a+2 b'(ij)=k6  ;ij=a+3 b'(ij)=-k2
iret=3503 ;GOTO 3500 ;:3503 ;:193 ;IF b0>=9 ;iret=4402
GOTO 4400 ;:4402 ;ENDIF ;s=0 ;:200
s=s+1 j=2*p0*(s-1)+1 j1=j+2*(p0-1) ij=j1+2 a=b'(j)+b'(ij)
m=4 i=j-2 ;:210 ;i=i+2 a=a+m*b'(i) m=6-m
IF i<j1 GOTO 210 ;ij=(b0-1)*91+s+2 p(ij)=a*s(s)/(3*p0)
IF s<nsp GOTO 200 ;j=1 i=b1-1 ;:220
i=i+1 ij=(b0-1)*91+i p(ij)=b'(j) j=j+4 ;IF i<b2 GOTO 220
ij=(b0-1)*91+1 p(ij)=t ij=ij+1 p(ij)=v b0=b0+1 ;GOTO 180
:230 ;GOTO 9000 ;:3500 ;i=0 ;:350 ;i=i+1 d=4*i-3 k=0
:360 ;k=k+1 ij=d+k m=a(ij)
IF m<>0 THEN ij=i+k b'(ij)=b'(ij)-b'(i)*m/a(d)
IF k<3 GOTO 360 ;IF i<e GOTO 350 ;ij=n+1 b'(ij)=0 ij=n+2
! Back substitute.
b'(ij)=0 ij=n+3 b'(ij)=0 i=0 ;:370 ;i=i+1 d=4*(n-i)+1
b=n-i+1 ij1=d+1 ij2=d+2 ij3=d+3 ik1=b+1 ik2=b+2
ik3=b+3 x=a(ij1)*b'(ik1)+a(ij2)*b'(ik2)+a(ij3)*b'(ik3)
b'(b)=(b'(b)-x)/a(d) ;IF i<n GOTO 370 ;GOTO iret ;:4400 ;o=0
* ORDINATES OF INFLUENCE LINES — IN ORDER ON 1 OR 2 LINES
* Joint     Muller-     NL-         Muller-     NL-        %age
* Number    Breslau     STRESS      Breslau     STRESS     diff.
j=b1 s=0 ;:440 ;s=s+1 iret1=5101 ;GOTO 5100 ;:5101
j=j+1 u=0 ;:450 ;u=u+1 iret1=5102 ;GOTO 5100 ;:5102
j=j+1 ;IF u<nseg-1 GOTO 450 ;IF s<nsp GOTO 440 ;iret1=5103
GOTO 5100 ;:5103 ;b0=2 ;GOTO iret ;:5100 ;o=o+1 ;nl1=0 nl2=0 nl3=0
nl4=0 ij1=j ij2=ij1+91 ij3=ij2+91 ij4=ij3+91 ij5=ij4+91
r=1 i=0 ;:460 ;i=i+1
IF opt(i)=1 THEN row=(jno(i)-1)*3+2 nl(i)=ARR(14,row,o)
IF opt(i)=2 THEN row=(o-1)*nm+jno(i) nl(i)=ARR(13,row,2)
IF opt(i)=3 THEN row=(o-1)*nm+jno(i) nl(i)=ARR(13,row,3)
p=ij(i+1) c(i)=p(p)
IF i<b0-2 GOTO 460 ;IF b0-1=1 ;* +o ;ENDIF
```

```
IF b0-1=2 ;d1=nl(1) d2=p(ij2)
#vmper.ndf !Compute percentage difference & any message.
* +o  ++p(ij2)     ++nl(1)                                      $ok
ENDIF
IF b0-1=3 ;d1=nl(1) d2=p(ij2)
#vmper.ndf
per1=per ok1=ok d1=nl(2) d2=p(ij3)
#vmper.ndf
IF per1>per THEN ok=ok1
* +o  ++p(ij2)     ++nl(1)      ++p(ij3)     ++nl(2)           $ok
ENDIF
IF b0-1=4 ;d1=nl(1) d2=p(ij2)
#vmper.ndf
per1=per ok1=ok d1=nl(2) d2=p(ij3)
#vmper.ndf
IF per1>per THEN ok=ok1
* +o  ++p(ij2)     ++nl(1)      ++p(ij3)     ++nl(2)           $ok
d1=nl(3) d2=p(ij4)
#vmper.ndf
*     ++p(ij4)     ++nl(3)                                      $ok
ENDIF
IF b0-1=5 ;d1=nl(1) d2=p(ij2)
#vmper.ndf
per1=per ok1=ok d1=nl(2) d2=p(ij3)
#vmper.ndf
IF per1>per THEN ok=ok1
* +o  ++p(ij2)     ++nl(1)      ++p(ij3)     ++nl(2)           $ok
d1=nl(3) d2=p(ij4)
#vmper.ndf
per1=per ok1=ok d1=nl(4) d2=p(ij5)
#vmper.ndf
IF per1>per THEN ok=ok1
*     ++p(ij4)     ++nl(3)      ++p(ij5)     ++nl(4)           $ok
ENDIF ;GOTO iret1 ;:9000 ;fnm=$(vm140.stk) ;#vmres.ndf !Conclude.
mjn=1 lcn=2 tot=1 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE N/PRATT THROUGH LATTICE TRUSS, OPTIONAL HEIGHT
TITLE TAPER, END DIAGONALS IN COMPRESSION, SUBJECTED TO
TITLE UDL ON TOP & BOTTOM CHORDS, SHARED TO JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 04.05.05 ;TYPE PLANE TRUSS run=0 ;REFNO VM150
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*  Y|                              ┌at centre
*       ┌───┬───┬───┬───┐       ─┼─   ┌───┬───┬───┬───┐        ─┬─  ──►st
*     / │ \ │ \ │ \ │ \ │        d    │ / │ / │ / │ / │ \       c
*   /───┴─X─┴───┴───┴───┘       ─┴─   └───┴───┴───┴───┴───\   ─┼─  ──►sb
*   _▲_N or Pratt lattice truss, even number of bays_▲_  └at ends
*   ///                                              ooo
*   ├───────────────────────a───────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2    d=3.2     ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
st=20    sb=15     ! Sway loading, top & bottom chord (+ve right).
ax(1)=75.8E-4 ax(3)=92.9E-4  ! Areas of bottom & top chords resp.
ax(4)=58.8E-4 ax(2)=47.4E-4  ! Areas of verticals & diagonals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*nb nj ;NUMBER OF MEMBERS nm=nb*4-3 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
IF nb/2<>INT(nb/2) THEN Even number of bays reqd.
JOINT COORDINATES ;b=a/nb x1=0 y1=0 i=0 x=0 ;:10 ;i=i+2 x=x+b yl=0
yt=d-(d-c)*ABS(a/2-x)/(a/2) x(i)=x x(i+1)=x ;IF i<nj/2 ;y(i)=yl
y(i+1)=yt ;ENDIF ;IF i>=nj/2 ;y(i)=yt y(i+1)=yl ;ENDIF
i x(i) y(i) ;i+1 x(i+1) y(i+1) ;IF i<nj-2 GOTO 10 ;x(nj)=a y(nj)=0
nj x(nj) y(nj) ;1 0 0 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js1=1 je1=2 mt1=1 js2=1 je2=3
mt2=2 i=-1 nmm=4*(nb-2) j=0 ;:20 ;i=i+4 j=j+2
IF i<nmm/2-1 ;js(i)=j je(i)=j+2 mt(i)=1 js(i+1)=j je(i+1)=j+1
mt(i+1)=4 js(i+2)=j+1 je(i+2)=j+2 mt(i+2)=2 js(i+3)=j+1
je(i+3)=j+3 mt(i+3)=3 ;ENDIF ;IF i=nmm/2-1 ;js(i)=j je(i)=j+3
mt(i)=1 js(i+1)=j je(i+1)=j+1 mt(i+1)=4 js(i+2)=j+1 je(i+2)=j+3
mt(i+2)=2 js(i+3)=j+1 je(i+3)=j+2 mt(i+3)=3 ;ENDIF ;IF i>nmm/2
js(i)=j je(i)=j+1 mt(i)=4 js(i+1)=j je(i+1)=j+2 mt(i+1)=3
js(i+2)=j+1 je(i+2)=j+3 mt(i+2)=1 js(i+3)=j+1 je(i+3)=j+2
mt(i+3)=2 ;ENDIF ;IF i<nm-6 GOTO 20 ;js(nm-2)=nj-2 je(nm-2)=nj-1
mt(nm-2)=4 js(nm-1)=nj-2 je(nm-1)=nj mt(nm-1)=2 js(nm)=nj-1
je(nm)=nj mt(nm)=1 i=0 ;:25 ;i=i+1 ;i js(i) je(i) ;IF i<nm GOTO 25
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
i=0 ;:30 ;i=i+1 mt=mt(i) i AX ax(mt) IZ 1 ;IF i<nm GOTO 30
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj h(nj-2)=st h(nj)=sb
wtp=wt*b wbp=wb*b i=nj/2-2 ;:32 ;i=i+2 v(i)=wtp v(i+1)=wbp
IF i<nj-2 GOTO 32 ;v(nj)=wbp/2 v(nj-2)=wtp/2 i=0 ;:34 ;i=i+2
v(i)=wbp v(i+1)=wtp ;IF i<nj/2-2 GOTO 34 ;v(1)=wbp/2 v(3)=wtp/2
i=0 ;:35 ;i=i+1 i FORCE Y v(i) ;IF i<nj GOTO 35
nj-2 FORCE X h(nj-2) ;nj FORCE X h(nj) ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i      +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm150.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE N/PRATT DECK LATTICE TRUSS, OPTIONAL HEIGHT TAPER,
TITLE END DIAGONALS IN TENSION, SUBJECTED TO UDL ON TOP
TITLE & BOTTOM CHORDS, SHARED TO JOINTS, AND SWAY LOADS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 14.04.05 ;TYPE PLANE TRUSS run=0 ;REFNO VM153
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*  Y|                                    ┌at centre
*    ┌─────────────────────┐  ─┬─  ┌──────────────────────┐  ─┬─  ──►st
*    │ \ │ \ │ \ │ \ │ \ │  d   │ / │ / │ / │ / │ / │  c
*    └────X────────────────┘  ─┴─  └──────────────────────┘  ─┼─  ──►sb
*   _▲_N or Pratt lattice truss, even number of bays_▲_  └at ends
*   ///                                            ooo
*    ├──────────────────────a──────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2    d=3.2     ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
st=20    sb=15     ! Sway loading, top & bottom chord (+ve right).
ax(1)=75.8E-4 ax(4)=92.9E-4  ! Areas of bottom & top chords res.
ax(2)=58.8E-4 ax(3)=47.4E-4  ! Areas of verticals & diagonals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*(nb+1) nj ;NUMBER OF MEMBERS nm=nb*4+1 nm
NUMBER OF LOADINGS 1 ;NUMBER OF SUPPORTS 0
IF nb/2<>INT(nb/2) THEN Even number of bays reqd.
JOINT COORDINATES ;b=a/nb i=-1 x=-b ;:10 ;i=i+2 ;x=x+b ;yl=0
yt=d-(d-c)*ABS(a/2-x)/(a/2) x(i)=x x(i+1)=x
IF i<nj/2 ;y(i)=yl y(i+1)=yt ;ENDIF ;IF i>=nj/2 ;y(i)=yt y(i+1)=yl
ENDIF ;i x(i) y(i) ;i+1 x(i+1) y(i+1) ;IF i<nj-1 GOTO 10
JOINT RELEASES ;1 FORCE X -1 Y -1 ;nj FORCE Y -1
MEMBER INCIDENCES ;i=-3 nmm=4*nb j=-1 ;:20 ;i=i+4 j=j+2
IF i<nmm/2-4 ;js(i)=j je(i)=j+2 mt(i)=1 js(i+1)=j je(i+1)=j+1
mt(i+1)=2 js(i+2)=j+1 je(i+2)=j+2 mt(i+2)=3 js(i+3)=j+1
je(i+3)=j+3 mt(i+3)=4 ;ENDIF ;IF i=nmm/2-3 ;js(i)=j je(i)=j+3
mt(i)=1 js(i+1)=j je(i+1)=j+1 mt(i+1)=2 js(i+2)=j+1 je(i+2)=j+3
mt(i+2)=3 js(i+3)=j+1 je(i+3)=j+2 mt(i+3)=4 ;ENDIF ;IF i>nmm/2
js(i)=j je(i)=j+1 mt(i)=2 js(i+1)=j je(i+1)=j+2 mt(i+1)=4
js(i+2)=j+1 je(i+2)=j+3 mt(i+2)=1 js(i+3)=j+1 je(i+3)=j+2
mt(i+3)=3 ;ENDIF ;k=-1 ;:25 ;k=k+1 ;i+k js(i+k) je(i+k)
IF k<3 GOTO 25 ;IF i<nm-4 GOTO 20 ;mt(nm)=2 ;nm nj nj-1
js(nm)=nj-1 je(nm)=nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
i=0 ;:30 ;i=i+1 mt=mt(i) i AX ax(mt) IZ 1 ;IF i<nm GOTO 30
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj h(nj-1)=VEC(st,sb)
wbp=wb*b wtp=wt*b v(1)=wbp/2 v(2)=wtp/2 v(nj-1)=wtp/2 v(nj)=wbp/2
n=INT(nb/2)-1 v(3)=VEC(wbp,wtp)*n i=2*n+3 n=nb-n-1
v(i)=VEC(wtp,wbp)*n i=0 ;:35 ;i=i+1 i FORCE Y v(i)
IF i<nj GOTO 35 ;nj-1 FORCE X h(nj-1) ;nj FORCE X h(nj) ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                        NL-STRESS     Classical       %age
*                                 analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i        +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm153.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf !Checksum
< ;FINISH
```

```
TITLE HOWE THROUGH LATTICE TRUSS, OPTIONAL HEIGHT
TITLE TAPER, END DIAGONALS IN COMPRESSION, SUBJECTED TO
TITLE UDL ON TOP & BOTTOM CHORDS, SHARED TO JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 18.04.05 ;TYPE PLANE TRUSS run=0 ;REFNO VM156
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*  Y|                           ┌at centre
*        ┌───┬───┬───┬───┬───┐  ─┼─  ┌───┬───┬───┬───┬───┐        ─┬─  ──►st
*     /  │ / │ / │ / │ / │ / │   d   │ \ │ \ │ \ │ \ │ \ │  \      c
*   /────┴───X───┴───┴───┴───┘  ─┴─  └───┴───┴───┴───┴───┴────\  ─┼─  ──►sb
*  _▲_   Howe lattice truss, even number of bays              _▲_  └at ends
*  ///                                                        ooo
*   ├──────────────────────────a──────────────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2    d=3.2     ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
st=20    sb=15     ! Sway loading, top & bottom chord (+ve right).
ax(1)=75.8E-4 ax(4)=92.9E-4  ! Areas of bottom & top chords resp.
ax(2)=58.8E-4 ax(3)=47.4E-4  ! Areas of diagonals and verticals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*nb nj ;NUMBER OF MEMBERS nm=nb*4-3 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
IF nb/2<>INT(nb/2) THEN Even number of bays reqd.
JOINT COORDINATES ;b=a/nb x(1)=0 x(nj)=a y(1)=VEC(0)*nj x=0 j=0
:2 ;j=j+2 x=x+b x(j)=x x(j+1)=x ;IF j<nj-1 GOTO 2
yinc=(d-c)/((nb-2)/2) y=c-yinc i=0 ;:4 ;i=i+2
IF i=nj/2+2 THEN yinc=-yinc ;y=y+yinc y(i)=y
IF i>nj/2-1 THEN y(i+1)=y(i) y(i)=0
IF i<nj-1 GOTO 4 ;y(nj)=0 j=0 ;:5 ;j=j+1 j x(j) y(j)
IF j<nj GOTO 5 ;JOINT RELEASES ;1 FORCE X -1 Y -1 ;nj FORCE Y -1
MEMBER INCIDENCES ;js(1)=1 je(1)=3 mt(1)=1 js(2)=1 je(2)=2 mt(2)=2
j=0 m=-2 ;:10 ;j=j+2 m=m+4 ;IF j<nj/2-2 ;js(m+1)=j
je(m+1)=j+1 mt(m+1)=3 js(m+2)=j je(m+2)=j+2 mt(m+2)=4 js(m+3)=j+1
je(m+3)=j+3 mt(m+3)=1 js(m+4)=j+1 je(m+4)=j+2 mt(m+4)=2 ;ENDIF
IF j=nj/2-2 ;js(m+1)=j je(m+1)=j+1 mt(m+1)=3 js(m+2)=j je(m+2)=j+3
mt(m+2)=4 js(m+3)=j+1 je(m+3)=j+2 mt(m+3)=1 js(m+4)=j+1
je(m+4)=j+3 mt(m+4)=2 ;ENDIF ;IF j>nj/2-2 ;js(m+1)=j
je(m+1)=j+2 mt(m+1)=1 js(m+2)=j je(m+2)=j+1 mt(m+2)=3 js(m+3)=j+1
je(m+3)=j+2 mt(m+3)=2 js(m+4)=j+1 je(m+4)=j+3 mt(m+4)=4 ;ENDIF
IF j<nj-4 GOTO 10 ;js(nm-2)=nj-2 je(nm-2)=nj mt(nm-2)=1
js(nm-1)=nj-2 je(nm-1)=nj-1 mt(nm-1)=3 js(nm)=nj-1 je(nm)=nj
mt(nm)=2 m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
m=0 ;:20 ;m=m+1 mt=mt(m) m AX ax(mt) IZ 1 ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj h(nj-1)=VEC(st,sb)
wbp=wb*b wtp=wt*b j=0 ;:25 ;j=j+1 ;IF y(j)=0 THEN v(j)=wbp
IF y(j)<>0 THEN v(j)=wtp ;IF j<nj GOTO 25 ;v(1)=v(1)/2 v(2)=v(2)/2
v(nj-1)=v(nj-1)/2 v(nj)=v(nj)/2 j=0 ;:30 ;j=j+1 j FORCE Y v(j)
IF j<nj GOTO 30 ;nj-1 FORCE X h(nj-1) ;nj FORCE X h(nj) ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                        NL-STRESS     Classical       %age
*                                 analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i        +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm156.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE HOWE DECK LATTICE TRUSS WITH OPTIONAL HEIGHT TAPER,
TITLE END DIAGONALS IN COMPRESSION, SUBJECTED TO UDL ON
TITLE TOP & BOTTOM CHORDS, SHARED TO JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 16.04.05 ;TYPE PLANE TRUSS run=0 ;;REFNO VM159
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*  Y│                                      ┌at centre
*     ┌───┬───┬───┬───┬───┬───┐  ─┼─  ┌───┬───┬───┬───┬───┬───┐  ─┬─
*     │ / │ / │ / │ / │ / │      d    │ \ │ \ │ \ │ \ │ \ │    c
*     └───┴──X┴───┴───┴───┴───┘  ─┴─  └───┴───┴───┴───┴───┴───┘  ─┼─
*   _▲_   Howe lattice truss, even number of bays.  _▲_  └at ends
*   ///                                             ooo
*     ├──────────────────────a───────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2    d=3.2     ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
ax(2)=92.9E-4 ax(3)=75.8E-4  ! Areas of top & bottom chords res.
ax(1)=47.4E-4 ax(4)=58.8E-4  ! Areas of verticals & diagonals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*(nb+1) nj ;NUMBER OF MEMBERS nm=nb*4+1 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
IF nb/2<>INT(nb/2) THEN Even number of bays reqd.
JOINT COORDINATES ;b=a/nb i=-1 x=-b ;:10 ;i=i+2 ;x=x+b ;yl=0
yt=d-(d-c)*ABS(a/2-x)/(a/2) x(i)=x x(i+1)=x
IF i<nj/2 ;y(i)=yt y(i+1)=yl ;ENDIF ;IF i>=nj/2 ;y(i)=yl y(i+1)=yt
ENDIF ;i x(i) y(i) ;i+1 x(i+1) y(i+1) ;IF i<nj-1 GOTO 10
JOINT RELEASES ;2 FORCE X -1 Y -1 ;nj-1 FORCE Y -1
MEMBER INCIDENCES ;i=-3 nmm=4*nb j=-1 ;:20 ;i=i+4 j=j+2
IF i<nmm/2-4 ;js(i)=j je(i)=j+1 mt(i)=1 js(i+1)=j je(i+1)=j+2
mt(i+1)=2 js(i+2)=j+1 je(i+2)=j+3 mt(i+2)=3 js(i+3)=j+1
je(i+3)=j+2 mt(i+3)=4 ;ENDIF ;IF i=nmm/2-3 ;js(i)=j je(i)=j+1
mt(i)=1 js(i+1)=j je(i+1)=j+3 mt(i+1)=2 js(i+2)=j+1 je(i+2)=j+2
mt(i+2)=3 js(i+3)=j+1 je(i+3)=j+3 mt(i+3)=4 ;ENDIF ;IF i>nmm/2
js(i)=j je(i)=j+2 mt(i)=3 js(i+1)=j je(i+1)=j+1 mt(i+1)=1
js(i+2)=j+1 je(i+2)=j+2 mt(i+2)=4 js(i+3)=j+1 je(i+3)=j+3
mt(i+3)=2 ;ENDIF ;k=-1 ;:25 ;k=k+1 ;i+k js(i+k) je(i+k)
IF k<3 GOTO 25 ;IF i<nm-4 GOTO 20 ;mt(nm)=1 ;nm nj-1 nj
js(nm)=nj-1 je(nm)=nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
i=0 ;:30 ;i=i+1 mt=mt(i) i AX ax(mt) IZ 1 ;IF i<nm GOTO 30
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj
wbp=wb*b wtp=wt*b v(1)=wtp/2 v(2)=wbp/2 v(nj-1)=wbp/2 v(nj)=wtp/2
n=INT(nb/2)-1 v(3)=VEC(wtp,wbp)*n i=2*n+3 n=nb-n-1
v(i)=VEC(wbp,wtp)*n i=0 ;:35 ;i=i+1 i FORCE Y v(i)
IF i<nj GOTO 35 ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
p(1)=p(1)-r1v p(nm)=p(nm)-r2v !Fix for supports at 2 & nj-1
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i      +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm159.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

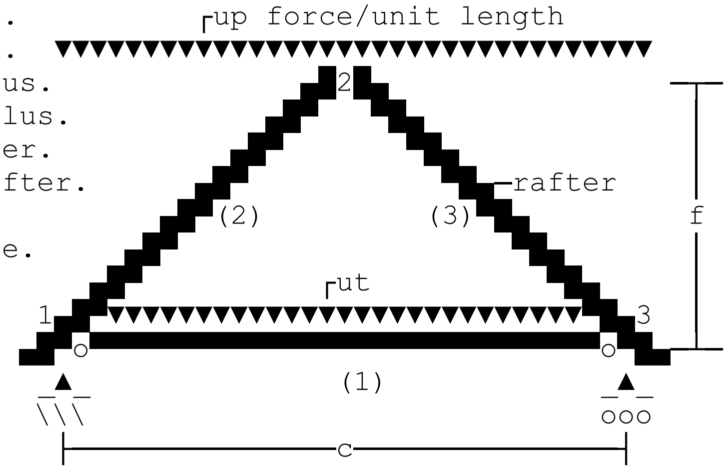
```
TITLE WARREN THROUGH LATTICE TRUSS, CONSTANT HEIGHT,
TITLE END DIAGONALS IN COMPRESSION, SUBJECTED TO
TITLE UDL ON TOP & BOTTOM CHORDS, SHARED TO JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 19.04.05 ;TYPE PLANE TRUSS run=0 ;REFNO VM162
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*                                       ┌at centre      ├───┼1 bay
*   Y   ·─────────·─────────·  ─┼─  ·─────────·─────────·     ─┬─ ──►st
*   │ /   \   /   \   /     c     \   /   \   /   \     c
*   ·───X───·─────────·─────  ─┴─  ─────·─────────·─────────·  ─┼─ ──►sb
*  _▲_   Warren lattice truss, even number of bays _▲_ └at ends
*  ///                                              ooo
*   ├───────────────────────a───────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2              ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
st=20    sb=15     ! Sway loading, top & bottom chord (+ve right).
ax(1)=75.8E-4 ax(3)=92.9E-4  ! Areas of bottom & top chords resp.
ax(2)=58.8E-4                ! Area of diagonals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=nb+1 nj ;NUMBER OF MEMBERS nm=nb*2-1 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
IF nb/2<>INT(nb/2) THEN Even number of bays reqd.
JOINT COORDINATES ;b=a/nb nbh=nb/2 y(1)=VEC(0,c)*nbh y(nj)=0
j=0 x=-b ;:2 ;j=j+1 x=x+b x(j)=x ;IF j<nj GOTO 2 ;j=0 ;:5 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 5 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;j=-1 m=-4 ;:10 ;j=j+2 m=m+4
js(m+1)=j je(m+1)=j+2 mt(m+1)=1 js(m+2)=j je(m+2)=j+1 mt(m+2)=2
js(m+3)=j+1 je(m+3)=j+2 mt(m+3)=2 js(m+4)=j+1 je(m+4)=j+3
mt(m+4)=3 ;IF j<nj-4 GOTO 10 ;js(nm-2)=nj-2 je(nm-2)=nj mt(nm-2)=1
js(nm-1)=nj-2 je(nm-1)=nj-1 mt(nm-1)=2 js(nm)=nj-1 je(nm)=nj
mt(nm)=2 m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
m=0 ;:20 ;m=m+1 mt=mt(m) m AX ax(mt) IZ 1 ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj h(nj-1)=VEC(st,sb)
wbp=wb*2*b wtp=wt*2*b j=0 ;:25 ;j=j+1 ;IF y(j)=0 THEN v(j)=wbp
IF y(j)<>0 THEN v(j)=wtp ;IF j<nj GOTO 25 ;v(1)=v(1)/2 v(2)=v(2)/2
v(nj-1)=v(nj-1)/2 v(nj)=v(nj)/2 j=0 ;:30 ;j=j+1 j FORCE Y v(j)
IF j<nj GOTO 30 ;nj-1 FORCE X h(nj-1) ;nj FORCE X h(nj) ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i      +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm162.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE WARREN THROUGH WITH VERTICALS, OPTIONAL HEIGHT
TITLE TAPER, END DIAGONALS IN COMPRESSION, SUBJECTED
TITLE TO UDL ON TOP & BOTTOM CHORDS, SHARED TO JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 14.04.05 ;TYPE PLANE TRUSS run=0 ;REFNO VM164
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*  Y│                          ┌at centre
*        ┌───┬───┬───┬───┬──  ─┼─  ──┬───┬───┬───┬───┐      ─┬─  ──►st
*     /  │ \ │ / │ \ │ / │     d    │ \ │ / │ \ │ / │ \     c
*   /────┴─X─┴───┴───┴───┴──  ─┴─  ──┴───┴───┴───┴───┴───\ ─┼─  ──►sb
*   _▲_N or Pratt lattice truss, even number of bays_▲_  └at ends
*   ///                                              ooo
*   ├───────────────────────a───────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2    d=3.2     ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
st=20    sb=15     ! Sway loading, top & bottom chord (+ve right).
ax(1)=75.8E-4 ax(3)=92.9E-4  ! Areas of bottom & top chords resp.
ax(4)=58.8E-4 ax(2)=47.4E-4  ! Areas of verticals & diagonals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*nb nj ;NUMBER OF MEMBERS nm=nb*4-3 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
IF nb/4<>INT(nb/4) THEN Quadruple number of bays required.
JOINT COORDINATES ;b=a/nb x(1)=0 x(nj)=a y(1)=VEC(0)*nj x=0 j=0
:2 ;j=j+2 x=x+b x(j)=x x(j+1)=x ;IF j<nj-1 GOTO 2
yinc=(d-c)/((nb-2)/2) y=c-yinc j=-1 ;:4 ;j=j+4
IF j=nj/2+3 THEN yinc=-yinc ;y=y+yinc y(j)=y y=y+yinc y(j+1)=y
IF j<nj-1 GOTO 4 ;y(nj)=0 j=0 ;:5 ;j=j+1 j x(j) y(j)
IF j<nj GOTO 5 ;JOINT RELEASES ;1 FORCE X -1 Y -1 ;nj FORCE Y -1
MEMBER INCIDENCES ;js(1)=1 je(1)=2 mt(1)=1 js(2)=1 je(2)=3 mt(2)=2
nset=(nb-2)/2 i=0 j=-2 m=-6 ;:10 ;i=i+1 j=j+4 m=m+8 js(m+1)=j
je(m+1)=j+3 mt(m+1)=1 js(m+2)=j je(m+2)=j+1 mt(m+2)=4 js(m+3)=j+1
je(m+3)=j+3 mt(m+3)=2 js(m+4)=j+1 je(m+4)=j+2 mt(m+4)=3
js(m+5)=j+2 je(m+5)=j+3 mt(m+5)=4 js(m+6)=j+2 je(m+6)=j+5
mt(m+6)=3 js(m+7)=j+3 je(m+7)=j+4 mt(m+7)=1 js(m+8)=j+3
je(m+8)=j+5 mt(m+8)=2 ;IF i<nset GOTO 10 ;js(nm-2)=nj-2
je(nm-2)=nj mt(nm-2)=1 js(nm-1)=nj-2 je(nm-1)=nj-1 mt(nm-1)=4
js(nm)=nj-1 je(nm)=nj mt(nm)=2 m=0 ;:15 ;m=m+1 m js(m) je(m)
IF m<nm GOTO 15 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES
m=0 ;:20 ;m=m+1 mt=mt(m) m AX ax(mt) IZ 1 ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj h(nj-1)=VEC(st,sb)
wbp=wb*b wtp=wt*b j=0 ;:25 ;j=j+1 ;IF y(j)=0 THEN v(j)=wbp
IF y(j)<>0 THEN v(j)=wtp ;IF j<nj GOTO 25 ;v(1)=v(1)/2 v(3)=v(3)/2
v(nj-1)=v(nj-1)/2 v(nj)=v(nj)/2 j=0 ;:30 ;j=j+1 j FORCE Y v(j)
IF j<nj GOTO 30 ;nj-1 FORCE X h(nj-1) ;nj FORCE X h(nj) ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                    NL-STRESS     Classical       %age
*                             analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i     +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm164.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE WARREN DECK LATTICE TRUSS, OPTIONAL HEIGHT
TITLE TAPER, END DIAGONALS IN TENSION, SUBJECTED TO
TITLE UDL ON TOP & BOTTOM CHORDS, SHARED TO JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 19.04.05 ;TYPE PLANE TRUSS run=0 ;REFNO VM165
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*   Y                          ┌at centre      ├───┤1 bay
*   .────────.────────.─────  ─┼─  ─────.────────.────────.  ─┬─  ──►st
*   | \   /   \   /   \        d      /   \   /   \   / |   c
*   .───. X─────.────────.    ─┴─  .────────.────────.───.  ─┼─  ──►sb
*  _▲_   Warren lattice truss, even number of bays _▲_  └at ends
*  ///                                              ooo
*   ├──────────────────────a──────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2    d=3.2     ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
st=20    sb=15     ! Sway loading, top & bottom chord (+ve right).
ax(1)=75.8E-4 ax(4)=92.9E-4  ! Areas of bottom & top chords resp.
ax(2)=58.8E-4 ax(3)=47.4E-4  ! Area of verticals and diagonals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=nb+3 nj ;NUMBER OF MEMBERS nm=nb*2+3 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
IF nb/4<>INT(nb/4) AND d<>c THEN Quadruple number of bays reqd.
JOINT COORDINATES ;b=a/nb x(1)=0 x(nj)=a x(nj-1)=a x=-2*b j=0
:2 ;j=j+2 x=x+2*b x(j)=x x(j+1)=x+b ;IF j<nj-3 GOTO 2
y(1)=VEC(0)*nj yinc=(d-c)/(nb/4) y=c-yinc i=0 ;:4 ;i=i+2
IF i=(nj-1)/2+3 THEN yinc=-yinc ;y=y+yinc y(i)=y ;IF i<nj-1 GOTO 4
j=0 ;:5 ;j=j+1 j x(j) y(j) ;IF j<nj GOTO 5
JOINT RELEASES ;1 FORCE X -1 Y -1 ;nj FORCE Y -1
MEMBER INCIDENCES ;j=-1 m=-4 ;:10 ;j=j+2 m=m+4
js(m+1)=j je(m+1)=j+2 mt(m+1)=1 js(m+2)=j je(m+2)=j+1 mt(m+2)=3
js(m+3)=j+1 je(m+3)=j+2 mt(m+3)=3 js(m+4)=j+1 je(m+4)=j+3
mt(m+4)=4 ;IF j<nj-4 GOTO 10 ;mt(2)=2 js(nm-2)=nj-2 je(nm-2)=nj
mt(nm-2)=1 js(nm-1)=nj-2 je(nm-1)=nj-1 mt(nm-1)=3 js(nm)=nj-1
je(nm)=nj mt(nm)=2 m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
m=0 ;:20 ;m=m+1 mt=mt(m) m AX ax(mt) IZ 1 ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj h(nj-1)=VEC(st,sb)
wbp=wb*2*b wtp=wt*2*b j=0 ;:25 ;j=j+1 ;IF y(j)=0 THEN v(j)=wbp
IF y(j)<>0 THEN v(j)=wtp ;IF j<nj GOTO 25 ;v(1)=v(1)/4 v(2)=v(2)/2
v(3)=v(3)*3/4 v(nj-2)=v(3) v(nj-1)=v(2) v(nj)=v(1) j=0 ;:30
j=j+1 j FORCE Y v(j) ;IF j<nj GOTO 30 ;nj-1 FORCE X h(nj-1)
nj FORCE X h(nj) ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                        NL-STRESS     Classical       %age
*                                 analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i        +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm165.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

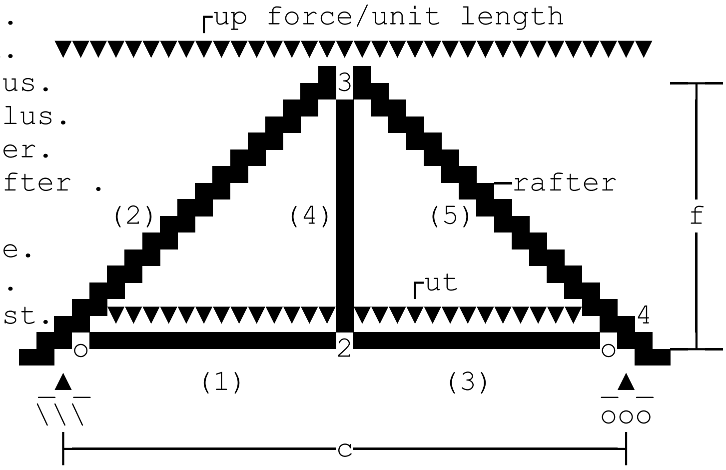
```
TITLE WARREN DECK LATTICE TRUSS WITH VERTICALS, OPTIONAL
TITLE HEIGHT TAPER, END DIAGONALS IN TENSION, SUBJECTED
TITLE TO UDL ON TOP & BOTTOM CHORDS, SHARED TO JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 20.04.05 ;TYPE PLANE TRUSS run=0 ;REFNO VM168
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;*/7
*   Y                        ┌at centre      ├───┼1 bay
*   .───.───.───.───.───.   ─┼─   .───.───.───.───.───.  ─┬─  ──►st
*   | \ | /   \ | /   \ |    d    | /   \ | /   \ | / |   c
*   .───. X───────.───────.   ─┴─   .───────.───────.───.  ─┼─  ──►sb
*  _▲_   Warren lattice truss, even number of bays _▲_ └at ends
*  ///                                             ooo
*   ├──────────────────────a───────────────────────┤
a=28     nb=8      ! Span of truss; even number of bays >=4.
c=2.2    d=3.2     ! Height bottom to top chords at ends & centre.
e=205E6  nu=0.3    ! Young's modulus & Poisson's ratio.
wt=-34   wb=-6     ! Plan loading, top & bottom chord (down -ve).
st=20    sb=15     ! Sway loading, top & bottom chord (+ve right).
ax(1)=75.8E-4 ax(4)=92.9E-4  ! Areas of bottom & top chords resp.
ax(2)=58.8E-4 ax(3)=47.4E-4  ! Area of verticals and diagonals.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=3*nb/2+3 nj ;NUMBER OF MEMBERS nm=nb*3+3 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
IF nb/4<>INT(nb/4) AND d<>c THEN Quadruple number of bays reqd.
JOINT COORDINATES ;b=a/nb x(1)=0 x(2)=0 x=-b j=0 ;:2 ;j=j+3
x=x+2*b x(j)=x x(j+1)=x x(j+2)=x+b ;IF j<nj-6 GOTO 2 ;x(nj-3)=a-b
x(nj-2)=a-b x(nj-1)=a x(nj)=a y(1)=VEC(0)*nj y(2)=c
yinc=(d-c)/(nb/2) y=c-yinc i=0 ;:4 ;i=i+3
IF i=(nj+1)/2+1 THEN y=y+2*yinc yinc=-yinc ;y=y+2*yinc y(i)=y
y(i+2)=y(i)+yinc ;IF i<nj-3 GOTO 4 ;y(nj)=0 j=0 ;:5 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 5 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js(1)=1 je(1)=4 mt(1)=1 js(2)=1
je(2)=2 mt(2)=2 js(3)=2 je(3)=4 mt(3)=3 js(4)=2 je(4)=3 mt(4)=4
j=0 m=-2 ;:10 ;j=j+3 m=m+6 js(m+1)=j je(m+1)=j+1 mt(m+1)=2
js(m+2)=j je(m+2)=j+2 mt(m+2)=4 js(m+3)=j+1 je(m+3)=j+4 mt(m+3)=1
js(m+4)=j+1 je(m+4)=j+2 mt(m+4)=3 js(m+5)=j+2 je(m+5)=j+4
mt(m+5)=3 js(m+6)=j+2 je(m+6)=j+3 mt(m+6)=4 ;IF j<nj-3 GOTO 10
je(nm-2)=nj js(nm)=nj-1 je(nm)=nj mt(nm)=2 m=0 ;:15
m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
m=0 ;:20 ;m=m+1 mt=mt(m) m AX ax(mt) IZ 1 ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;h(1)=VEC(0)*nj h(nj-1)=VEC(st,sb)
wbp=wb*2*b wtp=wt*b j=0 ;:25 ;j=j+1 ;IF y(j)=0 THEN v(j)=wbp
IF y(j)<>0 THEN v(j)=wtp ;IF j<nj GOTO 25 ;v(1)=v(1)/4 v(2)=v(2)/2
v(4)=v(4)*3/4 v(nj-2)=v(4) v(nj-1)=v(2) v(nj)=v(1) ;j=0 ;:30
j=j+1 j FORCE Y v(j) ;IF j<nj GOTO 30 ;nj-1 FORCE X h(nj-1)
nj FORCE X h(nj) ;SOLVE
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                    NL-STRESS     Classical       %age
*                             analysis      analysis        diff.
i=0 ;:90 ;i=i+1 nl(i)=ARR(13,i,1) d1=nl(i) p(i)=-p(i) d2=p(i)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +i     +nl(i)        +p(i)           $ok
IF i<nm GOTO 90 ;fnm=$(vm168.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE TWO RAFTERS WITH A SINGLE TIE FORMING A ROOF,
TITLE SUBJECTED TO UDL ON PLAN ON RAFTERS & TIE AND
TITLE WIND LOADS NORMAL TO THE RAFTERS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 20.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM171
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/14
c=3.5      ! Span of truss.
f=1.8      ! Ht. supp-apex.
e=9E6      ! Young's modulus.
g=e/16     ! Shearing modulus.
dr=0.250   ! Depth of rafter.
br=0.075   ! Breadth of rafter.
dt=0.225   ! Depth of tie.
bt=0.075   ! Breadth of tie.
up=-4.3    ! Udl on plan.
ut=-2.1    ! Udl on tie.
wlr=-1.8   ! Udl l. rafter
           ! +ve suction.
wrr=1.4    ! Udl r. rafter
           ! +ve suction.
```



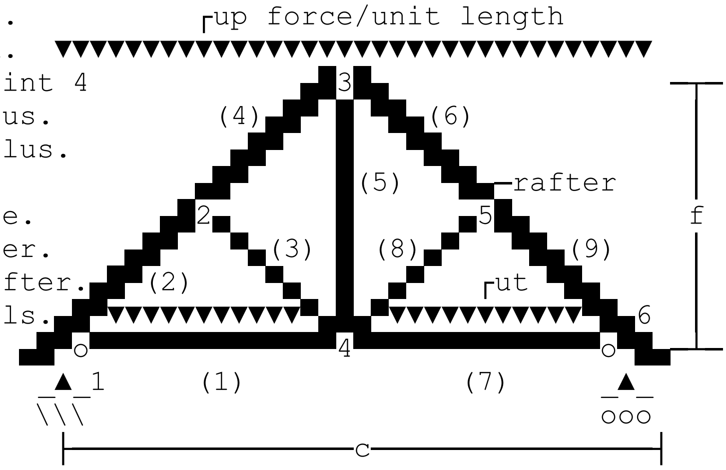

```
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=3 nj ;NUMBER OF MEMBERS nm=3 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE ;JOINT COORDINATES ;x(1)=0
y(1)=0 x(2)=c/2 y(2)=f x(3)=c y(3)=0 j=0 ;:5 ;j=j+1 j x(j) y(j)
IF j<nj GOTO 5 ;JOINT RELEASES ;1 FORCE X -1 Y -1 ;3 FORCE Y -1
MEMBER INCIDENCES ;js(1)=1 je(1)=3 js(2)=1 je(2)=2 js(3)=2 je(3)=3
m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
MEMBER RELEASES ;1 START MOMENT Z END MOMENT Z ;2 END MOMENT Z
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 RECTANGLE DY dt DZ bt ;2 THRU 3 RECTANGLE DY dr DZ br
LOADING CASE 1 ;MEMBER LOADS
2 THRU 3 FORCE Y PROJECTED UNIFORM W up
2 FORCE Y UNIFORM W wlr ;3 FORCE Y UNIFORM W wrr
1 FORCE Y UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/2 fwr=wrr*lr/2
v(1)=up*c/4+ut*c/2+fwl*COS(alp) h(1)=-fwl*SIN(alp)
v(3)=up*c/4+ut*c/2+fwr*COS(alp) h(3)=fwr*SIN(alp)
v(2)=up*c/2+(fwl+fwr)*COS(alp)  h(2)=-fwl*SIN(alp)+fwr*SIN(alp)
bp(1)=-ut*c^2/8 bp(2)=-(wlr+up*(0.5*c/lr)*COS(alp))*lr^2/8
bp(3)=-(wrr+up*(0.5*c/lr)*COS(alp))*lr^2/8
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                       NL-STRESS     Classical       %age
*                                analysis      analysis        diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m      +nl(m)        +p(m)           $ok
d1=bnl(m) d2=bp(m) ;#vmper.ndf !Compute percentage difference.
* Bending moment in member +m   +bnl(m)       +bp(m)          $ok
IF m<nm GOTO 90 ;fnm=$(vm171.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE TWO RAFTERS WITH TIE & KING-POST FORMING A ROOF
TITLE SUBJECTED TO UDL ON PLAN ON RAFTERS AND TIE,
TITLE AND WIND LOADS NORMAL TO THE RAFTERS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 21.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM172
METHOD STIFFNESS ;PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;*/14
c=3.5     ! Span of truss.
f=1.8     ! Ht. supp-apex.
e=9E6     ! Young's modulus.
g=e/16    ! Shearing modulus.
dr=0.250  ! Depth of rafter.
br=0.075  ! Breadth of rafter .
dt=0.225  ! Depth of tie.
bt=0.075  ! Breadth of tie.
dp=0.175  ! Depth of post.
bp=0.075  ! Breadth of post.
up=-4.3   ! Udl on plan.
ut=-2.1   ! Udl on tie.
wlr=-1.8  ! Udl l. rafter
wrr=1.4   ! Udl r. rafter
```



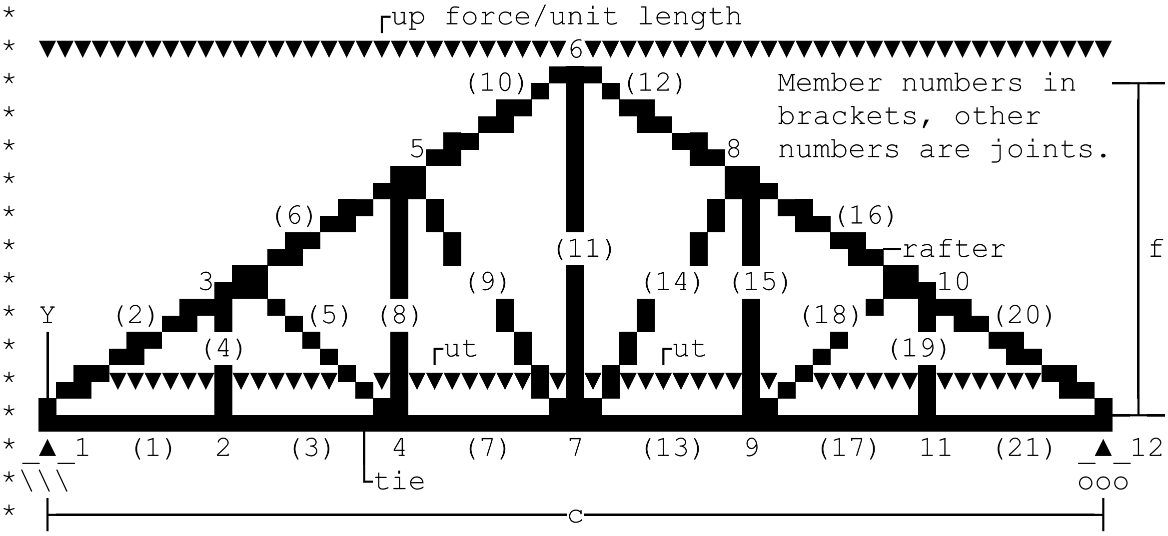

```
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=4 nj ;NUMBER OF MEMBERS nm=5 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE
JOINT COORDINATES ;x(1)=0 y(1)=0 x(2)=c/2 y(2)=0 x(3)=c/2 y(3)=f
x(4)=c y(4)=0 j=0 ;:5 ;j=j+1 j x(j) y(j) ;IF j<nj GOTO 5
JOINT RELEASES ;1 FORCE X -1 Y -1 ;4 FORCE Y -1 ;MEMBER INCIDENCES
js(1)=1 je(1)=2 js(2)=1 je(2)=3 js(3)=2 je(3)=4 js(4)=2 je(4)=3
js(5)=3 je(5)=4 m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
MEMBER RELEASES ;1 THRU 3 STEP 2 START MOMENT Z END MOMENT Z
2 END MOMENT Z ;5 START MOMENT Z ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 RECTANGLE DY dt DZ bt
2 RECTANGLE DY dr DZ br ;3 AS 1 ;4 RECTANGLE DY dp DZ bp ;5 AS 2
LOADING CASE 1 ;MEMBER LOADS
2 5 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 FORCE Y UNIFORM W wlr ;5 FORCE Y UNIFORM W wrr
1 3 INCLUSIVE FORCE Y UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/2 fwr=wrr*lr/2
v(1)=(up+ut)*c/4+fwl*COS(alp)    h(1)=-fwl*SIN(alp)
v(2)=ut*c/2                      h(2)=0
v(3)=up*c/2+(fwl+fwr)*COS(alp)   h(3)=(fwr-fwl)*SIN(alp)
v(4)=(up+ut)*c/4+fwr*COS(alp)    h(4)=fwr*SIN(alp)
bp(1)=-ut*(c/2)^2/8 bp(2)=-(wlr+up*(0.5*c/lr)*COS(alp))*lr^2/8
bp(3)=bp(1) bp(4)=0 bp(5)=-(wrr+up*(0.5*c/lr)*COS(alp))*lr^2/8
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m      +nl(m)        +p(m)           $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m   +bnl(m)       +bp(m)          $ok
IF m<nm GOTO 90 ;fnm=$(vm172.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE KING-POST ROOF TRUSS, WITH OPTIONAL RAISED CENTRE
TITLE FOR THE TIE, SUBJECTED TO UDL ON PLAN ON RAFTERS
TITLE AND TIE, WIND LOADS, AND VERTICAL LOADS ON JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 22.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM173
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/14
c=8.0      ! Span of truss.
f=4.2      ! Ht. supp-apex.
r4=0.5     ! Raised ht. joint 4
e=9E6      ! Young's modulus.
g=e/16     ! Shearing modulus.
d1=0.250   ! Depth of tie.
b1=0.075   ! Breadth of tie.
d2=0.225   ! Depth of rafter.
b2=0.075   ! Breadth of rafter.
d3=0.175   ! Depth internals.
b3=0.075   ! Breadth ints.
up=-4.3    ! Udl on plan.
ut=-2.1    ! Udl on tie.
wlr=-1.8   ! Udl l. rafter
wrr=1.4    ! Udl right rafter, +ve suction.
```




```
vjl(1)=VEC(-8,-16,-16,0,-16,-8)  ! Vert. joint loads e.g. purlins.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=6 nj ;NUMBER OF MEMBERS nm=9 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE
JOINT COORDINATES ;x(1)=0 y(1)=0 x(2)=c/4 y(2)=f/2 x(3)=c/2 y(3)=f
x(4)=c/2 y(4)=r4 x(5)=c*3/4 y(5)=f/2 x(6)=c y(6)=0 j=0 ;:5 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 5 ;JOINT RELEASES ;1 FORCE X -1 Y -1
6 FORCE Y -1 ;MEMBER INCIDENCES
js(1)=VEC(1,1,1)/4 js9=5 je1=VEC(4,2,4,3,4,5,6,5,6) m=0 ;:15
m=m+1 m js(m) je(m) ;IF m<nm GOTO 15 ;CONSTANTS E e ALL G g ALL
MEMBER RELEASES ;1 THRU 8 END MOMENT Z ;1 START MOMENT Z
3 5 7 INCLUSIVE START MOMENT Z ;MEMBER PROPERTIES
mt1=VEC(1,2,3,2,3,2,1,3,2) ;m=0 ;:20 ;m=m+1 mt=mt(m)
m RECTANGLE DY d(mt) DZ b(mt) ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:30 ;i=i+1 ;i FORCE Y vjl(i)
IF i<nj GOTO 30 ;MEMBER LOADS
2 4 6 9 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 4 INCLUSIVE FORCE Y UNIFORM W wlr
6 9 INCLUSIVE FORCE Y UNIFORM W wrr
1 7 INCLUSIVE FORCE Y PROJECTED UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/4 fwr=wrr*lr/4
v1=up*c/8+ut*c/4+fwl*COS(alp)+vjl(1) h1=-fwl*SIN(alp)
v2=up*c/4+2*fwl*COS(alp)+vjl(2) h2=-2*fwl*SIN(alp)
v3=up*c/4+fwl*COS(alp)+fwr*COS(alp)+vjl(3) h3=(fwr-fwl)*SIN(alp)
v4=ut*c/2+vjl(4) h4=0 v5=up*c/4+2*fwr*COS(alp)+vjl(5)
h5=2*fwr*SIN(alp) v6=up*c/8+ut*c/4+fwr*COS(alp)+vjl(6)
h6=fwr*SIN(alp) bp2=-(wlr+up*(0.5*c/lr)*COS(alp))*(lr/2)^2/8
bp1=-ut*(c/2)^2/8 bp3=0 bp4=bp2 bp5=0
bp6=-(wrr+up*(0.5*c/lr)*COS(alp))*(lr/2)^2/8 bp7=bp1 bp8=0 bp9=bp6
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) d2=-p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m      +nl(m)        +(-p(m))        $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m  +bnl(m)       +bp(m)          $ok
IF m<nm GOTO 90 ;fnm=$(vm173.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

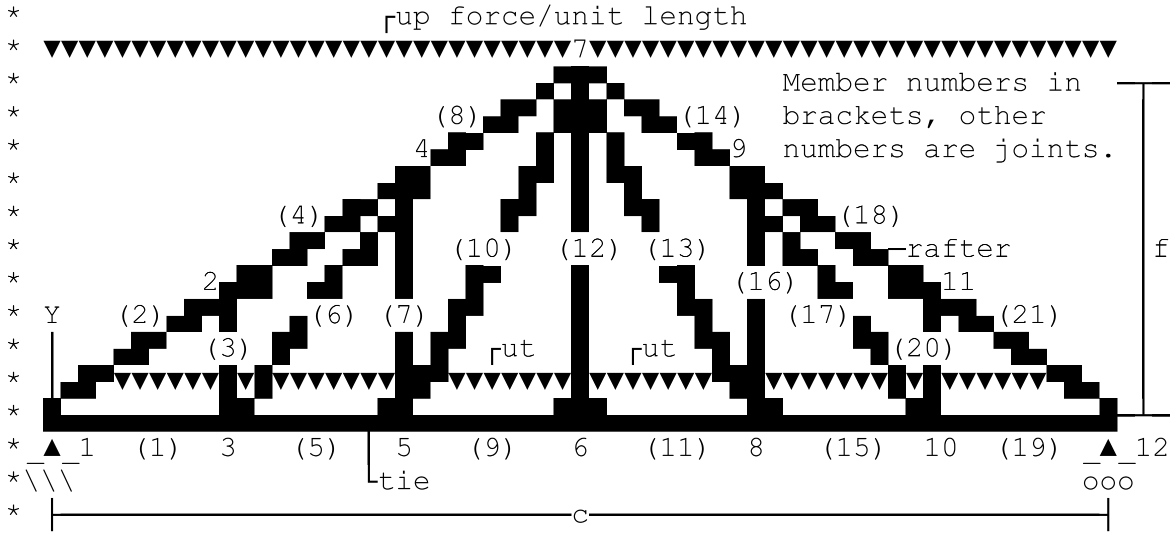
```
TITLE RIDGED ROOF TRUSS, WITH 3 SEGMENT RAFTERS, N/PRATT
TITLE INTERNALS, SUBJECTED TO UDL ON PLAN ON RAFTERS
TITLE AND TIE, WIND LOADS, AND VERTICAL LOADS ON JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 22.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM174
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/15
```




```
c=9.0     f=4.7      ! Span of truss, height to ridge.
r7=0.5               ! Raised height of joint 7.
e=9E6     g=e/16     ! Young's modulus & shearing modulus.
d1=0.250  b1=0.075   ! Depth of tie & breadth of tie.
d2=0.225  b2=0.075   ! Depth of rafter & breadth of rafter.
d3=0.175  b3=0.075   ! Depth of internals & breadth of internals.
up=-4.3   ut=-2.1    ! Uniformly distributed load on plan & tie.
wlr=-1.8  wrr=1.4    ! Udl on left & right rafters, +ve suction.
vjl(1)=-8 vjl(2)=VEC(-16)*10 vjl(12)=-8  ! Vertical joint loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=12 nj ;NUMBER OF MEMBERS nm=21 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE ;JOINT COORDINATES ;xinc=c/6
x0=VEC(0,0,xinc)/6 x12=c y1=0 y2=r7/3 y4=2*y2 y7=r7 y9=y4 y11=y2
y12=0 y3=f/3 y5=2*y3 y6=f y8=y5 y10=y3 j=0 ;:10 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js(1)=VEC(1,1,1)/10 js(21)=11
je(1)=VEC(2,3,4,3,4,5,7,5,7,6,7,8,9,8,9,10,11,10,11,12,12)
m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
CONSTANTS E e ALL G g ALL ;MEMBER RELEASES
1 4 5 8 9 11 14 15 18 19 INCLUSIVE START MOMENT Z END MOMENT Z
2 3 6 7 10 12 13 16 17 21 INCLUSIVE END MOMENT Z ;MEMB PROPERTIES
mt(1)=VEC(1,2,1,3,3,2,1,3,3,2,3,2,1,3,3,2,1,3,3,2,1) ;m=0 ;:20
m=m+1 mt=mt(m) m RECTANGLE DY d(mt) DZ b(mt) ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:30 ;i=i+1 ;i FORCE Y vjl(i)
IF i<nj GOTO 30 ;MEMBER LOADS
2 6 10 12 16 20 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 6 10 INCLUSIVE FORCE Y UNIFORM W wlr
12 16 20 INCLUSIVE FORCE Y UNIFORM W wrr
1 3 7 13 17 21 INCLUSIVE FORCE Y PROJECTED UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/6 fwr=wrr*lr/6
! Compute vertical joint loads ;v1=up*c/12+ut*c/12+fwl*COS(alp)
v2=ut*c/6 v3=up*c/6+2*fwl*COS(alp) v4=v2 v5=v3
v6=up*c/6+fwl*COS(alp)+fwr*COS(alp) v7=v2 v8=up*c/6+2*fwr*COS(alp)
v9=v2 v10=v8 v11=v2 v12=up*c/12+ut*c/12+fwr*COS(alp) ;j=0 ;:35
j=j+1 v(j)=v(j)+vjl(j) ;IF j<nj GOTO 35 ;h2=VEC(0)*10
h1=-fwl*SIN(alp) h3=-2*fwl*SIN(alp) h5=h3 h6=(fwr-fwl)*SIN(alp)
h8=2*fwr*SIN(alp) h10=h8 h12=h8/2
```

```
! Compute BM at centre of members ;bp1=VEC(0)*nm bp1=-ut*(c/6)^2/8
bp2=-(wlr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8 bp3=bp1 bp6=bp2
bp7=bp1 bp10=bp2 bp12=-(wrr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8
bp13=bp1 bp16=bp12 bp17=bp1 bp20=bp12 bp21=bp1
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m     +nl(m)        +p(m)           $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m  +bnl(m)       +bp(m)          $ok
IF m<nm GOTO 90 ;fnm=$(vm174.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

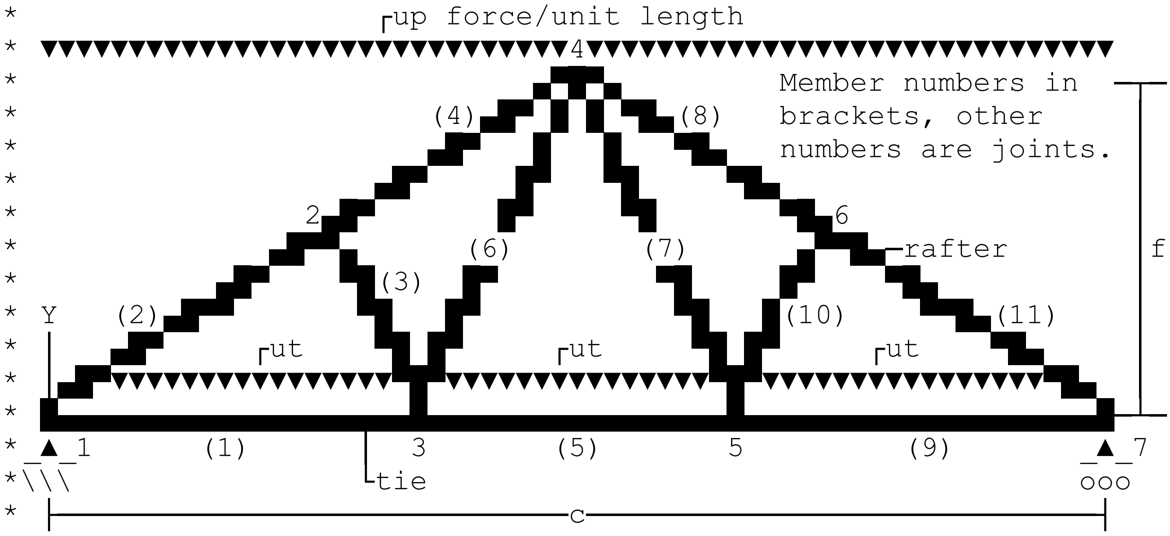
```
TITLE RIDGED ROOF TRUSS, WITH 3 SEGMENT RAFTERS, HOWE
TITLE INTERNALS, SUBJECTED TO UDL ON PLAN ON RAFTERS
TITLE AND TIE, WIND LOADS, AND VERTICAL LOADS ON JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 24.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM175
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/15
```




```
c=9.0     f=4.7      ! Span of truss, height to ridge.
r6=0.5               ! Raised height of joint 6.
e=9E6     g=e/16     ! Young's modulus & shearing modulus.
d1=0.250  b1=0.075   ! Depth of tie & breadth of tie.
d2=0.225  b2=0.075   ! Depth of rafter & breadth of rafter.
d3=0.175  b3=0.075   ! Depth of internals & breadth of internals.
up=-4.3   ut=-2.1    ! Uniformly distributed load on plan & tie.
wlr=-1.8  wrr=1.4    ! Udl on left & right rafters, +ve suction.
vjl(1)=-8 vjl(2)=VEC(-16)*10 vjl(12)=-8  ! Vertical joint loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=12 nj ;NUMBER OF MEMBERS nm=21 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE ;JOINT COORDINATES ;xinc=c/6
x0=VEC(0,0,xinc)/6 x12=c y1=0 y3=r6/3 y5=2*y3 y6=r6 y8=y5 y10=y3
y12=0 y2=f/3 y4=2*y2 y7=f y9=y4 y11=y2 j=0 ;:10 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js(1)=VEC(1,1,1)/10 js(21)=11
je(1)=VEC(3,2,3,4,5,4,5,7,6,7,8,7,8,9,10,9,10,11,12,11,12)
m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
CONSTANTS E e ALL G g ALL ;MEMBER RELEASES
1 3 6 7 10 12 13 16 17 20 INCLUSIVE START MOMENT Z END MOMENT Z
2 4 5 8 9 11 14 15 18 21 INCLUSIVE END MOMENT Z ;MEMB PROPERTIES
mt(1)=VEC(1,2,3,2,1,3,3,2,1,3,1,3,3,2,1,3,3,2,1,3,2) ;m=0 ;:20
m=m+1 mt=mt(m) m RECTANGLE DY d(mt) DZ b(mt) ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:30 ;i=i+1 ;i FORCE Y vjl(i)
IF i<nj GOTO 30 ;MEMBER LOADS
2 4 8 14 18 21 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 4 8 INCLUSIVE FORCE Y UNIFORM W wlr
14 18 21 INCLUSIVE FORCE Y UNIFORM W wrr
1 5 9 11 15 19 INCLUSIVE FORCE Y PROJECTED UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/6 fwr=wrr*lr/6
! Compute vertical joint loads ;v1=up*c/12+ut*c/12+fwl*COS(alp)
v3=ut*c/6 v2=up*c/6+2*fwl*COS(alp) v4=v2 v5=v3
v7=up*c/6+fwl*COS(alp)+fwr*COS(alp) v6=v3 v9=up*c/6+2*fwr*COS(alp)
v8=v3 v11=v9 v10=v3 v12=up*c/12+ut*c/12+fwr*COS(alp) ;j=0 ;:35
j=j+1 v(j)=v(j)+vjl(j) ;IF j<nj GOTO 35 ;h2=VEC(0)*10
h1=-fwl*SIN(alp) h2=-2*fwl*SIN(alp) h4=h2 h7=(fwr-fwl)*SIN(alp)
h9=2*fwr*SIN(alp) h11=h9 h12=h9/2
```

```
! Compute BM at centre of members ;bp1=VEC(0)*nm bp1=-ut*(c/6)^2/8
bp2=-(wlr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8 bp4=bp2 bp5=bp1
bp8=bp4 bp9=bp1 bp14=-(wrr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8
bp11=bp1 bp18=bp14 bp15=bp1 bp19=bp1 bp21=bp14
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                     NL-STRESS     Classical       %age
*                              analysis      analysis        diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m     +nl(m)        +p(m)           $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m  +bnl(m)       +bp(m)          $ok
IF m<nm GOTO 90 ;fnm=$(vm175.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

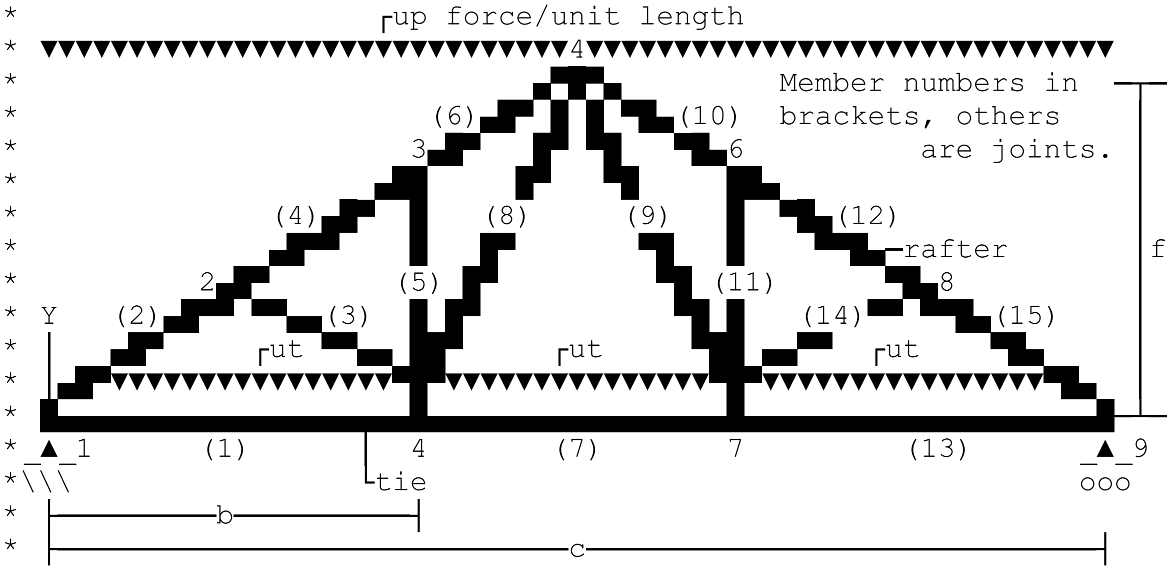
```
TITLE TRUSSED RAFTER (OR FINK) ROOF TRUSS WITH WARREN
TITLE INTERNALS, SUBJECTED TO UDL ON PLAN ON RAFTERS
TITLE AND TIE, WIND LOADS, AND VERTICAL LOADS ON JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 22.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM177
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/15
```




```
c=9.0     f=4.7      ! Span of truss, height to ridge.
r35=0.5              ! Raised height of joints 3 & 5.
e=9E6     g=e/16     ! Young's modulus & shearing modulus.
d1=0.250  b1=0.075   ! Depth of tie & breadth of tie.
d2=0.225  b2=0.075   ! Depth of rafter & breadth of rafter.
d3=0.175  b3=0.075   ! Depth of internals & breadth of internals.
up=-4.3   ut=-2.1    ! Uniformly distributed load on plan & tie.
wlr=-1.8  wrr=1.4    ! Udl on left & right rafters, +ve suction.
vjl(1)=-8 vjl(2)=VEC(-16)*5 vjl(7)=-8  ! Vertical joint loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=7 nj ;NUMBER OF MEMBERS nm=11 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE ;JOINT COORDINATES ;x1=0 x2=c/4
x3=c/3 x4=c/2 x5=c*2/3 x6=c*3/4 x7=c y1=0 y2=f/2 y3=r35 y4=f y5=y3
y6=f/2 y7=0 j=0 ;:10 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js(1)=VEC(1,1,1)/6 js(11)=6
je(1)=VEC(3,2,3,4,5,4,5,6,7,6,7) m=0 ;:15 ;m=m+1 m js(m) je(m)
IF m<nm GOTO 15 ;CONSTANTS E e ALL G g ALL ;MEMBER RELEASES
1 3 6 7 10 INCLUSIVE START MOMENT Z END MOMENT Z
2 4 5 8 9 INCLUSIVE END MOMENT Z ;MEMBER PROPERTIES
mt(1)=VEC(1,2,3,2,1,3,3,2,1,3,2) ;m=0 ;:20
m=m+1 mt=mt(m) m RECTANGLE DY d(mt) DZ b(mt) ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:30 ;i=i+1 ;i FORCE Y vjl(i)
IF i<nj GOTO 30 ;MEMBER LOADS
2 4 8 11 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 4 INCLUSIVE FORCE Y UNIFORM W wlr
8 11 INCLUSIVE FORCE Y UNIFORM W wrr
1 5 9 INCLUSIVE FORCE Y PROJECTED UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/4 fwr=wrr*lr/4
! Compute vertical joint loads ;v1=up*c/8+ut*c/6+fwl*COS(alp)
v3=ut*c/3 v2=up*c/4+2*fwl*COS(alp)
v4=up*c/4+fwl*COS(alp)+fwr*COS(alp) v5=v3 v6=up*c/4+2*fwr*COS(alp)
v7=up*c/8+ut*c/6+fwr*COS(alp) ;j=0 ;:35
j=j+1 v(j)=v(j)+vjl(j) ;IF j<nj GOTO 35 ;h2=VEC(0)*5
h1=-fwl*SIN(alp) h2=-2*fwl*SIN(alp) h4=(fwr-fwl)*SIN(alp)
h6=2*fwr*SIN(alp) h7=h6/2
! Compute BM at centre of members ;bp1=VEC(0)*nm bp1=-ut*(c/3)^2/8
bp2=-(wlr+up*(0.5*c/lr)*COS(alp))*(lr/2)^2/8 bp4=bp2 bp5=bp1
bp8=-(wrr+up*(0.5*c/lr)*COS(alp))*(lr/2)^2/8 bp9=bp1 bp11=bp8
```

```
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                    NL-STRESS     Classical      %age
*                             analysis      analysis       diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m     +nl(m)        +p(m)          $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m  +bnl(m)       +bp(m)         $ok
IF m<nm GOTO 90 ;fnm=$(vm177.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

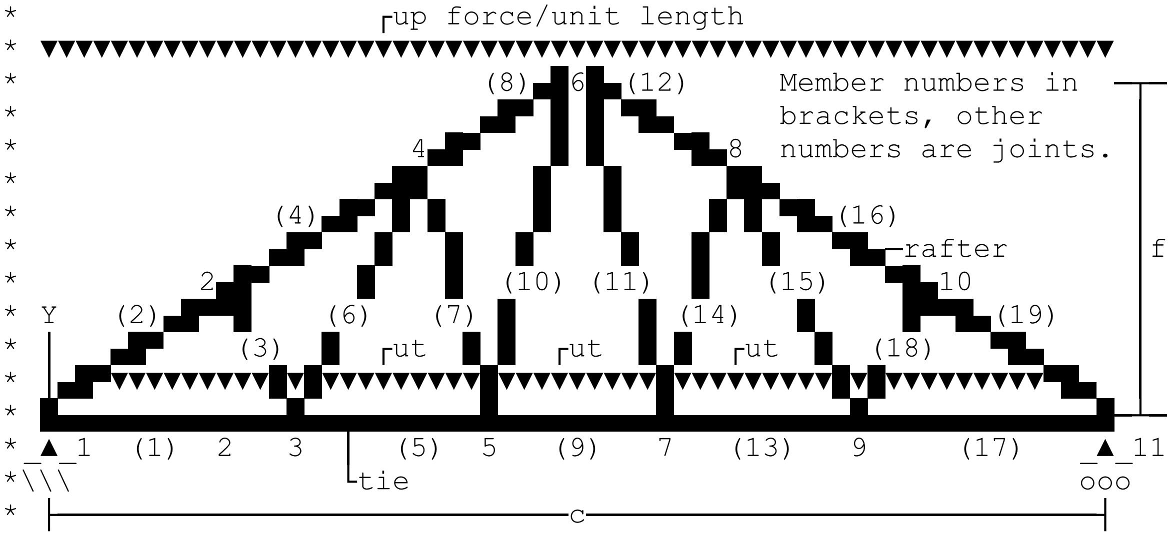
```
TITLE EACH RAFTER SUPPORTED BY ITS OWN TRUSS, 2 TRUSSES
TITLE TIED TOGETHER, SUBJECTED TO UDL ON PLAN ON RAFTERS
TITLE AND TIE, WIND LOADS, AND VERTICAL LOADS ON JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 27.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM178
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/15
```




```
c=9.0     f=4.7      ! Span of truss, height to ridge.
r47=0.5   b=3        ! Raised ht joints 4 & 7, dist. to joint 4.
e=9E6     g=e/16     ! Young's modulus & shearing modulus.
d1=0.250  b1=0.075   ! Depth of tie & breadth of tie.
d2=0.225  b2=0.075   ! Depth of rafter & breadth of rafter.
d3=0.175  b3=0.075   ! Depth of internals & breadth of internals.
up=-4.3   ut=-2.1    ! Uniformly distributed load on plan & tie.
wlr=-1.8  wrr=1.4    ! Udl on left & right rafters, +ve suction.
vjl(1)=-8 vjl(2)=VEC(-16)*7 vjl(9)=-8  ! Vertical joint loads.
#cc924.stk !Import verification data from cc924.stk if available.
IF b>0.46*c THEN Dimension b is out of range.
NUMBER OF JOINTS nj=9 nj ;NUMBER OF MEMBERS nm=15 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE ;JOINT COORDINATES ;x1=0 x2=c/6
x3=c/3 x4=b x5=c/2 x6=2*c/3 x7=c-b x8=5*c/6 x9=c y1=0 y2=f/3
y3=2*f/3 y4=r47 y5=f y6=y3 y7=y4 y8=y2 y9=0 j=0 ;:10 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js(1)=VEC(1,1,1)/7 js(15)=8
je(1)=VEC(4,2,4,3,4,5,7,5,7,6,7,8,9,8,9) m=0 ;:15 ;m=m+1
m js(m) je(m) ;IF m<nm GOTO 15 ;CONSTANTS E e ALL G g ALL
MEMBER RELEASES ;1 3 5 8 9 11 14 INCL START MOMENT Z END MOMENT Z
2 4 6 7 10 12 13 INCLUSIVE END MOMENT Z ;MEMBER PROPERTIES
mt(1)=VEC(1,2,3,2,3,2,1,3,3,2,3,2,1,3,2) ;m=0 ;:20
m=m+1 mt=mt(m) m RECTANGLE DY d(mt) DZ b(mt) ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:30 ;i=i+1 ;i FORCE Y vjl(i)
IF i<nj GOTO 30 ;MEMBER LOADS
2 4 6 10 12 15 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 4 6 INCLUSIVE FORCE Y UNIFORM W wlr
10 12 15 INCLUSIVE FORCE Y UNIFORM W wrr
1 7 13 INCLUSIVE FORCE Y PROJECTED UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/6 fwr=wrr*lr/6
! Compute vertical joint loads ;v1=up*c/12+ut*b/2+fwl*COS(alp)
b'=c-2*b v4=ut*(b+b')/2 v7=v4 v2=up*c/6+2*fwl*COS(alp) v3=v2
v5=up*c/6+fwl*COS(alp)+fwr*COS(alp) v6=up*c/6+2*fwr*COS(alp)
v8=v6 v9=up*c/12+ut*b/2+fwr*COS(alp) ;j=0 ;:35
j=j+1 v(j)=v(j)+vjl(j) ;IF j<nj GOTO 35 ;h2=VEC(0)*7
h1=-fwl*SIN(alp) h2=-2*fwl*SIN(alp) h3=h2 h5=(fwr-fwl)*SIN(alp)
h6=2*fwr*SIN(alp) h8=h6 h9=h6/2
```

```
! Compute BM at centre of members ;bp1=VEC(0)*nm bp1=-ut*b^2/8
bp2=-(wlr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8 bp4=bp2 bp6=bp2
bp7=-ut*b'^2/8 bp13=bp1
bp10=-(wrr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8 bp12=bp10
bp15=bp10
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                    NL-STRESS     Classical      %age
*                             analysis      analysis       diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m    +nl(m)        +p(m)          $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m  +bnl(m)      +bp(m)         $ok
IF m<nm GOTO 90 ;fnm=$(vm178.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

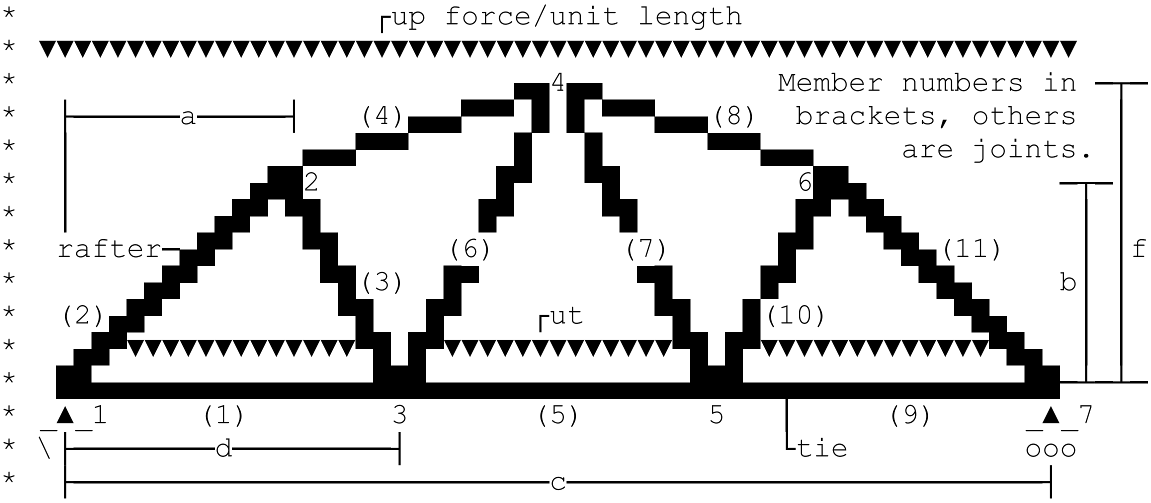
```
TITLE GANG-NAIL TYPE ROOF TRUSS, WITH 3 SEGMENT RAFTERS,
TITLE WARREN INTERNALS, SUBJECTED TO UDL ON PLAN ON RAFTERS
TITLE AND TIE, WIND LOADS, AND VERTICAL LOADS ON JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 22.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM179
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/15
```




```
c=9.0     f=4.7      ! Span of truss, height to ridge.
r57=0.5              ! Raised height of joints 5 & 7.
e=9E6     g=e/16     ! Young's modulus & shearing modulus.
d1=0.225  b1=0.075   ! Depth of tie & breadth of tie.
d2=0.250  b2=0.075   ! Depth of rafter & breadth of rafter.
d3=0.175  b3=0.075   ! Depth of internals & breadth of internals.
up=-4.3   ut=-2.1    ! Uniformly distributed load on plan & tie.
wlr=-1.8  wrr=1.4    ! Udl on left & right rafters, +ve suction.
vjl(1)=-8 vjl(2)=VEC(-16)*9 vjl(11)=-8  ! Vertical joint loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=11 nj ;NUMBER OF MEMBERS nm=19 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE ;JOINT COORDINATES ;xinc=c/5
j=-1 x=-xinc ;:5 ;j=j+2 x=x+xinc x(j)=x ;IF j<11 GOTO 5
xinc=c/6 j=0 x=0 ;:7 ;j=j+2 x=x+xinc x(j)=x ;IF j<10 GOTO 7
y1=0 y2=f/3 y3=r57/2 y4=2*y2 y5=r57 y6=f y7=y5 y8=y4 y9=y3
y10=y2 y11=0 j=0 ;:10 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js(1)=VEC(1,1,1)/9 js(19)=10
je(1)=VEC(3,2,3,4,5,4,5,6,7,6,7,8,9,8,9,10,11,10,11)
m=0 ;:15 ;m=m+1 m js(m) je(m) ;IF m<nm GOTO 15
CONSTANTS E e ALL G g ALL ;MEMBER RELEASES
1 3 6 7 10 11 14 15 18 INCLUSIVE START MOMENT Z END MOMENT Z
2 4 5 8 9 12 13 16 17 INCLUSIVE END MOMENT Z ;MEMBER PROPERTIES
mt(1)=VEC(1,2,3,2,1,3,3,2,1,3,3,2,1,3,3,2,1,3,2) ;m=0 ;:20
m=m+1 mt=mt(m) m RECTANGLE DY d(mt) DZ b(mt) ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:30 ;i=i+1 ;i FORCE Y vjl(i)
IF i<nj GOTO 30 ;MEMBER LOADS
2 4 8 12 16 19 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 4 8 INCLUSIVE FORCE Y UNIFORM W wlr
12 16 19 INCLUSIVE FORCE Y UNIFORM W wrr
1 5 9 13 17 INCLUSIVE FORCE Y PROJECTED UNIFORM W ut ;SOLVE
alp=ATN(2*f/c) lr=0.5*c/COS(alp) fwl=wlr*lr/6 fwr=wrr*lr/6
! Compute vertical joint loads ;v1=up*c/12+ut*c/10+fwl*COS(alp)
v3=ut*c/5 v2=up*c/6+2*fwl*COS(alp) v4=v2 v5=v3
v6=up*c/6+fwl*COS(alp)+fwr*COS(alp) v7=v3 v8=up*c/6+2*fwr*COS(alp)
v9=v3 v10=v8 v11=up*c/12+ut*c/10+fwr*COS(alp) ;j=0 ;:35
j=j+1 v(j)=v(j)+vjl(j) ;IF j<nj GOTO 35 ;h2=VEC(0)*9
h1=-fwl*SIN(alp) h2=-2*fwl*SIN(alp) h4=h2 h6=(fwr-fwl)*SIN(alp)
h8=2*fwr*SIN(alp) h10=h8 h11=h8/2
```

```
! Compute BM at centre of members ;bp1=VEC(0)*nm bp1=-ut*(c/5)^2/8
bp2=-(wlr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8 bp5=bp1 bp4=bp2
bp8=bp2 bp9=bp1 bp12=-(wrr+up*(0.5*c/lr)*COS(alp))*(lr/3)^2/8
bp13=bp1 bp16=bp12 bp17=bp1 bp16=bp12 bp19=bp12
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                     NL-STRESS     Classical       %age
*                              analysis      analysis        diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m     +nl(m)        +p(m)           $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m  +bnl(m)       +bp(m)          $ok
IF m<nm GOTO 90 ;fnm=$(vm179.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

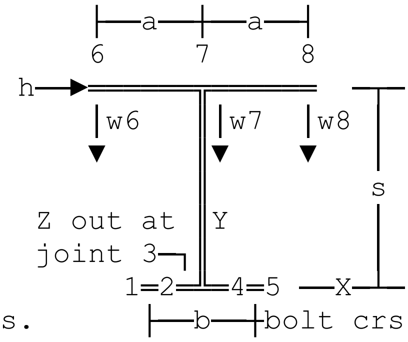
```
TITLE MANSARD (OR HOG BACK) ROOF TRUSS WITH WARREN
TITLE INTERNALS, SUBJECTED TO UDL ON PLAN ON RAFTERS
TITLE AND TIE, WIND LOADS, AND VERTICAL LOADS ON JOINTS.
TITLE STIFFNESS METHOD CHECKED BY THE 'METHOD OF JOINTS'.
MADEBY DWB ;DATE 22.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM181
METHOD STIFFNESS ;PRINT DATA, RESULTS, FROM 1 ;TABULATE ALL ;*/15
```




```
c=9.0      f=4.7       ! Span of truss, height to ridge.
a=2.0      b=4         ! Dimensions to locate joints 2 & 6.
d=3.0      r35=0.5     ! Dim. to joint 3, raised height of 3 & 5.
e=9E6      g=e/16      ! Young's modulus & shearing modulus.
d1=0.225   b1=0.075    ! Depth of tie & breadth of tie.
d2=0.250   b2=0.075    ! Depth of rafter & breadth of rafter.
d3=0.175   b3=0.075    ! Depth of internals & breadth of internals.
up=-4.3    ut=-2.1     ! Uniformly distributed load on plan & tie.
wlr=-1.8   wrr=1.4     ! Udl normal to l. & r. rafters, +ve suction.
vjl(1)=-8 vjl(2)=VEC(-16)*5 vjl(7)=-8  ! Vertical joint loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=7 nj ;NUMBER OF MEMBERS nm=11 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF SEGMENTS nsg=2 nsg TRACE ;JOINT COORDINATES ;x1=0 x2=a
x3=d x4=c/2 x5=c-d x6=c-a x7=c y1=0 y2=b y3=r35 y4=f y5=y3
y6=b y7=0 j=0 ;:10 ;j=j+1
j x(j) y(j) ;IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;MEMBER INCIDENCES ;js(1)=VEC(1,1,1)/6 js(11)=6
je(1)=VEC(3,2,3,4,5,4,5,6,7,6,7) m=0 ;:15 ;m=m+1 m js(m) je(m)
IF m<nm GOTO 15 ;CONSTANTS E e ALL G g ALL ;MEMBER RELEASES
1 3 6 7 10 INCLUSIVE START MOMENT Z END MOMENT Z
2 4 5 8 9 INCLUSIVE END MOMENT Z ;MEMBER PROPERTIES
mt(1)=VEC(1,2,3,2,1,3,3,2,1,3,2) ;m=0 ;:20
m=m+1 mt=mt(m) m RECTANGLE DY d(mt) DZ b(mt) ;IF m<nm GOTO 20
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:30 ;i=i+1 ;i FORCE Y vjl(i)
IF i<nj GOTO 30 ;MEMBER LOADS
2 4 8 11 INCLUSIVE FORCE Y PROJECTED UNIFORM W up
2 4 INCLUSIVE FORCE Y UNIFORM W wlr
8 11 INCLUSIVE FORCE Y UNIFORM W wrr
1 5 9 INCLUSIVE FORCE Y PROJECTED UNIFORM W ut ;SOLVE
alp=ATN(b/a) lr1=a/COS(alp) fwl1=wlr*lr1/2 fwr1=wrr*lr1/2
a'=c/2-a d'=c-2*d bet=ATN((f-b)/a') lr2=a'/COS(bet) fwl2=wlr*lr2/2
fwr2=wrr*lr2/2 !Compute vertical joint loads
v1=up*a/2+ut*d/2+fwl1*COS(alp)
v2=up*(a/2+a'/2)+fwl1*COS(alp)+fwl2*COS(bet) v3=ut*(d/2+d'/2)
v4=up*a'+fwl2*COS(bet)+fwr2*COS(bet) v5=v3
v6=up*(a/2+a'/2)+fwr1*COS(alp)+fwr2*COS(bet)
v7=up*a/2+ut*d/2+fwr1*COS(alp) ;j=0 ;:35 ;j=j+1 v(j)=v(j)+vjl(j)
IF j<nj GOTO 35 ;h2=VEC(0)*5 ;h1=-fwl1*SIN(alp)
h2=-fwl1*SIN(alp)-fwl2*SIN(bet) h4=(fwr2-fwl2)*SIN(bet)
h6=fwr1*SIN(alp)+fwr2*SIN(bet) h7=fwr1*SIN(alp)
```

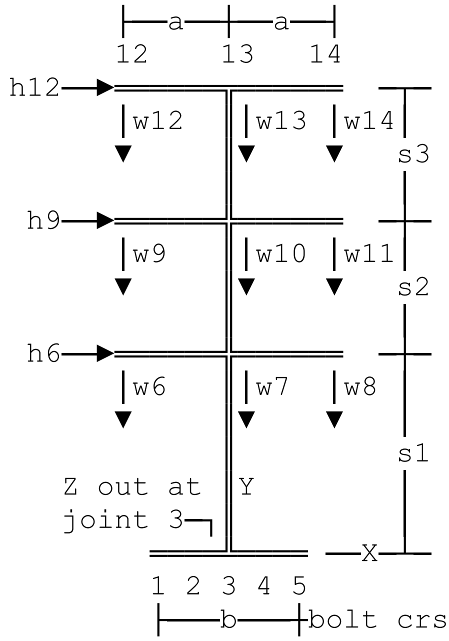
```
! Compute BM at centre of members ;bp1=VEC(0)*nm bp1=-ut*d^2/8
bp2=-(wlr+up*(a/lr1)*COS(alp))*lr1^2/8
bp4=-(wlr+up*(a'/lr2)*COS(bet))*lr2^2/8 bp5=-ut*d'^2/8
bp8=-(wrr+up*(a'/lr2)*COS(bet))*lr2^2/8 bp9=bp1
bp11=-(wrr+up*(a/lr1)*COS(alp))*lr1^2/8
#vmmoj.ndf !Compute axial forces by 'Method of Joints'.
* Location                     NL-STRESS     Classical       %age
*                              analysis      analysis        diff.
m=0 ;:90 ;m=m+1 i=nsg*(m-1)+1 fs=ARR(13,i,1) bnl(m)=ARR(13,i,6)
i=i+nsg-1 fe=ARR(13,i,4) nl(m)=(fs-fe)/2 d1=nl(m) p(m)=-p(m) d2=p(m)
#vmper.ndf !Compute percentage difference & any message.
* Axial force in member +m     +nl(m)        +p(m)           $ok
d1=bnl(m) d2=bp(m)
#vmper.ndf !Compute percentage difference & any message.
* Bending moment in member +m  +bnl(m)       +bp(m)          $ok
IF m<nm GOTO 90 ;fnm=$(vm181.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE RHS PIPE TREE WITH TWO HORIZONTAL BRANCHES,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM202
TYPE PLANE FRAME run=0 ;DATE 09.09.05 ;NUMBER OF JOINTS 8
NUMBER OF MEMBERS 7 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/10
s=1.8        ! Height of tree.
a=0.3        ! Branch length.
nsg=8        ! Number of segments.
e=205E6      ! Young's modulus.
nu=0.3       ! Poisson's ratio.
h=5.2        ! Horizontal load, +ve right.
w6=-1.0      ! Vertical load, -ve down.
w7=-8.5      ! Vertical load, -ve down.
w8=-23.6     ! Vertical load, -ve down.
b=0.40 bpy=0.040 bpz=0.30 ! Base plate sizes.
dyt=0.25 dzt=0.15 tt=0.010 rt=0.010 ! RHS trunk sizes, rt=radius.
dyb=0.15 dzb=0.15 tb=0.010 rb=0.010 ! RHS branch sizes, rb=radius.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 -b/2 0 ;5 b/2 0
2 THRU 4 X -dyt/2 Y 0 XL dyt/2 ;6 THRU 8 X -a Y s XL a
JOINT RELEASES ;1 THRU 5 STEP 2 FORCE Y -1 ;3 FORCE X -1
MEMBER INCIDENCES ;1 THRU 4 RANGE 1,2  4,5 ;5 3 7
6 THRU 7 CHAIN 6,7,8 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 RECT DY bpy DZ bpz ;2 RECT DY dyt/4 DZ bpz
3 THRU 4 AS 2 THRU 1 ;5 RECTANGLE DY dyt DZ dzt T tt R rt
6 THRU 7 RECTANGLE DY dyb DZ dzb T tb R rb
LOADING CASE 1 ;JOINT LOADS ;w1=VEC(0)*5 j=5 ;:5 ;j=j+1
j FORCE Y w(j) ;IF j<8 GOTO 5 ;6 FORCE X h
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=8 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=VEC(0)*7 vc1=VEC(0)*7 vjl1=VEC(0)*5 vjl6=VEC(w6,w7,w8)
hjl1=VEC(0)*8 hjl6=h ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm202.stk) ;#vmres.ndf !Conclude results.
mjn=6 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

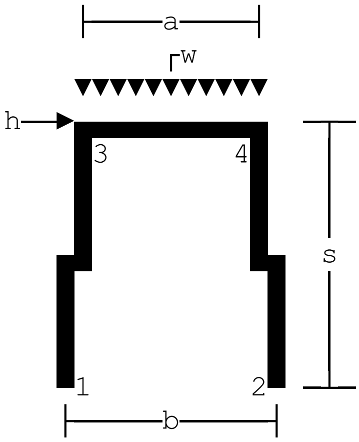


```
TITLE RHS PIPE TREE WITH TWO HORIZONTAL BRANCHES,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM203
TYPE PLANE FRAME run=0 ;DATE 09.09.05 ;NUMBER OF JOINTS 11
NUMBER OF MEMBERS 10 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/15
s1=1.8       ! Height of 1st storey.           ├──a──┼──a──┤
s2=0.6       ! Height of 2nd storey.           9     10    11
a=0.3        ! Branch length.             h9──►════════╦══════   ┬──
nsg=8        ! Number of segments.             │w9     ║│w10 │w11│
e=205E6      ! Young's modulus.                ▼       ║▼    ▼   s2
nu=0.3       ! Poisson's ratio.                        ║         │
h6=5.2       ! Horiz. load, at joint 6.   h6──►════════╬══════  ─┼─
h9=5.2       ! Horiz. load, at joint 9.        │w6     ║│w7  │w8 │
w6=-1.0      ! Vertical load, -ve down.        ▼       ║▼    ▼   │
w7=-8.5      ! Vertical load, -ve down.                ║         s1
w8=-23.6     ! Vertical load, -ve down.     Z out at   ║Y        │
w9=-1.0      ! Vertical load, -ve down.     joint 3─┐  ║         │
w10=-8.5     ! Vertical load, -ve down.          ═════╩═════  ──X──┴─
w11=-23.6    ! Vertical load, -ve down.          1 2 3 4 5
b=0.40 bpy=0.040 bpz=0.30 ! Base plate sizes.    ├───b───┼bolt crs
dyt=0.25 dzt=0.15 tt=0.010 rt=0.010 ! RHS trunk sizes, rt=radius.
dyb=0.15 dzb=0.15 tb=0.010 rb=0.010 ! RHS branch sizes, rb=radius.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 -b/2 0 ;5 b/2 0
2 THRU 4 X -dyt/2 Y 0 XL dyt/2 ;6 THRU 8 X -a Y s1 XL a
9 THRU 11 X -a Y s1+s2 XL a
JOINT RELEASES ;1 THRU 5 STEP 2 FORCE Y -1 ;3 FORCE X -1
MEMBER INCIDENCES ;1 THRU 4 RANGE 1,2 4,5 ;5 3 7
6 THRU 7 CHAIN 6,7,8 ;8 7 10 ;9 THRU 10 CHAIN 9,10,11
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 RECT DY bpy DZ bpz ;2 RECT DY dyt/4 DZ bpz
3 THRU 4 AS 2 THRU 1 ;5 THRU 8 STEP 3 RECT DY dyt DZ dzt T tt R rt
6 THRU 7 RECTANGLE DY dyb DZ dzb T tb R rb ;9 THRU 10 AS 6
LOADING CASE 1 ;JOINT LOADS ;w1=VEC(0)*5 j=5 ;:5 ;j=j+1
j FORCE Y w(j) ;IF j<11 GOTO 5 ;6 FORCE X h6 ;9 FORCE X h9
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=11 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=VEC(0)*10 vc1=VEC(0)*10 hjl1=VEC(0)*11 hjl6=h6 hjl9=h9
vjl1=VEC(0)*5 vjl6=VEC(w6,w7,w8,w9,w10,w11) ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm203.stk) ;#vmres.ndf !Conclude results.
mjn=6 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE RHS PIPE TREE WITH TWO HORIZONTAL BRANCHES,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM204
TYPE PLANE FRAME run=0 ;DATE 09.09.05 ;NUMBER OF JOINTS 14
NUMBER OF MEMBERS 13 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/19
s1=1.8        ! Height of 1st storey.
s2=0.6        ! Height of 2nd storey.
s3=0.6        ! Height of 2nd storey.
a=0.3         ! Branch length.
nsg=8         ! Number of segments.
e=205E6       ! Young's modulus.
nu=0.3        ! Poisson's ratio.
h6=3.2        ! Horiz. load, at joint 6.
h9=3.2        ! Horiz. load, at joint 9.
h12=3.2       ! Horiz. load, at joint 12.
w6=-1.0       ! Vertical load, -ve down.
w7=-8.5       ! Vertical load, -ve down.
w8=-23.6      ! Vertical load, -ve down.
w9=-1.0       ! Vertical load, -ve down.
w10=-8.5      ! Vertical load, -ve down.
w11=-23.6     ! Vertical load, -ve down.
w12=-1.0      ! Vertical load, -ve down.
w13=-8.5      ! Vertical load, -ve down.
w14=-23.6     ! Vertical load, -ve down.
b=0.40 bpy=0.040 bpz=0.30 ! Base plate sizes.
dyt=0.25 dzt=0.15 tt=0.010 rt=0.010 ! RHS trunk sizes, rt=radius.
dyb=0.15 dzb=0.15 tb=0.010 rb=0.010 ! RHS branch sizes, rb=radius.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 -b/2 0 ;5 b/2 0
2 THRU 4 X -dyt/2 Y 0 XL dyt/2 ;6 THRU 8 X -a Y s1 XL a
9 THRU 11 X -a Y s1+s2 XL a ;12 THRU 14 X -a Y s1+s2+s3 XL a
JOINT RELEASES ;1 THRU 5 STEP 2 FORCE Y -1 ;3 FORCE X -1
MEMBER INCIDENCES ;1 THRU 4 RANGE 1,2 4,5 ;5 3 7
6 THRU 7 CHAIN 6,7,8 ;8 7 10 ;9 THRU 10 CHAIN 9,10,11
11 10 13 ;12 THRU 13 CHAIN 12,13,14
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 RECT DY bpy DZ bpz ;2 RECT DY dyt/4 DZ bpz ;3 THRU 4 AS 2 THRU 1
5 THRU 11 STEP 3 RECT DY dyt DZ dzt T tt R rt
6 7 9 10 12 13 INCLUSIVE RECTANGLE DY dyb DZ dzb T tb R rb
LOADING CASE 1 ;JOINT LOADS ;w1=VEC(0)*5 j=5 ;:5 ;j=j+1
j FORCE Y w(j) ;IF j<14 GOTO 5 ;6 FORCE X h6 ;9 FORCE X h9
12 FORCE X h12
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=14 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=VEC(0)*13 vc1=VEC(0)*13 hjl1=VEC(0)*14 hjl6=h6 hjl9=h9
hjl12=h12 vjl1=VEC(0)*5
vjl6=VEC(w6,w7,w8,w9,w10,w11,w12,w13,w14) ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm204.stk) ;#vmres.ndf !Conclude results.
mjn=6 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

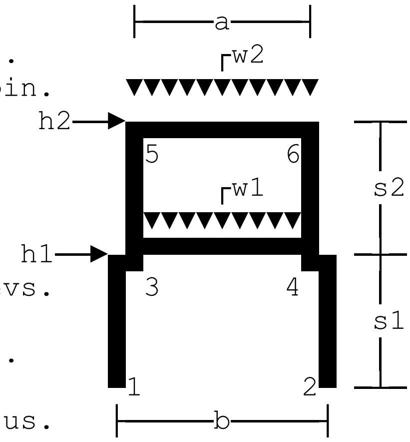


```
TITLE ONE STOREY BENT WITH VERTICAL OR RAKING COLUMNS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM207
TYPE PLANE FRAME run=0 ;DATE 15.09.05 ;NUMBER OF JOINTS 4
NUMBER OF MEMBERS 3 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/13
```

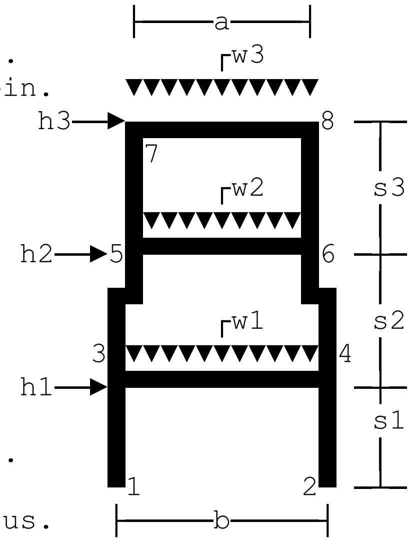



```
s=2.6      ! Storey height.
a=1.0      ! Top dimension.
b=1.5      ! Bottom dimension.
fix=0      ! Base fixity, 1=fix, 0=pin.
nsg=8      ! Number of segments.
e=205E6    ! Young's modulus.
nu=0.3     ! Poisson's ratio.
w=-23.6    ! UDL on transom.
h=12       ! Sway load.
dyc=0.25 dzc=0.15 ! RHS column dimensions.
tc=0.010 rc=0.010 ! Col. thickness & radius.
dyt=0.25 dzt=0.15 ! RHS transom dimensions.
tt=0.010 rt=0.010 ! Trans. thickness & radius.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg
JOINT COORDINATES ;1 THRU 2 X 0 Y 0 XL b SUPPORT
3 THRU 4 X x=(b-a)/2 x Y s XL b-x ;IF fix=0 ;JOINT RELEASES
1 THRU 2 MOMENT Z ;ENDIF ;MEMBER INCIDENCES ;1 1 3 ;2 2 4 ;3 3 4
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU 2 RECTANGLE DY dyc DZ dzc T tc R rc
3 RECTANGLE DY dyt DZ dzt T tt R rt
LOADING CASE 1 ;JOINT LOADS ;3 FORCE X h ;MEMBER LOADS
3 FORCE Y UNIFORM W w
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=4 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=VEC(0)*3 vc1=VEC(0,0,w) vjl1=VEC(0)*4
hjl1=VEC(0,0,h,0) ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm207.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE TWO STOREY BENT WITH VERTICAL OR RAKING COLUMNS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM208
TYPE PLANE FRAME run=0 ;DATE 15.09.05 ;NUMBER OF JOINTS 6
NUMBER OF MEMBERS 6 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/13
s1=4.2   s2=3.6   ! Storey heights.
a=1.0    b=2.4    ! Top & bottom dimensions.
fix=0             ! Base fixity, 1=fix, 0=pin.
nsg=8             ! Number of segments.
e=205E6           ! Young's modulus.
nu=0.3            ! Poisson's ratio.
w1=-23.6          ! UDl at 1st level.
w2=-23.6          ! UDL at 2nd level.
h1=12    h2=6     ! Sway loads 1st & 2nd levs.
dyc=0.25 dzc=0.15 ! RHS column dimensions.
tc=0.010 rc=0.010 ! Col. thickness & radius.
dyt=0.25 dzt=0.15 ! RHS transom dimensions.
tt=0.010 rt=0.010 ! Trans. thickness & radius.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES
1 THRU 2 X 0 Y 0 XL b SUPPORT ;c=a+(b-a)*s2/(s1+s2) x=(b-c)/2
3 THRU 4 X x Y s1 XL b-x ;5 THRU 6 X x=(b-a)/2 x Y s1+s2 XL b-x
IF fix=0 ;JOINT RELEASES ;1 THRU 2 MOMENT Z ;ENDIF
MEMBER INCIDENCES ;1 1 3 ;2 2 4 ;3 3 4 ;4 3 5 ;5 4 6 ;6 5 6
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU 2 RECTANGLE DY dyc DZ dzc T tc R rc
3 RECTANGLE DY dyt DZ dzt T tt R rt ;4 THRU 6 AS 1 THRU 3
LOADING CASE 1 ;JOINT LOADS ;3 FORCE X h1 ;5 FORCE X h2
MEMBER LOADS ;3 FORCE Y UNIFORM W w1 ;6 FORCE Y UNIFORM W w2
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=6 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=VEC(0)*6 vc1=VEC(0,0,w1,0,0,w2) vjl1=VEC(0)*6
hjl1=VEC(0,0,h1,0,h2,0) ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm208.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```



```
TITLE THREE STOREY BENT WITH VERTICAL OR RAKING COLUMNS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM209
TYPE PLANE FRAME run=0 ;DATE 16.09.05 ;NUMBER OF JOINTS 8
NUMBER OF MEMBERS 9 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/16
s1=4.2   s2=1.8   s3=1.8  ! Storey heights.
a=1.0    b=2.4    ! Top & bottom dimensions.
fix=0             ! Base fixity, 1=fix, 0=pin.
nsg=8             ! Number of segments.
e=205E6           ! Young's modulus.
nu=0.3            ! Poisson's ratio.
w1=-23.6          ! UDl at 1st level.
w2=-23.6          ! UDL at 2nd level.
w3=-23.6          ! UDL at 3rd level.
h1=12             ! Sway load at 1st level.
h2=12             ! Sway load at 2nd level.
h3=6              ! Sway load at 3rd level.
dyc=0.25 dzc=0.15 ! RHS column dimensions.
tc=0.010 rc=0.010 ! Col. thickness & radius.
dyt=0.25 dzt=0.15 ! RHS transom dimensions.
tt=0.010 rt=0.010 ! Trans. thickness & radius.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;s=s1+s2+s3 c=(b-a)/2
1 THRU 2 X 0 Y 0 XL b SUPPORT ;3 THRU 4 X x=c*s1/s x Y s1 XL b-x
5 THRU 6 X x=c*(s1+s2)/s x Y s1+s2 XL b-x
7 THRU 8 X c Y s XL b-c ;IF fix=0 ;JOINT RELEASES ;1 THRU 2 MOME Z
ENDIF ;MEMBER INCIDENCES ;1 THRU 7 STEP 3 RANGE 1,3 5,7
2 THRU 8 STEP 3 RANGE 2,4 6,8 ;3 THRU 9 STEP 3 RANGE 3,4 7,8
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU 2 RECTANGLE DY dyc DZ dzc T tc R rc
3 RECTANGLE DY dyt DZ dzt T tt R rt ;4 THRU 6 AS 1 THRU 3
7 THRU 9 AS 1 THRU 3
LOADING CASE 1 ;JOINT LOADS ;3 FORCE X h1 ;5 FORCE X h2
7 FORCE X h3 ;MEMBER LOADS ;3 FORCE Y UNIFORM W w1
6 FORCE Y UNIFORM W w2 ;9 FORCE Y UNIFORM W w3
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=8 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=VEC(0)*9 vc1=VEC(0,0,w1,0,0,w2,0,0,w3) vjl1=VEC(0)*8
hjl1=VEC(0,0,h1,0,h2,0,0,h3,0) ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm209.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```



```
TITLE RECTANGULAR PORTAL FRAME, OR BENT, WITH
TITLE FIXED FEET, SUBJECTED TO VERTICAL AND SWAY LOADS
TITLE INCLUDING A CHECK OF RESULTS AGAINST THE COLUMN
TITLE ANALOGY METHOD DEVISED BY PROF. HARDY CROSS.
MADEBY DWB ;DATE 31.10.04 ;TABULATE ALL ;REFNO VM210
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE FRAME run=0
NUMBER OF JOINTS 4 ;NUMBER OF MEMBERS 3 ;NUMBER OF SUPPORTS 2
NUMBER OF LOADINGS 1 ;*/10
*          ├─b(n)─┤fy(n) kN, for n'th load, -ve down.
*                 ▼
* ──┬──  fx──►╔═════════════════════╗       Member numbers are in
*   │    kN   ║2        (2)        3║       brackets, other numbers
*   │         ║                     ║       are joints.
* h │     (1) ║                     ║(3)
*   │         ║                     ║       Section properties for
*   │         ║                     ║       member 3 are as for
* ──┴──      1╨  ──X                ╨4      member 1.
*             ├──────────l──────────┤
h=8  l=12  fx=0  nvl=1  ! Sizes, sway load kN, No. of vert. loads.
e=205E6 nu=.3    ! Young's mod; nu=Poisson, 0 to ignore shear def.
ax1=47.5E-4 ay1=16E-4 iz1=7162E-8  ! Area/shear area m2 inert m4.
ax2=47.5E-4 ay2=16E-4 iz2=7162E-8  ! Area/shear area m2 inert m4.
dp1=.304 dp2=.304                  ! Beam depths m.
fy(1)=VEC(-250)            ! Vertical loads kN for n=1-►nvl loads.
b(1)=VEC(5)                ! Dist. of vert. loads from joint 2, m.
#cc924.stk !Import set of parameters if available from cc924.stk.
JOINT COORDINATES         ;* All units are kN & m & combinations.
1 0 0 SUPPORT  ;2 0 h   ;3 l h ;4 l 0 SUPPORT
MEMBER INCIDENCES ;1 THRU 3 RANGE 1,2 3,4
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
IF nu=1E-12 THEN ay1=0 ay2=0 ;! If nu=1E-12 shear deform. ignored.
1 AX ax1 AY ay1 IZ iz1 CY dp1/2
2 AX ax2 AY ay2 IZ iz2 CY dp2/2 ;3 AS 1
LOADING CASE 1: VERTICAL & SWAY LOADS ;MEMBER LOADS ;i=0
:50 ;i=i+1 ;IF i>nvl GOTO 55
2 FORCE Y CONCENTRATED P fy(i) L b(i) ;GOTO 50 ;:55
IF fx<>0 ;JOINT LOADS ;2 FORCE X fx ;ENDIF ;SOLVE ;gtot=0 nur=0
! Compute area and second moments of area of analogous column.
status=1 a=h/(e*iz1)+l/(e*iz2)+h/(e*iz1)
ax'=h^2/(e*iz1*2)+l*h/(e*iz2)+h^2/(e*iz1*2)
ay'=h*l/(e*iz1)+l*l/(e*iz2*2) cy=ax'/a cx=ay'/a
ix=h^3/(e*iz1*3)+l*h^2/(e*iz2)+h^3/(e*iz1*3)-a*cy^2
iy=h*l^2/(e*iz1)+l^3/(e*iz2*3)-a*cx^2 m1=0 m2=0 m3=0 m4=0 i=0
iy=h*l^2/(e*iz1)+l^3/(e*iz2*3)-a*cx^2 m1=0 m2=0 m3=0 m4=0 i=0
:60 ;i=i+1 ;IF i>nvl GOTO 65 ;! Compute moments and stresses.
fy=fy(i) b=b(i) pt=(fy*(l-b)^2)/(2*e*iz2) ps=fy*(l-b)*h/(e*iz1)
mx=pt*(h-cy)+ps*(h/2-cy) my=pt*(l-(l-b)/3-cx)+ps*(l-cx)
x=-cx y=-cy pa=(pt+ps)/a+mx*y/ix+my*x/iy x=-cx y=h-cy
pb=(pt+ps)/a+mx*y/ix+my*x/iy x=l-cx y=h-cy
pc=(pt+ps)/a+mx*y/ix+my*x/iy x=l-cx y=-cy
pd=(pt+ps)/a+mx*y/ix+my*x/iy m(1)=m(1)-(0-pa)
m(2)=m(2)-(0-pb) m(3)=m(3)-(fy*(l-b)-pc) m(4)=m(4)+fy*(l-b)-pd
GOTO 60 ;:65 ;IF fx=0 GOTO 80
ps=fx*h^2/(2*e*iz1) mx=ps*(h/3-cy) my=ps*(l-cx) x=-cx y=-cy
pa=ps/a+mx*y/ix+my*x/iy x=-cx y=h-cy pb=ps/a+mx*y/ix+my*x/iy
x=l-cx y=h-cy pc=ps/a+mx*y/ix+my*x/iy x=l-cx y=-cy
pd=ps/a+mx*y/ix+my*x/iy m(1)=m(1)-(0-pa) m(2)=m(2)-(0-pb)
m(3)=m(3)-(0-pc) m(4)=m(4)+fx*h-pd ;:80 ;*/7
IF run>0 GOTO 88 ;*/5 ;i=0 ;:84 ;i=i+1 f(i)=ARR(13,i,9)
* Stress, at start of member +i = +f(i)/1000 N/mm2
IF i<3 GOTO 84 ;f(i)=ARR(13,i,12)
```

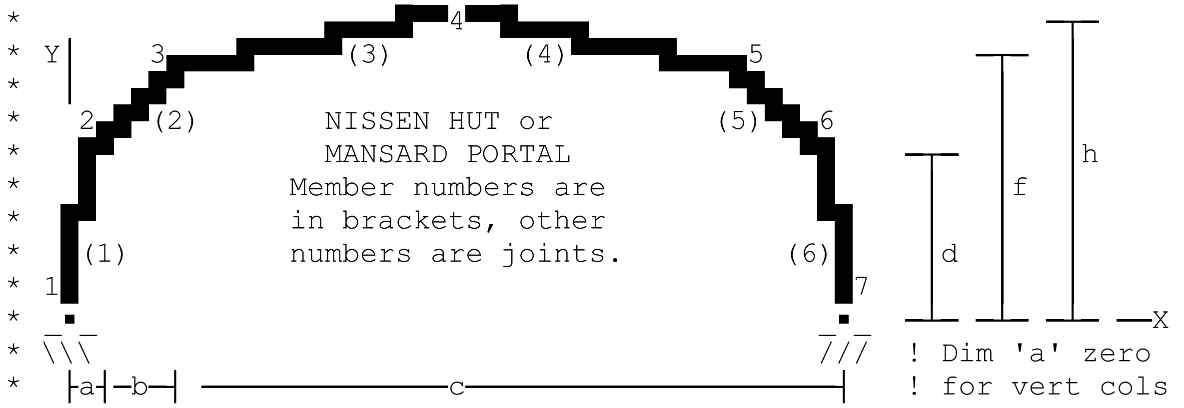
```
* Stress, at end   of member +i = +f(i)/1000 N/mm2 ;:88 ;*/10
* Member  Position    Bending moment  Bending moment         %age
* number  on member   by NL-STRESS    by column analogy      diff.
! Array 13 contains moments at start & end in columns 3 & 6.
i=0 ;:100 ;i=i+1 bm(i)=ARR(13,i,3) d1=bm(i) d2=m(i)
#vmper.ndf !Compute percentage difference & any message.
*   +i    start       +bm(i)          +m(i)                  $ok
IF i<3 GOTO 100 ;i=4 bm(i)=ARR(13,3,6) d1=bm(i) d2=m(i)
#vmper.ndf
*   4     end         +bm(i)          +m(i)                  $ok
IF m<nm GOTO 90 ;fnm=$(vm210.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE LOADS ON VERTICAL PILES IN GROUPS.  PILE CAP
TITLE SUBJECTED TO VERTICAL AND SWAY LOADS, INCLUDING
TITLE COMPARISON OF RESULTS WITH THE CLASSICAL METHOD IN
TITLE REINFORCED CONCRETE DESIGNERS' HANDBOOK, 5TH EDTN.
MADEBY DWB ;DATE 06.04.05 ;METHOD ELASTIC ;REFNO VM211
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE FRAME run=0 ;*/15
*  |-----------e'--------->|w
*  d                        v
*  a     |      Rigid pile cap        |  hl +ve
*  t     |||     ||            ||     ||  --->    ___
*  u      ||     ||            ||     ||            |
*  m      ||     ||            ||     ||            |    h is depth to full
*  |a(1)  ||     ||            ||     ||            |    fixity, thus pile
*  |--->  ||a(2) ||            ||     ||           h|    bends in double
*  |------||---->||    a(n)    ||     ||            |    curvature due to
*  |------||-----||------------||---->||            |    horizontal load hl.
*  ^Y     ||     ||            ||     ||            |
*  L_ // ||\\//\\||//\\//\\//\\//||\\//\\||//    ---   --->X
*         ||     ||            ||     ||             np() is number of piles
*       np(1)  np(2)       np(n-1)  np(n)            in each row into media.
n=4      vss=1E-36  ! No. of rows; vertical spring stiffness/pile.
h=9.8    wp=38      ! Pile length to full fixity; wt. of one pile.
w=-800   hl=-40     ! Vertical & horizontal load on group.
dc=2     bc=2       ! Depth and breadth of cap.
ep=28E6  nup=0.2    ! Young's modulus & Poisson's ratio for piles.
ec=28E6  nuc=0.2    ! Young's modulus & Poisson's ratio for cap.
ax=0.4^2 ay=ax*5/6 iz=0.4^4/12  ! Section properties of each pile.
vss=1E36            ! Vertical spring stiffness beneath each pile.
np(1)=VEC(1,2,2,1)  ! Number of piles in each row: np(1) to np(n).
a(1)=VEC(0,1,3,4)   ! Dist. to each row of piles: a(1) to a(n).
e'=3.0   mcr=1      ! Dist. to line of w; mcr=1 makes cap rigid.
#cc924.stk !Import set of parameters if available from cc924.stk.
w=0 wp=0
NUMBER OF JOINTS nj=2*n nj ;NUMBER OF MEMBERS nm=2*n-1 nm
NUMBER OF SUPPORTS n ;NUMBER OF LOADINGS 1 ;JOINT COORDINATES
i=0 j=-1 ;:10 ;i=i+1 j=j+2 ;j a(i) 0 SUPPORT ;j+1 a(i) h
IF i<n GOTO 10 ;JOINT RELEASES ;i=0 j=-1 ;:15 ;i=i+1 j=j+2
j FORCE Y vss*np(i) ;IF i<n GOTO 15 ;MEMBER INCIDENCES
1 THRU n RANGE 1,2 nj-1,nj ;n+1 THRU nm RANGE 2,4 nj-2,nj
CONSTANTS E ep 1 THRU n G gp=ep/(2*(1+nup)) gp 1 THRU n
ec'=ec ;IF mcr=1 THEN ec'=ec*1E3 ;CONSTANTS E ec' ALL
CONSTANTS G ec'/(2*(1+nuc)) ALL ;MEMBER PROPERTIES
i=0 ;:20 ;i=i+1 i AX np(i)*ax AY np(i)*ay IZ np(i)*iz
IF i<n GOTO 20 ;n+1 THRU nm AX dc*bc AY dc*bc*5/6 IZ bc*dc^3/12
LOADING CASE 1 ;JOINT LOADS ;IF hl<>0 THEN nj FORCE X hl
MEMBER LOADS ;IF w=0 GOTO 30 ;i=0 mld=0 ;:25 ;i=i+1
IF e'<a(i) THEN mld=i-1 ;IF mld<>0 THEN i=n ;IF i<n GOTO 25
mld+n FORCE Y CONCENTRATED P w L e'-a(mld) ;:30 ;IF wp=0 GOTO 40
i=0 ;:35 ;i=i+1 i FORCE Y GLOBAL UNIFORM W -wp*np(i)/h
IF i<n GOTO 35 ;:40 ;SOLVE ;status=1 gtot=0 nur=0 ;i=0 signp=0
sigan=0 ;:45 ;i=i+1 ;signp=signp+np(i) ;sigan=sigan+np(i)*a(i)
IF i<n GOTO 45 ;xbar=sigan/signp e=e'-xbar i=0 signi=0 ;:50 ;i=i+1
signi=signi+np(i)*(a(i)-xbar)^2 ;IF i<n GOTO 50 ;m=w*e-hl*h/2
kw=1/signp i=0 ;:60 ;i=i+1 j=i-INT((i-1)/3)*3 k=1+INT((i-1)/3)
km=(a(k)-xbar)/signi ;IF j=1 ;cla(i)=-kw*w-km*m+wp nl=ARR(13,k,1)
ENDIF ;IF j=2 THEN cla(i)=hl/signp nl=ARR(13,k,2)
IF j=3 THEN cla(i)=h*hl/2/signp nl=ARR(13,k,3)
nl(i)=nl/np(k) ;IF i<3*n GOTO 60 ;*/10
```

```
* Location                      NL-STRESS     Classical       %age
* Forces/pile                   analysis      analysis        diff.
b(1)=$(Axial) b(2)=$(Shear) b(3)=$(Bend mmt) i=0 ;:70
i=i+1 j=i-INT((i-1)/3)*3 k=1+INT((i-1)/3) d1=nl(i) d2=cla(i)
#vmper.ndf !Compute percentage difference & any message.
* Row +k  $(b(j))               +nl(i)        +cla(i)         $ok
IF i<3*n GOTO 70 ;fnm=$(vm211.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

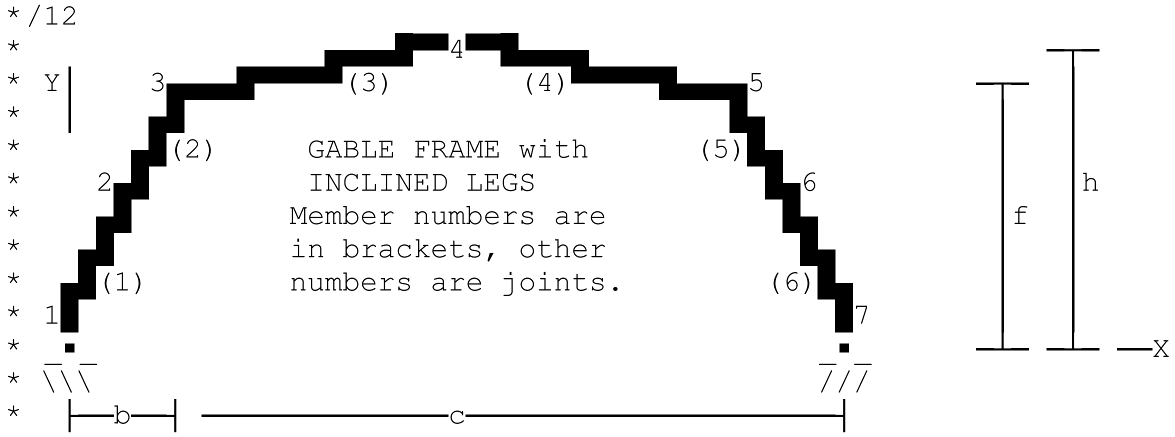
```
TITLE RIDGED PORTAL FRAME WITH FIXED OR PINNED FEET,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM215
TYPE PLANE FRAME run=0 ;DATE 12.05.05 ;NUMBER OF JOINTS 5
NUMBER OF MEMBERS 4 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/9
h=4       ! Ht to eaves intersection.   up─▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
f=2       ! Height eaves to ridge.   f      Y          3
l=10      ! Span of portal.        ─┼─     │2  (2)═════╪═════(3)   4
fix=0     ! Support, 1=fix, 0=pin. ─┼─   ─►╔════              ════╗
e=205E6   ! Young's modulus.         │     ─►║  Members Nos.  ─►   ║
nu=.3     ! Poisson's ratio.        h│     ─►║ ┌in brackets.  ─►(4)║
nsg=16    ! No. of segments/member.  │u1g─►║(1)          u4─►    ║
up=-8.6   ! UDL on plan.           ──┴─   ─►╨1───X         ─►   5╨
u1g=2.8 u4=1   ! UDL sway, +ve X dirn.     ├───────────l───────────┤
u2=-2 u3=4.5   ! Wind normal to left & right rafters, +ve suction.
vjl1=VEC(0,-5,-10,-5,0)  ! Vertical joint loads for joints 1 to 5.
dp1=0.304 dp2=0.304 nli=1   ! Col & rafter depths; No. load incrs.
ax1=47.5E-4 ay1=16E-4 iz1=7162E-8  ! Col. area, shear area, inert.
ax2=47.5E-4 ay2=16E-4 iz2=7162E-8  ! Raf. area, shear area, inert.
#cc924.stk !Import set of parameters if available from cc924.stk.
IF nli>1 ;METHOD SWAY ;NUMBER OF INCREMENTS nli ;ENDIF
NUMBER OF SEGMENTS nsg TRACE ;JOINT COORDINATES ;1 0 0 SUPPORT
2 0 h ;3 l/2 h+f ;4 l h ;5 l 0 SUPPORT
JOINT RELEASES ;IF fix=0 THEN 1 5 INCLUSIVE MOMENT Z
MEMBER INCIDENCES ;1 THRU 4 RANGE 1,2 4,5
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
IF nu=1E-12 THEN ay1=0 ay2=0 ;1 AX ax1 AY ay1 IZ iz1 CY dp1/2
2 THRU 3 AX ax2 AY ay2 IZ iz2 CY dp2/2 ;4 AS 1
LOADING CASE 1 ;JOINT LOADS ;i=0 ;:5 ;i=i+1 ;IF vjl(i)<>0
i FORCE Y vjl(i) ;ENDIF ;IF i<5 GOTO 5
MEMBER LOADS ;u1=-u1g i=0 ;:6 ;i=i+1
IF u(i)<>0 THEN i FORCE Y UNIFORM W u(i) ;IF i<4 GOTO 6
IF up<>0 THEN 2 THRU 3 FORCE Y PROJECTED UNIFORM W up
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=5 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
! Convert to local axes ;lr=SQR((l/2)^2+f^2) cx=(l/2)/lr
cy=f/lr va1=0 vc1=u1 va2=up*cx*cy vc2=u2+up*cx^2 va3=-up*cx*cy
vc3=u3+up*cx^2 va4=0 vc4=u4 hjl1=VEC(0)*5 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm215.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=10 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE NISSEN HUT/MANSARD PORTAL FRAME SUBJECTED TO WIND
TITLE NORMAL TO MEMBERS, UDL ON PLAN AND JOINT LOADS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM216
TYPE PLANE FRAME run=0 ;DATE 04.06.05 ;NUMBER OF JOINTS 7
NUMBER OF MEMBERS 6 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/12
```




```
a=1  b=2  c=20  p1=0  ! Frame X dims from jt 1; pin at 1&7, 1=yes.
d=4  f=5  h=6   p4=0  ! Frame Y dims from jnt 1; pin at 4, 1=yes.
ax=190.1E-4 ay=72.5E-4 iz=124660E-8 dy=.610  ! Memb. props; depth.
e=205E6  g=79E6 nsg=16   ! Young's & shear mod.; No. of segments.
up=-32   ul=8.5 ur=-8.8  ! Plan load; wind l. & r. membrs +ve suct.
vjl1=VEC(0,-5,-5,-5,-5,-5,0)   ! Vert. loads joints 1-7, -ve down.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES SYMMETRY X c/2
1 0 0 S ;2 a d ;3 b f ;4 c/2 h ;5 THRU 6 SYMM 3 THRU 2 ;7 c 0 S
IF p1=1 ;JOINT RELEASES ;1 7 INCLUSIVE MOMENT Z ;ENDIF
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;MEMBER RELEASES
IF p4=1 THEN 3 END MOMENT Z ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU 6 AX ax AY ay IZ iz CY dy/2
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:15 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<7 GOTO 15 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIFORM W ur
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=7 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*7 ;! Conv.to local axes ;lr=SQR(a^2+d^2)
cx=a/lr cy=d/lr va1=up*cx*cy vc1=ul+up*cx^2 va6=-up*cx*cy
vc6=ur+up*cx^2 b'=b-a f'=f-d lr=SQR(b'^2+f'^2) cx=b'/lr cy=f'/lr
va2=up*cx*cy vc2=ul+up*cx^2 va5=-up*cx*cy vc5=ur+up*cx^2 c'=c/2-b
h'=h-f lr=SQR(c'^2+h'^2) cx=c'/lr cy=h'/lr va3=up*cx*cy
vc3=ul+up*cx^2 va4=-up*cx*cy vc4=ur+up*cx^2 ch9=1 nj=7
ch10=0 ;#vmecp.ndf !Equilibrium, compatibility and energy checks.
fnm=$(vm216.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE GABLE FRAME WITH INCLINED LEGS, SUBJECTED TO WIND
TITLE NORMAL TO MEMBERS, UDL ON PLAN AND JOINT LOADS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM217
TYPE PLANE FRAME run=0 ;DATE 04.06.05 ;NUMBER OF JOINTS 7
NUMBER OF MEMBERS 6 ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/12
```




```
b=1   c=20  p1=0   ! Frame X dims from joint 1; pin at 1&7, 1=yes.
f=5   h=6   p4=0   ! Frame Y dims from joint 1; pin at 4, 1=yes.
ax=190.1E-4 ay=72.5E-4 iz=124660E-8 dy=.610  ! Memb. props; depth.
e=205E6  g=79E6 nsg=16   ! Young's & shear mod.; No. of segments.
up=-32   ul=8.5 ur=-8.8  ! Plan load; wind l. & r. membrs +ve suct.
vjl1=VEC(0,-5,-5,-5,-5,-5,0)   ! Vert. loads joints 1-7, -ve down.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES SYMMETRY X c/2 ;a=b/2
1 0 0 S ;2 a d=f/2 d ;3 b f ;4 c/2 h ;5 THRU 6 SYMM 3 THRU 2
7 c 0 S ;IF p1=1 ;JOINT RELEASES ;1 7 INCLUSIVE MOMENT Z ;ENDIF
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;MEMBER RELEASES
IF p4=1 THEN 3 END MOMENT Z ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU 6 AX ax AY ay IZ iz CY dy/2
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<7 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIFORM W ur
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=7 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*7 ;! Conv.to local axes ;lr=SQR(a^2+d^2)
cx=a/lr cy=d/lr va1=up*cx*cy vc1=ul+up*cx^2 va6=-up*cx*cy
vc6=ur+up*cx^2 b'=b-a f'=f-d lr=SQR(b'^2+f'^2) cx=b'/lr cy=f'/lr
va2=up*cx*cy vc2=ul+up*cx^2 va5=-up*cx*cy vc5=ur+up*cx^2 c'=c/2-b
h'=h-f lr=SQR(c'^2+h'^2) cx=c'/lr cy=h'/lr va3=up*cx*cy
vc3=ul+up*cx^2 va4=-up*cx*cy vc4=ur+up*cx^2 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility and energy checks.
fnm=$(vm217.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PORTAL FRAME WITH SKEWED CORNERS, SUBJECTED TO WIND
TITLE NORMAL TO MEMBERS, UDL ON PLAN AND JOINT LOADS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM218
TYPE PLANE FRAME run=0 ;DATE 04.06.05 ;NUMBER OF JOINTS 7
NUMBER OF MEMBERS 6  ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/12
*      3                                  5
*  Y          (3)        4        (4)
*       (2)
*  2              PORTAL FRAME with                   6
*                 SKEWED CORNERS
*                 Member numbers are                         h
*      (1)        in brackets, other            (6)       d
*                 numbers are joints.
*  1                                               7
*                                                                      X
* \\\                                             ///
*   |-b-|  -----------------c------------------|
b=1.5  c=20  p1=0  ! Frame X dims from joint 1; pin at 1&7, 1=yes.
d=4.5  h=6   p4=0  ! Frame Y dims from joint 1; pin at 4, 1=yes.
ax=190.1E-4 ay=72.5E-4 iz=124660E-8 dy=.610  ! Memb. props; depth.
e=205E6  g=79E6 nsg=16   ! Young's & shear mod.; No. of segments.
up=-32   ul=8.5 ur=-8.8  ! Plan load; wind l. & r. membrs +ve suct.
vjl1=VEC(0,-5,-5,-5,-5,-5,0)   ! Vert. loads joints 1-7, -ve down.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES SYMMETRY X c/2 ;a=0 f=h
1 0 0 S ;2 a d ;3 b f ;4 c/2 h ;5 THRU 6 SYMM 3 THRU 2 ;7 c 0 S
IF p1=1 ;JOINT RELEASES ;1 7 INCLUSIVE MOMENT Z ;ENDIF
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;MEMBER RELEASES
IF p4=1 THEN 3 END MOMENT Z ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU 6 AX ax AY ay IZ iz CY dy/2
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<7 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIFORM W ur
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=7 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*7 ;! Conv.to local axes ;lr=SQR(a^2+d^2)
cx=a/lr cy=d/lr va1=up*cx*cy vc1=ul+up*cx^2 va6=-up*cx*cy
vc6=ur+up*cx^2 b'=b-a f'=f-d lr=SQR(b'^2+f'^2) cx=b'/lr cy=f'/lr
va2=up*cx*cy vc2=ul+up*cx^2 va5=-up*cx*cy vc5=ur+up*cx^2 c'=c/2-b
h'=h-f lr=SQR(c'^2+h'^2) cx=c'/lr cy=h'/lr va3=up*cx*cy
vc3=ul+up*cx^2 va4=-up*cx*cy vc4=ur+up*cx^2 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility and energy checks.
fnm=$(vm218.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE TRAPEZOIDAL FRAME, SUBJECTED TO WIND NORMAL TO
TITLE MEMBERS, UDL ON PLAN AND JOINT LOADS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM219
TYPE PLANE FRAME run=0 ;DATE 04.06.05 ;NUMBER OF JOINTS 7
NUMBER OF MEMBERS 6  ;NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 3
*/12
*         3                             5
* Y          (3)       4       (4)
*     (2)                                  (5)
*                TRAPEZIODAL FRAME                        h
*      2                                    6
*                Member numbers are
*  (1)           in brackets, other           (6)
*                numbers are joints.
* 1                                             7
*  _._                                         _._          ____X
* \\\                                          ///
*  |---b---|  ----------------c-----------------|
b=1.5  c=20  h=6  p1=0   ! Frame dims; pin at 1&7, 1=yes, 0=No.
ax=190.1E-4 ay=72.5E-4 iz=124660E-8 dy=.610  ! Memb. props; depth.
e=205E6  g=79E6 nsg=16   ! Young's & shear mod.; No. of segments.
up=-32   ul=8.5 ur=-8.8  ! Plan load; wind l. & r. membrs +ve suct.
vjl1=VEC(0,-5,-5,-5,-5,-5,0)   ! Vert. loads joints 1-7, -ve down.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES SYMM X c/2 ;a=b/2 d=h/2
f=h ;1 0 0 S ;2 a d ;3 b f ;4 c/2 h ;5 THRU 6 SYMM 3 THRU 2
7 c 0 S ;IF p1=1 ;JOINT RELEASES ;1 7 INCLUSIVE MOMENT Z ;ENDIF
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;MEMBER RELEASES
CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU 6 AX ax AY ay IZ iz CY dy/2
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<7 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIFORM W ur
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=7 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*7 ;! Conv.to local axes ;lr=SQR(a^2+d^2)
cx=a/lr cy=d/lr va1=up*cx*cy vc1=ul+up*cx^2 va6=-up*cx*cy
vc6=ur+up*cx^2 b'=b-a f'=f-d lr=SQR(b'^2+f'^2) cx=b'/lr cy=f'/lr
va2=up*cx*cy vc2=ul+up*cx^2 va5=-up*cx*cy vc5=ur+up*cx^2 c'=c/2-b
h'=h-f lr=SQR(c'^2+h'^2) cx=c'/lr cy=h'/lr va3=up*cx*cy
vc3=ul+up*cx^2 va4=-up*cx*cy vc4=ur+up*cx^2 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility and energy checks.
fnm=$(vm219.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

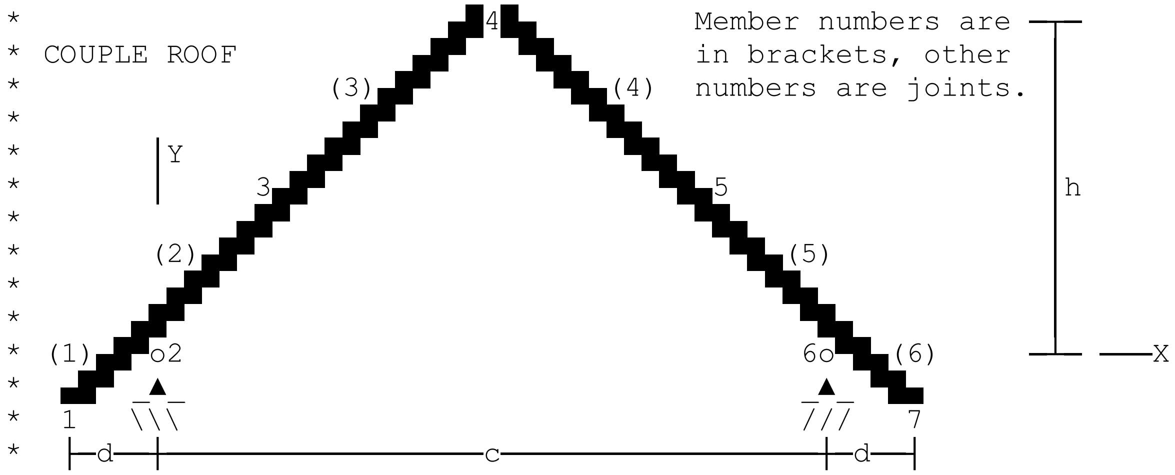
```
TITLE TWO BAY RIDGED PORTAL FRAME (TWIN GABLE FRAME)
TITLE WITH FIXED FEET, INCLUDING CHECK OF RESULTS
TITLE AGAINST KLEINLOGEL'S MEHRSTIELIGE RAHMEN,
TITLE BAND I+II, PUB. WILHELM ERNST & SOHN. ;MADEBY DWB
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;REFNO VM220
TYPE PLANE FRAME run=0 ;DATE 30.10.04 ;NUMBER OF JOINTS 8
NUMBER OF MEMBERS 7 ;NUMBER OF SUPPORTS 3 ;NUMBER OF LOADINGS 1
*/9
*          ▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼  uy kN/m,
*  f        Y          3                     6           -ve down.
* ─┼─       |2  (2)════┼════(3)   4   (5)════┼════(6)
* ─┼─ fx──►╔════          ════╦════          ════╗7
*  │   kN  ║    Member Nos. in    ║                ║
* h│       ║    brackets, other   ║(4)          (7)║
*  │   (1) ║    Nos. are joints.  ║                ║
* ─┴─     1╨ ──X                 ╨5              8╨
*          ├──────────l──────────┼──────────l──────────┤
h=4    f=1       l=10    ! Frame dimensions m.
e=205E6  nu=.3           ! Young's modulus kN/m2, Poisson's ratio.
ax1=47.5E-4 ay1=16E-4 iz1=7162E-8  ! Area, shear area m2 inert m4.
ax2=47.5E-4 ay2=16E-4 iz2=7162E-8  ! Area, shear area m2 inert m4.
ax4=40.0E-4 ay4=15E-4 iz4=4439E-8  ! Area, shear area m2 inert m4.
dp1=.304 dp2=.304 dp4=.252         ! Beam depths m.
uy=-8.6     fx=12.8   ! UDL kN/m on plan, sway load kN at joint 2.
#cc924.stk !Import set of parameters if available from cc924.stk.
JOINT COORDINATES    ;* All units are kN & m & combinations.
1 0 0 SUPPORT  ;2 0 h   ;3 l/2 h+f ;4 l h ;5 l 0 SUPPORT
6 1.5*l h+f    ;7 2*l h ;8 2*l 0 SUPPORT
JOINT RELEASES ;1 THRU 5 STEP 4 MOMENT Z ;8 MOMENT Z
MEMBER INCIDENCES ;1 THRU 4 RANGE 1,2 4,5 ;5 THRU 7 CHAIN 4,6,7,8
g=e/(2*(1+nu)) ;CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
IF nu=1E-12 THEN ay1=0 ay2=0 ay4=0
1 AX ax1 AY ay1 IZ iz1 CY dp1/2
2 THRU 3 AX ax2 AY ay2 IZ iz2 CY dp2/2
4 AX ax4 AY ay4 IZ iz4 CY dp4/2 ;5 THRU 7 AS 3 THRU 1
LOADING CASE 1: PLAN LOAD WITH OPTIONAL SWAY LOAD
IF uy<>0 ;MEMBER LOADS ;2 THRU 3 FORCE Y PROJECTED UNIFORM W uy
5 THRU 6 FORCE Y PROJECTED UNIFORM W uy ;ENDIF
JOINT LOADS ;IF fx<>0 THEN 2 FORCE X fx ;SOLVE
! Array 13 contains: for each loading & member, 7 rows, with each
! row containing 6 member end forces & 6 member end stresses.
i=0 ;:20 ;i=i+1 ;bms(i)=ARR(13,i,3) bme(i)=ARR(13,i,6)
sts(i)=ARR(13,i,9) sts(i)=sts(i)/1000 ste(i)=ARR(13,i,12)
ste(i)=ste(i)/1000 ;IF i<7 GOTO 20 ;status=1 gtot=0 nur=0
s=SQR(f^2+(l/2)^2) ii=iz1 i1=iz2 iii=iz4 !Kleinlogel's constants:
phi=f/h xi=i1*h/(ii*s) xii=i1*h/(iii*s) costhe=l/(2*s) sinthe=f/s
n1=8*xi+12*(1+phi)+7*phi^2 n2=2*xi+12*(1+phi)+4*phi^2+4*xii
! Kleinlogel's influence & composite influence coefficients:
n11=2/n1 n12=(2+3*phi)/(2*n1) n21=n12 n22=(xi+2+3*phi+2*phi^2)/n1
l1=(2+5*phi)/(4*n1) c1=(6+7*phi)/(4*n1) y11=2*phi*c1
l2=(2*xi+2-phi-2*phi^2)/(4*n1) c2=(6*xi+6+phi)/(4*n1)
y12=2*phi*c2 n=1/(2*n2) l'=(1+phi)/(2*n2) c=(3+2*phi)/(2*n2)
r=1/n2 y13=(6+9*phi+4*phi^2+4*xii)/(2*n2)
y23=(6+3*phi+4*xii)/(2*n2) y14=(2*xi+6+3*phi)/(2*n2)
y24=(2*xi+6+9*phi+4*phi^2)/(2*n2) s1=n11+l1 s'1=-l1+n21
r1=-n11+n21 s2=n12-l2 s'2=l2+n22 r2=-n12+n22 s'=l'+n r=2*n
```

```
m1=(s1-s'1)/2 m2=(-s2+s'2)/2 y3=y23+c*phi y4=y24-c*phi
m1a=uy*l^2*(m1+2*c1)/4 m1c=uy*l^2*(m2+2*c2)/4
m1b=-uy*l^2/8+(m1a*(1+2*phi)+m1c)/2 m2a=fx*h*(y11+y13)
m2a'=fx*h*(y11-y13) m2c1=fx*h*(y12-y14) m2c2=fx*h*(y12+y14)
m2c'=-2*fx*h*y14 m2b=(m2a*(1+2*phi)+m2c1)/2-fx*f
m2b'=(m2c2+m2a'*(1+2*phi))/2 mks(1)=0 mke(1)=m1a+m2a
mks(2)=-mke(1) mke(2)=m1b+m2b mks(3)=-mke(2) mke(3)=m1c+m2c1
mks(4)=-m2c' mke(4)=0 mks(5)=-m1c-m2c2 mke(5)=m1b+m2b'
mks(6)=-mke(5) mke(6)=m1a+m2a' mks(7)=-mke(6) mke(7)=0 ;*/5
* Member &   Stress N/mm2    Moment kNm      Moment kNm      %age
* location   bending NL-ST   NL-STRESS       Kleinlogel      diff.
i=0 ;:100 ;i=i+1 d1=bms(i) d2=mks(i)
#vmper.ndf !Compute percentage difference & any message.
* +i  start  +sts(i)         +bms(i)         +mks(i)         $ok
d1=bme(i) d2=mke(i)
#vmper.ndf
*     end    +ste(i)         +bme(i)         +mke(i)         $ok
IF i<7 GOTO 100 ;fnm=$(vm220.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE SINGLE/MULTI-BAY PORTAL FRAME/S WITH HAUNCHES
TITLE HAVING 'nb' BAYS SUBJECTED TO VERTICAL AND SWAY
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM223
TYPE PLANE FRAME ;DATE 29.10.05 ;*/9
*        ▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼ w=udl, plan load
*   f    Y         5                  12          per unit length,
*  ─┼─   │     ════╪════        9     ════╪════      16       -ve down.
*  ─┼─ 2╔3=4══         ══6=7╤10=11         13=14╤17=18
*   │    ║                    ║                  ║    Joint Nos.
*   b    ║├──┼ha          ha┼─┤                  ║    shown, member
*   │    ║                    ║                  ║    Nos. omitted
*  ─┴─ 1╨  ───X               ╨8              15╨   for clarity.
*        ├──────────a─────────┼─────────a───────┼────►
a=28  b=4.8    f=1.2   ! Frame sizes.
nb=3  e=205E6  nu=0.3  ! No. of bays, Young's mod., Poisson's rat.
de1=VEC(0.8407,0.2924,0.0147,0.0217)  ! External columns, D,B,t,T.
di1=VEC(0.8407,0.2924,0.0147,0.0217)  ! Internal columns, D,B,t,T.
dr1=VEC(0.7698,0.2680,0.0156,0.0254)  ! Rafters, D,B,t,T.
nsg=16  ha=2.8 dmax=1.540 ! No of segs; haunch length & max depth.
bfix=0  w=-16  sl=20 ! Base fixity, plan udl, sway load at jnt 2.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj=7*nb+2 nj ;NUMBER OF MEMBERS nm=7*nb+1 nm
NUMBER OF SUPPORTS ns=nb+1 ns ;NUMBER OF LOADINGS 3
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;dy=f*(ha/2)/(a/2)
1 THRU nj-1 STEP 7 X 0 Y 0 XL nb*a SUPPORT
2 THRU nj STEP 7 X 0 Y b XL nb*a ;IF nb=1
3 X ha/2 Y b+dy ;4 X ha Y b+2*dy ;5 X a/2 Y b+f ;6 X a-ha Y b+2*dy
7 X a-ha/2 Y b+dy ;ENDIF ;IF nb>1
3 THRU nj-6 STEP 7 X ha/2 Y b+dy XL (nb-1)*a+ha/2
4 THRU nj-5 STEP 7 X ha Y b+2*dy XL (nb-1)*a+ha
5 THRU nj-4 STEP 7 X a/2 Y b+f XL (nb-1)*a+a/2
6 THRU nj-3 STEP 7 X a-ha Y b+2*dy XL nb*a-ha
7 THRU nj-2 STEP 7 X a-ha/2 Y b+dy XL nb*a-ha/2 ;ENDIF
JOINT RELEASES ;IF bfix=0 THEN 1 THRU nj-1 STEP 7 MOMENT Z
MEMBER INCIDENCES ;1 THRU nm STEP 7 RANGE 1,2 nj-1,nj
IF nb=1 THEN 2 THRU nm-1 CHAIN 2,3,4,5,6,7,9
IF nb>1 ;2 THRU nm-6 STEP 7 RANGE 2,3 nj-7,nj-6
3 THRU nm-5 STEP 7 RANGE 3,4 nj-6,nj-5
4 THRU nm-4 STEP 7 RANGE 4,5 nj-5,nj-4
5 THRU nm-3 STEP 7 RANGE 5,6 nj-4,nj-3
6 THRU nm-2 STEP 7 RANGE 6,7 nj-3,nj-2
7 THRU nm-1 STEP 7 RANGE 7,9 nj-2,nj ;ENDIF
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nm STEP nm-1 ISECTION DY de1 DZ de2 TZ de3 TY de4
IF nb=2 THEN 8 ISECTION DY di1 DZ di2 TZ di3 TY di4
IF nb>2 ;8 THRU nm-7 STEP 7 ISECTION DY di1 DZ di2 TZ di3 TY di4
ENDIF ;4 THRU 5 ISECTION DY dr1 DZ dr2 TZ dr3 TY dr4
IF nb=2 THEN 11 THRU 12 AS 4
IF nb>2 THEN 11 THRU nm-4 STEP 7 AS 4
IF nb>2 THEN 12 THRU nm-3 STEP 7 AS 4
2 THRU 3 ISECTION DY dmax DYL dr1 DZ dr2 DZL dr2 TZ dr3 TY dr4
6 THRU 7 AS 3 THRU 2 ;IF nb>1 ;i=1 m=2 ;REPEAT ;i=i+1 m=m+7
m THRU m+1 AS 2 THRU 3 ;m+4 THRU m+5 AS 3 THRU 2 ;UNTIL i=nb
ENDREPEAT ;ENDIF
```

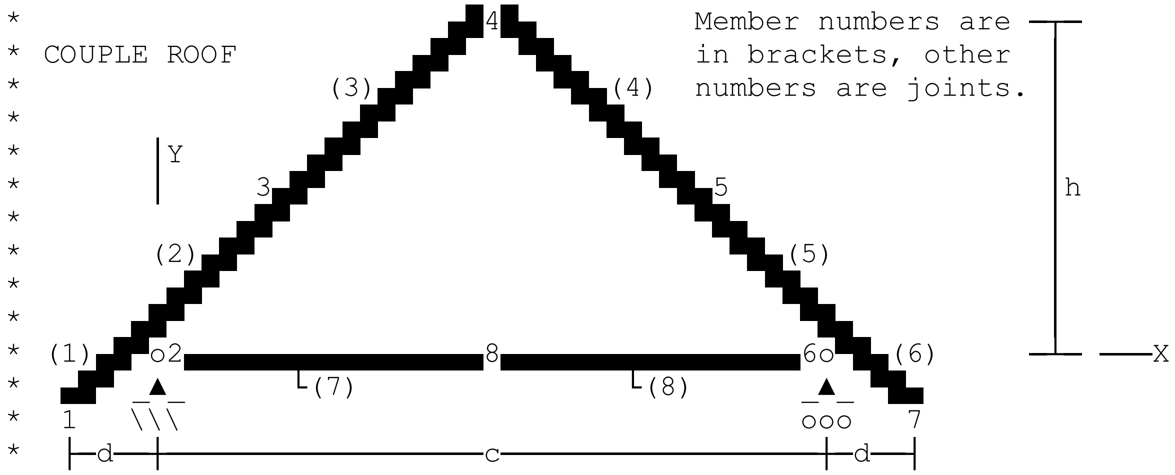
```
LOADING CASE 1 ;MEMBER LOADS ;i=0 m=-5 ;REPEAT ;i=i+1 m=m+7
m THRU m+5 FORCE Y PROJECTED UNIFORM W w ;UNTIL i=nb ;ENDREPEAT
JOINT LOADS ;2 FORCE X sl
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*nj hjl2=sl vjl1=VEC(0)*nj lr=SQR((a/2)^2+f^2) i=0
cx=(a/2)/lr cy=f/lr m=-5 ;:20 ;i=i+1 m=m+7 va=w*cy*cx vc=w*cx^2
va(m)=VEC(va)*3 vc(m)=VEC(vc)*3 va=-va m'=m+3 va(m')=VEC(va)*3
vc(m')=VEC(vc)*3 ;IF i<nb GOTO 20 ;ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm223.stk) ;#vmres.ndf !Conclude results.
mjn=5 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE COUPLE ROOF FRAME SUBJECTED TO JOINT LOADS, PLAN UDL
TITLE ON ROOF & WIND LOADING NORMAL TO RAFTERS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM225
TYPE PLANE FRAME run=0 ;DATE 20.05.05 ;NUMBER OF JOINTS 7
NUMBER OF MEMBERS 6 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
```



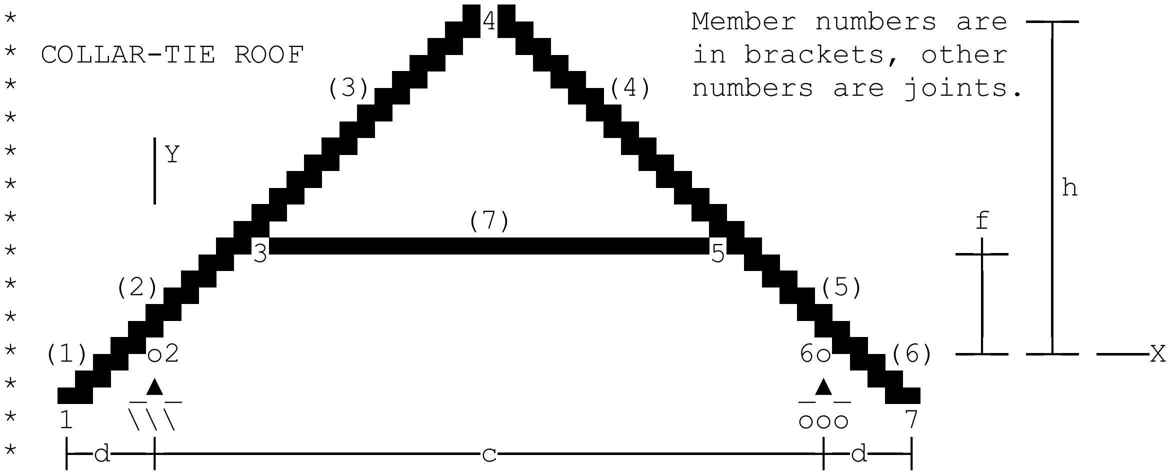

```
c=4  d=0.4  h=3  p4=0      ! Frame dimensions; pin at 4 (1=yes)
e=9E6  g=e/16     nsg=4    ! Young's & shear mod; No. of segments.
dy1=.200 dz1=.050 up=-2.3  ! Rafter sizes; udl on plan (-ve down)
ul=-0.5  ur=1       ! Wind load on l. & r. rafters. (+ve suction).
vjl1=VEC(-1.5,-3,-3,-3,-3,-3,-1.5)  ! Vertical loads joints 1-7.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDS SYMM X c/2 ;1 -d -d*h/(c/2)
2 THRU 4 X 0 Y 0 XL c/2 YL h ;5 THRU 7 SYMMETRY 3 THRU 1
JOINT RELEASES ;2 FORCE X -1 Y -1 ;6 FORCE X -1 Y -1
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;IF p4=1
MEMBER RELEASES ;3 END MOMENT Z ;ENDIF ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU 6 RECTANGLE DY dy1 DZ dz1
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<7 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIF W ur
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=7 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*7 ;! Convert to local axes ;lr=SQR((c/2)^2+h^2)
cx=(c/2)/lr cy=h/lr va=up*cy*cx vc=ul+up*cx^2 va1=VEC(va)*3
vc1=VEC(vc)*3 va=-up*cy*cx vc=ur+up*cx^2 va4=VEC(va)*3
vc4=VEC(vc)*3 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm225.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE COUPLE CLOSE ROOF FRAME SUBJECTED TO JOINT LOADS,
TITLE PLAN UDL ON ROOF & WIND LOADING NORMAL TO RAFTERS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM226
TYPE PLANE FRAME run=0 ;DATE 27.05.05 ;NUMBER OF JOINTS 8
NUMBER OF MEMBERS 8 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
```




```
c=4  d=0.4  h=3  p4=1       ! Frame dims; pin jnt. 4 (1=yes, 0=No).
e=9E6  g=e/16    nsg=16     ! Young's & shear mod; No. of segments.
dy1=.200 dz1=.050 up=-2.3   ! Rafter sizes; udl on plan (-ve down)
dy7=.200 dz7=.050 p8=1      ! Floor joist sizes, pin ends memb. 8.
uf=-2 ul=-0.5  ur=1       ! Floor udl; wind l. & r. raft. +ve suct.
vjl1=VEC(-1.5,-3,-3,-3,-3,-3,-1.5,0)  ! Vertical loads joints 1-8.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDS SYMM X c/2 ;1 -d -d*h/(c/2)
2 THRU 4 X 0 Y 0 XL c/2 YL h ;5 THRU 7 SYMMETRY 3 THRU 1 ;8 c/2 0
JOINT RELEASES ;2 FORCE X -1 Y -1 ;6 FORCE Y -1
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;7 THRU 8 CHAIN 2,8,6
MEMBER RELEASES ;IF p4=1 THEN 3 END MOMENT Z ;IF p8=1
7 START MOMENT Z ;8 END MOMENT Z ;ENDIF ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU 6 RECTANGLE DY dy1 DZ dz1
7 THRU 8 RECTANGLE DY dy7 DZ dz7
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<8 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIF W ur
7 THRU 8 FORCE Y UNIFORM W uf
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=8 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*8 ;! Convert to local axes ;lr=SQR((c/2)^2+h^2)
cx=(c/2)/lr cy=h/lr va=up*cy*cx vc=ul+up*cx^2 va1=VEC(va)*3
vc1=VEC(vc)*3 va=-up*cy*cx vc=ur+up*cx^2 va4=VEC(va)*3
vc4=VEC(vc)*3 va(7)=VEC(0)*2 vc(7)=VEC(uf)*2 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm226.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE COLLAR-TIE ROOF FRAME SUBJECTED TO JOINT LOADS, PLAN
TITLE UDL ON ROOF & WIND LOADING NORMAL TO RAFTERS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM227
TYPE PLANE FRAME run=0 ;DATE 25.05.05 ;NUMBER OF JOINTS 7
NUMBER OF MEMBERS 7 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
```

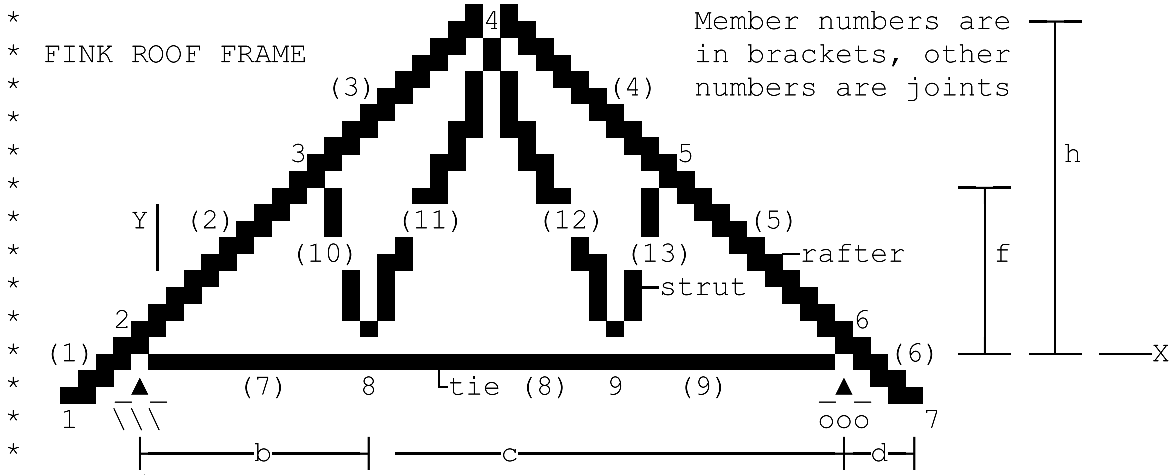



```
c=4   d=0.4   f=1   h=3    ! Frame dimensions.
e=9E6  g=e/16     nsg=16   ! Young's & shear mod; No. of segments.
dy1=.200 dz1=.050 up=-2.3  ! Rafter sizes; udl on plan, -ve down.
dy7=.150 dz7=.050 ut=-3.0  ! Tie sizes; udl on tie, -ve down.
p4=0     p35=0    ! Pins at joint: 4 and/or both 3&5; 1=Yes, 0=No.
ul=-0.5  ur=1     ! Wind normal to l. & r. rafters, +ve suction.
vjl1=VEC(-1.5,-3,-3,-3,-3,-3,-1.5)  ! Vertical loads joints 1-7.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDS SYMM X c/2 ;1 -d -d*h/(c/2)
2 THRU 3 X 0 Y 0 XL (c/2)*f/h YL f ;4 c/2 h
5 THRU 7 SYMMETRY 3 THRU 1 ;JOINT RELEASES ;2 FORCE X -1 Y -1
6 FORCE Y -1 ;MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;7 3 5
MEMBER RELEASES ;IF p4=1 THEN 3 END MOMENT Z
IF p35=1 THEN 7 START MOMENT Z END MOMENT Z
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 THRU 6 RECTANGLE DY dy1 DZ dz1 ;7 RECTANGLE DY dy7 DZ dz7
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<7 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
7 FORCE Y PROJECTED UNIFORM W ut
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIF W ur
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=7 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*7 ;! Convert to local axes ;lr=SQR((c/2)^2+h^2)
cx=(c/2)/lr cy=h/lr va=up*cx*cy vc=ul+up*cx^2 va1=VEC(va)*3
vc1=VEC(vc)*3 va=-up*cx*cy vc=ur+up*cx^2 va4=VEC(va)*3
vc4=VEC(vc)*3 va7=0 vc7=ut ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm227.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE COLLAR & TIE ROOF WITH UDL ON THEM; JOINT LOADS,
TITLE PLAN UDL ON ROOF & WIND LOADING NORMAL TO RAFTERS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM228
TYPE PLANE FRAME run=0 ;DATE 27.05.05 ;NUMBER OF JOINTS 9
NUMBER OF MEMBERS 10 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
*                          4                Member numbers are
*  COLLAR & TIE ROOF                        in brackets, other
*                (3)              (4)       numbers are joints.
*
*        |Y
*        |        3 o          9          o 5                        h
*                       (9)collar(10)
*       (2)                                   (5)rafter        f
*
*
*  (1)     o2              8              6o    (6)                        X
*         _▲_       (7)tie         (8)tie  _▲_
*   1     \\\                              ooo    7
*   |-d---|---------------c----------------|---d--|
c=4  d=0.4  h=3 f=2  p4=1  ! Frame dims; pin jnt. 4 (1=yes, 0=No).
e=9E6  g=e/16    nsg=16    ! Young's & shear mod; No. of segments.
dy1=.200 dz1=.050 up=-2.3  ! Rafter sizes; udl on plan (-ve down)
dy7=.200 dz7=.050 p8=1     ! Tie joist sizes, pinned ends memb. 8.
dy9=.200 dz9=.050 p9=1     ! Collar sizes, pinned ends memb. 9.
ut=-2    uc=-0.5           ! Tie & collar udls.
ul=-0.5  ur=1              ! Wind on l. & r. rafters, +ve suction.
vjl1=VEC(-1.5,-3,-3,-3,-3,-3,-1.5,0,0)  ! Vert. loads joints 1-9.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDS SYMM X c/2 ;1 -d -d*h/(c/2)
2 THRU 3 X 0 Y 0 XL (c/2)*f/h YL f ;4 c/2 h
5 THRU 7 SYMMETRY 3 THRU 1 ;8 c/2 0
9 c/2 f ;JOINT RELEASES ;2 FORCE X -1 Y -1 ;6 FORCE Y -1
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;7 THRU 8 CHAIN 2,8,6
9 THRU 10 CHAIN 3,9,5 ;MEMBER RELEASES ;IF p4=1 ;3 END MOMENT Z
ENDIF ;IF p8=1 ;7 START MOMENT Z ;8 END MOMENT Z ;ENDIF
IF p9=1 ;9 START MOMENT Z ;10 END MOMENT Z ;ENDIF
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 THRU 6 RECTANGLE DY dy1 DZ dz1 ;7 THRU 8 RECTANGLE DY dy7 DZ dz7
9 THRU 10 RECTANGLE DY dy9 DZ dz9
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<9 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIF W ur
7 THRU 8 FORCE Y UNIFORM W ut ;9 THRU 10 FORCE Y UNIFORM W uc
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=9 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*9 ;! Convert to local axes ;lr=SQR((c/2)^2+h^2)
cx=(c/2)/lr cy=h/lr va=up*cy*cx vc=ul+up*cx^2 va1=VEC(va)*3
vc1=VEC(vc)*3 va=-up*cy*cx vc=ur+up*cx^2 va4=VEC(va)*3
vc4=VEC(vc)*3 va(7)=VEC(0)*2 vc(7)=VEC(ut)*2 va(9)=VEC(0)*2
vc(9)=VEC(uc)*2 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compat & energy checks.
fnm=$(vm228.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

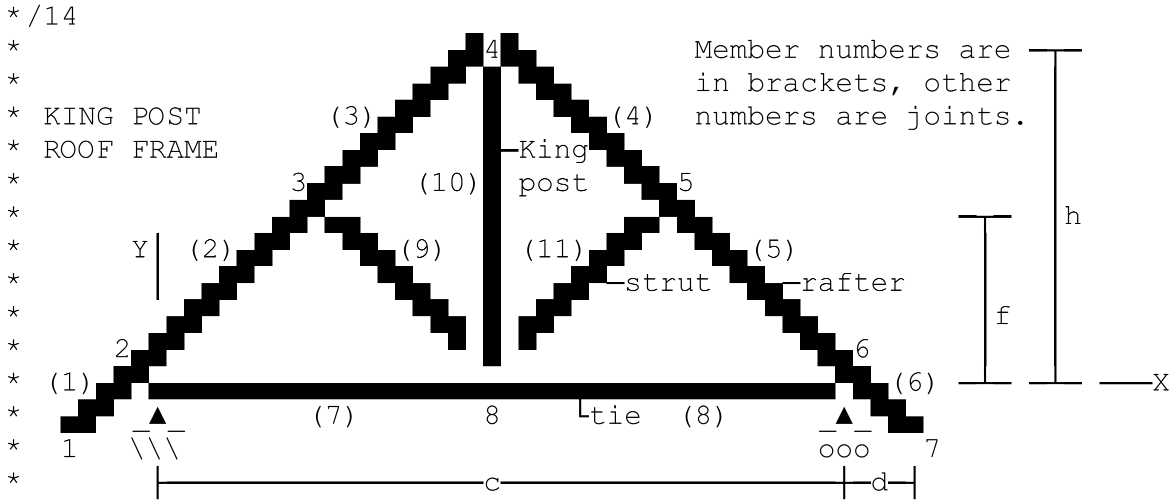
```
TITLE ATTIC ROOM ROOF FRAME SUBJECTED TO UDL ON FLOOR &
TITLE CEILING, WIND NORMAL TO RAFTERS & JOINT LOADS,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM230
TYPE PLANE FRAME run=0 ;DATE 20.05.05 ;NUMBER OF JOINTS 14
NUMBER OF MEMBERS 17 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
*                                  o         Member numbers in
*  ATTIC ROOM ROOF  (10)          11  (11)   brackets, joints
*                                            not. o denotes pin.
*                                ┌(17)
*        |Y     (9)    o                    o    (12)
*        |              10  └ceiling   12        ┌rafter          h
*                    o9       joist            13o
*       (8)          (15)                  (16)     (13)          f
*                                      stud┤
*     1   ┌(1) o      ┌floor joist        o  (6)┐   7
* (7)   o    2   3          4              5    6  o   (14)        X
* 8     _▲_   (2)    (3)           (4)     (5)   _▲_  14    Support
*       \\\                                      ooo ←──────positions
*   |   ├a┤ ├b┤   ───c───┤                                    can vary.
*   ├k──┼───────────────────d────────────────────┼─k─┤
a=0.4  b=1.2  c=3  d=6   k=0.6   ! Distances 1-2 1-3 1-4 1-7 7-14.
sl=2  sr=6   ! Supported joints left & right from: 1 2 3 or 5 6 7.
f=2.3 h=3  p4=0   ! Hts floor to ceiling & ridge; pin at 4, 1=yes.
dy1=.200  dz1=.050    dy7=.150  dz7=.050   ! Floor & rafter sizes.
dy15=.100 dz15=.050   dy17=.100 dz17=.050  ! Stud & ceiling sizes.
e=9E6    g=e/16  nsg=16   ! Young's & shear mod.; No. of segments.
up=-2.3  uc=-1  uf=-3.5   ! Plan load on rafters, ceiling & floor.
ul=0.5   ur=-1            ! Wind on l. & r. rafters (+ve suction).
vjl1=VEC(0,0,0,0,0,0,0,0,-3,0,0,0,-3,0)  ! Vert loads joints 1-14.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES SYMMETRY X d/2 ;1 0 0
2 a 0 ;3 b 0 ;4 c 0 ;5 THRU 7 SYMMETRY 3 THRU 1 ;8 -k -k*h/(d/2)
9 b b*h/(d/2) ;10 f*(d/2)/h f ;11 d/2 h
12 THRU 14 SYMMETRY 10 THRU 8 ;JOINT RELEASES ;sl FORCE X -1 Y -1
sr FORCE Y -1 ;MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7
7 THRU 14 CHAIN 8,1,9,10,11,12,13,7,14 ;15 3 9 ;16 5 13 ;17 10 12
MEMBER RELEASES ;1 START MOMENT Z ;IF p4=1 THEN 3 END MOMENT Z
6 END MOMENT Z ;10 END MOMENT Z
15 THRU 17 START MOMENT Z END MOMENT Z
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 THRU 6 RECTANGLE DY dy1 DZ dz1 ;7 THRU 14 RECT DY dy7 DZ dz7
15 THRU 16 RECTANGLE DY dy15 DZ dz15 ;17 RECTANGLE DY dy17 DZ dz17
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<14 GOTO 16 ;MEMBER LOADS
1 THRU 6 FORCE Y GLOBAL UNIFORM W uf
7 THRU 14 FORCE Y PROJECTED UNIFORM W up
7 THRU 10 FORCE Y UNIFORM W ul
11 THRU 14 FORCE Y UNIFORM W ur ;17 FORCE Y UNIFORM W uc
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=14 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*14 ;! Conv.to local axes ;lr=SQR((d/2)^2+h^2)
cx=(d/2)/lr cy=h/lr va1=VEC(0)*6 vc1=VEC(uf)*6 va=up*cx*cy
vc=ul+up*cx^2 va7=VEC(va)*4 vc7=VEC(vc)*4 va=-up*cx*cy
vc=ur+up*cx^2 va11=VEC(va)*4 vc11=VEC(vc)*4 va15=VEC(0)*3
vc15=VEC(0)*2 vc(17)=uc ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatability and energy checks.
fnm=$(vm230.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE FINK ROOF FRAME (WARREN INTERNALS) WITH JOINT LOADS,
TITLE PLAN UDL ON ROOF & WIND LOADING NORMAL TO RAFTERS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM232
TYPE PLANE FRAME run=0 ;DATE 01.06.05 ;NUMBER OF JOINTS 9
NUMBER OF MEMBERS 13 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
```



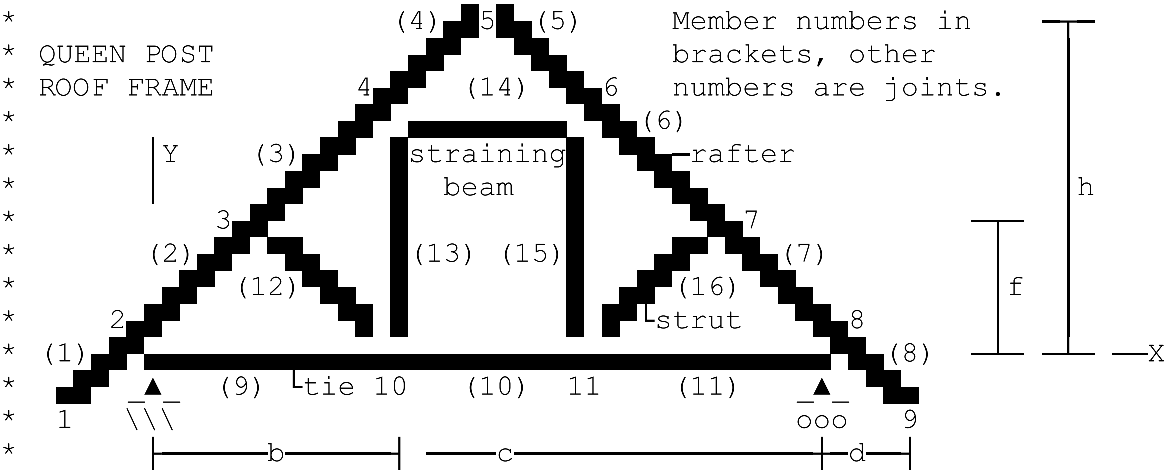

```
c=4 b=4/3 d=0.4 h=3 f=1.5 p4=1  ! Frame dims; pin jt at 4 (1=yes).
e=9E6  g=e/16    nsg=16     ! Young's & shear mod; No. of segments.
dy1=.200 dz1=.050 up=-2.3  ! Rafter sizes; udl on plan (-ve down)
dy7=.200 dz7=.050 p79=1    ! Tie joist sizes, pinned ends of tie.
dy10=.200 dz10=.050 pin=1  ! Internal sizes, pinned internals.
ul=-0.5  ur=1  ut=-2  ! Wind on l. & r. rafters +ve suct, tie udl.
vjl1=VEC(-1.5,-3,-3,-3,-3,-3,-1.5,2,2)  ! Vert. loads joints 1-9.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDS SYMM X c/2 ;1 -d -d*h/(c/2)
2 THRU 3 X 0 Y 0 XL (c/2)*f/h YL f ;4 c/2 h
5 THRU 7 SYMMETRY 3 THRU 1 ;8 b 0 ;9 c-b 0
JOINT RELEASES ;2 FORCE X -1 Y -1 ;6 FORCE Y -1
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;7 THRU 9 CHAIN 2,8,9,6
10 THRU 13 CHAIN 3,8,4,9,5 ;MEMBER RELEASES ;IF p4=1
3 END MOMENT Z ;ENDIF ;IF p79=1 ;7 START MOMENT Z ;9 END MOMENT Z
ENDIF ;IF pin=1 ;10 THRU 13 START MOMENT Z END MOMENT Z ;ENDIF
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 THRU 6 RECTANGLE DY dy1 DZ dz1 ;7 THRU 9 RECTANGLE DY dy7 DZ dz7
10 THRU 13 RECTANGLE DY dy10 DZ dz10
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<9 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIF W ur
7 THRU 9 FORCE Y UNIFORM W ut
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=9 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*9 ;! Convert to local axes ;lr=SQR((c/2)^2+h^2)
cx=(c/2)/lr cy=h/lr va=up*cy*cx vc=ul+up*cx^2 va1=VEC(va)*3
vc1=VEC(vc)*3 va=-up*cy*cx vc=ur+up*cx^2 va4=VEC(va)*3
vc4=VEC(vc)*3 va(7)=VEC(0)*3 vc(7)=VEC(ut)*3 va(10)=VEC(0)*4
vc(10)=VEC(0)*4 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm232.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE KING POST ROOF FRAME SUBJECTED TO JOINT LOADS, PLAN
TITLE UDL ON ROOF & WIND LOADING NORMAL TO RAFTERS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM233
TYPE PLANE FRAME run=0 ;DATE 02.06.05 ;NUMBER OF JOINTS 8
NUMBER OF MEMBERS 11 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
```



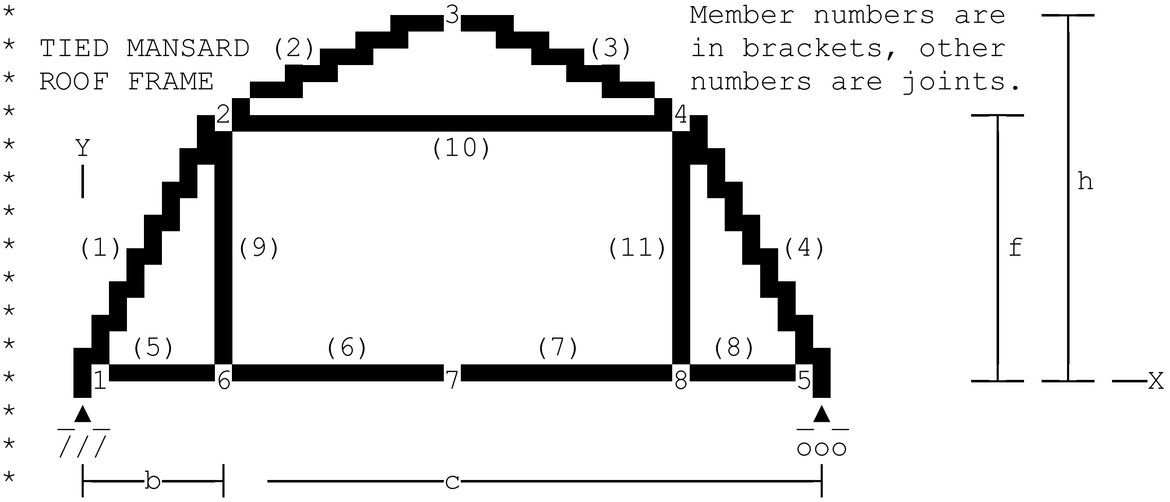

```
c=4 d=0.4  h=3 f=1.5 p4=1  ! Frame dims; pin jt at 4 (1=yes,0=No).
e=9E6  g=e/16    nsg=16    ! Young's & shear mod; No. of segments.
dy1=.200 dz1=.050 up=-2.3  ! Rafter sizes; udl on plan (-ve down)
dy7=.200 dz7=.050 p78=1    ! Tie joist sizes, pinned ends of tie.
dy9=.200  dz9=.050  pin=1  ! Internal sizes, pinned internals.
ul=-0.5  ur=1  ut=-2  ! Wind on l. & r. rafters +ve suct, tie udl.
vjl1=VEC(-1.5,-3,-3,-3,-3,-3,-1.5,-2)  ! Vert. loads joints 1-8.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDS SYMM X c/2 ;1 -d -d*h/(c/2)
2 THRU 3 X 0 Y 0 XL (c/2)*f/h YL f ;4 c/2 h
5 THRU 7 SYMMETRY 3 THRU 1 ;8 c/2 0
JOINT RELEASES ;2 FORCE X -1 Y -1 ;6 FORCE Y -1
MEMBER INCIDENCES ;1 THRU 6 RANGE 1,2 6,7 ;7 THRU 8 CHAIN 2,8,6
9 THRU 11 RANGE 3,8 5,8 ;MEMBER RELEASES ;IF p4=1
3 END MOMENT Z ;ENDIF ;IF p78=1 ;7 START MOMENT Z ;8 END MOMENT Z
ENDIF ;IF pin=1 ;9 THRU 11 START MOMENT Z END MOMENT Z ;ENDIF
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 THRU 6 RECTANGLE DY dy1 DZ dz1 ;7 THRU 8 RECTANGLE DY dy7 DZ dz7
9 THRU 11 RECTANGLE DY dy9 DZ dz9
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<8 GOTO 16 ;MEMBER LOADS ;1 THRU 6 FORCE Y PROJ UNIFORM W up
1 THRU 3 FORCE Y UNIFORM W ul ;4 THRU 6 FORCE Y UNIF W ur
7 THRU 8 FORCE Y UNIFORM W ut
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=8 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*8 ;! Convert to local axes ;lr=SQR((c/2)^2+h^2)
cx=(c/2)/lr cy=h/lr va=up*cy*cx vc=ul+up*cx^2 va1=VEC(va)*3
vc1=VEC(vc)*3 va=-up*cy*cx vc=ur+up*cx^2 va4=VEC(va)*3
vc4=VEC(vc)*3 va(7)=VEC(0)*2 vc(7)=VEC(ut)*2 va(9)=VEC(0)*3
vc(9)=VEC(0)*3 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm233.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE QUEEN POST ROOF FRAME SUBJECTED TO UDL ON FLOOR &
TITLE CEILING, WIND NORMAL TO RAFTERS & JOINT LOADS,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM234
TYPE PLANE FRAME run=0 ;DATE 03.06.05 ;NUMBER OF JOINTS 11
NUMBER OF MEMBERS 16 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
```




```
c=4  d=0.4  b=1.5  h=3   f=1  p5=1  ! Frame dims; pin at 5, 1=yes.
dy1=.200  dz1=.050  dy9=.150  dz9=.050   ! Rafter & floor sizes.
dy12=.100 dz12=.050 dy13=.100 dz13=.050  ! Strut & post/beam size.
e=9E6   g=e/16 nsg=16   ! Young's & shear mod.; No. of segments.
up=-2.3 ub=-1  uf=-3.5  ! Plan load on rafters, beam & floor.
ul=0.5   ur=-1   ! Wind on l. & r. raft. +ve suction.
p911=1   pin=1   ! Pinned tie ends; pinned internals, 1=Yes, 0=No.
vjl1=VEC(0,0,-5,-5,-5,-5,-5,0,0,-2,-2)   ! Vert loads joints 1-11.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES SYMMETRY X c/2
1 -d -d*h/(c/2) ;2 0 0 ;3 f*(c/2)/h f ;4 b b*h/(c/2) ;5 c/2 h
6 THRU 9 SYMMETRY 4 THRU 1 ;10 b 0 ;11 c-b 0
JOINT RELEASES ;2 FORCE X -1 Y -1 ;8 FORCE Y -1
MEMBER INCIDENCES ;1 THRU 8 RANGE 1,2 8,9
9 THRU 11 CHAIN 2,10,11,8 ;12 THRU 16 CHAIN 3,10,4,6,11,7
MEMBER RELEASES ;IF p5=1 THEN 4 END MOMENT Z
IF p911=1 ;9 START MOMENT Z ;11 END MOMENT Z ;ENDIF
IF pin=1 ;12 THRU 16 START MOMENT Z END MOMENT Z ;ENDIF
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 THRU 8 RECTANGLE DY dy1 DZ dz1 ;9 THRU 11 RECT DY dy9 DZ dz9
12 THRU 16 STEP 4 RECTANGLE DY dy12 DZ dz12
13 THRU 15 RECTANGLE DY dy13 DZ dz13
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<11 GOTO 16 ;MEMBER LOADS ;1 THRU 8 FORCE Y PROJ UNIFORM W up
9 THRU 11 FORCE Y UNIFORM W uf ;1 THRU 4 FORCE Y UNIFORM W ul
5 THRU 8 FORCE Y UNIFORM W ur ;14 FORCE Y UNIFORM W ub
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=11 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*11 ;! Conv.to local axes ;lr=SQR((c/2)^2+h^2)
cx=(c/2)/lr cy=h/lr va=up*cx*cy vc=ul+up*cx^2 va1=VEC(va)*4
vc1=VEC(vc)*4 va=-up*cx*cy vc=ur+up*cx^2 va5=VEC(va)*4
vc5=VEC(vc)*4 va9=VEC(0)*3 vc9=VEC(uf)*3 va12=VEC(0)*5
vc12=VEC(0)*5 vc(14)=ub ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm234.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

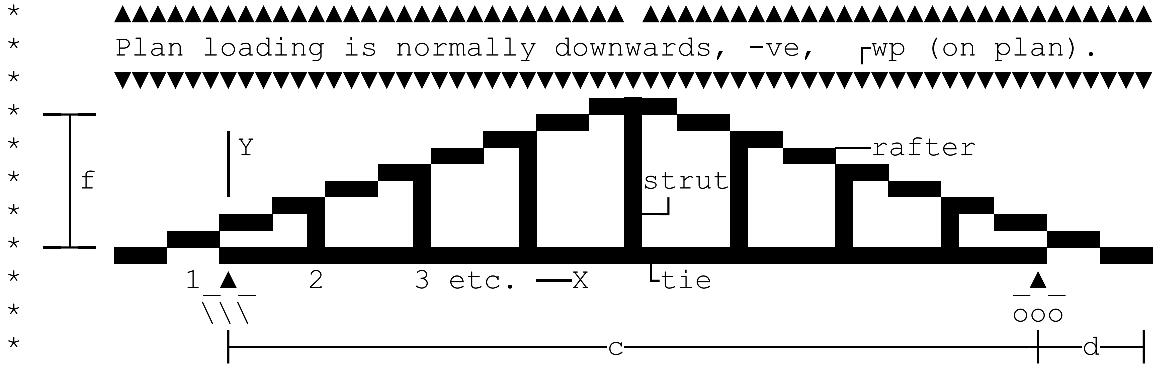
```
TITLE TIED MANSARD ROOF FRAME SUBJECTED TO UDL ON FLOOR &
TITLE CEILING, WIND NORMAL TO RAFTERS & JOINT LOADS,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM235
TYPE PLANE FRAME run=0 ;DATE 03.06.05 ;NUMBER OF JOINTS 8
NUMBER OF MEMBERS 11 ;NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
*/14
```




```
c=4 b=1 h=3 f=2.5 p2=1 p3=1  ! Frame dims; pin 2 & 3, 1=yes 0=No.
dy1=.200  dz1=.050  dy5=.150  dz5=.050   ! Rafter & floor sizes.
dy9=.100  dz9=.050  dy10=.100 dz10=.050  ! Post & beam sizes.
e=9E6   g=e/16 nsg=16   ! Young's & shear mod.; No. of segments.
up=-2.3 uc=-1  uf=-3.5  ! Plan load on rafters, ceiling & floor.
ul=-0.5   ur=1   ! Wind on left & right rafters, +ve suction.
p15=1    pin=1   ! Pinned tie ends; pinned internals, 1=Yes, 0=No.
vjl1=VEC(-2,-5,-5,-5,-2,-2,0,-2)   ! Vert loads joints 1-8.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES SYMMETRY X c/2 ;1 0 0
2 b f ;3 c/2 h ;4 THRU 5 SYMM 2 THRU 1 ;6 THRU 8 X b Y 0 XL c-b
JOINT RELEASES ;1 FORCE X -1 Y -1 ;5 FORCE Y -1 ;MEMBER INCIDENCES
1 THRU 4 RANGE 1,2 4,5 ;5 THRU 8 CHAIN 1,6,7,8,5
9 THRU 11 CHAIN 6,2,4,8 ;MEMBER RELEASE ;IF p2=1 ;2 START MOMENT Z
3 END MOMENT Z ;ENDIF ;IF p3=1 THEN 2 END MOMENT Z ;IF p15=1
5 START MOMENT Z ;8 END MOMENT Z ;ENDIF
IF pin=1 ;9 THRU 11 START MOMENT Z END MOMENT Z ;ENDIF
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES
1 THRU 4 RECTANGLE DY dy1 DZ dz1 ;5 THRU 8 RECT DY dy5 DZ dz5
9 THRU 11 STEP 2 RECTANGLE DY dy9 DZ dz9
10 RECTANGLE DY dy10 DZ dz10
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1 jn FORCE Y vjl(jn)
IF jn<8 GOTO 16 ;MEMBER LOADS ;1 THRU 4 FORCE Y PROJ UNIFORM W up
5 THRU 8 FORCE Y UNIFORM W uf ;1 THRU 2 FORCE Y UNIFORM W ul
3 THRU 4 FORCE Y UNIFORM W ur ;10 FORCE Y UNIFORM W uc
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=8 jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*8 ;! Convert to local axes ;d=c/2-b a=h-f
lr=SQR(b^2+f^2) cx=b/lr cy=f/lr va1=up*cx*cy vc1=ul+up*cx^2
va4=-up*cx*cy vc4=ur+up*cx^2 lr=SQR(d^2+a^2) cx=d/lr cy=a/lr
va2=up*cx*cy vc2=ul+up*cx^2 va3=-up*cx*cy vc3=ur+up*cx^2
va5=VEC(0)*4 vc5=VEC(uf)*4 va9=VEC(0)*3 vc9=VEC(0)*3 vc(10)=uc
ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm235.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE VIERENDEEL GIRDER WITH UDL'S & POINT LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM241
TYPE PLANE FRAME run=0 ;DATE 20.09.05 ;*/12
*        |wt        |wt        |wt        |wt        |wt
*        ▼   ┌ut    ▼   ┌ut    ▼   ┌ut    ▼   ┌ut    ▼
*        ▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
*                   4                  ┌top chord
*    ──── 2█████████████████  ███████████████████████nj
*     │    █        █                 █         █    █◄──nv=No. of
*     │ f  █  ┌ub    █  ┌ub    █  ┌ub    █  ┌ub    █   verticals
*     │    █▼▼▼▼▼▼▼▼▼█▼▼▼▼▼▼▼▼▼█▼▼▼▼▼▼▼▼▼█▼▼▼▼▼▼▼▼▼█   including
*    ────  █████████████████  ███████████████████████nj-1   ends.
*       1_▲_        3|         |    └bottom|chord     _▲_
*        \\\        ▼wb        ▼wb         ▼wb        ooo
*        ├─────────────────────c─────────────────────┤
c=18   f=1     ! Span of girder, height tie c.l. btm to top chord.
nv=19  nsg=12  ! No. of verticals including ends, No. of segments.
e=205E6 nu=0.3 ! Young's modulus and Poisson's ratio.
! For solid section give zero thickness and zero corner radius.
dyt=0.15 dzt=0.15 tt=0.01 rt=0.01 ! Top chord dims, thickn. & rad.
dyb=0.15 dzb=0.15 tb=0.01 rb=0.01 ! Btm chord dims, thickn. & rad.
dye=0.15 dze=0.15 te=0.01 re=0.01 ! End virts dims, thickn. & rad.
dyi=0.15 dzi=0.15 ti=0.01 ri=0.01 ! Internals dims, thickn. & rad.
ut=-1 ub=-1 wt=-5 wb=-5 ! UDL's & joint loads on top & btm chords.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj=2*nv nj ;NUMBER OF MEMBERS nm=nv+2*(nv-1) nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3 ;NUMBER OF SEGMENTS nsg
JOINT COORDINATES
1 THRU nj-1 STEP 2 X 0 Y 0 XL c ;2 THRU nj STEP 2 X 0 Y f XL c
JOINT RELEASES ;1 FORCE X -1 Y -1 ;nj-1 FORCE Y -1
MEMBER INCIDENCES ;1 THRU nv-1 RANGE 1,3 nj-3,nj-1
nv THRU 2*(nv-1) RANGE 2,4 nj-2,nj
2*nv-1 THRU nm RANGE 1,2 nj-1,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nv-1 RECTANGLE DY dyb DZ dzb T tb R rb
nv THRU 2*(nv-1) RECTANGLE DY dyt DZ dzt T tt R rt
2*nv-1 THRU nm STEP nv-1 RECTANGLE DY dye DZ dze T te R re
2*nv THRU nm-1 RECTANGLE DY dyi DZ dzi T ti R ri
LOADING CASE 1 ;MEMBER LOADS ;1 THRU nv-1 FORCE Y UNIFORM W ub
nv THRU 2*(nv-1) FORCE Y UNIFORM W ut ;JOINT LOADS
1 THRU nj-1 STEP 2 FORCE Y wb ;2 THRU nj STEP 2 FORCE Y wt
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va(1)=(0)*nm vc(1)=VEC(0)*nm nhm=nv-1 vc(1)=VEC(ub)*nhm ch9=1
vc(nv)=VEC(ut)*nhm vjl(1)=VEC(wb,wt)*nv hjl(1)=VEC(0)*nj ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm241.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PITCHED VIERENDEEL ROOF WITH UDL, POINT & WIND LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM242
TYPE PLANE FRAME run=0 ;DATE 20.09.05 ;*/12
*    wl┐  Wind loads are normal to rafters and positive up.  ┌wr
```




```
c=6   d=0.4   f=2    ! Span, overhang & height dimensions.
nv=7      vp=1       ! Odd No. of struts, pinned ends(1=Yes,0=No).
rp=1      tp=1       ! Pin at apex, pin ends of tie (1=Yes, 0=No).
wp=-2.3   wt=-1      ! Rafter & tie loading on plan (-ve down).
wl=-0.5   wr=1       ! Wind on left & right rafters (+ve suction).
e=9E6     nu=7       ! Young's mod. & Poisson's ratio, see Notes.
nsg=12               ! No. of segments for strain energy check.
dyr=.150  dzr=.050  tr=0  ! Rafter sizes, thickness tr=0 if solid.
dyt=.150  dzt=.050  tt=0  ! Tie size, thickness tt=0 if solid.
dyv=.100  dzv=.050  tv=0  ! Strut sizes, thickness tv=0 if solid.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj=2*(nv+2) nj
NUMBER OF MEMBERS nm=3*(nv+1)+1 nm ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 3 ;NUMBER OF SEGMENTS nsg
nb=nv+1 cv=c/nb nbh=nb/2 hv=f/nbh ;JOINT COORDINATES
1 THRU nv+2 X 0 Y 0 XL c ;nv+3 -d -d*f/(c/2)
nv+4 THRU nv+(nv+1)/2+3 X cv Y hv XL c/2 YL f
nv+(nv+1)/2+4 THRU nj-1 X c/2+cv Y f-hv XL c-cv YL hv
nj c+d -d*f/(c/2) ;JOINT RELEASES ;1 FORCE X -1 Y -1
nv+2 FORCE Y -1 ;MEMBER INCIDENCES ;1 THRU nb RANGE 1,2 nb,nb+1
nb+1 THRU nb+2 CHAIN nv+3,1,nv+4
nb+3 THRU 2*nb RANGE nv+4,nv+5 2*nb,2*nb+1
2*nb+1 THRU 2*nb+2 CHAIN nj-1,nb+1,nj
2*nb+3 THRU nm RANGE 2,nv+4 nb,nj-1
MEMBER RELEASES ;IF tp=1 THEN 1 START MOMENT Z ;nb END MOMENT Z
IF rp=1 THEN nb+nbh+1 END MOMENT Z ;nb+nbh+2 START MOMENT Z
IF vp=1 ;nm-nv+1 THRU nm-nbh START MOMENT Z END MOMENT Z
nm-nbh+1 START MOMENT Z
nm-nbh+2 THRU nm START MOMENT Z END MOMENT Z ;ENDIF
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nb RECTANGLE DY dyt DZ dzt T tt
nb+1 THRU 2*nb+2 RECTANGLE DY dyr DZ dzr T tr
2*nb+3 THRU nm RECTANGLE DY dyv DZ dzv T tv
LOADING CASE 1 ;MEMBER LOADS ;1 THRU nb FORCE Y UNIFORM W wt
nb+1 THRU 2*nb+2 FORCE Y PROJECTED UNIFORM W wp
nb+1 THRU nb+nbh+1 FORCE Y UNIFORM W wl
nb+nbh+2 THRU 2*nb+2 FORCE Y UNIFORM W wr
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
```

```
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
hjl1=VEC(0)*nj vjl1=VEC(0)*nj ;! Convert udl's to local axes.
va1=VEC(0)*nm vc1=VEC(0)*nm vc1=VEC(wt)*nb lr=SQR((c/2)^2+f^2)
cx=(c/2)/lr cy=f/lr va=wp*cy*cx vc=wl+wp*cx^2 nbh1=nbh+1
va(nb+1)=VEC(va)*nbh1 vc(nb+1)=VEC(vc)*nbh1 nst=nb+nbh+2
va=-wp*cy*cx vc=wr+wp*cx^2 va(nst)=VEC(va)*nbh1
vc(nst)=VEC(vc)*nbh1 ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm242.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE N/PRATT LATTICE PORTAL/GIRDER WITH/WITHOUT TAPER
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM244
TYPE PLANE FRAME run=0 ;DATE 21.09.05 ;*/7
*    |Y                         ┌at centre                ┌at ends.
*    ┌─────────────────────┐  ─┼─  ┌─────────────────────┐  ─┼─  ──►st
*    │ \ │ \ │ \ │ \ │ \ │  d   │ / │ / │ / │ / │ / │   c
*    ├───X─────────────────┤  ─┴─  ├─────────────────────┤  ─┼─  ──►sb
*    │                                                  │   │h=zero if
*  ──┴──                                              ──┴── ─┴─ girder.
*    ├────────────────────a───────────────────────────────┤
a=28  c=2.2  d=3.2  h=8  ! Dims of lattice girder, see above.
nb=8      nsg=8     ! No. of bays & segments for strain energy.
e=205E6   nu=0.3    ! Young's modulus, Poisson's ratio.
wt=-16    wb=-16    ! Plan loading on top/bottom chord, -ve down.
st=50     sb=50     ! Sway loading at top/bottom chord, +ve right.
dyt=.250  dzt=.250  tt=0.01 ! Top chord, thickness tt=0 if solid.
dyb=.250  dzb=.250  tb=0.01 ! Bot. chord, thickness tb=0 if solid.
dyi=.150  dzi=.150  ti=.008 ! Internals, thickness ti=0 if solid.
dye=.400  dze=.400  te=.012 ! End verts., thickness tv=0 if solid.
#cc924.stk !Import set of parameters if available from cc924.stk.
nj=(nb+1)*2 njt=nj odd=0 ;IF nb/2<>INT(nb/2) THEN odd=1 d=c !Fix.
nm=1+nb*4+odd nmt=nm ;IF h<>0 THEN njt=nj+2 nmt=nm+2 ;p=a/nb
IF c<>d AND odd=1 ;Odd number of bays cannot be tapered. ;ENDIF
NUMBER OF JOINTS njt ;NUMBER OF MEMBERS nmt ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 3 ;NUMBER OF SEGMENTS nsg ;JOINT COORDINATES
1 0 0 ;nj-1 a 0 ;3 THRU nj-3 STEP 2 X p Y 0 XL a-p
IF c=d THEN 2 THRU nj STEP 2 X 0 Y c XL a ;IF c<>d
njcl=1+nj/2 ;2 THRU njcl STEP 2 X 0 Y c XL a/2 YL d
njcl+2 THRU nj STEP 2 X a/2+p Y d-(d-c)/(nb/2) XL a YL c ;ENDIF
IF h<>0 ;njt-1 0 -h ;njt a -h ;ENDIF ;JOINT RELEASES ;IF h=0
1 FORCE X -1 Y -1 ;nj-1 FORCE Y -1 ;ENDIF ;IF h<>0
njt-1 THRU njt FORCE X -1 Y -1 ;ENDIF ;MEMBER INCIDENCES
1 THRU nb RANGE 2 4 nj-2 nj ;nb+1 THRU 2*nb RANGE 1 3 nj-3 nj-1
IF odd=0 ;* Internals for even number of bays.
2*nb+1 THRU 3*nb RANGE 1 2 nb nb+1
3*nb+1 THRU nm STEP 2 RANGE nb+1 nb+2 nj-1 nj
3*nb+2 THRU nm-1 STEP 2 RANGE nb+1 nb+4 nj-3 nj ;ENDIF
IF odd=1 ;* Internals for odd number of bays.
2*nb+1 THRU 3*nb+1 RANGE 1 2 nb+1 nb+2 ;3*nb+2 nb nb+3
3*nb+3 THRU nm STEP 2 RANGE nb+2 nb+3 nj-1 nj
3*nb+4 THRU nm-1 STEP 2 RANGE nb+2 nb+5 nj-3 nj ;ENDIF
IF h<>0 ;nmt-1 njt-1 1 ;nmt njt nj-1 ;ENDIF
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nb RECTANGLE DY dyt DZ dzt T tt
nb+1 THRU 2*nb RECTANGLE DY dyb DZ dzb T tb ;j1=nj-1 j2=nj-2
j3=nj-3 h1=nb+1 h2=nb+2 h3=2*nb-1 h4=2*nb i1=2*nb+1 i2=i1+1
i3=i1+2 i4=i1+3 i5=i1+4 i6=nm-4 i7=nm-3 i8=nm-2 i9=nm-1 l=nm
i2 THRU i9 RECTANGLE DY dyi DZ dzi T ti
i1 RECTANGLE DY dye DZ dze T te ;nm AS i1
IF h<>0 THEN nmt-1 THRU nmt AS i1
LOADING CASE 1 ;JOINT LOADS ;1 FORCE X sb ;2 FORCE X st
MEMBER LOADS ;1 THRU nb FORCE Y PROJECTED UNIFORM W wt
nb+1 THRU 2*nb FORCE Y PROJECTED UNIFORM W wb
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<njt GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=njt-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<njt GOTO 19
SOLVE ;*/10
```

```
*    2      1      4      2      6           +nb-1  +j2   +nb   +nj
*
*     \              \                 +i6          /           /
*       \              \      +i5                 /    +i8   /
*   +i1    +i2            +i4       +d           +i7       +i9      +c
*             \        +i3   \      at centre/              /      +l
*               \               \          /            /
*   1
*          +h1   3      +h2   5            +h3   +j3   +h4      +j1
*                        N (or PRATT) LATTICE
hjl1=VEC(0)*njt hjl1=sb hjl2=st vjl1=VEC(0)*njt va1=VEC(0)*nmt
vc1=VEC(0)*nmt vc(h1)=VEC(wb)*nb
IF odd=1 THEN vc1=VEC(wt)*nb ;! Convert udl's to local axes.
IF odd=0 ;ah=a/2 f=d-c lr=SQR(ah^2+f^2) cx=ah/lr cy=f/lr
va=wt*cy*cx vc=wt*cx^2 nbh=nb/2 va(1)=VEC(va)*nbh
vc(1)=VEC(vc)*nbh nst=nbh+1 va=-wt*cy*cx vc=wt*cx^2
va(nst)=VEC(va)*nbh vc(nst)=VEC(vc)*nbh ;ENDIF ;ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm244.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE HOWE LATTICE PORTAL/GIRDER WITH/WITHOUT TAPER
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM245
TYPE PLANE FRAME run=0 ;DATE 22.09.05 ;*/7
*  │Y                              ┌at centre                 ┌at ends.
*  ┌───┬───┬───┬───┬───┬───┐  ─┼─  ┌───┬───┬───┬───┬───┐  ─┼─  ──►st
*  │ / │ / │ / │ / │ / │    d    │ \ │ \ │ \ │ \ │ \ │    c
*  ├───┴─X─┴───┴───┴───┴───┘  ─┴─  └───┴───┴───┴───┴───┤  ─┼─  ──►sb
*  │                                                  │   │h=zero if
* ─┴─                                                ─┴─ ─┴─ girder.
*  ├─────────────────────────a────────────────────────┤
a=28  c=2.2  d=3.2  h=8  ! Dims of lattice girder, see above.
nb=8      nsg=8     ! No. of bays & segments for strain energy.
e=205E6   nu=0.3    ! Young's modulus, Poisson's ratio.
wt=-16    wb=-16    ! Plan loading on top/bottom chord, -ve down.
st=50     sb=50     ! Sway loading at top/bottom chord, +ve right.
dyt=.250  dzt=.250  tt=0.01 ! Top chord, thickness tt=0 if solid.
dyb=.250  dzb=.250  tb=0.01 ! Bot. chord, thickness tb=0 if solid.
dyi=.150  dzi=.150  ti=.008 ! Internals, thickness ti=0 if solid.
dye=.400  dze=.400  te=.012 ! End verts., thickness tv=0 if solid.
#cc924.stk !Import set of parameters if available from cc924.stk.
nj=(nb+1)*2 njt=nj odd=0 ;IF nb/2<>INT(nb/2) THEN odd=1 d=c !Fix.
nm=1+nb*4+odd nmt=nm ;IF h<>0 THEN njt=nj+2 nmt=nm+2 ;p=a/nb
IF c<>d AND odd=1 ;Odd number of bays cannot be tapered. ;ENDIF
NUMBER OF JOINTS njt ;NUMBER OF MEMBERS nmt ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 3 ;NUMBER OF SEGMENTS nsg ;JOINT COORDINATES
1 0 0 ;nj-1 a 0 ;3 THRU nj-3 STEP 2 X p Y 0 XL a-p
IF c=d THEN 2 THRU nj STEP 2 X 0 Y c XL a ;IF c<>d
njcl=1+nj/2 ;2 THRU njcl STEP 2 X 0 Y c XL a/2 YL d
njcl+2 THRU nj STEP 2 X a/2+p Y d-(d-c)/(nb/2) XL a YL c ;ENDIF
IF h<>0 ;njt-1 0 -h ;njt a -h ;ENDIF ;JOINT RELEASES ;IF h=0
1 FORCE X -1 Y -1 ;nj-1 FORCE Y -1 ;ENDIF ;IF h<>0
njt-1 THRU njt FORCE X -1 Y -1 ;ENDIF ;MEMBER INCIDENCES
1 THRU nb RANGE 2 4 nj-2 nj ;nb+1 THRU 2*nb RANGE 1 3 nj-3 nj-1
IF odd=0 ;* Internals for even number of bays.
2*nb+1 THRU 3*nb+1 STEP 2 RANGE 1 2 nb+1 nb+2
2*nb+2 THRU 3*nb STEP 2 RANGE 1 4 nb-1 nb+2
3*nb+2 THRU nm RANGE nb+2 nb+3 nj-1 nj ;ENDIF ;IF odd=1
* Internals for odd number of bays.
2*nb+1 THRU 3*nb STEP 2 RANGE 1 2 nb nb+1
2*nb+2 THRU 3*nb+1 STEP 2 RANGE 1 4 nb nb+3
3*nb+2 THRU nm RANGE nb+1 nb+2 nj-1 nj ;ENDIF
j2=nj-1 j3=nj-2 j4=nj-3 h1=nb-1 h3=nb+1 h4=nb+2 h5=2*nb-1
h6=2*nb i1=2*nb+1 i2=i1+1 i3=i1+2 i4=i1+3 i5=i1+4 i6=nm-4
i7=nm-3 i8=nm-2 i9=nm-1 l=nm ;IF h<>0 ;nmt-1 njt-1 1 ;nmt njt j2
ENDIF ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU nb RECTANGLE DY dyt DZ dzt T tt
nb+1 THRU 2*nb RECTANGLE DY dyb DZ dzb T tb
i2 THRU i9 RECTANGLE DY dyi DZ dzi T ti
i1 RECTANGLE DY dye DZ dze T te ;nm AS i1
IF h<>0 THEN nmt-1 THRU nmt AS i1
LOADING CASE 1 ;JOINT LOADS ;1 FORCE X sb ;2 FORCE X st
MEMBER LOADS ;1 THRU nb FORCE Y PROJECTED UNIFORM W wt
nb+1 THRU 2*nb FORCE Y PROJECTED UNIFORM W wb
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<njt GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=njt-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<njt GOTO 19
SOLVE ;*/10
```

```
*   2     1     4     2     6           +h1   +j3   +nb   +nj
*   ┌─────┬─────┬─────┬─────┐ ─── ┌─────┬─────┬─────┬─────┐┌──
*   │         / │         / │  │  │ \         │ \         ││
*   │       /   ├ +i3    / +i5 │ +i6  \       │   \     +l ││
*  +i1    +i2   │     +i4   │ +d          +i7 │     +i9   │ +c
*   │   /       │   /         at centre  \    ├ +i8   \   │ │
*   │ /         │ /         │  │  │         \ │         \ │ │
*  1└─────┴─────┴─────┴─────┘ ─── └─────┴─────┴─────┴─────┼─┴─
*   │     +h3   3     +h4   5           +h5   +j4   +h6   │ +j2
*                        HOWE LATTICE
hjl1=VEC(0)*njt hjl1=sb hjl2=st vjl1=VEC(0)*njt va1=VEC(0)*nmt
vc1=VEC(0)*nmt vc(h3)=VEC(wb)*nb
IF odd=1 THEN vc1=VEC(wt)*nb ;! Convert udl's to local axes.
IF odd=0 ;ah=a/2 f=d-c lr=SQR(ah^2+f^2) cx=ah/lr cy=f/lr
va=wt*cy*cx vc=wt*cx^2 nbh=nb/2 va(1)=VEC(va)*nbh
vc(1)=VEC(vc)*nbh nst=nbh+1 va=-wt*cy*cx vc=wt*cx^2
va(nst)=VEC(va)*nbh vc(nst)=VEC(vc)*nbh ;ENDIF ;ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm245.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE WARREN PORTAL/GIRDER WITH END DIAGONALS IN TENSION
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM246
TYPE PLANE FRAME run=0 ;DATE 21.09.05 ;*/7
*    |Y                        ┌at centre                  ┌at ends.
*    ┌───────┬───────┬──────  ─┼─  ──────┬───────┬──────┐  ─┼─  ──►st
*    │ \   /   \   /   \       d      /   \   /   \   /  │   c
*    ├─────┴─X─────┴───────┴─ ─┴─  ┴───────┴───────┴────┤  ─┼─  ──►sb
*    │                                                  │   │h=zero if
*  ──┴──                                              ──┴── ─┴─ girder.
*    ├──────────────────────a───────────────────────────┤
a=28  c=2.2  d=3.2  h=8  ! Dims of lattice girder, see above.
nb=8  nsg=8  ! No. of bays i.e. diags, & segments for str. energy.
e=205E6   nu=0.3    ! Young's modulus, Poisson's ratio.
wt=-16    wb=-16    ! Plan loading on top/bottom chord, -ve down.
st=50     sb=50     ! Sway loading at top/bottom chord, +ve right.
dyt=.250  dzt=.250  tt=0.01 ! Top chord, thickness tt=0 if solid.
dyb=.250  dzb=.250  tb=0.01 ! Bot. chord, thickness tb=0 if solid.
dyi=.150  dzi=.150  ti=.008 ! Internals, thickness ti=0 if solid.
dye=.400  dze=.400  te=.012 ! End verts., thickness tv=0 if solid.
#cc924.stk !Import set of parameters if available from cc924.stk.
nj=3+nb njt=nj odd=0 ;IF nb/2<>INT(nb/2) THEN odd=1 ;nm=2*nb+3
nmt=nm ;IF h<>0 THEN njt=nj+2 nmt=nm+2 ;p=a/nb
IF odd=1 THEN Must have even number of bays.
IF INT(nb/4)<>nb/4 AND c<>d ;* Can't be tapered so fix d=c ;d=c
ENDIF ;NUMBER OF JOINTS njt ;NUMBER OF MEMBERS nmt
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3 ;NUMBER OF SEGMENTS nsg
JOINT COORDINATES
1 0 0 ;nj a 0 ;3 THRU nj-2 STEP 2 X p Y 0 XL a-p
IF c=d THEN 2 THRU nj-1 STEP 2 X 0 Y c XL a ;IF c<>d
njcl=(1+nj)/2 2 THRU njcl STEP 2 X 0 Y c XL a/2 YL d
i=njcl+2 j=nj-1 x=a/2+2*p y=d-2*(d-c)/(nb/2)
IF njcl+2<>nj-1 THEN i THRU j STEP 2 X x Y y XL a YL c
IF njcl+2=nj-1 THEN nj-1 X a Y c ;ENDIF ;IF h<>0 ;njt-1 0 -h
njt a -h ;ENDIF ;JOINT RELEASES ;IF h=0 ;1 FORCE X -1 Y -1
nj FORCE Y -1 ;ENDIF ;IF h<>0 ;njt-1 THRU njt FORCE X -1 Y -1
ENDIF ;MEMBER INCIDENCES ;1 THRU nb/2 RANGE 2 4 nj-3 nj-1
nb/2+1 THRU nb+1 RANGE 1 3 nj-2 nj ;nb+2 THRU nm RANGE 1 2 nj-1 nj
IF h<>0 ;nmt-1 njt-1 1 ;nmt njt nj ;ENDIF
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nb/2 RECTANGLE DY dyt DZ dzt T tt
nb/2+1 THRU nb+1 RECTANGLE DY dyb DZ dzb T tb
nb+3 THRU nm-1 RECTANGLE DY dyi DZ dzi T ti
nb+2 RECTANGLE DY dye DZ dze T te ;nm AS nb+2
IF h<>0 THEN nmt-1 THRU nmt AS nb+2
LOADING CASE 1 ;MEMBER LOADS
1 THRU nb/2 FORCE Y PROJECTED UNIFORM W wt
nb/2+1 THRU nb+1 FORCE Y PROJECTED UNIFORM W wb
JOINT LOADS ;1 FORCE X sb ;2 FORCE X st
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<njt GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=njt-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<njt GOTO 19
SOLVE ;j1=nj-1 j2=nj-2 h1=nb/2+1 h2=h1+1 h3=nb h4=nb+1 l=nm ;*/10
```

```
*    2           1           4                 +nb/2         +j1
*    ┌───────────┴───────────┬  ──┬─  ┬─────────────┴───────────┬─┬──
*    │ \                   / │   │    \                    /  │ │
*    │   \               /       │      \                /    │ │
*   +nb+2  +nb+3       +nb+4  +d          +nm-2       +nm-1   │ +c
*    │         \       /        at centre    \          /     │ │
*    │           \   /           │             \   /        +l  │
*   1├─────┬─────┴─────────┬─  ─┴─  ─┬─────────┴─────┬───────┤─┴──
*    │     +h1   3         +h2       +h3          +j2   +h4   │ +nj
*                WARREN - END DIAGONALS IN TENSION
hjl1=VEC(0)*njt hjl1=sb hjl2=st vjl1=VEC(0)*njt va1=VEC(0)*nmt
vc1=VEC(0)*nmt nbh=nb/2 nbb=nbh+1 nmb=nb/2+1 vc(h1)=VEC(wb)*nmb
IF c=d THEN vc1=VEC(wt)*nbh ;! Convert udl's to local axes.
IF c<>d ;ah=a/2 f=d-c lr=SQR(ah^2+f^2) cx=ah/lr cy=f/lr
va=wt*cy*cx vc=wt*cx^2 nbq=nbh/2 va(1)=VEC(va)*nbq
vc(1)=VEC(vc)*nbq nst=nbq+1 va=-wt*cy*cx vc=wt*cx^2
va(nst)=VEC(va)*nbq vc(nst)=VEC(vc)*nbq ;ENDIF ;ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm246.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE WARREN PORTAL/GIRDER, END DIAGONALS IN COMPRESSION
TITLE INCLUDING CHECKS FOR: COMPATIBILITY & EQUILIBRIUM
TITLE FOR EACH MEMBER, OVERALL EQUILIBRIUM, AND CHECK
TITLE THAT STRAIN ENERGY EQUALS EXTERNAL WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM247
TYPE PLANE FRAME run=0 ;DATE 21.09.05 ;*/7
*    |Y                          ┌at centre                    ┌at ends.
*    ┌──────┬────────┬────────┬─  ─┼─  ─┬────────┬────────┬───┐  ─┼─  ──►st
*    |  /   \   /   \   /      d     \   /   \   /   \  |   c
*    ├─────X─┴────────┴────────  ─┴─  ──────┴────────┴────── |  ─┼─  ──►sb
*    |                                                       |   |h=zero if
*  ──┴──                                                   ──┴── ─┴─ girder.
*    ├──────────────────────────a────────────────────────────┤
a=28  c=2.2  d=3.2  h=8  ! Dims of lattice girder, see above.
nb=10 nsg=8  ! No. of bays i.e. diags, & segments for str. energy.
e=205E6   nu=0.3    ! Young's modulus, Poisson's ratio.
wt=-16    wb=-16    ! Plan loading on top/bottom chord, -ve down.
st=50     sb=50     ! Sway loading at top/bottom chord, +ve right.
dyt=.250  dzt=.250  tt=0.01 ! Top chord, thickness tt=0 if solid.
dyb=.250  dzb=.250  tb=0.01 ! Bot. chord, thickness tb=0 if solid.
dyi=.150  dzi=.150  ti=.008 ! Internals, thickness ti=0 if solid.
dye=.400  dze=.400  te=.012 ! End verts., thickness tv=0 if solid.
#cc924.stk !Import set of parameters if available from cc924.stk.
nj=3+nb njt=nj odd=0 ;IF nb/2<>INT(nb/2) THEN odd=1 ;nm=2*nb+3
nmt=nm ;IF h<>0 THEN njt=nj+2 nmt=nm+2 ;p=a/nb
IF odd=1 THEN Must have even number of bays. ;nb'=nb-2
IF INT(nb'/4)<>nb'/4 AND c<>d ;* Can't be tapered so fix d=c ;d=c
ENDIF ;NUMBER OF JOINTS njt ;NUMBER OF MEMBERS nmt
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
NUMBER OF SEGMENTS nsg ;j1=nj-1 j2=nj-2 l=nm
JOINT COORDINATES ;2 0 0 ;nj-1 a 0 ;IF nb=4 THEN 4 a/2 0
IF nb<>4 THEN 4 THRU nj-3 STEP 2 X 2*p Y 0 XL a-2*p ;1 0 c ;nj a c
IF c=d THEN 3 THRU nj-2 STEP 2 X p Y c XL a-p ;IF c<>d
njcl=(1+nj)/2 3 THRU njcl STEP 2 X p Y c+(d-c)/(nb/2) XL a/2 YL d
y'=(d-c)/(nb/2) i=njcl+2 j=nj-2 x=a/2+2*p y=d-2*y'
IF njcl+2<>nj-2 THEN i THRU j STEP 2 X x Y y XL a-p YL c+y'
IF njcl+2=nj-2 THEN nj-2 a-p c+(d-c)/(nb/2) ;ENDIF ;IF h<>0
njt-1 0 -h ;njt a -h ;ENDIF ;JOINT RELEASES ;IF h=0
2 FORCE X -1 Y -1 ;nj-1 FORCE Y -1 ;ENDIF
IF h<>0 THEN njt-1 THRU njt FORCE X -1 Y -1
MEMBER INCIDENCES ;1 THRU nb/2 RANGE 2 4 nj-3 nj-1
nb/2+1 THRU nb+1 RANGE 1 3 nj-2 nj ;nb+2 THRU nm RANGE 1 2 nj-1 nj
IF h<>0 ;nmt-1 njt-1 2 ;nmt njt nj-1 ;ENDIF
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nb/2 RECTANGLE DY dyt DZ dzt T tt
nb/2+1 THRU nb+1 RECTANGLE DY dyb DZ dzb T tb
nb+3 THRU nm-1 RECTANGLE DY dyi DZ dzi T ti
nb+2 RECTANGLE DY dye DZ dze T te ;nm AS nb+2
IF h<>0 THEN nmt-1 THRU nmt AS nb+2
LOADING CASE 1 ;MEMBER LOADS
1 THRU nb/2+1 FORCE Y PROJECTED UNIFORM W wt
nb/2+2 THRU nb+1 FORCE Y PROJECTED UNIFORM W wb
JOINT LOADS ;1 FORCE X st ;2 FORCE X sb
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<njt GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=njt-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<njt GOTO 19
SOLVE ;h1=nb/2+2 h ;*/10
```

```
*    1     1     3           2     +nb/2        +j2  +nb/2+1  +nj
*
*             /   \                      /   \        +1
*           /       \                  /       \
*   +nb+2  +nb+3       +nb+4   +d        +nm-2      +nm-1     +c
*       /               \    at centre/              \
*     /                   \        /                   \
*   2
*                +nb/2+2     4     +nj-3       +nb+1          +j1
*               WARREN - END DIAGONALS IN COMPRESSION
hjl1=VEC(0)*njt hjl1=st hjl2=sb vjl1=VEC(0)*njt va1=VEC(0)*nmt
vc1=VEC(0)*nmt nbh=nb/2 vc(h1)=VEC(wb)*nbh nbh1=nbh+1
IF c=d THEN vc1=VEC(wt)*nbh1 ;! Convert udl's to local axes.
IF c<>d ;ah=a/2 f=d-c lr=SQR(ah^2+f^2) cx=ah/lr cy=f/lr
va=wt*cy*cx vc=wt*cx^2 nbq=(nbh+1)/2 va(1)=VEC(va)*nbq
vc(1)=VEC(vc)*nbq nst=nbq+1 va=-wt*cy*cx vc=wt*cx^2
va(nst)=VEC(va)*nbq vc(nst)=VEC(vc)*nbq ;ENDIF ;ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm247.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE MULTI-STOREY FRAME WITH nbay BAYS & ns STOREYS
TITLE WITH FIXED OR PINNED FEET, SUBJECTED TO VERTICAL
TITLE AND SWAY LOADS, INCLUDING A CHECK OF RESULTS
TITLE AGAINST THOSE FROM MOMENT DISTRIBUTION.
MADEBY DWB ;DATE 03.11.04 ;METHOD ELASTIC ;REFNO VM260
PRINT DATA RESULTS FROM 1 ;TYPE PLANE FRAME run=0 ;*/13
*  Loading        ▲pv()  ├───ub()────┤       Beams numbered left to
*  case 2┐  ├─pa()┤      ├─ua()─►▲▲▲▲▲uw()   right, top to bottom.
*  ph(1)─┼──►╔═══════════╦═════════════╦══   ══╦══════════╗  ──┬──
*        │   ║└beam 1    ║└beam 2      ║       ║          ║    │ht(1)
*  ph(2)─┼──►╠═══════════╬═════════════╬══   ══╬══════════╣  ──┼──
*        │   ║           ║             ║       ║          ║    │ht(2)
*  ph(3)─┼──►╠═══════════╬═════════════╬══   ══╬══════════╣  ──┼──
*
*  ph(ns)┴──►╟───────────╫─────────────╫──   ──╫──────────╢  ──┬──
*            ║           ║             ║       ║          ║    │ht(ns)
*           _o_         _o_           _o_     _o_        _o_ ──┴──
*           ///         ///           ///     ///        ///
*            ├───sp(1)───┼────sp(2)────┼►►  ─┼────sp(nb)───┤
! All parameters are contained within the next 20 lines; US units.
nb=4 ns=3 fix=0  ! No. of bays & storeys; fix=0(pin) or 1(fixed).
sp(1)=VEC(18.75,22,24,20.25)   ! Spans of bays left to right.
ht(1)=VEC(15,16,18.75)         ! Storey heights top to bottom.
ey=0.58E6/3 nu=0.2  ! Young's, /3 for creep factor=2,; nu=Poisson.
* Beam & column moments of inertia, left to right, top to bottom.
bi(1)=VEC(1,1,1E-6,1E-6,2,2,4,1E-6,3,3.6,4.25,3)
ci(1)=VEC(1,1,1,1E-6,1E-6,2,3,3,2.5,1E-6,3,4,4,3.75,2.5)
* Beam & column cross sect. areas, left to right, top to bottom.
ba(1)=VEC(1,1,1E-6,1E-6,2,2,4,1E-6,3,3.6,4.25,3)
ca(1)=VEC(1,1,1,1E-6,1E-6,2,3,3,2.5,1E-6,3,4,4,3.75,2.5)
u(1)=VEC(-2,-2,0,0,-4,0,-4,0,-4,0,-4.25,0)  ! Full udl's in order.
np=0 n=8                      ! No. partial & concentrated loads.
ph(1)=VEC(0,0,0)              ! Sway loads, top to bottom.
bu(1)=VEC(6,10,12)            ! Beam Nos. with partial udl's.
uw(1)=VEC(-4,-4,-4)           ! Magnitudes of partial udl's.
ua(1)=VEC(0,11,11)            ! Dist. to start of partial udl's.
ub(1)=VEC(11,22,13)           ! Dist. to end of partial udl's.
bc(1)=VEC(1,2,2,7,7,11,12,12)  ! Beam Nos. containing conc loads.
pv(1)=VEC(-10,5,-5,-10,-2,-20,-5,1)    ! Magnitudes of conc loads.
pa(1)=VEC(18.75,5,10,4,11.5,7.125,11,13)   ! Dist. to conc loads.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nc=nb+1 nl=ns+1 nj=nc*nl nj
NUMBER OF MEMBERS nbm=nb*ns ncl=nc*ns nm=nbm+ncl nm
NUMBER OF SUPPORTS nc ;NUMBER OF LOADINGS 1
JOINT COORDINATES  !Left to right top to bottom.
y=0 i=0 sp(0)=0 ht(0)=0 ;:10 ;i=i+1 y=y+ht(i) ;IF i<ns GOTO 10
j=0 i=0 ;:20 ;i=i+1 y=y-ht(i-1) x=-sp(1) col=0
:30 ;j=j+1 ;x=x+sp(col) ;col=col+1 ;IF i<nl THEN j x y
IF i=nl THEN j x y SUPPORT ;IF col<nc GOTO 30 ;IF i<nl GOTO 20
IF fix=0 ;JOINT RELEASES ;nj-nb THRU nj MOMENT Z ;ENDIF
MEMBER INCIDENCES ;i=0 js=1-nc ms=1-nb ms'=1+nb*ns-nc
:40 ;i=i+1 ms=ms+nb ms'=ms'+nc js=js+nc
ms THRU ms+nb-1 RANGE js js+1 js+nb-1 js+nb
ms' THRU ms'+nb RANGE js+nc js js+nb+nc js+nb ;IF i<ns GOTO 40
CONSTANTS E ey ALL G g=ey/(2*(1+nu)) g ALL ;MEMBER PROPERTIES ;i=0
:50 ;i=i+1 ax=ba(i) ay=ax*5/6 ;IF nu=1E-12 THEN ax=ax*1E6 ay=0
i AX ax AY ay IZ bi(i) ;IF i<nbm GOTO 50 ;j=0 ;:60 ;i=i+1 j=j+1
ax=ca(j) ay=ax*5/6 ;IF nu=1E-12 THEN ax=ax*1E6 ay=0
i AX ax AY ay IZ ci(j) ;IF j<ncl GOTO 60
```

```
LOADING CASE 1 ;MEMBER LOADS ;i=0 ;:70 ;i=i+1
IF u(i)<>0 THEN i FORCE Y UNIFORM W u(i) ;IF i<nbm GOTO 70
IF np>0 ;i=0 ;:75 ;i=i+1
bu(i) FORCE Y UNIFORM W uw(i) LA ua(i) LB ub(i)
IF i<np GOTO 75 ;ENDIF ;IF n>0 ;i=0 ;:80 ;i=i+1
bc(i) FORCE Y CONCENTRATED P pv(i) L pa(i) ;IF i<n GOTO 80 ;ENDIF
JOINT LOADS ;j=1-nc i=0 ;:90 ;j=j+nc i=i+1
j FORCE X ph(i) ;IF i<ns GOTO 90 ;SOLVE ;status=1 gtot=0 nur=0
i1=ns imax=nl j1=nb jmax=nc m=np+nbm !Distrib. factors.
i=0 ;:100 ;i=i+1 suk(i)=0 j=0 ;:110 ;j=j+1 jm1=j-1 im1=i-1
i'j=im1*jmax+j b'j=im1*(jmax-1)+j im1'j=i'j-jmax skl=0
IF j>1 THEN b'jm1=b'j-1 skl=bi(b'jm1)/sp(jm1) ;ska=0
IF i>1 THEN ska=ci(im1'j)/ht(im1) ;skr=0
IF j<jmax THEN skr=bi(b'j)/sp(j) ;skb=ci(i'j)/ht(i)
IF i>=i1 AND fix<=0 THEN skb=0.75*ci(i'j)/ht(i)
sumk=skl+ska+skr+skb
IF sumk<=0 THEN dfl(i'j)=0 dfa(i'j)=0 dfr(i'j)=0 dfb(i'j)=0
IF sumk>0 ;dfl(i'j)=skl/sumk dfa(i'j)=ska/sumk dfr(i'j)=skr/sumk
dfb(i'j)=skb/sumk ;ENDIF ;ssk(i'j)=skb/ht(i)
suk(i)=suk(i)+ssk(i'j) ;IF j<jmax GOTO 110 ;IF i<i1 GOTO 100
i=0 ;:120 ;i=i+1 j=0 ;:130 ;j=j+1 i'j=(i-1)*jmax+j
sdf(i'j)=ssk(i'j)/suk(i) ;IF j<jmax GOTO 130 ;IF i<i1 GOTO 120
i=0 ;:140 ;i=i+1 j=0 ;:150 ;j=j+1 i'j=(i-1)*jmax+j vr(i'j)=0
vl(i'j)=0 eml(i'j)=0 emr(i'j)=0 ema(i'j)=0 emb(i'j)=0
IF j<jmax GOTO 150 ;IF i<i1 GOTO 140 ;! UDL's ;k=0 ;:170 ;k=k+1
IF k<=np ;ui(k)=INT((bu(k)-1)/nb)+1 uj(k)=bu(k)-INT(ui(k)-1)*nb
ENDIF ;IF k>np ;i=k-np ui(k)=INT((i-1)/nb)+1 ua(k)=0
uj(k)=i-INT(ui(k)-1)*nb uw(k)=u(i) d=uj(k) ub(k)=sp(d) ;ENDIF
j'=uj(k) i'j=(ui(k)-1)*jmax+j' i'jp1=i'j+1 a=ua(k) d=ub(k) b=d-a
l=sp(j') c=l-d e=b+c w=uw(k)*b
ma=-w*(e^3*(4*l-3*e)-c^3*(4*l-3*c))/(12*l^2*b)
mb=-w*(d^3*(4*l-3*d)-a^3*(4*l-3*a))/(12*l^2*b)
ra=w*(c+b/2)/l rb=w*(a+b/2)/l vr(i'j)=vr(i'j)+ra
vl(i'jp1)=vl(i'jp1)-rb emr(i'j)=emr(i'j)+ma
eml(i'jp1)=eml(i'jp1)-mb ;IF k<m GOTO 170
! Concentrated ;k=0 ;:180 ;k=k+1 ;IF k>n GOTO 190
pi(k)=INT((bc(k)-1)/nb)+1 pj(k)=bc(k)-INT(pi(k)-1)*nb
j'=pj(k) b=sp(j')-pa(k) i'j=(pi(k)-1)*jmax+j'
vr(i'j)=vr(i'j)+pv(k)*b/sp(j') i'jp1=i'j+1
vl(i'jp1)=vl(i'jp1)-pv(k)*pa(k)/sp(j')
emr(i'j)=emr(i'j)-pv(k)*pa(k)*b*b/(sp(j')*sp(j'))
eml(i'jp1)=eml(i'jp1)+pv(k)*pa(k)*pa(k)*b/(sp(j')*sp(j'))
! pv(k)*pa(k)/sp(j') pv(k)*b/sp(j')
GOTO 180 ;:190 ;thv=0 i=0 ;:200 ;i=i+1 thv=thv+ph(i) j=0
:210 ;j=j+1 i'j=(i-1)*jmax+j emb(i'j)=0.5*sdf(i'j)*thv*ht(i)
ip1'j=i'j+jmax ema(ip1'j)=emb(i'j) ;IF j<jmax GOTO 210
IF i<i1 GOTO 200 ;IF fix>0 GOTO 230 ;j=0 ;:220 ;j=j+1
i1'j=(i1-1)*jmax+j emb(i1'j)=2*emb(i1'j) imx'j=(imax-1)*jmax+j
ema(imx'j)=0 ;IF j<jmax GOTO 220 ;:230 ;l=0 f371=0
eps=1E-6 !Accuracy limit ;:270 ;thv=0 i=0
:280 ;i=i+1 j=0 ;:290 ;j=j+1 i'j=(i-1)*jmax+j
IF i<=1 THEN dfa(i'j)=0 ema(i'j)=0
IF j<=1 THEN dfl(i'j)=0 eml(i'j)=0
IF j>1 AND j>=jmax THEN dfr(i'j)=0 emr(i'j)=0
sum1=eml(i'j)+ema(i'j)+emr(i'j)+emb(i'j) f269=0
IF sum1<=eps AND -sum1<=eps ;l=l+1 ;IF l<i1*jmax THEN f269=1
IF l>=i1*jmax THEN f371=1 ;ENDIF
IF -sum1>eps OR sum1>eps THEN l=0
```

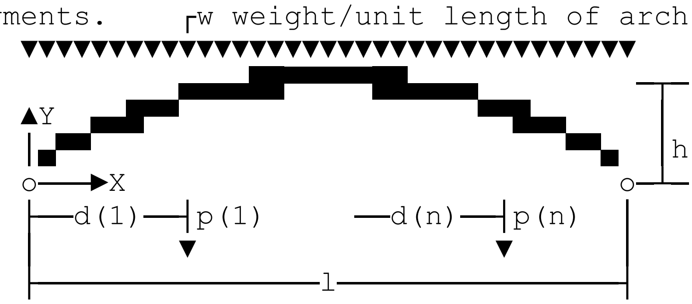
```
IF f269<>0 OR f371<>0 GOTO 295
eml(i'j)=eml(i'j)-dfl(i'j)*sum1 ema(i'j)=ema(i'j)-dfa(i'j)*sum1
emr(i'j)=emr(i'j)-dfr(i'j)*sum1 emb(i'j)=emb(i'j)-dfb(i'j)*sum1
IF j>1 ;i'jm1=i'j-1 emr(i'jm1)=emr(i'jm1)-0.5*dfl(i'j)*sum1
IF j<jmax THEN i'jp1=i'j+1 eml(i'jp1)=eml(i'jp1)-0.5*dfr(i'j)*sum1
ENDIF
IF j<=1 THEN i'jp1=i'j+1 eml(i'jp1)=eml(i'jp1)-0.5*dfr(i'j)*sum1
IF i>1 THEN im1'j=i'j-jmax emb(im1'j)=emb(im1'j)-0.5*dfa(i'j)*sum1
IF i>=i1 AND fix>0
ip1'j=i'j+jmax ema(ip1'j)=ema(ip1'j)-0.5*dfb(i'j)*sum1 ;ENDIF
IF i<i1 ;ip1'j=i'j+jmax ema(ip1'j)=ema(ip1'j)-0.5*dfb(i'j)*sum1
ENDIF ;:295 ;IF j<jmax GOTO 290 ;IF f371<>0 GOTO 315 ;sum2=0
thv=thv+ph(i) j=0 ;:300 ;j=j+1 i'j=(i-1)*jmax+j ip1'j=i'j+jmax
sum2=sum2+emb(i'j)+ema(ip1'j) ;IF j<jmax GOTO 300
sum2=sum2-thv*ht(i) j=0 ;:310 ;j=j+1
i'j=(i-1)*jmax+j ip1'j=i'j+jmax
emb(i'j)=emb(i'j)-0.5*sdf(i'j)*sum2 ;IF i>=i1 AND fix<=0
ema(ip1'j)=0 emb(i'j)=emb(i'j)-0.5*sdf(i'j)*sum2 ;ENDIF
IF i>=i1 AND fix>0 THEN ema(ip1'j)=ema(ip1'j)-0.5*sdf(i'j)*sum2
IF i<i1 THEN ema(ip1'j)=ema(ip1'j)-0.5*sdf(i'j)*sum2
IF j<jmax GOTO 310 ;:315 ;IF i<i1 GOTO 280 ;IF f371<=0 GOTO 270
i=0 ;:320 ;i=i+1 j=0 ;:330 ;j=j+1 i'j=(i-1)*jmax+j i'jp1=i'j+1
! Change sign. ;vr(i'j)=-vr(i'j)+(emr(i'j)+eml(i'jp1))/sp(j)
vl(i'jp1)=vl(i'jp1)-(emr(i'j)+eml(i'jp1))/sp(j) ;IF j<j1 GOTO 330
IF i<i1 GOTO 320 ;i=0 ;:340 ;i=i+1 j=0 ;:350 ;j=j+1
i'j=(i-1)*jmax+j ip1'j=i'j+jmax
vb(i'j)=-(emb(i'j)+ema(ip1'j))/ht(i)
! Change sign. ;va(ip1'j)=-vb(i'j) ;IF j<jmax GOTO 350
IF i<i1 GOTO 340 ;*/20
* Beam BMs/SFs, left/right top/bottom, N=NL-STRESS, M=Mmt Dist.
* Method Left BM       Right BM     Left SF      Right SF     %age
* & beam                                                      diff
i=0 k=0 ;:460 ;i=i+1 j=0 ;:470
j=j+1 i'j=(i-1)*jmax+j i'jp1=i'j+1 k=k+1
dn1=ARR(13,k,3) dn2=ARR(13,k,6) dn3=ARR(13,k,2) dn4=ARR(13,k,5)
c1=emr(i'j) c2=eml(i'jp1) c3=vr(i'j) c4=vl(i'jp1)
l=0 per1=0 ok1=0 ;:475 ;l=l+1 d1=dn(l) d2=c(l)
#vmper.ndf !Compute percentage difference as text message.
IF per>per1 THEN per1=per ok1=ok ;IF l<4 GOTO 475 ;ok=ok1
* M  +k  +c1           +c2          +c3          +c4
* N  +k  +dn1          +dn2         +dn3         +dn4         $ok
IF j<j1 GOTO 470 ;IF i<i1 GOTO 460 ;* ;*/20
* Colm. BMs/SFs, left/right top/bottom, N=NL-STRESS, M=Mmt Dist.
* Method Top BM         Bottom BM    Top SF       Bottom SF   %age
* & col.                                                      diff
i=0 k=0 ;:480 ;i=i+1 j=0 ;:490
j=j+1 i'j=(i-1)*jmax+j ip1'j=i'j+jmax k=k+1
c1=emb(i'j) c2=ema(ip1'j) c3=vb(i'j) c4=va(ip1'j) e=k+nbm
dn1=ARR(13,e,6) dn2=ARR(13,e,3) dn3=ARR(13,e,5) dn4=ARR(13,e,2)
l=0 per1=0 ok1=0 ;:495 ;l=l+1 d1=dn(l) d2=c(l)
#vmper.ndf !Compute percentage difference as text message.
IF per>per1 THEN per1=per ok1=ok ;IF l<4 GOTO 495 ;ok=ok1
* M  +k  +c1           +c2          +c3          +c4
* N  +k  +dn1          +dn2         +dn3         +dn4         $ok
IF j<jmax GOTO 490 ;IF i<i1 GOTO 480 ;fnm=$(vm260.stk) ;#vmres.ndf
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE MULTI-STOREY FRAME HAVING nb BAYS & ns STOREYS
TITLE SUBJECTED TO U.D.L. & VERTICAL & HORIZ POINT LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 11.11.05 ;TYPE PLANE FRAME run=0 ;REFNO VM262
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;*/14
nb=1   ns=2   ! No. of bays & storeys.          ╔════════╦═══    ═══╗
s=3.82 h=3.82 ! Bay span & storey height.       ║        ║nj-1   nj║
axb=38.2E-4   ! X-sect. area of beams.     ▲
ayb=23.9E-4   ! Shear area of beams.       │    ║        ║         ║
izb=558E-8    ! Mom. of inert. of beams.───┼─   ╠════════╬═══   ═══╣
dpb=111.2E-6  ! Depth of beams.           .│    ║        ║         ║
axc=38.2E-4   ! X-sect. area of cols.      h    ║Y       ║         ║
ayc=23.9E-4   ! Shear area of columns.     │    ║        ║         ║
izc=558E-8    ! Mom. of inert. of cols. ───┼─   ╠════════╬═══   ═══╣
dpc=111.2E-6  ! Depth of columns.          │    ║nb+2    ║nb+3 etc.║
e=206E6       ! Young's modulus.           h    ║ └joint numbers.  ║
nu=0.3        ! Poisson's ratio.           │    ║        ║         ║
fix=0         ! Bases, -1=fixed, 0=pin. ───┴─ 1╨ ───X   2╨     nb+1╨
nsg=32        ! Number of segments.             ├────s─────┼►►nb bays.
hjl1=VEC(0,0,4,0,4,0)  ! Hor. joint loads, l. to r., bot. to top.
vjl1=VEC(0,0,-10,-10,-10,-10) ! Vir. joint loads, l to r, b to t.
udl=-10   nc=2  ! Udl on all beams; No. of conc. loads on members.
! Members are numbered left to right ground floor columns, left to
! right first floor beams, left to right 1st floor cols and so on.
IF nc>0                    !          ▲cn(n)
nc(1)=VEC(3,6)             !  ├─cs(n)─┤         ┌n'th load occurs on
cs(1)=VEC(1.91)*2          !  ╪═══════╧═════════╪ on member nc(n).
cn(1)=VEC(-40)*2           !  start-joint        end-joint.
ENDIF                      !  lower or left      higher or right.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj=(nb+1)*(ns+1) nj
NUMBER OF MEMBERS nm=ns*(2*nb+1) nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3 ;NUMBER OF SEGMENTS nsg
JOINT COORDINATES ;n=0 j=-nb ;:40 ;n=n+1 j=j+nb+1
j THRU j+nb X 0 Y h*(n-1) XL nb*s ;IF n<ns+1 GOTO 40
JOINT RELEASES ;1 THRU nb+1 FORCE X -1 FORCE Y -1 MOMENT Z fix
MEMBER INCIDENCES ;n=0 j=-nb m=-2*nb ;:50 ;n=n+1 j=j+nb+1
m=m+2*nb+1 m+nb+1 THRU m+2*nb RANGE j+nb+1 j+nb+2 j+2*nb j+2*nb+1
m THRU m+nb RANGE j j+nb+1 j+nb j+2*nb+1 ;IF n<ns GOTO 50
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;n=0 m=-2*nb ;:70 ;n=n+1  m=m+2*nb+1
m+nb+1 THRU m+2*nb AX axb AY ayb IZ izb CY dpb/2
m THRU m+nb AX axc AY ayc IZ izc CY dpc/2 ;IF n<ns GOTO 70
LOADING CASE 1 ;MEMBER LOADS ;n=0 m=-2*nb ;:48 ;n=n+1  m=m+2*nb+1
m+nb+1 THRU m+2*nb FORCE Y UNIFORM W udl
IF n<ns GOTO 48 ;i=0 ;:52 ;i=i+1 ;IF i>nc GOTO 53
nc(i) FORCE Y CONCENTRATED P cn(i) L cs(i) ;GOTO 52 ;:53
JOINT LOADS ;i=0 ;:56 ;i=i+1 ;IF hjl(i)<>0 THEN i FORCE X hjl(i)
IF vjl(i)<>0 THEN i FORCE Y vjl(i) ;IF i<nj GOTO 56
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19
SOLVE ;status=1E-36 !Tells vmecp point loads on members
nb'=nb+1 n=0 m=-2*nb ;:67 ;n=n+1 m=m+2*nb+1 m'=m+nb+1
s(m')=VEC(s)*nb ct(m')=VEC(0)*nb va(m')=VEC(0)*nb
vc(m')=VEC(udl)*nb s(m)=VEC(h)*nb' ct(m)=VEC(0)*nb'
va(m)=VEC(0)*nb vc(m)=VEC(0)*nb ;IF n<ns GOTO 67 ;ch9=1 ch10=0
i=0 ;:68 ;i=i+1 m=nc(i) ;IF i<nc GOTO 68 ;i=0 ;:78 ;i=i+1
IF i<nm GOTO 78  ;#vmecp.ndf !Equilib., compat. & energy checks.
fnm=$(vm262.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE COUPLED OR PIERCED SHEAR WALLS SUBJECTED TO HORIZONTAL
TITLE WIND LOAD.  RESULTS COMPARE THE MATRIX STIFFNESS
TITLE METHOD WITH 'ANALYSIS AND DESIGN OF PIERCED SHEAR-WALLS'
TITLE by D MAGNUS, CONCRETE PUBLICATIONS, LONDON 1968.
METHOD ELASTIC ;MADEBY DWB ;DATE 08.03.05 ;REFNO VM270
TYPE PLANE FRAME run=0 ;PRINT DATA, RESULTS FROM 1 ;*/11
*                    Joint numbering              Member numbering
*  ->    ___   |        |_______|       |        |----|   ____  |----|
*  ->     ^    |    7  8 _______ 9   10 |        |--8-|    9    |-10-|
*  ->    ho    |  Wall A |  d   | Wall B|       6|                     |7
* p->    _v_   |         |______|       |        |----|   ____  |----|
*  ->     ^    |    3  4 ___|___ 5    6 |        |--3-|    4    |--5-|
*  ->    hg    |        |        |      |       1|                     |2
*  ->    _|_  _|______1_|_      _|____2_|_     __|__                 __|__
*          ////////////    ////////////        /////                 /////
*                  la        l       lb          | wall section     |
*              |---------|-------|--------|      |<-centre lines--->|
n=12         p=80          ! No. of storeys & wind load/unit height.
hg=3         ho=3          ! Ground fl. storey ht, & all other hts.
la=5.4  lb=4.4  l=2.2      ! Walls A & B widths, lintel length.
va=-3180     vb=-2580      ! Vertical loads each floor walls A & B.
t=0.2        d=1           ! Wall thickness and depth of lintel.
e=28E6       nu=1E-12      ! Young's modulus & Poisson's ratio.
#cc924.stk !Import verification data from cc924.stk if available.
la=11 lb=3 l=5
NUMBER OF JOINTS nj=n*4+2 nj ;NUMBER OF MEMBERS nm=n*5 nm
NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;1 THRU 2 X 0 Y 0 XL la/2+l+lb/2 SUPPORT ;i=1
j=-1 ;:5 ;i=i+1 j=j+4 ;IF i=2 THEN lev=hg ;IF i>2 THEN lev=lev+ho
j 0 lev ;j+1 la/2 lev ;j+2 la/2+l lev ;j+3 la/2+l+lb/2 lev
IF i<n+1 GOTO 5 ;MEMBER INCIDENCES ;s=5 ;IF n<2 THEN s=1
1 THRU 2 RANGE 1,3 2,6 ;3 THRU nm-2 STEP s RANGE 3,4 nj-3,nj-2
4 THRU nm-1 STEP s RANGE 4,5 nj-2,nj-1
5 THRU nm STEP s RANGE 5,6 nj-1,nj
IF n=2 THEN 6 THRU 7 RANGE 3,7 6,10 ;IF n>2
6 THRU nm-4 STEP 5 RANGE 3,7 nj-7,nj-3
7 THRU nm-3 STEP 5 RANGE 6,10 nj-4,nj ;ENDIF ;g=e/(2*(1+nu))
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES ;aa=la*t sa=aa*5/6
ia=t*la^3/12 ab=lb*t sb=ab*5/6 ib=t*lb^3/12 alin=d*t slin=alin*5/6
ilin=t*d^3/12 ;IF nu=1E-12 THEN sa=0 sb=0 slin=0
1 AX aa AY sa IZ ia ;2 AX ab AY sb IZ ib
IF n=2 THEN 6 THRU 7 AS 1 THRU 2
IF n>2 ;6 THRU nm-4 STEP 5 AS 1 ;7 THRU nm-3 STEP 5 AS 2 ;ENDIF
3 THRU nm-2 STEP s AX t*ho*n IZ t*(ho*n)^3/12
4 THRU nm-1 STEP s AX alin AY slin IZ ilin ;5 THRU nm STEP s AS 3
LOADING UDL ON SIDE FROM LEFT TO RIGHT ;JOINT LOADS
IF n=1 ;3 FORCE Y va ;6 FORCE Y vb ;ENDIF
IF n>1 ;3 THRU nj-3 STEP 4 FORCE Y va
6 THRU nj STEP 4 FORCE Y vb ;ENDIF ;MEMBER LOADS
1 THRU nm-4 STEP s FORCE X GLOBAL UNIFORM W p ;SOLVE
status=1 gtot=0 nur=0 h=ho*n al=l^3/(12*e*ilin) x=l+(la+lb)/2
inn=(x^2*aa*ab+(ia+ib)*(aa+ab))/(aa+ab)
a=h/(n*e*al)*(aa+ab)/(aa*ab) b=inn/(ia+ib) rab=SQR(a*b)
en=2.71828 nra=n*rab num=en^nra+1/nra den=en^nra+en^(-rab)
bn=num/den r=0 m=-1 mnls=0 rnls=0 mmag=0 rmag=0 ;:50 ;r=r+1 m=m+5
snls=ARR(13,m,5) ;IF ABS(snls)>mnls THEN mnls=ABS(snls) rnls=r
fr=-(1-bn)*en^(r*rab)-bn*en^(-r*rab)+(n-r)/n
smag=p*h^2/(n*x)*(1-1/b)*fr ;IF ABS(smag)>mmag
mmag=ABS(smag) rmag=r ;ENDIF ;IF r<n GOTO 50 ;d1=mnls d2=mmag
```

```
* Magnus method assumes all storey heights are equal.
* Floor  Lintel shear force    Floor  Lintel shear force     %age
* level  maximum NL-STRESS     level  maximum D Magnus       diff.
#vmper.ndf !Compute percentage difference & any message.
* +rnls  +mnls                 +rmag  +mmag                  $ok
fnm=$(vm270.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

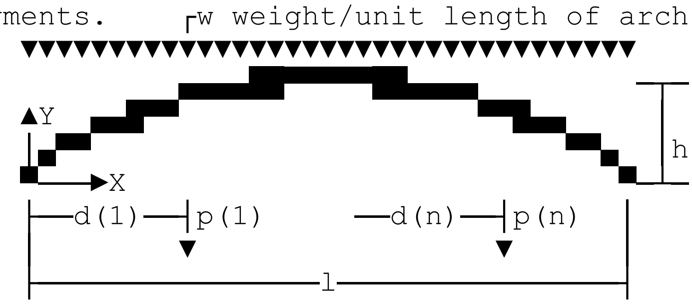
```
TITLE TWO PINNED SEGMENTAL (CIRCULAR) ARCH SUBJECTED TO
TITLE DISTRIBUTED LOADING AND VARIOUS CONCENTRATED LOADS
TITLE CF. CLASSICAL ANALYSIS BY PIPPARD & BAKER, THE
TITLE ANALYSIS OF ENGINEERING STRUCTURES. ;TYPE PLANE FRAME run=0
MADEBY DWB ;DATE 16.03.05 ;PRINT DATA RESULTS FROM 1 ;REFNO VM280
METHOD ELASTIC ;*/9
nsg=64    ! Even No. of segments.
l=20      ! Span of arch.
h=10      ! Height.
e=8E6     ! Young's mod.
nu=0.1    ! Poisson's rat.
ax=1.44   ! X-sect. area.
iz=0.17   ! Mmt Inertia.
w=-15     ! Plan loading.
n=3   ! No. of point loads.
```




```
d(1)=VEC(5,10,15)   ! Distance to point loads d(1)-d(n) l to r.
p(1)=VEC(-50)*3     ! Magnitude of point loads p(1)-p(n) l to r.
#cc924.stk !Import set of parameters if available from cc924.stk.
IF h>l/2 THEN Arch height must not exceed half the span.
NUMBER OF JOINTS nj=nsg+1 nj ;NUMBER OF MEMBERS nsg
NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;di=l^2/(4*h)+h phi=ASN(l/di) alp=2*phi/nsg
r=di/2 h'=r*COS(phi) i=1 x1=0 y1=0 ;i x1 y1 SUPPORT ;:10 ;i=i+1
c=phi-(i-1)*alp x(i)=l/2-r*SIN(c) y(i)=r*COS(c)-h' ;i x(i) y(i)
IF i<nsg GOTO 10 ;x(nj)=l y(nj)=0 ;nj x(nj) y(nj) SUPPORT
JOINT RELEASES ;1 MOMENT Z ;nj MOMENT Z ;MEMBER INCIDENCES
1 THRU nsg RANGE 1,2 nj-1,nj ;MEMBER PROPERTIES
1 THRU nsg AX ax IZ iz ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
LOADING CASE 1 UDL w ON PLAN AND n POINT LOADS ;MEMBER LOADS
IF w<>0 THEN 1 THRU nsg FORCE Y GLOBAL UNIFORM W w ;IF n<1 GOTO 30
i=0 ;:20 ;i=i+1 d=d(i) j=0 ;:25 ;j=j+1 x=x(j) y=y(j) x'=x(j+1)
y'=y(j+1) ;IF d>=x AND d<x' ;ls=SQR((x'-x)^2+(y'-y)^2)
j FORCE Y GLOBAL CONCENTRATED P p(i) L ls*(d-x)/(x'-x) ;ENDIF
IF j<nj-1 GOTO 25 ;IF i<n GOTO 20 ;:30 ;SOLVE ;status=1 gtot=0
bmc=0 nur=0 phi=PI/2-ACS(l/di) be=(di/2)^2*ax/iz ha=0 va=0 vb=0
IF n<1 GOTO 40 ;i=0 ;:35 ;i=i+1 d=d(i) the=ASN((l/2-d)/(di/2))
c=4*COS(phi)*(COS(the)+the*SIN(the)-COS(phi)-phi*SIN(phi))
num=be*(c+COS(2*the)-COS(2*phi))-COS(2*the)+COS(2*phi)
c=4*phi-3*SIN(2*phi)+2*phi*COS(2*phi)
den=2*be*c+2*(2*phi+SIN(2*phi)) ha=ha-p(i)*num/den
va'=p(i)/2*(1+SIN(the)/SIN(phi)) va=va-va' vb=vb-p(i)+va'
IF d(i)<l/2 THEN bmc=bmc+p(i)*(l/2-d(i)) ;IF i<n GOTO 35 ;:40
i=0 ;:50 ;i=i+1 ;IF i>1 AND i<nj
lu=w*SQR((x(i+1)-x(i-1))^2+(y(i+1)-y(i-1))^2)/2 ;ENDIF
IF i=1 THEN lu=w*SQR((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)/2
IF i=nj THEN lu=w*SQR((x(i)-x(i-1))^2+(y(i)-y(i-1))^2)/2 ;d=x(i)
IF i=1 THEN d=(x(i+1)-x(i))/4 ;IF i=nj THEN d=l-(x(nj)-x(nj-1))/4
the=ASN((l/2-d)/(di/2))
c=4*COS(phi)*(COS(the)+the*SIN(the)-COS(phi)-phi*SIN(phi))
num=be*(c+COS(2*the)-COS(2*phi))-COS(2*the)+COS(2*phi)
c=4*phi-3*SIN(2*phi)+2*phi*COS(2*phi)
den=2*be*c+2*(2*phi+SIN(2*phi)) ha=ha-lu*num/den
va'=lu/2*(1+SIN(the)/SIN(phi)) va=va-va' vb=vb-lu+va'
IF i<(nj-1)/2+1 THEN bmc=bmc+lu*(l/2-d)
IF i=(nj-1)/2+1 THEN bmc=bmc+lu/2*(l/2-x(i-1))/4 ;IF i<nj GOTO 50
bmc=bmc+va*(l/2)-ha*h pb1=ha pb2=va pb3=0 pb4=-ha pb5=vb pb6=0
pb7=bmc nl1=ARR(14,1,1) nl2=ARR(14,2,1) nl3=ARR(14,3,1)
i=3*(nj-1)+1 nl4=ARR(14,i,1) i=i+1 nl5=ARR(14,i,1) i=i+1
nl6=ARR(14,i,1) jc=INT(nsg/2)+1 nl7=ARR(13,jc,3) nl7=-nl7 ;*/10
```

```
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
a(1)=$(Force X) a(2)=$(Force Y) a(3)=$(Moment Z)
i=0 ;:70 ;i=i+1 j=i-INT((i-1)/3)*3 ;IF i=7 THEN j=3
jn=1 ;IF i>3 THEN jn=nj ;IF i=7 THEN jn=jc ;IF i=3 OR i=6 GOTO 90
d1=nl(i) d2=pb(i)
#vmper.ndf !Compute percentage difference & any message.
* $a(j)    at joint +jn        +nl(i)        +pb(i)          $ok
:90 ;IF i<7 GOTO 70 ;fnm=$(vm280.stk) ;#vmres.ndf !Conclude.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

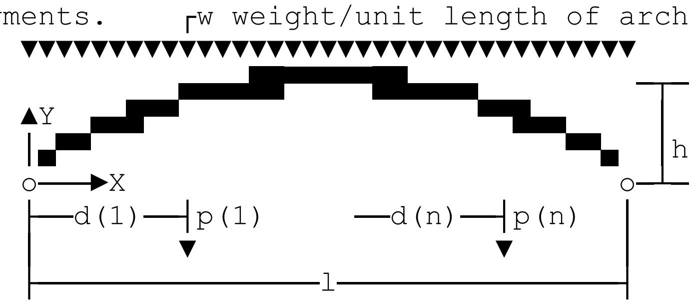
```
TITLE ENCASTRE SEGMENTAL (CIRCULAR) ARCH SUBJECTED TO
TITLE DISTRIBUTED LOADING AND VARIOUS CONCENTRATED LOADS
TITLE CF. CLASSICAL ANALYSIS BY PIPPARD & BAKER, THE
TITLE ANALYSIS OF ENGINEERING STRUCTURES. ;TYPE PLANE FRAME run=0
MADEBY DWB ;DATE 19.03.05 ;PRINT DATA RESULTS FROM 1 ;REFNO VM281
METHOD ELASTIC ;*/9
nsg=64    ! Even No. of segments.
l=20      ! Span of arch.
h=10      ! Height.
e=8E6     ! Young's mod.
nu=0.1    ! Poisson's rat.
ax=1.44   ! X-sect. area.
iz=0.17   ! Mmt Inertia.
w=-15     ! Plan loading.
n=3   ! No. of point loads.
```




```
d(1)=VEC(5,10,15)   ! Distance to point loads d(1)-d(n) l to r.
p(1)=VEC(-50)*3     ! Magnitude of point loads p(1)-p(n) l to r.
#cc924.stk !Import set of parameters if available from cc924.stk.
IF h>l/2 THEN Arch height must not exceed half the span.
NUMBER OF JOINTS nj=nsg+1 nj ;NUMBER OF MEMBERS nsg
NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;di=l^2/(4*h)+h phi=ASN(l/di) alp=2*phi/nsg
r=di/2 h'=r*COS(phi) i=1 x1=0 y1=0 ;i x1 y1 SUPPORT ;:10 ;i=i+1
c=phi-(i-1)*alp x(i)=l/2-r*SIN(c) y(i)=r*COS(c)-h' ;i x(i) y(i)
IF i<nsg GOTO 10 ;x(nj)=l y(nj)=0 ;nj x(nj) y(nj) SUPPORT
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nj-1,nj ;MEMBER PROPERTIES
1 THRU nsg AX ax IZ iz ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
LOADING CASE 1 UDL w ON PLAN AND n POINT LOADS ;MEMBER LOADS
IF w<>0 THEN 1 THRU nsg FORCE Y GLOBAL UNIFORM W w ;IF n<1 GOTO 30
i=0 ;:20 ;i=i+1 d=d(i) j=0 ;:25 ;j=j+1 x=x(j) y=y(j) x'=x(j+1)
y'=y(j+1) ;IF d>=x AND d<x' ;ls=SQR((x'-x)^2+(y'-y)^2)
j FORCE Y GLOBAL CONCENTRATED P p(i) L ls*(d-x)/(x'-x) ;ENDIF
IF j<nj-1 GOTO 25 ;IF i<n GOTO 20 ;:30 ;SOLVE ;status=1 gtot=0
nur=0 bmc=0 be=(di/2)^2*ax/iz ha=0 va=0 ma=0
r=di/2 hb=0 vb=0 mb=0 ;IF n<1 GOTO 40 ;i=0 ;:35 ;i=i+1 d=d(i)
the=ASN((l/2-d)/(di/2)) c=phi*(COS(2*phi)+COS(2*the)-2)
c=c+4*SIN(phi)*(the*SIN(the)-COS(phi)+COS(the))
num=be*c-phi*(COS(2*the)-COS(2*phi))
c=2*phi^2+phi*SIN(2*phi)-4*(SIN(phi))^2
den=2*be*c+2*phi*(2*phi+SIN(2*phi)) ho=p(i)*num/den the'=phi-the
c=ho*(phi-SIN(phi))-p(i)/2*(COS(the)-COS(phi)-the'*SIN(the))
mo=r/phi*c c=2*the'-SIN(2*phi)-SIN(2*the)+4*SIN(the)*COS(phi)
num=be*c+2*the'-SIN(2*phi)+SIN(2*the)
den=2*(1+be)*(2*phi-SIN(2*phi)) vo=p(i)*num/den
c=mo-ho*r*(1-COS(phi))-vo*r*SIN(phi)+p(i)*r*(SIN(phi)-SIN(the))
ma=ma-c mb=mb+mo-ho*r*(1-COS(phi))+vo*r*SIN(phi) ha=ha-ho hb=hb+ho
va=va+vo-p(i) vb=vb-vo ;IF d(i)<l/2 THEN bmc=bmc+p(i)*(l/2-d(i))
IF i<n GOTO 35 ;:40 ;IF w=0 GOTO 60 ;i=0 ;:50 ;i=i+1
IF i>1 AND i<nj ;lu=w*SQR((x(i+1)-x(i-1))^2+(y(i+1)-y(i-1))^2)/2
ENDIF ;IF i=1 THEN lu=w*SQR((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)/2
IF i=nj THEN lu=w*SQR((x(i)-x(i-1))^2+(y(i)-y(i-1))^2)/2 ;d=x(i)
IF i=1 THEN d=(x(i+1)-x(i))/4 ;IF i=nj THEN d=l-(x(nj)-x(nj-1))/4
the=ASN((l/2-d)/(di/2)) c=phi*(COS(2*phi)+COS(2*the)-2)
c=c+4*SIN(phi)*(the*SIN(the)-COS(phi)+COS(the))
num=be*c-phi*(COS(2*the)-COS(2*phi))
c=2*phi^2+phi*SIN(2*phi)-4*(SIN(phi))^2
den=2*be*c+2*phi*(2*phi+SIN(2*phi)) ho=lu*num/den the'=phi-the
c=ho*(phi-SIN(phi))-lu/2*(COS(the)-COS(phi)-the'*SIN(the))
mo=r/phi*c c=2*the'-SIN(2*phi)-SIN(2*the)+4*SIN(the)*COS(phi)
num=be*c+2*the'-SIN(2*phi)+SIN(2*the)
den=2*(1+be)*(2*phi-SIN(2*phi)) vo=lu*num/den
c=mo-ho*r*(1-COS(phi))-vo*r*SIN(phi)+lu*r*(SIN(phi)-SIN(the))
```

```
ma=ma-c mb=mb+mo-ho*r*(1-COS(phi))+vo*r*SIN(phi) ha=ha-ho hb=hb+ho
va=va+vo-lu vb=vb-vo ;IF i<(nj-1)/2+1 THEN bmc=bmc+lu*(l/2-d)
IF i=(nj-1)/2+1 THEN bmc=bmc+lu/2*(l/2-x(i-1))/4 ;IF i<nj GOTO 50
:60 ;bmc=bmc+va*(l/2)-ha*h-ma pb1=ha pb2=va pb3=ma pb4=hb pb5=vb
pb6=mb pb7=bmc nl1=ARR(14,1,1) nl2=ARR(14,2,1) nl3=ARR(14,3,1)
i=3*(nj-1)+1 nl4=ARR(14,i,1) i=i+1 nl5=ARR(14,i,1) i=i+1
nl6=ARR(14,i,1) jc=INT(nsg/2)+1 nl7=ARR(13,jc,3) nl7=-nl7 ;*/10
* Location                    NL-STRESS     Classical      %age
*                             analysis      analysis       diff.
a(1)=$(Force X) a(2)=$(Force Y) a(3)=$(Moment Z)
i=0 ;:70 ;i=i+1 j=i-INT((i-1)/3)*3 ;IF i=7 THEN j=3
jn=1 ;IF i>3 THEN jn=nj ;IF i=7 THEN jn=jc
d1=nl(i) d2=pb(i)
#vmper.ndf !Compute percentage difference & any message.
* $a(j)    at joint +jn        +nl(i)        +pb(i)         $ok
IF i<7 GOTO 70 ;fnm=$(vm281.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

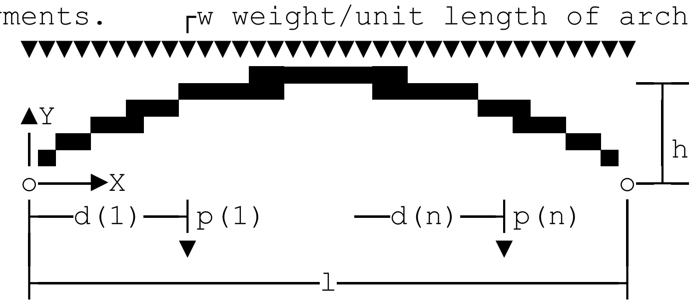
```
TITLE TWO PINNED PARABOLIC ARCH SUBJECTED TO DISTRIBUTED
TITLE LOADING AND VARIOUS CONCENTRATED LOADS
TITLE CF. CLASSICAL ANALYSIS BY PIPPARD & BAKER, THE
TITLE ANALYSIS OF ENGINEERING STRUCTURES. ;TYPE PLANE FRAME run=0
MADEBY DWB ;DATE 16.03.05 ;PRINT DATA RESULTS FROM 1 ;REFNO VM282
METHOD ELASTIC ;*/9
nsg=64    ! Even No. of segments.
l=20      ! Span of arch.
h=10      ! Height.
e=8E6     ! Young's mod.
nu=0.1    ! Poisson's rat.
ax=1.44   ! X-sect. area.
iz=0.17   ! Mmt Inertia.
w=-15     ! Plan loading.
n=3   ! No. of point loads.
```




```
d(1)=VEC(5,10,15)   ! Distance to point loads d(1)-d(n) l to r.
p(1)=VEC(-50)*3     ! Magnitude of point loads p(1)-p(n) l to r.
#cc924.stk !Import set of parameters if available from cc924.stk.
IF nsg/2<>INT(nsg/2) THEN Number of segments should be even.
IF h>l/2 THEN Arch height must not exceed half the span.
NUMBER OF JOINTS nj=nsg+1 nj ;NUMBER OF MEMBERS nsg
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;inc=l/nsg x=-l/2-inc i=0 ;:10 ;i=i+1 x=x+inc
y(i)=h-4*h*x^2/l^2 x(i)=x+l/2 i x(i) y(i) ;IF i<nj GOTO 10
JOINT RELEASES ;1 FORCE X -1 Y -1 ;nj FORCE X -1 Y -1
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nj-1,nj
MEMBER PROPERTIES ;i=0 ;:15 ;i=i+1 dx=x(i+1)-x(i) dy=y(i+1)-y(i)
ds=SQR(dx^2+dy^2) ;i AX ax IZ iz !*ds/dx ;IF i<nsg GOTO 15
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
LOADING CASE 1 UDL w ON PLAN AND n POINT LOADS ;MEMBER LOADS
IF w<>0 THEN 1 THRU nsg FORCE Y GLOBAL UNIFORM W w ;IF n<1 GOTO 30
i=0 ;:20 ;i=i+1 d=d(i) j=0 ;:25 ;j=j+1 x=x(j) y=y(j) x'=x(j+1)
y'=y(j+1) ;IF d>=x AND d<x' ;ls=SQR((x'-x)^2+(y'-y)^2)
j FORCE Y GLOBAL CONCENTRATED P p(i) L ls*(d-x)/(x'-x) ;ENDIF
IF j<nj-1 GOTO 25 ;IF i<n GOTO 20 ;:30 ;SOLVE ;status=1 gtot=0
nur=0 bmc=0 ha=0 va=0 vb=0 ;IF n<1 GOTO 40 ;i=0 ;:35 ;i=i+1 d=d(i)
bl=l/2 l'=bl-d x=bl-d b=l-d f=h
! Grassie +k=.25 +hag=.625*p(i)*20/10*(k-2*k^3+k^4)
! Morley +ham=0.625*p(i)*l/h*.25*.75*(1.25-.25^2)
! Roark 4th +har=0.625*p(i)*5*b/(8*f)*(1-2*(b/l)^2+(b/l)^3)
ha=ha-p(i)*5/(64*h*bl^3)*(5*bl^2-l'^2)*(bl^2-l'^2)
va=va-p(i)*(bl+l')/l vb=vb-p(i)*(bl-l')/l
IF d(i)<l/2 THEN bmc=bmc+p(i)*(l/2-d(i)) ;IF i<n GOTO 35 ;:40
IF w=0 GOTO 55 ;i=0 ;:50 ;i=i+1 ;IF i>1 AND i<nj
lu=w*SQR((x(i+1)-x(i-1))^2+(y(i+1)-y(i-1))^2)/2 ;ENDIF
IF i=1 THEN lu=w*SQR((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)/2
IF i=nj THEN lu=w*SQR((x(i)-x(i-1))^2+(y(i)-y(i-1))^2)/2 ;d=x(i)
IF i=1 THEN d=(x(i+1)-x(i))/4 ;IF i=nj THEN d=l-(x(nj)-x(nj-1))/4
bl=l/2 l'=bl-d ha=ha-lu*5/(64*h*bl^3)*(5*bl^2-l'^2)*(bl^2-l'^2)
va=va-lu*(bl+l')/l vb=vb-lu*(bl-l')/l
IF i<nsg/2+1 THEN bmc=bmc+lu*(l/2-d)
IF i=nsg/2+1 THEN bmc=bmc+lu/2*(l/2-x(i-1))/4 ;IF i<nj GOTO 50
:55 ;bmc=bmc+va*(l/2)-ha*h pb1=ha pb2=va pb3=0 pb4=-ha pb5=vb
pb6=0 pb7=bmc nl1=ARR(14,1,1) nl2=ARR(14,2,1) nl3=ARR(14,3,1)
i=3*(nj-1)+1 nl4=ARR(14,i,1) i=i+1 nl5=ARR(14,i,1) i=i+1
nl6=ARR(14,i,1) jc=INT(nsg/2)+1 nl7=ARR(13,jc,3) nl7=-nl7 ;*/10
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
a(1)=$(Force X) a(2)=$(Force Y) a(3)=$(Moment Z)
i=0 ;:70 ;i=i+1 j=i-INT((i-1)/3)*3 ;IF i=7 THEN j=3
jn=1 ;IF i>3 THEN jn=nj ;IF i=7 THEN jn=jc ;IF i=3 OR i=6 GOTO 90
d1=nl(i) d2=pb(i)
```

```
#vmper.ndf !Compute percentage difference & any message.
* $a(j)    at joint +jn        +nl(i)        +pb(i)          $ok
:90 ;IF i<7 GOTO 70 ;fnm=$(vm282.stk) ;#vmres.ndf !Conclude.
mjn=1 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

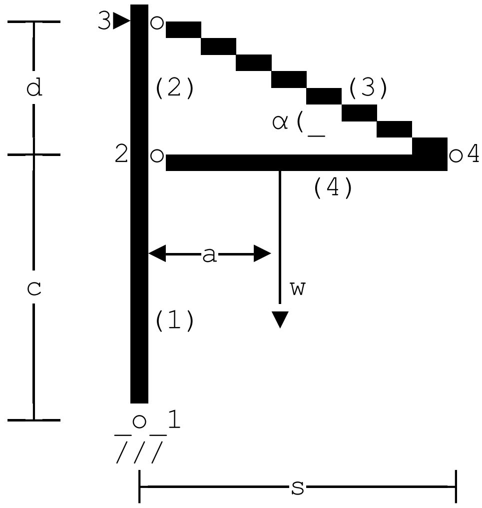
```
TITLE ENCASTRE PARABOLIC ARCH SUBJECTED TO DISTRIBUTED
TITLE LOADING AND VARIOUS CONCENTRATED LOADS
TITLE CF. CLASSICAL ANALYSIS BY PIPPARD & BAKER, THE
TITLE ANALYSIS OF ENGINEERING STRUCTURES. ;TYPE PLANE FRAME run=0
MADEBY DWB ;DATE 21.03.05 ;PRINT DATA RESULTS FROM 1 ;REFNO VM283
METHOD ELASTIC ;*/9
nsg=64     ! Even No. of segments.
l=20       ! Span of arch.
h=10       ! Height.
e=8E6      ! Young's mod.
nu=0.1     ! Poisson's rat.
ax=1.44    ! X-sect. area.
iz=0.17    ! Mmt Inertia.
w=-15      ! Plan loading.
n=3   ! No. of point loads.
```




```
d(1)=VEC(5,10,15)    ! Distance to point loads d(1)-d(n) l to r.
p(1)=VEC(-50)*3      ! Magnitude of point loads p(1)-p(n) l to r.
#cc924.stk !Import set of parameters if available from cc924.stk.
IF nsg/2<>INT(nsg/2) THEN Number of segments should be even.
IF h>l/2 THEN Arch height must not exceed half the span.
NUMBER OF JOINTS nj=nsg+1 nj ;NUMBER OF MEMBERS nsg
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;inc=l/nsg x=-l/2-inc i=0 ;:10 ;i=i+1 x=x+inc
y(i)=h-4*h*x^2/l^2 x(i)=x+l/2 i x(i) y(i) ;IF i<nj GOTO 10
JOINT RELEASES ;1 THRU nj STEP nj-1 FORCE X -1 Y -1 MOMENT Z -1
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nj-1,nj
MEMBER PROPERTIES ;i=0 ;:15 ;i=i+1 dx=x(i+1)-x(i) dy=y(i+1)-y(i)
ds=SQR(dx^2+dy^2) ;i AX ax IZ iz !*ds/dx ;IF i<nsg GOTO 15
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
LOADING CASE 1 UDL w ON PLAN AND n POINT LOADS ;MEMBER LOADS
IF w<>0 THEN 1 THRU nsg FORCE Y GLOBAL UNIFORM W w ;IF n<1 GOTO 30
i=0 ;:20 ;i=i+1 d=d(i) j=0 ;:25 ;j=j+1 x=x(j) y=y(j) x'=x(j+1)
y'=y(j+1) ;IF d>=x AND d<x' ;ls=SQR((x'-x)^2+(y'-y)^2)
j FORCE Y GLOBAL CONCENTRATED P p(i) L ls*(d-x)/(x'-x) ;ENDIF
IF j<nj-1 GOTO 25 ;IF i<n GOTO 20 ;:30 ;SOLVE ;status=1 gtot=0
nur=0 ha=0 ma=0 mb=0 va=0 vb=0 bmc=0 ;IF n<1 GOTO 40 ;i=0 ;:35
i=i+1 d=d(i) ha=ha-15*p(i)*d^2*(l-d)^2/(4*l^3*h) hb=-ha
mb=mb-p(i)*d^2*(3*l-5*d)*(l-d)/(2*l^3)
d'=l-d ma=ma+p(i)*d'^2*(3*l-5*d')*(l-d')/(2*l^3)
v=-p(i)*d^2*(3*l-2*d)/l^3 va=va-p(i)-v vb=vb+v
IF d(i)<l/2 THEN bmc=bmc+p(i)*(l/2-d) ;IF i<n GOTO 35 ;:40
IF w=0 GOTO 55 ;i=0 ;:50 ;i=i+1 ;IF i>1 AND i<nj
lu=w*SQR((x(i+1)-x(i-1))^2+(y(i+1)-y(i-1))^2)/2 ;ENDIF
IF i=1 THEN lu=w*SQR((x(i+1)-x(i))^2+(y(i+1)-y(i))^2)/2
IF i=nj THEN lu=w*SQR((x(i)-x(i-1))^2+(y(i)-y(i-1))^2)/2 ;d=x(i)
IF i=1 THEN d=(x(i+1)-x(i))/4 ;IF i=nj THEN d=l-(x(nj)-x(nj-1))/4
ha=ha-15*lu*d^2*(l-d)^2/(4*l^3*h) hb=-ha
mb=mb-lu*d^2*(3*l-5*d)*(l-d)/(2*l^3)
d'=l-d ma=ma+lu*d'^2*(3*l-5*d')*(l-d')/(2*l^3)
v=-lu*d^2*(3*l-2*d)/l^3 va=va-lu-v vb=vb+v
IF i<nsg/2+1 THEN bmc=bmc+lu*(l/2-d)
IF i=nsg/2+1 THEN bmc=bmc+lu/2*(l/2-x(i-1))/4 ;IF i<nj GOTO 50
:55 ;bmc=bmc+va*(l/2)-ha*h-ma pb1=ha pb2=va pb3=ma pb4=hb pb5=vb
pb6=mb pb7=bmc nl1=ARR(14,1,1) nl2=ARR(14,2,1) nl3=ARR(14,3,1)
i=3*(nj-1)+1 nl4=ARR(14,i,1) i=i+1 nl5=ARR(14,i,1) i=i+1
nl6=ARR(14,i,1) jc=INT(nsg/2)+1 nl7=ARR(13,jc,3) nl7=-nl7 ;*/10
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
a(1)=$(Force X) a(2)=$(Force Y) a(3)=$(Moment Z)
i=0 ;:70 ;i=i+1 j=i-INT((i-1)/3)*3 ;IF i=7 THEN j=3
jn=1 ;IF i>3 THEN jn=nj ;IF i=7 THEN jn=jc
d1=nl(i) d2=pb(i)
```

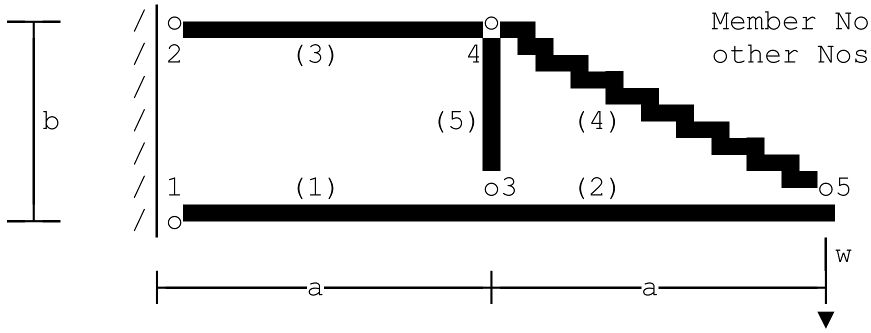
```
#vmper.ndf !Compute percentage difference & any message.
* $a(j)    at joint +jn        +nl(i)        +pb(i)          $ok
IF i<7 GOTO 70 ;fnm=$(vm283.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE OUTRIGGED FRAME CARRYING POINT LOAD ON MEMBER (4)
TITLE WITH PINNED SUPPORT AT JOINT 1 AND HORIZONTAL PROP
TITLE AT JOINT 3, INCLUDING A CHECK OF RESULTS
TITLE USING CASTIGLIANO'S THEOREM. ;MADEBY DWB ;DATE 15.08.05
METHOD ELASTIC ;PRINT DATA RESULTS FROM 1 ;REFNO VM290
TYPE PLANE FRAME run=0 ;NUMBER OF JOINTS 4 ;NUMBER OF MEMBERS 4
NUMBER OF SUPPORTS 2   ;NUMBER OF LOADINGS 1 ;*/15
s=8          ! Span of beam.
c=12         ! Height to beam.
d=6          ! Height beam to top.
a=4          ! Distance to load.
w=-4         ! Magnitude of load.
ax1=.069444  ! Area of column.
ay1=0        ! Shear area of column.
iz1=.004340  ! Inertia of column.
ax3=.027777  ! Area of tie.
iz3=1E-12    ! Inertia of tie.
ax4=.034722  ! Area of beam.
ay4=0        ! Shear area of beam.
iz4=.002411  ! Inertia of beam.
e=1728000    ! Young's modulus.
nu=0.3       ! Poisson's ratio.
```




```
#cc924.stk !Import set of parameters if available from cc924.stk.
JOINT COORDINATES ;1 0 0 SUPPORT ;2 0 c ;3 0 c+d SUPPORT ;4 s c
JOINT RELEASES ;1 MOMENT Z ;3 FORCE Y MOMENT Z ;MEMBER INCIDENCES
1 THRU 3 CHAIN 1,2,3,4 ;4 2 4 ;MEMBER RELEASES
3 THRU 4 START MOMENT Z ;4 END MOMENT Z ;g=e/(2*(1+nu))
CONSTANTS E e ALL G g ALL ;MEMBER PROPERTIES ;3 AX ax3 IZ iz3
1 THRU 2 AX ax1 AY ay1 IZ iz1 ;4 AX ax4 AY ay4 IZ iz4
LOADING CASE 1 ;MEMBER LOADS ;4 FORCE Y GLOBAL CONCENTR P w L a
SOLVE ;status=1 gtot=0 nur=0 l=SQR(s^2+d^2)
da=(w*c/ax1+w*a*d/(s*ax1)+w*a*l^3/(s*d^2*ax3)+w*a*s^2/(d^2*ax4))/e
db=w*a*s*c^3/(3*(c+d)^2*e*iz1)+w*a*c^2*s*d^3/(3*(d*(c+d))^2*e*iz1)
dy4=da+db
nn=ARR(8,4,2) nr=3*(nn-1)+2 ndy4=ARR(6,nr,1) d1=ndy4 d2=dy4
#vmper.ndf !Compute percentage difference & any message.
*
*                                NL-STRESS    Castigliano's  %age
*                                             first theorem  diff.
* Deflection Y at joint 4        +ndy4        +dy4           $ok
fnm=$(vm290.stk) ;#vmres.ndf !Conclude results.
mjn=4 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE OUTRIGGED FRAME CARRYING POINT LOAD AT JOINT 5, WITH
TITLE PINNED SUPPORTS AT JOINTS 1 & 2, BRACED BY PINNED
TITLE CONNECTED MEMBERS 3-5, INCLUDING A CHECK OF RESULTS
TITLE USING CASTIGLIANO'S THEOREM. ;MADEBY DWB ;DATE 16.08.05
METHOD ELASTIC ;PRINT DATA RESULTS FROM 1 ;REFNO VM291
TYPE PLANE FRAME run=0 ;NUMBER OF JOINTS 5 ;NUMBER OF MEMBERS 5
NUMBER OF SUPPORTS 2   ;NUMBER OF LOADINGS 1 ;*/15
*
*                                                   Member Nos in brackets,
*                                                   other Nos are joints.
*
*
*
*
*
*
*
*
```




```
a=2     b=2       w=-50         ! Frame dimensions & end load.
ax1=68.8E-4 ay1=0 iz1=11700E-8  ! Beam section properties.
ax3=38.2E-4 ay3=0 iz3=2900E-8   ! Section props for members 3-5.
e=205E6     nu=0.3              ! Young's mod. & Poisson's ratio.
#cc924.stk !Import set of parameters if available from cc924.stk.
JOINT COORDS ;1 THRU 2 X 0 Y 0 YL b SUPPORT ;3 a 0 ;4 a b ;5 2*a 0
JOINT RELEASES ;1 THRU 2 MOMENT Z ;MEMBER INCIDENCES ;1 1 3 ;2 3 5
3 2 4 ;4 4 5 ;5 3 4 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER RELEASES ;4 THRU 5 START MOMENT Z END MOMENT Z
MEMBER PROPERTIES ;1 THRU 2 AX ax1 AY ay1 IZ iz1
3 THRU 5 AX ax3 AY ay3 IZ iz3
LOADING CASE 1 ;JOINT LOADS ;5 FORCE Y w ;SOLVE
dy5=2*w*a^3/(3*e*iz1)+8*a^3*w/(b^2*e*ax1)+4*w*a^3/(b^2*e*ax3)
dy5=dy5+4*w*(a^2+b^2)*SQR(a^2+b^2)/(b^2*e*ax3)+4*w*b/(e*ax3)
dx5=4*w*a^2/(b*e*ax1) status=1 gtot=0 nur=0
nn=ARR(8,5,2) nr=3*(nn-1)+2 ndy5=ARR(6,nr,1) d1=ndy5 d2=dy5
#vmper.ndf !Compute percentage difference & any message.
*
*                               NL-STRESS    Castigliano's  %age
*                                            first theorem  diff.
* Deflection Y at joint 5       +ndy5        +dy5           $ok
nr=nr-1 ndx5=ARR(6,nr,1) d1=ndx5 d2=dx5
#vmper.ndf !Compute percentage difference & any message.
* Deflection X at joint 5       +ndx5        +dx5           $ok
fnm=$(vm291.stk) ;#vmres.ndf !Conclude results.
mjn=5 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CANTILEVER/PROPPED GRILLAGE BEAM SUBJECTED TO
TITLE UNIFORMLY DISTRIBUTED LOADING & END VERTICAL LOAD,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 11.06.05 ;TYPE PLANE GRID run=0 ;REFNO VM300
PRINT DATA, RESULTS FROM 1 ;NUMBER OF JOINTS 3
NUMBER OF MEMBERS 2 ;NUMBER OF SUPPORTS 1 ;NUMBER OF LOADINGS 3
*/8
sp=3.0          ! Span of cantilever.  Span is divided into 2*nsg
nsg=16          ! No. of segments.     segments of length sp/2/nsg.
dz=0.36         ! Depth of beam.              Y +ve          |v
dy=0.3          ! Breadth of beam.             |      ┌w      ▼
e=28E6/3        ! Young's modulus.           / |▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
nu=0.2          ! Poisson's ratio.           /═╪════════════════ ──X
w=-36           ! Load/unit length.          / |1      2       3
t=3             ! Torque per unit length.      ├───────sp──────┤
v=-8            ! End vertical load.
prop=0          ! Prop at end of cantilever, 1=yes, 0=no.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 0 0 SUPPORT ;2 sp/2 0
3 sp 0 ;IF prop=1 ;JOINT RELEASES ;3 FORCE Z -1 ;ENDIF
MEMBER INCIDENCES ;1 1 2 ;2 2 3
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU 2 RECTANGLE DY dy DZ dz az=dy*dz*5/6 iy=dy*dz^3/12
LOADING CASE 1 ;MEMBER LOADS
IF w<>0 THEN 1 THRU 2 FORCE Z UNIFORM W w
IF t<>0 THEN 1 THRU 2 MOMENT X UNIFORM W t
JOINT LOADS ;IF v<>0 THEN 3 FORCE Z v
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;nj=3 jn=0 ;:18 ;jn=jn+1
jn MOMENT X jn Y jn FORCE Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn MOMENT X jn' Y jn' FORCE Z jn' ;IF jn<nj GOTO 19 ;SOLVE
va1=VEC(t)*2 vc1=VEC(w)*2 vl1=VEC(0,0,v) ch9=1 ch10=0
#vmecg.ndf !Equilibrium/energy & compatibility checks for grid.
fnm=$(vm300.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

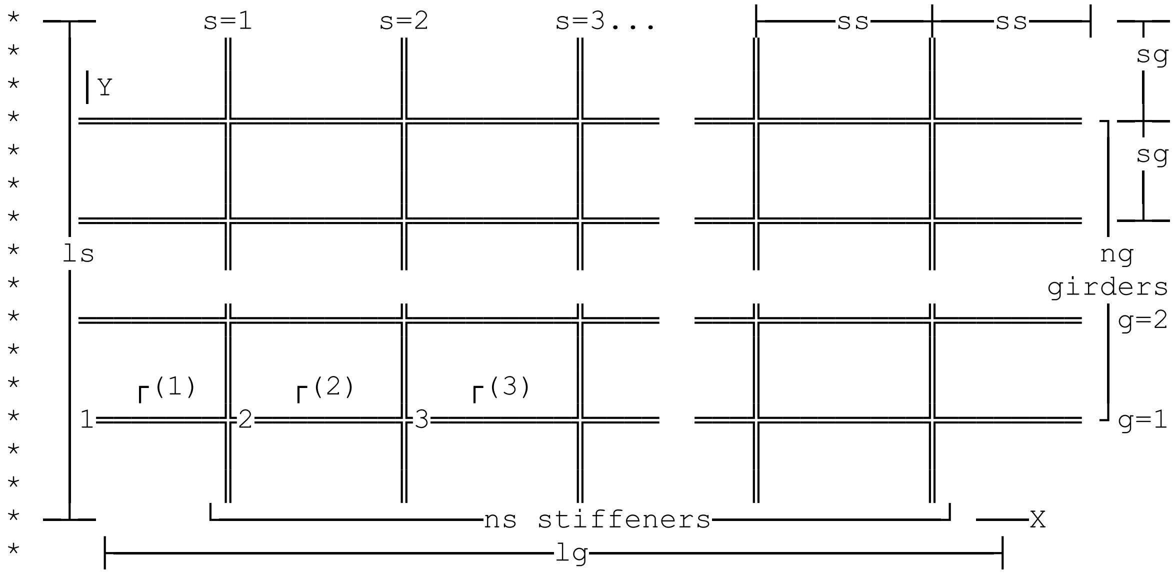
```
TITLE CIRCULAR-ARC CANTILEVER SUBJECTED TO CONCENTRATED
TITLE & DISTRIBUTED LOADS; CHECKING OF RESULTS AGAINST
TITLE CLASSICAL THEORY OF PIPPARD & BAKER, THE ANALYSIS
TITLE OF ENGINEERING STRUCTURES, THIRD EDITION, 1957.
MADEBY DWB ;DATE 21.03.05 ;METHOD ELASTIC ;REFNO VM301
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE GRID run=0 ;*/16
r=3.6        ! Radius.  /A              AC represents a cantilever
e=205E6      ! Young's  /|1       w(n)  of radius r built-in at A
nu=0.3       ! Poisson's.|      /       and subtending an angle
iy=9141E-8   ! Inertia.  r    /         AOC =phid° at the centre
ix=14090E-8  ! Tors.con.|  /            O.  At an angular distance
az=96.5E-4   ! Shr area. O/┐td(n)       td from OC a +ve load w()
fz=-1        ! Frce Z out.\┘            acts upwards i.e. out of
mx=1         ! Mmt about X. \           media.  The free end of
my=1         ! Mmt about Y.   X         the girder, C, carries a
nsg=64       ! No. of segments. \C      bending moment 'mx', a
udl=-0.75    ! Wt/unit length.  / fz    twisting moment 'my' & an
phid=120     ! Angle AOC°.   /my    \mx upward, out of force 'fz'.
nw=3         ! No. of concentrated loads, into media is negative.
w(1)=VEC(-1)*3          ! Loads w(1)-w(nw) anticlockwise from OC.
td(1)=VEC(40,60,80)     ! Angles° td(1)-td(nw) anticlock. from OC.
#cc924.stk !Import verification data from cc924.stk if available.
phi=RAD(phid) nj=nsg+1 i=0 ;:5 ;i=i+1 t(i)=RAD(td(i))
IF i<nw GOTO 5 ;NUMBER OF JOINTS nj ;NUMBER OF MEMBERS nsg
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1 ;JOINT COORDINATES
i=0 ga=phi/nsg alp=phi+ga ;:10 ;i=i+1 alp=alp-ga
i r*COS(alp)-r r*SIN(alp) ;IF i<nj GOTO 10 ;lseg=2*r*SIN(ga/2)
JOINT RELEASES ;1 MOMENT X -1 MOMENT Y -1 FORCE Z -1
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nsg,nj ;g=e/(2*(1+nu))
CONSTANTS E e ALL G g ALL ;az'=az ;IF nu=1E-12 THEN az'=az*1E6
MEMBER PROPERTIES ;1 THRU nsg AZ az' IX ix IY iy
LOADING DYNAMIC G 9.80665
JOINT LOADS ;nj FORCE Z fz MOMENT X mx MOMENT Y my
MEMBER LOADS ;IF udl<>0 THEN 1 THRU nsg FORCE Z UNIFORM W udl
IF nw<1 GOTO 25 ;i=0 ;:20 ;i=i+1 rn=(phi-t(i))/ga m=INT(rn)
l=lseg*(rn-m) ;IF m<0 THEN Load wrong. ;IF m=nsg THEN m=m-1 l=lseg
m+1 FORCE Z CONCENTRATED P w(i) L l ;IF i<nw GOTO 20 ;:25 ;SOLVE
status=1 gtot=0 nur=0 t0=0 mx'=mx my'=my fz'=fz muot=0 taot=0
dlot=0 gam=e*iy/(g*ix) c=r/(4*e*iy) i=-1 ;:30 ;i=i+1
IF udl=0 AND i>0 AND i<=nj GOTO 30 ;IF i>nj AND nw=0 GOTO 40
IF i<1 THEN w=0 t=0 ;IF i>0 THEN mx'=0 my'=0 fz'=0
IF i>0 AND i<=nj THEN w=udl*lseg t=phi-ga*(i-1)
IF i=1 OR i=nj THEN w=w/2 ;IF i>nj THEN w=w(i-nj) t=t(i-nj)
t'=phi-t ;c1=mx'*((gam+1)*2*phi-(gam-1)*SIN(2*phi))
c2=my'*(gam-1)*(1-COS(2*phi)) c4=SIN(phi)*SIN(t')-t'*SIN(t)
c3=fz'*r*((gam-1)*(1-COS(2*phi))-4*gam*(1-COS(phi)))
c5=gam*(2*COS(t)-2*COS(phi)-SIN(phi)*SIN(t')-t'*SIN(t))
muo=c*(c1-c2+c3-2*w*r*(c4+c5)) c1=-mx'*(gam-1)*(1-COS(2*phi))
c2=my'*((gam+1)*2*phi+(gam-1)*SIN(2*phi))
c3=fz'*r*((gam-1)*(2*phi-SIN(2*phi))-4*gam*(phi-SIN(phi)))
c4=t'*COS(t)-COS(phi)*SIN(t')
c5=gam*(2*SIN(phi)-2*SIN(t)-COS(phi)*SIN(t')-t'*COS(t))
tao=c*(c1+c2+c3-2*w*r*(c4-c5)) !Rotn in direction of my.
c1=mx'*(1-COS(phi))-my'*SIN(phi)-fz'*r*(phi-SIN(phi))
dlo=-r*tao-r^2/(g*ix)*(c1-w*r*(t'-SIN(t'))) !Displ. Z.
muot=muot+muo taot=taot+tao dlot=dlot+dlo ;IF i<nj+nw GOTO 30
:40 ;pb1=muot pb2=taot pb3=dlot j=3*(nj-1)+1 nl1=ARR(6,j,1)
j=j+1 nl2=ARR(6,j,1) j=j+1 nl3=ARR(6,j,1) ;*/6
* Location                     NL-STRESS     Classical       %age
*                              analysis      analysis        diff.
a(1)=$(RotationX) a(2)=$(RotationY) a(3)=$(DisplaceZ)
i=0 ;:70 ;i=i+1 d1=nl(i) d2=pb(i)
```

```
#vmper.ndf !Compute percentage difference & any message.
* $a(i)     at joint +nj       +nl(i)        +pb(i)          $ok
IF i<3 GOTO 70 ;fnm=$(vm301.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CIRCULAR-ARC BOW GIRDER SUBJECTED TO CONCENTRATED
TITLE & DISTRIBUTED LOADS; CHECKING OF RESULTS AGAINST
TITLE CLASSICAL THEORY OF PIPPARD & BAKER, THE ANALYSIS
TITLE OF ENGINEERING STRUCTURES, THIRD EDITION, 1957.
MADEBY DWB ;DATE 25.03.05 ;METHOD ELASTIC ;REFNO VM302
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE GRID run=0 ;*/16
r=6.0       ! Radius.  /A▀▀▀▄▄▄             AC represents a bow girder
e=205E6     ! Young's  /│1     ▀▀▄w(n)    of radius r built-in at A
nu=0.3      ! Poisson's.│       /  ▀▄     and C, subtending an angle
iy=9141E-8  ! Inertia.  r     /      █    AOC =phid° at the centre
ix=14090E-8  ! Tors.con.│   /         █   O.  At an angular distance
az=96.5E-4  ! Shr area. O/┐td(n)      █   td° from OC a +ve load w()
phid=120    ! Angle AOC°.\┘           █   acts upwards i.e. out of
nsg=64      ! No. segments.\         █    the paper/screen.  Bow is
udl=-0.75  ! Wt/unit length. X      █     modelled by nsg segments.
nw=3    ! No. of conc. loads.  \C▄▀
w(1)=VEC(-10)*3        ! Loads w(1)-w(nw) anticlockwise from OC.
td(1)=VEC(40,60,80)    ! Angles° td(1)-td(nw) anticlock. from OC.
#cc924.stk !Import verification data from cc924.stk if available.
IF phid>359 THEN Subtended angle AOC° too high.
phi=RAD(phid) nj=nsg+1 i=0 ;:5 ;i=i+1 t(i)=RAD(td(i))
IF i<nw GOTO 5 ;NUMBER OF JOINTS nj ;NUMBER OF MEMBERS nsg
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1 ;JOINT COORDINATES
ga=phi/nsg alp=phi+ga i=0 ;:10 ;i=i+1 alp=alp-ga
i r*COS(alp)-r r*SIN(alp) ;IF i<nj GOTO 10 ;lseg=2*r*SIN(ga/2)
JOINT RELEASES ;1,nj INCLUSIVE MOMENT X -1 MOMENT Y -1 FORCE Z -1
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nsg,nj ;g=e/(2*(1+nu))
CONSTANTS E e ALL G g ALL ;az'=az ;IF nu=1E-12 THEN az'=az*1E6
MEMBER PROPERTIES ;1 THRU nsg AZ az' IX ix IY iy
LOADING DYNAMIC 9.80665 ;MEMBER LOADS
IF udl<>0 THEN 1 THRU nsg FORCE Z UNIFORM W udl
IF nw<1 GOTO 25 ;i=0 ;:20 ;i=i+1 rn=(phi-t(i))/ga m=INT(rn)
l=lseg*(rn-m) ;IF m<0 THEN Load wrong. ;IF m=nsg THEN m=m-1 l=lseg
m+1 FORCE Z CONCENTRATED P w(i) L l ;IF i<nw GOTO 20 ;:25 ;SOLVE
status=1 gtot=0 nur=0 ph2=phi/2 i=0 ;:30 ;i=i+1 pb(i)=0
IF i<6 GOTO 30 ;fat=0 mbt=0 tbt=0 fbt=0 gam=e*iy/(g*ix) i=0 ;:35
i=i+1 ;IF udl=0 AND i<=nj GOTO 35 ;IF i>nj AND nw=0 GOTO 40
IF i<=nj THEN w=udl*lseg/2 t=ga*(i-1)-ph2
IF i=1 OR i=nj THEN w=w/2 ;IF i>nj THEN w=w(i-nj)/2 t=t(i-nj)-ph2
t'=ph2-t c1=(gam+1)*t'*SIN(t)+(gam-1)*SIN(ph2)*SIN(t')
c2=2*gam*(COS(t)-COS(ph2))
mo=2*w*r*(c1-c2)/((gam+1)*2*ph2-(gam-1)*SIN(2*ph2))
c1=2*(gam+1)*(t'*COS(t)*SIN(ph2)-ph2*SIN(t')-t*ph2)
c2=4*gam*SIN(t)*SIN(ph2)-(gam-1)*t*SIN(2*ph2)
c3=(gam+1)*2*ph2^2+(gam-1)*ph2*SIN(2*ph2)-2*gam*(1-COS(2*ph2))
fo=w*(1+(c1+c2)/c3)
to=w*r*(t'-SIN(t')-fo/w*(ph2-SIN(ph2)))/SIN(ph2)
ma=mo*COS(ph2)+to*SIN(ph2)-fo*r*SIN(ph2)+2*w*r*SIN(t')
ta=-mo*SIN(ph2)+to*COS(ph2)+fo*r*(1-COS(ph2))-2*w*r*(1-COS(t'))
fa=fo-2*w mb=mo*COS(ph2)-to*SIN(ph2)+fo*r*SIN(ph2)
tb=mo*SIN(ph2)+to*COS(ph2)+fo*r*(1-COS(ph2)) fb=fo
pb1=pb1-ta pb2=pb2+ma pb3=pb3+fa pb4=pb4-mb pb5=pb5-tb pb6=pb6-fb
IF i<nj+nw GOTO 35 ;alp=phi-PI/2 ;mx=pb1*COS(alp)-pb2*SIN(alp)
my=pb1*SIN(alp)+pb2*COS(alp) pb1=mx pb2=my ;:40 ;i=0 ;:50 ;i=i+1
j=0 ;IF i>3 THEN j=nsg-1 ;k=3*j+i nl(i)=ARR(14,k,1)
IF i<6 GOTO 50 ;*/9
```

```
* Location                      NL-STRESS     Classical      %age
*                               analysis      analysis       diff.
a(1)=$(Moment X) a(2)=$(Moment Y) a(3)=$(Force  Z)
i=0 ;:70 ;i=i+1 d1=nl(i) d2=pb(i) j=1 k=i ;IF i>3 THEN j=nj k=i-3
#vmper.ndf !Compute percentage difference & any message.
* $a(k)     at joint +j         +nl(i)        +pb(i)         $ok
IF i<6 GOTO 70 ;fnm=$(vm302.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE GRILLAGE OF GIRDERS AND STIFFENERS SUBJECTED TO
TITLE CONCENTRATED & DISTRIBUTED LOADS; CHECKING OF
TITLE RESULTS AGAINST MODERN FORMULAS FOR STATICS &
TITLE DYNAMICS BY PILKEY & CHANG, PUB. MCGRAW-HILL.
MADEBY DWB ;DATE 22.02.05 ;METHOD ELASTIC JOINTS ;REFNO VM310
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE GRID run=0 ;*/17
```




```
lg=10    ls=4        ! Length of girders & length of stiffeners.
ng=4     ns=5        ! No of girders & stiffeners.
e=205E6  nu=0.3      ! Young's modulus & Poisson's ratio.
ig=1E-3  is=ig/2     ! Moment of inertia for girders & stiffeners.
wsg=-30              ! Load at each intersection, zero to omit.
udl(1)=VEC(-10)*ns  ! UDL along 'ns' stiffeners, left to right.
#cc924.stk !Import verification data from cc924.stk if available.
! The beams parallel to the X-axis are denoted 'girders', those
! parallel to the Y-axis are denoted 'stiffeners', the stiffeners
! may be larger/smaller or more/less numerous that the girders,
! but as the udl's are always distributed to the stiffeners, make
! the stiffeners greater than or equal to the number of girders.
IF ng<2 OR ns<2 THEN Number of girders/stiffeners too low.
sg=ls/(ng+1) ss=lg/(ns+1) ny=ns+2
NUMBER OF JOINTS nj=ng*(ns+2)+2*ns nj ;nm=ng*(ns+1)+ns*(ng+1)
NUMBER OF MEMBERS nm ;NUMBER OF SUPPORT 2*(ns+ng)
NUMBER OF LOADINGS 1 ;JOINT COORDINATES
1 THRU 1+ny*(ng-1) STEP ny X 0 Y sg YL ls-sg SUPPORT
ny THRU ny*ng STEP ny X lg Y sg YL ls-sg SUPPORT
nj-2*ns+1 THRU nj-ns X ss Y 0 XL lg-ss SUPPORT
nj-ns+1 THRU nj X ss Y ls XL lg-ss SUPPORT
i=0 j=2-ny y=0 ;:10 ;i=i+1 j=j+ny y=y+sg
j THRU j+ns-1 X ss XL lg-ss Y y ;IF i<ng GOTO 10 ;JOINT RELEASES
1 THRU 1+ny*(ng-1) STEP ny MOMENT X Y
ny THRU ny*ng STEP ny MOMENT X Y ;nj-2*ns+1 THRU nj-ns MOMENT X Y
nj-ns+1 THRU nj MOMENT X Y ;MEMBER INCIDENCES
i=0 minc=2*ns+1 m=1-minc j=1-ny ;:20 ;i=i+1 m=m+minc j=j+ny
m THRU m+ns RANGE j,j+1 j+ns,j+ns+1 ;IF i<ng GOTO 20
i=0 m=ny-minc j=2-ny ;:30 ;i=i+1 m=m+minc j=j+ny
m THRU m+ns-1 RANGE j,j+ny j+ns-1,j+ny+ns-1 ;IF i<ng-1 GOTO 30
nm-2*ns+1 THRU nm-ns RANGE nj-2*ns+1,2 nj-ns,ns+1
j=(ng-1)*ny+2 nm-ns+1 THRU nm RANGE j,nj-ns+1 j+ns-1,nj
CONSTANTS E e ALL G sm=e/(2*(1+nu)) sm ALL ;MEMBER PROPERTIES
i=0 m=1-minc ;:40 ;i=i+1 m=m+minc
m THRU m+ns IY ig IX 1E-6 ;IF i<ng GOTO 40 ;i=0 m=ny-minc ;:45
i=i+1 m=m+minc ;m THRU m+ns-1 IY is IX 1E-6 ;IF i<ng-1 GOTO 45
nm-2*ns+1 THRU nm-ns IY is IX 1E-6 ;nm-ns+1 THRU nm IY is IX 1E-6
```

```
LOADING CASE 1 ;MEMBER LOADS ;i=0 ;:46 ;i=i+1
ny-1+i THRU nm-2*ns+i STEP minc FORCE Z UNIFORM W udl(i)
nm-ns+i FORCE Z UNIFORM W udl(i) ;IF i<ns GOTO 46 ;IF wsg<>0
JOINT LOADS ;g=0 ;:47 ;g=g+1 s=0 ;:48 ;s=s+1 j=s+1+ny*(g-1)
j FORCE Z wsg ;IF s<ns GOTO 48 ;IF g<ng GOTO 47 ;ENDIF
SOLVE ;status=1 gtot=0 nur=0
*/10 ;* Verification with WD Pilkey & PY Chang in 'Modern
* formulas for statics and dynamics', McGraw-Hill New York 1978.
* Position      Deflection      Deflection by     %age
* ref. No.      by NL-STRESS    Pilkey & Chang    diff.
i=0 ;:80 ;i=i+1 dj(i)=0 ;IF i<ns*ng GOTO 80
j=0 ;:83 ;j=j+1 s=0 sum=0 ;:84 ;s=s+1 ws=udl(s)
num=4*ls^4/(e*is*PI^5) den=(ng+1)/2*j^4*(ls/lg)^3*(ig/is)+(ns+1)/2
sum=sum+ws*SIN(j*PI*s/(ns+1)) ;IF s<ns GOTO 84 ;k(j)=num*sum/den
IF j<(ns*ng) GOTO 83 ;s=0 i=0 ;:86 ;s=s+1 g=0 ;:87 ;g=g+1 sum=0
i=i+1 jno=s+1+ny*(g-1) x=ss+(s-1)*ss ;j=0 ;:88 ;j=j+1
sum=sum+k(j)*SIN(PI*g/(ng+1))*SIN(j*PI*x/lg) ;IF j<(ns*ng) GOTO 88
dj(i)=dj(i)+sum ;IF g<ng GOTO 87 ;IF s<ns GOTO 86
IF wsg=0 GOTO 99 ;j=0 ;:92 ;j=j+1 s=0 sum=0 ;:93 ;s=s+1 g=0 ;:94
g=g+1 num=2*ls^3/(e*is*PI^4)
den=(ng+1)/2*j^4*(ls/lg)^3*(ig/is)+(ns+1)/2
sum=sum+wsg*SIN(PI*g/(ng+1))*SIN(j*PI*s/(ns+1))
IF g<ng GOTO 94 ;IF s<ns GOTO 93 ;k(j)=num*sum/den
IF j<(ns*ng) GOTO 92 ;s=0 i=0 ;:96 ;s=s+1 g=0 ;:97 ;g=g+1 i=i+1
jno=s+1+ny*(g-1) x=ss+(s-1)*ss ;j=0 sum=0 ;:98 ;j=j+1
sum=sum+k(j)*SIN(PI*g/(ng+1))*SIN(j*PI*x/lg)
IF j<(ns*ng) GOTO 98
dj(i)=dj(i)+sum ;IF g<ng GOTO 97 ;IF s<ns GOTO 96 ;:99
s=0 i=0 ;:100 ;s=s+1 g=0 ;:110 ;g=g+1 i=i+1 jno=s+1+ny*(g-1)
x=ss+(s-1)*ss j=3*jno dy=ARR(6,j,1) d1=dy d2=dj(i)
#vmper.ndf !Compute percentage difference & any message.
*   +jno        +dy             +dj(i)           $ok
IF g<ng GOTO 110 ;IF s<ns GOTO 100 ;fnm=$(vm310.stk) ;#vmres.ndf
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE GRILLAGE OF GIRDERS AND STIFFENERS SUBJECTED TO
TITLE POINT LOADS AT INTERSECTIONS & DISTRIBUTED LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 18.06.05 ;METHOD ELASTIC JOINTS ;REFNO VM311
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE GRID run=0 ;*/17
* ──┬──     s=1      s=2      s=3...    ├────ss────┼────ss────┤  ─┬─
*   │        ║        ║        ║         ║          ║             │
*   │ |Y     ║        ║        ║         ║          ║            sg
*   │ ═══════╬════════╬════════╬═══   ═══╬══════════╬══════════  ┼
*   │        ║        ║        ║         ║          ║          │ sg
*   │        ║        ║        ║         ║          ║          │  │
*   │ ═══════╬════════╬════════╬═══   ═══╬══════════╬══════════│ ─┴─
*  ls        ║        ║        ║         ║          ║          ng
*   │                                                       girders
*   │ ═══════╬════════╬════════╬═══   ═══╬══════════╬══════════ |g=2
*   │        ║        ║        ║         ║          ║          |
*   │    ┌(1)║    ┌(2)║    ┌(3)║         ║          ║          |
*   │ 1══════2════════3════════╬═══   ═══╬══════════╬══════════┘g=1
*   │        ║        ║        ║         ║          ║
*   │        ║        ║        ║         ║          ║
* ──┴──      └──────────────ns stiffeners───────────┘    ───X
*     ├──────────────────────────lg──────────────────────────┤
lg=10    ls=4        ! Length of girders & length of stiffeners.
ng=4     ns=5        ! No of girders & stiffeners.
e=205E6  nu=0.3      ! Young's modulus & Poisson's ratio.
ig=1E-3  is=ig/2     ! Moment of inertia for girders & stiffeners.
nsg=4    wsg=-30     ! No. of segments; load at each intersection.
udl(1)=VEC(-10)*ns   ! UDL along 'ns' stiffeners, left to right.
#cc924.stk !Import verification data from cc924.stk if available.
IF ng<2 OR ns<2 THEN Number of girders/stiffeners too low.
sg=ls/(ng+1) ss=lg/(ns+1) ny=ns+2 nj=ng*(ns+2)+2*ns
NUMBER OF JOINTS nj ;NUMBER OF MEMBERS nm=ng*(ns+1)+ns*(ng+1) nm
NUMBER OF SUPPORT 2*(ns+ng) ;NUMBER OF LOADINGS 3
NUMBER OF SEGMENTS nsg
JOINT COORDS ;1 THRU 1+ny*(ng-1) STEP ny X 0 Y sg YL ls-sg SUPPORT
ny THRU ny*ng STEP ny X lg Y sg YL ls-sg SUPPORT
nj-2*ns+1 THRU nj-ns X ss Y 0 XL lg-ss SUPPORT
nj-ns+1 THRU nj X ss Y ls XL lg-ss SUPPORT
i=0 j=2-ny y=0 ;:10 ;i=i+1 j=j+ny y=y+sg
j THRU j+ns-1 X ss XL lg-ss Y y ;IF i<ng GOTO 10
JOINT RELEASES ;1 THRU 1+ny*(ng-1) STEP ny MOMENT X Y
ny THRU ny*ng STEP ny MOMENT X Y ;nj-2*ns+1 THRU nj-ns MOMENT X Y
nj-ns+1 THRU nj MOMENT X Y ;MEMBER INCIDENCES
i=0 minc=2*ns+1 m=1-minc j=1-ny ;:20 ;i=i+1 m=m+minc j=j+ny
m THRU m+ns RANGE j,j+1 j+ns,j+ns+1 ;IF i<ng GOTO 20
i=0 m=ny-minc j=2-ny ;:30 ;i=i+1 m=m+minc j=j+ny
m THRU m+ns-1 RANGE j,j+ny j+ns-1,j+ny+ns-1 ;IF i<ng-1 GOTO 30
nm-2*ns+1 THRU nm-ns RANGE nj-2*ns+1,2 nj-ns,ns+1
j=(ng-1)*ny+2 nm-ns+1 THRU nm RANGE j,nj-ns+1 j+ns-1,nj
CONSTANTS E e ALL G sm=e/(2*(1+nu)) sm ALL ;MEMBER PROPERTIES
i=0 m=1-minc ;:40 ;i=i+1 m=m+minc m THRU m+ns IY ig IX 1E-6 AY 0
IF i<ng GOTO 40 ;i=0 m=ny-minc ;:45
i=i+1 m=m+minc ;m THRU m+ns-1 IY is IX 1E-6 ;IF i<ng-1 GOTO 45
nm-2*ns+1 THRU nm-ns IY is IX 1E-6 ;nm-ns+1 THRU nm IY is IX 1E-6
```

```
LOADING CASE 1 ;MEMBER LOADS ;i=0 ;:46 ;i=i+1
ny-1+i THRU nm-2*ns+i STEP minc FORCE Z UNIFORM W udl(i)
nm-ns+i FORCE Z UNIFORM W udl(i) ;IF i<ns GOTO 46 ;IF wsg<>0
JOINT LOADS ;g=0 ;:47 ;g=g+1 s=0 ;:48 ;s=s+1 j=s+1+ny*(g-1)
j FORCE Z wsg ;IF s<ns GOTO 48 ;IF g<ng GOTO 47 ;ENDIF
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn MOMENT X jn Y jn FORCE Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn MOMENT X jn' Y jn' FORCE Z jn' ;IF jn<nj GOTO 19 ;SOLVE
ch9=1 ch10=0 va1=VEC(0)*nm g=sm vc1=VEC(0)*nm i=0 ;:60 ;i=i+1
m=ny-1+i-minc ;:65 ;m=m+minc vc(m)=udl(i) ;IF m<nm-2*ns+i GOTO 65
m=nm-ns+i vc(m)=udl(i) ;IF i<ns GOTO 60 ;vl1=VEC(0)*nj ;IF wsg<>0
g=0 ;:67 ;g=g+1 s=0 ;:68 ;s=s+1 j=s+1+ny*(g-1)
vl(j)=wsg ;IF s<ns GOTO 68 ;IF g<ng GOTO 67 ;ENDIF
#vmecg.ndf !Equilibrium/energy & compatibility checks for grid.
fnm=$(vm311.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PLASTIC ANALYSIS OF CANTILEVER I-BEAM SUBJECTED
TITLE TO UNIFORMLY DISTRIBUTED LOADING & END POINT LOAD
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 23.09.05 ;TYPE PLANE FRAME run=0 ;REFNO VM410
PRINT DATA, RESULTS, FROM 1 ;METHOD PLASTIC ;TABULATE ALL
NUMBER OF JOINTS 2 ;NUMBER OF MEMBERS 1 ;NUMBER OF SUPPORTS 1
NUMBER OF LOADINGS 1 ;*/8
dy=0.2068    dz=0.1339      ! Depth & breadth of I beam.
tz=0.0064    ty=0.0096      ! Web & flange thickness of I beam.
sp=3.0  nsg=16 ! Span, No. of segments.      Y +ve           |v
e=205E6        ! Young's modulus.            |     ┌w        ▼
nu=0.3         ! Poisson's ratio.          / |▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
fy=265E3       ! Yield strength.           /╪════════════════  ──X
nli=100        ! No. of load increments.   / |1               2
w=-9   v=-20   ! Udl & end working load.     ├───────sp───────┤
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF INCREMENTS nli sense=2 ;NUMBER OF SEGMENTS nsg TRACE
JOINT COORDINATES ;1 0 0 SUPPORT ;2 sp 0 ;MEMBER INCIDENCES ;1 1 2
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL YIELD fy ALL
MEMBER PROPERTIES ;1 ISECTION DY dy DZ dz TY ty TZ tz
LOADING CASE 1 ;MEMBER LOADS ;1 FORCE Y UNIFORM W w
JOINT LOADS ;2 FORCE Y v ;SOLVE
va1=0 vc1=w hjl1=VEC(0)*2 vjl1=VEC(0,v) ch9=0 ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm410.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PLASTIC ANALYSIS OF CANTILEVER I-BEAM SUBJECTED
TITLE TO UNIFORMLY DISTRIBUTED LOADING & END POINT LOAD
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 26.09.05 ;TYPE PLANE FRAME run=0 ;REFNO VM411
PRINT DATA, RESULTS, FROM 1 ;METHOD PLASTIC 1 ;TABULATE ALL
NUMBER OF JOINTS 3 ;NUMBER OF MEMBERS 2 ;NUMBER OF SUPPORTS 2
NUMBER OF LOADINGS 1 ;*/8
dy=0.2068    dz=0.1339      ! Depth & breadth of I beam.
tz=0.0064    ty=0.0096      ! Web & flange thickness of I beam.
sp=3.0  nsg=16 ! Span, No. of segments.      ├───a───┤p
e=205E6        ! Young's modulus.          Y │       ▼    ┌w
nu=0.3         ! Poisson's ratio.          / │▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
fy=265E3       ! Yield strength.           /╪════════════════ ──X
nli=100 w=-120 ! No. of load incs. & udl.  / │1      2       3▲
p=-200  a=1.5  ! Point load & distance.      ├───────sp───────┤
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF INCREMENTS nli sense=2 ;NUMBER OF SEGMENTS nsg TRACE
JOINT COORDINATES ;1 0 0 SUPPORT ;2 a 0 ;3 sp 0 SUPPORT
JOINT RELEASES ;3 FORCE X MOMENT Z ;MEMBER INCIDENCES ;1 1 2
2 2 3 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL YIELD fy ALL
MEMBER PROPERTIES ;1 THRU 2 ISECTION DY dy DZ dz TY ty TZ tz
LOADING CASE 1 ;MEMBER LOADS ;1 THRU 2 FORCE Y UNIFORM W w
JOINT LOADS ;2 FORCE Y p ;SOLVE ;va1=VEC(0)*2 vc1=VEC(w)*2
hjl1=VEC(0)*3 vjl1=VEC(0,p,0) ch9=0 ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm411.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PLASTIC ANALYSIS OF CONTINUOUS I-BEAM SUBJECTED
TITLE TO UNIFORMLY DISTRIBUTED LOADING & POINT LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 15.10.05 ;TYPE PLANE FRAME run=0 ;REFNO VM420
PRINT DATA, RESULTS, FROM 1 ;METHOD PLASTIC 1 ;TABULATE ALL ;*/4
* Y│  member 1      member 2 etc.
*  ═════════════════════════════ ──X     s(1)=s1, s(2)=s2... i.e.
*  ├──────s(1)─────┼──────s(2)─────┼──►►     brackets may be omitted.
*  1              2              3
nm=4   nsg=32  nli=200    ! No.: spans, segments, load increments.
s1=VEC(4.8,4.8,4.8,4.8)   ! Spans left to right.
lsc=0        rsc=0        ! Left/right end fixity: 0=pin, 1=fixed.
dy=0.2068    dz=0.1339    ! Depth & breadth of I beam.
tz=0.0064    ty=0.0096    ! Web & flange thickness of I beam.
e=205E6      nu=0.3       ! Young's modulus; Poisson's ratio.
nc=4         fy=265E3     ! No. of conc. loads; yield strength.
vc1=VEC(-22,-22,-22,-22)  ! UDLs, l. to r., down is -ve.
IF nc>0
cs(1)=VEC(2.4)*nc         !        ▲cn(i)
cn(1)=VEC(-56)*nc         ! ├─cs(i)─┤        ┌i'th load occurs
nc(1)=VEC(1,2,3,4)        ! ╪═══════╧════════╧═════╪ on span nc(i).
ENDIF
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=nm+1 nj ;NUMBER OF MEMBERS nm
NUMBER OF SUPPORTS nj ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS nli sense=2 ;NUMBER OF SEGMENTS nsg TRACE
JOINT COORDINATES ;1 0 0 SUPPORT i=0 d=0
REPEAT ;i=i+1 d=d+s(i) ;i+1 d 0 SUPPORT ;UNTIL i=nm ;ENDREPEAT
JOINT RELEASES ;2 THRU nj FORCE X
IF lsc=0 AND rsc=0 THEN 1 THRU nj MOMENT Z
IF lsc=1 AND rsc=0 THEN 2 THRU nj MOMENT Z
IF lsc=0 AND rsc=1 THEN 1 THRU nj-1 MOMENT Z
IF lsc=1 AND rsc=1 AND nm>1 THEN 2 THRU nj-1 MOMENT Z
MEMBER INCIDENCES ;1 THRU nm RANGE 1,2 nj-1,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g*1E12 ALL YIELD fy ALL
MEMBER PROPERTIES ;1 THRU nm ISECTION DY dy DZ dz TZ tz TY ty
LOADING 1 ;MEMBER LOADS ;i=0 ;:100 ;i=i+1
i FORCE Y UNIFORM W vc(i) ;IF nc>0 ;j=0 ;:120 ;j=j+1
IF nc(j)=i THEN i FORCE Y CONCENTRATED P cn(j) L cs(j)
IF j<nc GOTO 120 ;ENDIF ;IF i<nm GOTO 100 ;SOLVE
status=1E-36 !Tells vmecp point loads in spans. ;va1=VEC(0)*nm
hjl1=VEC(0)*nj vjl1=VEC(0)*nj ch9=0 ch10=0
ct(1)=VEC(0)*nc ;#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm420.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PLASTIC ANALYSIS OF RECTANGULAR PORTAL FRAME
TITLE I-BEAM SECTION SUBJECTED TO U.D.L. & POINT LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 18.10.05 ;TYPE PLANE FRAME run=0 ;REFNO VM430
PRINT DATA, RESULTS FROM 1 ;METHOD PLASTIC 1 ;TABULATE ALL
*/10
*         ├─cs(n)─┤cn(n), for n'th load on member nc(n), -ve down.
*                 ▼                               └joint loads separate.
* ───     ╔═════════┼═════════╗     Member numbers are in
*  │      ║2   (2)   3   (3)  4║     brackets, other numbers
*  │      ║                   ║     are joints.
* h│  (1) ║                   ║(4)  Start of member 1 is at
*  │      ║Y                  ║     joint 1, start of member
*  │      ║                   ║     2 is at joint 2 and so on.
* ───    1╨  ──X              ╨5
*         ├──────────l────────┤
h=4.571  l=15.239  bfix=0  ! Sizes; base fixity, 1=fixed, 0=pin.
nsg=16   nli=200  nc=1     ! No. of segs, load incs & point loads.
e=205E6  nu=0.3   py=247E3 ! Young's mod., Poisson's ratio, yield.
hjl1=VEC(0,0,0,0,0)        ! Forces global X direction joints 1-5.
vjl1=VEC(0,0,0,0,0)        ! Forces global Y direction joints 1-5.
va1=VEC(0)*4               ! UDL's local x axes for each member.
vc1=VEC(0,-130.76,-130.76,0) ! UDL's local y axes for each member.
IF nc>0
nc(1)=VEC(2)               !        ▲cn(n)
cs(1)=VEC(3.05)            ! ├─cs(n)─┤       ┌n'th load occurs
cn(1)=VEC(-996.4)          ! ╪═══════════════╪ on member nc(n).
ENDIF
! Section depth, breadth, web & flange thickness in order follow.
db1=VEC(.9184,.3055,.0173,.0279) ! Transom beam, D,B,t,T in order.
dc1=VEC(.9184,.3055,.0173,.0279) ! Columns, D,B,t,T in order.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj=5 nj ;NUMBER OF MEMBERS nm=4 nm
NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS nli sense=2 ;NUMBER OF SEGMENTS nsg
JOINT COORDINATES ;1 0 0 SUPPORT ;2 0 h ;3 l/2 h ;4 l h
5 l 0 SUPPORT ;JOINT RELEASES ;IF bfix=0 ;1 MOMENT Z ;nj MOMENT Z
ENDIF ;MEMBER INCIDENCES ;1 THRU 4 RANGE 1,2 4,5
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL YIELD py ALL
MEMBER PROPERTIES ;1 ISECTION DY dc1 DZ dc2 TZ dc3 TY dc4
2 THRU 3 ISECTION DY db1 DZ db2 TZ db3 TY db4 ;4 AS 1
LOADING CASE 1 ;MEMBER LOADS ;i=0 ;:48 ;i=i+1
IF va(i)<>0 THEN i FORCE X UNIFORM W va(i)
IF vc(i)<>0 THEN i FORCE Y UNIFORM W vc(i) ;IF i<nm GOTO 48
i=0 ;:50 ;i=i+1 ;IF i>nc GOTO 55
nc(i) FORCE Y CONCENTRATED P cn(i) L cs(i) ;GOTO 50 ;:55
JOINT LOADS ;i=0 ;:56 ;i=i+1 ;IF hjl(i)<>0 THEN i FORCE X hjl(i)
IF vjl(i)<>0 THEN i FORCE Y vjl(i) ;IF i<nj GOTO 56
SOLVE ;status=1E-36 !Tells vmecp point loads on members
IF run=1 THEN %del vmres.stk !Clears reporting of load factors.
l2=l/2 s1=VEC(h,l2,l2,h) ch9=0 ch10=0
ct1=VEC(0)*nc ;#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm430.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PLASTIC ANALYSIS OF SINGLE BAY RIDGED PORTAL FRAME
TITLE I-BEAM SECTION SUBJECTED TO U.D.L. & POINT LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 18.10.05 ;TYPE PLANE FRAME run=0 ;REFNO VM435
METHOD PLASTIC 1 ;TABULATE DISPLACEMENTS STRESSES ;*/11
*                 ▲cY(n), for n'th load on member nc(n), distances
*   f     ├─cs(n)─┤       measured from start, loads cY(n) applied
*  ─┼─       ═══════╪═══════            in Y dirn, -ve if down.
*  ─┼─     ╔═══ (2)   3   (3) ═══╗      Member numbers are in
*   │      ║2                   4║      brackets, other numbers
*   │      ║                     ║      are joints.
*  h│  (1) ║                     ║ (4)  Start of member 1 is at
*   │      ║Y                    ║      joint 1, start of member
*   │      ║                     ║      2 is at joint 2 and so on.
*  ─┴─    1╨  ──X                ╨5
*          ├──────────l──────────┤
h=4.267 l=12.191 f=2.524 bfix=0 ! Sizes; base fixity, 1=fixed, 0=pin.
nsg=16   nli=100  nc=0     ! No. of segs, load incs & point loads.
e=205E6  nu=0.3   py=247E3 ! Young's mod., Poisson's ratio, yield.
up=-17.05                  ! UDL on plan; negative is downwards.
wl(1)=VEC(0)*4             ! Wind loads on members (+ve suction).
hjl(1)=VEC(0,0,0,0,0)      ! Forces global X direction joints 1-5.
vjl(1)=VEC(0,0,0,0,0)      ! Forces global Y direction joints 1-5.
IF nc>0
cs(1)=VEC(0)               !        ▲cY(n) global Y direction.
cY(1)=VEC(0)               ! ├─cs(n)─┤        ┌n'th load occurs
nc(1)=VEC(0)               ! ╪═══════╧════════╧═╪ on member nc(n).
ENDIF
!        ▲cY(n) global Y direction.
! ├─cs(n)─┤        ┌n'th load occurs on
! ╪═══════╧════════╧═╪ on member nc(n).
! Section depth, breadth, web & flange thickness in order follow.
db1=VEC(.2514,.1461,.0060,.0086) ! Rafters.
dc1=VEC(.2514,.1461,.0060,.0086) ! Columns.
#cc924.stk !Import set of parameters if available from cc924.stk.
PRINT DATA, RESULTS FROM 1 LENGTH gen
NUMBER OF JOINTS nj=5 nj ;NUMBER OF MEMBERS nm=4 nm
NUMBER OF SUPPORTS 2 ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS nli sense=2 ;NUMBER OF SEGMENTS nsg 0.2
JOINT COORDINATES ;1 0 0 SUPPORT ;2 0 h ;3 l/2 h+f ;4 l h
5 l 0 SUPPORT ;JOINT RELEASES ;IF bfix=0 ;1 MOMENT Z ;nj MOMENT Z
ENDIF ;MEMBER INCIDENCES ;1 THRU 3 RANGE 1,2 3,4 ;4 5 4
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL YIELD py ALL
MEMBER PROPERTIES ;1 ISECTION DY dc1 DZ dc2 TZ dc3 TY dc4
2 THRU 3 ISECTION DY db1 DZ db2 TZ db3 TY db4 ;4 AS 1
LOADING CASE 1 ;MEMBER LOADS
IF up<>0 THEN 1 THRU nm FORCE Y PROJECTED UNIFORM W up
i=0 ;:48 ;i=i+1 ;IF wl(i)<>0 AND i<nm ;i FORCE Y UNIFORM W wl(i)
ENDIF ;IF wl(i)<>0 AND i=nm ;i FORCE Y UNIFORM W -wl(i) ;ENDIF
IF i<nm GOTO 48 ;i=0 ;:50 ;i=i+1 ;IF i>nc GOTO 55
nc(i) FORCE Y GLOBAL CONCENTRATED P cY(i) L cs(i) ;GOTO 50 ;:55
JOINT LOADS ;i=0 ;:56 ;i=i+1 ;IF hjl(i)<>0 THEN i FORCE X hjl(i)
IF vjl(i)<>0 THEN i FORCE Y vjl(i) ;IF i<nj GOTO 56
SOLVE ;status=1E-36 !Tells vmecp point loads on members
IF run=1 THEN %del vmres.stk !Clears reporting of load factors.
l'=l/2 h'=h+f x1=VEC(0,0,l',l,l) y1=VEC(0,h,h',h,0) i=0 ;:57
i=i+1 x=x(i+1)-x(i) y=y(i+1)-y(i) s(i)=SQR(x^2+y^2) cx(i)=x/s(i)
cy(i)=y/s(i) va(i)=up*cy(i)*cx(i) vc(i)=wl(i)+up*cx(i)^2
IF i<4 GOTO 57 ;i=0 ;:58 ;i=i+1 ;IF i>nc GOTO 59 ;mn=nc(i)
ct(i)=cy(mn)*cY(i) cn(i)=cx(mn)*cY(i) ;GOTO 58 ;:59
njn=nj+nm*(nsg-1)
```

```
ch9=0 ch10=0 ;#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm435.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
IF gen=-1 ! If called from sc461.pro, pipe results to sc461.res.
%del sc461.res
mn=0 ;:902 ;mn=mn+1 ;! End node Nos.
nst=ARR(1,mn,1) nen=ARR(1,mn,2) ;! Coordinates ;xcs=ARR(8,nst,3)
ycs=ARR(8,nst,4) xce=ARR(8,nen,3) yce=ARR(8,nen,4) ;! Sectn props.
ax=ARR(11,mn,1) ay=ARR(11,mn,2) iz=ARR(11,mn,6) e=ARR(11,mn,11)
g=ARR(11,mn,12) ;! Displ in global axes ;rn=3*(nst-1)+1
xds'=ARR(6,rn,lli) rn=rn+1 yds'=ARR(6,rn,lli) rn=rn+1
zrs'=ARR(6,rn,lli) rn=3*(nen-1)+1 xde'=ARR(6,rn,lli) rn=rn+1
yde'=ARR(6,rn,lli) rn=rn+1 zre'=ARR(6,rn,lli)
! Memb forces ;rn=(lli-1)*nm+mn xfs(mn)=ARR(13,rn,1) yfs(mn)=ARR(13,rn,2)
zms(mn)=ARR(13,rn,3) xfe(mn)=ARR(13,rn,4) yfe(mn)=ARR(13,rn,5)
zme(mn)=ARR(13,rn,6) xds(mn)=ARR(13,rn,13) yds(mn)=ARR(13,rn,14)
zrs=(mn)ARR(13,rn,15) xde(mn)=ARR(13,rn,16) yde(mn)=ARR(13,rn,17)
zre(mn)=ARR(13,rn,18) ks(mn)=ARR(1,mn,6) ke(mn)=ARR(1,mn,7)
IF meth=3 THEN mzcs=ARR(10,rn,4) mzce=ARR(10,rn,8)
>sc461.res ! +xfs(mn)= +yfs(mn)= +zms(mn)=
>sc461.res ! +xfe(mn)= +yfe(mn)= +zme(mn)=
IF mn<nm GOTO 902
ENDIF
< ;FINISH
```

```
TITLE SINGLE/MULTI-BAY PORTAL FRAME/S WITH HAUNCHES
TITLE HAVING 'nb' BAYS SUBJECTED TO VERTICAL AND SWAY
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;TABULATE ALL ;MADEBY DWB ;REFNO VM436
TYPE PLANE FRAME run=0 ;DATE 30.10.05 ;*/9
*       ▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼ udl, plan load
*   f    Y          5                  12           per unit length,
*  ─┼─   │     ═══╪═══      9     ═══╪═══      16        -ve down.
*  ─┼─ 2╓3=4══        ══6=7╥10=11        13=14╥17=18
*   │   ║                  ║                  ║    Joint Nos.
*   b   ║──┼ha          ha┼──║                  ║    shown, member
*   │   ║                  ║                  ║    Nos. omitted
*  ─┴─ 1╨  ──X             ╨8              15╨    for clarity.
*       ├─────────a────────┼──────────a──────────┼──►
a=12.191 b=4.267 f=2.524 nli=100  ! Frame sizes, No. of load incs.
nb=1  e=205E6  nu=0.3  ! No. of bays, Young's mod., Poisson's rat.
de1=VEC(.2514,.1461,.0060,.0086)      ! External columns, D,B,t,T.
di1=VEC(.2514,.1461,.0060,.0086)      ! Internal columns, D,B,t,T.
dr1=VEC(.2514,.1461,.0060,.0086)      ! Rafters, D,B,t,T.
nsg=8  ha=2.8  dmax=0.2514 ! No of segs; haunch length & max depth.
bfix=1 udl=-17.05 sl=0 ! Base fixity, plan udl, sway load at jnt 2.
py=247E3   ! Yield stress.
#cc924.stk !Import set of parameters if available from cc924.stk.
METHOD PLASTIC 1 ;p=.2 ;IF nb>1 THEN p=0
NUMBER OF JOINTS nj=7*nb+2 nj ;NUMBER OF MEMBERS nm=7*nb+1 nm
NUMBER OF SUPPORTS ns=nb+1 ns ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS nli sense=2 ;NUMBER OF SEGMENTS nsg p
JOINT COORDINATES ;dy=f*(ha/2)/(a/2)
1 THRU nj-1 STEP 7 X 0 Y 0 XL nb*a SUPPORT
2 THRU nj STEP 7 X 0 Y b XL nb*a ;IF nb=1
3 X ha/2 Y b+dy ;4 X ha Y b+2*dy ;5 X a/2 Y b+f ;6 X a-ha Y b+2*dy
7 X a-ha/2 Y b+dy ;ENDIF ;IF nb>1
3 THRU nj-6 STEP 7 X ha/2 Y b+dy XL (nb-1)*a+ha/2
4 THRU nj-5 STEP 7 X ha Y b+2*dy XL (nb-1)*a+ha
5 THRU nj-4 STEP 7 X a/2 Y b+f XL (nb-1)*a+a/2
6 THRU nj-3 STEP 7 X a-ha Y b+2*dy XL nb*a-ha
7 THRU nj-2 STEP 7 X a-ha/2 Y b+dy XL nb*a-ha/2 ;ENDIF
JOINT RELEASES ;IF bfix=0 THEN 1 THRU nj-1 STEP 7 MOMENT Z
MEMBER INCIDENCES ;IF nb>1 THEN 1 THRU nm STEP 7 RANGE 1,2 nj-1,nj
IF nb=1 ;1 1 2 ;8 8 9 ;ENDIF
IF nb=1 THEN 2 THRU nm-1 CHAIN 2,3,4,5,6,7,9
IF nb>1 ;2 THRU nm-6 STEP 7 RANGE 2,3 nj-7,nj-6
3 THRU nm-5 STEP 7 RANGE 3,4 nj-6,nj-5
4 THRU nm-4 STEP 7 RANGE 4,5 nj-5,nj-4
5 THRU nm-3 STEP 7 RANGE 5,6 nj-4,nj-3
6 THRU nm-2 STEP 7 RANGE 6,7 nj-3,nj-2
7 THRU nm-1 STEP 7 RANGE 7,9 nj-2,nj ;ENDIF
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL YIELD py ALL
MEMBER PROPERTIES
1 THRU nm STEP nm-1 ISECTION DY de1 DZ de2 TZ de3 TY de4
IF nb=2 THEN 8 ISECTION DY di1 DZ di2 TZ di3 TY di4
IF nb>2 ;8 THRU nm-7 STEP 7 ISECTION DY di1 DZ di2 TZ di3 TY di4
ENDIF ;4 THRU 5 ISECTION DY dr1 DZ dr2 TZ dr3 TY dr4
IF nb=2 THEN 11 THRU 12 AS 4
IF nb>2 THEN 11 THRU nm-4 STEP 7 AS 4
IF nb>2 THEN 12 THRU nm-3 STEP 7 AS 4
2 THRU 3 ISECTION DY dmax DYL dr1 DZ dr2 DZL dr2 TZ dr3 TY dr4
6 THRU 7 AS 3 THRU 2 ;IF nb>1 ;i=1 m=2 ;REPEAT ;i=i+1 m=m+7
m THRU m+1 AS 2 THRU 3 ;m+4 THRU m+5 AS 3 THRU 2 ;UNTIL i=nb
ENDREPEAT ;ENDIF
```

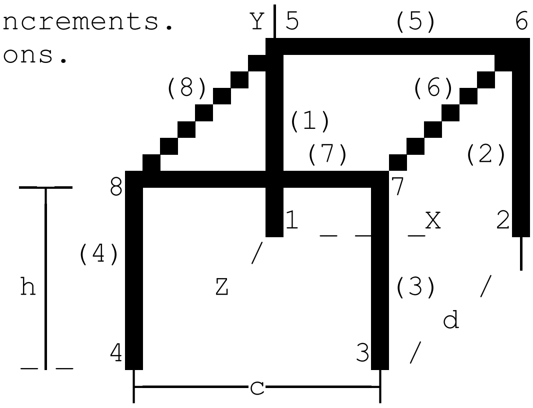
```
LOADING CASE 1 ;MEMBER LOADS ;i=0 m=-5 ;REPEAT ;i=i+1 m=m+7
m THRU m+5 FORCE Y PROJECTED UNIFORM W udl ;UNTIL i=nb ;ENDREPEAT
JOINT LOADS ;2 FORCE X sl ;SOLVE
IF run=1 THEN %del vmres.stk !Clears reporting of load factors.
hjl1=VEC(0)*nj hjl2=sl vjl1=VEC(0)*nj lr=SQR((a/2)^2+f^2) i=0
cx=(a/2)/lr cy=f/lr m=-5 ;:20 ;i=i+1 m=m+7 va=udl*cy*cx
vc=udl*cx^2 va(m)=VEC(va)*3 vc(m)=VEC(vc)*3 va=-va m'=m+3
va(m')=VEC(va)*3 vc(m')=VEC(vc)*3 ;IF i<nb GOTO 20 ;ch9=0 ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm436.stk) ;#vmres.ndf !Conclude results.
mjn=5 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE PLASTIC ANALYSIS OF MULTI-STOREY FRAME SUBJECTED TO
TITLE U.D.L. & VERTICAL & HORIZONTAL POINT LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 07.11.05 ;TYPE PLANE FRAME run=0 ;REFNO VM440
PRINT DATA, RESULTS FROM 1 ;METHOD PLASTIC 1 ;*/13
nb=1   ns=2   ! No. of bays & storeys.
s=3.82 h=3.82 ! Bay span & storey height.              nj-1   nj
axb=38.2E-4   ! X-sect. area of beams.    ▲
ayb=23.9E-4   ! Shear area of beams.
izb=558E-8    ! Mom. of inert. of beams.
zpb=111.2E-6  ! Plastic modulus of beams.
axc=38.2E-4   ! X-sect. area of cols.     h     Y
ayc=23.9E-4   ! Shear area of columns.
izc=558E-8    ! Mom. of inert. of cols.
zpc=111.2E-6  ! Plastic modulus of cols.      nb+2      nb+3 etc.
e=206E6       ! Young's modulus.          h    └joint numbers.
nu=0.3        ! Poisson's ratio.
py=240E3      ! Yield stress.                1   ──X   2      nb+1
nsg=32        ! Number of segments.            ├───s───┼►►nb bays.
nli=200  fix=-1  ! No. of load increments; bases -1=fixed, 0=pin.
hjl1=VEC(0,0,10,0,10,0) ! Hor. joint loads, l. to r., bot. to top.
vjl1=VEC(0,0,-40,-40,-40,-40) ! Vir. joint loads, l to r, b to t.
udl=0     nc=2  ! Udl on all beams; No. of conc. loads on members.
! Members are numbered left to right ground floor columns, left to
! right first floor beams, left to right 1st floor cols and so on.
IF nc>0
nc(1)=VEC(3,6)              !  ├─cs(n)─▲cn(n)   ┌n'th load occurs on
cs(1)=VEC(1.91)*2           !  ╪═══════════════╪ on member nc(n).
cn(1)=VEC(-40)*2            !  lower or left      higher or right.
ENDIF
!  ├─cs(n)─▲cn(n)   ┌n'th load occurs on
!  ╪═══════════════╪ on member nc(n).
!  lower or left      higher or right.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj=(nb+1)*(ns+1) nj
NUMBER OF MEMBERS nm=ns*(2*nb+1) nm ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 1 ;NUMBER OF INCREMENTS nli
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES
n=0 j=-nb ;:40 ;n=n+1 j=j+nb+1 ;j THRU j+nb X 0 Y h*(n-1) XL nb*s
IF n<ns+1 GOTO 40 ;JOINT RELEASES
1 THRU nb+1 FORCE X -1 FORCE Y -1 MOMENT Z fix ;MEMBER INCIDENCES
n=0 j=-nb m=-2*nb ;:50 ;n=n+1 j=j+nb+1 m=m+2*nb+1
m+nb+1 THRU m+2*nb RANGE j+nb+1 j+nb+2 j+2*nb j+2*nb+1
m THRU m+nb RANGE j j+nb+1 j+nb j+2*nb+1 ;IF n<ns GOTO 50
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL YIELD py ALL
MEMBER PROPERTIES ;n=0 m=-2*nb ;:70 ;n=n+1  m=m+2*nb+1
m+nb+1 THRU m+2*nb AX axb AY ayb IZ izb FXP axb*py MZP zpb*py
m THRU m+nb AX axc AY ayc IZ izc FXP axc*py MZP zpc*py
IF n<ns GOTO 70
LOADING CASE 1 ;MEMBER LOADS
n=0 m=-2*nb ;:48 ;n=n+1  m=m+2*nb+1
m+nb+1 THRU m+2*nb FORCE Y UNIFORM W udl
IF n<ns GOTO 48 ;i=0 ;:52 ;i=i+1 ;IF i>nc GOTO 53
nc(i) FORCE Y CONCENTRATED P cn(i) L cs(i) ;GOTO 52 ;:53
JOINT LOADS ;i=0 ;:56 ;i=i+1 ;IF hjl(i)<>0 THEN i FORCE X hjl(i)
IF vjl(i)<>0 THEN i FORCE Y vjl(i) ;IF i<nj GOTO 56
SOLVE ;status=1E-36 !Tells vmecp point loads on members
IF run=1 THEN %del vmres.stk !Clears reporting of load factors.
nb'=nb+1 n=0 m=-2*nb ;:67 ;n=n+1 m=m+2*nb+1 m'=m+nb+1
s(m')=VEC(s)*nb ct(m')=VEC(0)*nb va(m')=VEC(0)*nb
vc(m')=VEC(udl)*nb s(m)=VEC(h)*nb' ct(m)=VEC(0)*nb'
```

```
va(m)=VEC(0)*nb vc(m)=VEC(0)*nb ;IF n<ns GOTO 67 ;ch9=0 ch10=0
i=0 ;:68 ;i=i+1 m=nc(i) ;IF i<nc GOTO 68 ;i=0 ;:78 ;i=i+1
IF i<nm GOTO 78 ;#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm440.stk) ;#vmres.ndf !Conclude results.
mjn=nb+2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

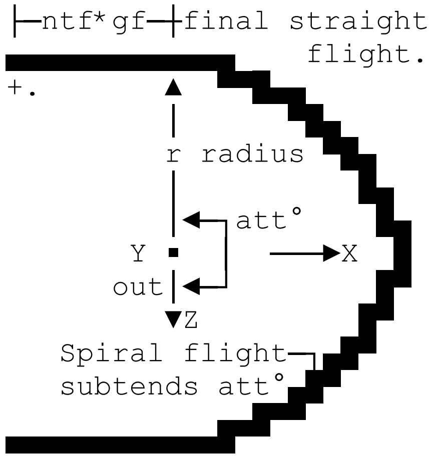
```
TITLE CANTILEVER BEAM IN SPACE SUBJECTED TO 'n' POINT LOADS
TITLE MAGNITUDE w(1:n) POSITIONS a(1:n) DIRECTIONS d(1:n)
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 26.01.06 ;TYPE SPACE FRAME run=0 ;REFNO VM501
PRINT DATA, RESULTS, FROM 1 ;NUMBER OF JOINTS 2
NUMBER OF MEMBERS 1 ;NUMBER OF SUPPORTS 1 ;NUMBER OF LOADINGS 3
*/6
s=3.0          ! Span of cantilever.
dy=0.36        ! Depth of beam.              Y
dz=0.3         ! Breadth of beam.            ├─acl(n)─┤pcl(n)
e=28E6/3       ! Young's modulus.           /│        ▼
nu=0.2         ! Poisson's ratio.        Z /╞═══════════════ ──X
nsg=64         ! No. of segments.      out /├───────s───────┤
ejl1=VEC(1)*6  ! End joint loads in global X,Y,Z & about X,Y,Z.
nli=2  ncl=3   ! Nos. of load incrs. & concentrated loads ncl=0-5.
pcl1=VEC(-30,-20,10)      ! Magnitude of point loads.
acl1=VEC(1.5,2,3.0)       ! Distances to point loads.
dcl1=VEC(1,2,3)           ! Dir. of global point loads 1-3 ≡ X-Z.
ux1=1  uy1=-0.6  uz1=0.2  ! UDL on cant. in global directions X-Z.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF INCREMENTS nli ;IF nli=1 THEN METHOD ELASTIC NODES
IF nli>1 THEN METHOD SWAY NODES ;NUMBER OF SEGMENTS nsg TRACE
JOINT COORDINATES ;1 0 0 0 SUPPORT ;2 s 0 0 ;MEMBER INCIDENCES
1 1 2 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 RECTANGLE DY dy DZ dz
LOADING CASE 1 ;MEMBER LOADS ;IF ncl<1 GOTO 6 ;i=0 ;:5 ;i=i+1
IF dcl(i)=1 THEN 1 FORCE X GLOBAL CONCENTRATED P pcl(i) L acl(i)
IF dcl(i)=2 THEN 1 FORCE Y GLOBAL CONCENTRATED P pcl(i) L acl(i)
IF dcl(i)=3 THEN 1 FORCE Z GLOBAL CONCENTRATED P pcl(i) L acl(i)
IF i<ncl GOTO 5 ;:6 ;1 FORCE X GLOBAL UNIFORM W ux1
1 FORCE Y GLOBAL UNIFORM W uy1
1 FORCE Z GLOBAL UNIFORM W uz1 ;JOINT LOADS ;njo=2 nmo=1
2 FORCE X ejl1 Y ejl2 Z ejl3 MOMENT X ejl4 Y ejl5 Z ejl6
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:8 ;jn=jn+1
jn FORCE X jn Y jn Z jn MOMENT X jn Y jn Z jn ;IF jn<njo GOTO 8
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:9 ;jn'=njo-jn
jn=jn+1 jn FORCE X jn' Y jn' Z jn' MOMENT X jn' Y jn' Z jn'
IF jn<njo GOTO 9 ;SOLVE ;nm=nmo*nsg nj=njo+(nsg-1)*nmo
IF nsg<4 THEN FINISH ;mcl1=VEC(1)*ncl jst1=1 jen1=2 xc(1)=VEC(0,s)
yc(1)=VEC(0)*2 zc(1)=VEC(0)*2 xjl(1)=VEC(0,ejl1)
yjl(1)=VEC(0,ejl2) zjl(1)=VEC(0,ejl3) xjm(1)=VEC(0,ejl4)
yjm(1)=VEC(0,ejl5) zjm(1)=VEC(0,ejl6) ;#vmecs.ndf
fnm=$(vm501.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE FOUR LEGGED STOOL SUBJECTED TO JOINT LOADS AND UDL
TITLE RACKING LOADS ON COLUMNS AND CONNECTING BEAMS.
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
PRINT DATA, RESULTS FROM 1 ;MADEBY DWB ;REFNO VM510
TYPE SPACE FRAME run=0 ;DATE 24.01.06 ;NUMBER OF JOINTS 8
NUMBER OF MEMBERS 8 ;NUMBER OF SUPPORTS 4 ;NUMBER OF LOADINGS 3
*/12
nsg=32  nli=4 ! No. of segs & load increments.
c=8.6  d=8.6  h=6.4  ! Frame dimensions.
dyc=.4   dzc=.4      ! Column sizes.
dyb=.4   dzb=.4      ! Beam sizes.
e=9E6          ! Young's modulus.
nu=0.2         ! Poisson's ratio.
fix=0          ! Fixed feet 1=yes.
ubx=5.6        ! Udl beams global X.
uby=-8.6       ! Udl beams global Y.
ubz=5.6        ! Udl beams global Z.
ucx=5.6        ! Udl cols global X.
ucz=5.6        ! Udl cols global Z.
```

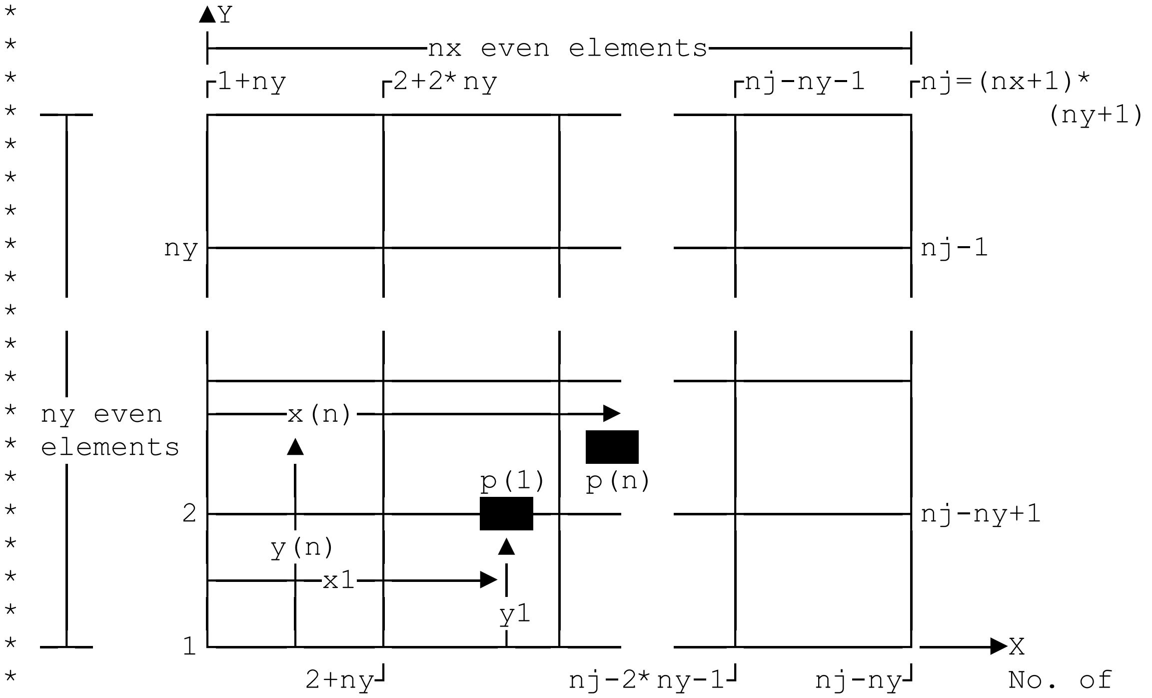



```
xjl1=VEC(0,0,0,0,100,0,-100,0)          ! Dirn. X joint loads 1-8.
yjl1=VEC(-4.5)*8                        ! Dirn. Y joint loads 1-8.
zjl1=VEC(0,0,0,0,0,100,0,-100)          ! Dirn. Z joint loads 1-8.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF INCREMENTS nli ;IF nli=1 THEN METHOD ELASTIC NODES
IF nli>1 THEN METHOD SWAY NODES ;NUMBER OF SEGMENTS nsg TRACE
JOINT COORDINATES ;1 0 0 0 SUPPORT ;2 c 0 0 SUPPORT
3 c 0 d SUPPORT ;4 0 0 d SUPPORT ;5 0 h 0 ;6 c h 0 ;7 c h d
8 0 h d ;JOINT RELEASES ;IF fix=0 THEN 1 THRU 4 MOMENT Y Z
MEMBER INCIDENCES ;1 THRU 4 RANGE 1,5 4,8
5 THRU 8 CHAIN 5,6,7,8,5 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU 4 RECTANGLE DY dyc DZ dzc
5 THRU 8 RECTANGLE DY dyb DZ dzb
LOADING CASE 1 ;JOINT LOADS ;jn=0 ;:16 ;jn=jn+1
jn FORCE X xjl(jn) Y yjl(jn) Z zjl(jn) ;IF jn<8 GOTO 16
MEMBER LOADS ;5 THRU 8 FORCE X GLOBAL UNIFORM W ubx
5 THRU 8 FORCE Y GLOBAL UNIFORM W uby
5 THRU 8 FORCE Z GLOBAL UNIFORM W ubz
1 THRU 4 FORCE X GLOBAL UNIFORM W ucx
1 THRU 4 FORCE Z GLOBAL UNIFORM W ucz ;njo=8 nmo=8
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn Z jn MOMENT X jn Y jn Z jn ;IF jn<njo GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=njo-jn
jn=jn+1 jn FORCE X jn' Y jn' Z jn' MOMENT X jn' Y jn' Z jn'
IF jn<njo GOTO 19 ;SOLVE ;IF nsg<4 THEN FINISH ;nm=nmo*nsg
nj=njo+(nsg-1)*nmo ncl=0 xc1=VEC(0,c,c,0)*2 yc1=VEC(0)*4
yc5=VEC(h)*4 zc1=VEC(0,0,d,d)*2 jst1=VEC(1,1)/njo
jen1=VEC(5,6,7,8,6,7,8,5) ux1=VEC(ucx)*4 ux5=VEC(ubx)*4
uy1=VEC(0)*4 uy5=VEC(uby)*4 uz1=VEC(ucz)*4 uz5=VEC(ubz)*4
xjm(1)=VEC(0)*njo yjm(1)=VEC(0)*njo zjm(1)=VEC(0)*njo ;#vmecs.ndf
fnm=$(vm510.stk) ;#vmres.ndf !Conclude results.
mjn=5 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE SPIRAL CONCRETE STAIRS FORMED FROM INITIAL STRAIGHT
TITLE FLIGHT, THEN SPIRAL, THEN FINAL STRAIGHT FLIGHT;
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WO2K DONE.
PRINT DATA, RESULTS FROM 1 ;TYPE SPACE FRAME run=0 ;DATE 30.01.06
MADEBY DWB ;NUMBER OF LOADING 3 ;NUMBER OF SUPPORT 0 ;REFNO VM520
nti=5       ! No. treads initial flight.   ├ntf*gf┼final straight
ri=0.174    ! Riser/going initial flight ±.              flight.
gi=0.250    ! Going/riser initial straight +.
nts=16      ! No. treads spiral flight.
rs=0.174    ! Riser/going spiral flight ±.         r radius
r=1.45      ! Radius to centre of flight.
att=180     ! Angle (+ve anticl. abt Y).              att°
ntf=3       ! No. treads final straight.       Y           X
rf=0.174    ! Riser/going final straight ±.      out
gf=0.250    ! Going/riser final straight +.          Z
wd=1.100    ! Width of stair.                Spiral flight
ed=0.150    ! Effective depth of stair.      subtends att°
e=28E6/3    ! Young's modulus.
nu=0.2      ! Poisson's ratio.
udl=-19.8   ! Factored crush UDL on plan.  ├nti*gi┼initial flight
nli=4       ! Number of loading increments.        in X direction.
#cc924.stk !Import set of parameters if available from cc924.stk.
nm=nti+nts+ntf ;NUMBER OF JOINTS nj=nm+1 nj ;NUMBER OF MEMBERS nm
NUMBER OF INCREMENTS nli ;IF nli=1 THEN METHOD ELASTIC JOINTS
IF nli>1 THEN METHOD SWAY JOINTS ;JOINT COORDINATES ;IF nti=0
1 X 0 Y 0 Z 0 ;ENDIF ;IF nti>0 ;* First straight flight in X dirn.
1 THRU nti+1 X 0 Y 0 Z 0 XL nti*gi YL nti*ri ZL 0 ;ENDIF
* Spiral flight. ;ast=RAD(ABS(att/nts)) xi=nti*gi y=nti*ri i=0
:10 ;i=i+1 x=xi+r*SIN(ABS(ast)*i) y=y+rs z=-r+r*COS(ABS(ast)*i)
IF att<0 THEN z=-z ;j=nti+1+i xc(j)=x yc(j)=y zc(j)=z j x y z
IF i<nts GOTO 10 ;IF ntf>0 ;* Final straight flight. ;alp=ast*nts
i=0 ;:12 ;i=i+1 x=x+gf*COS(alp) y=y+rf z=z-gf*SIN(alp)
j=nti+1+nts+i xc(j)=x yc(j)=y zc(j)=z j x y z
IF i<ntf GOTO 12 ;ENDIF ;JOINT RELEASES ;1 FORCE X -1 Y -1 Z -1
1 MOMENT X -1 Y -1 Z -1 ;nj FORCE X -1 Y -1 Z -1
nj MOMENT X -1 Y -1 Z -1 ;MEMBER INCIDENCES
1 THRU nm RANGE 1,2 nm,nj ;CONSTANT E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU nm RECTANGLE DY ed DZ wd
LOADING CASE 1 - FACTORED DL+LL ON PLAN (PROJECTED).
MEMBER LOADS ;1 THRU nm FORCE Y PROJECTED UNIFORM W udl*wd
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn Z jn MOMENT X jn Y jn Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' Z jn' MOMENT X jn' Y jn' Z jn'
IF jn<nj GOTO 19 ;SOLVE ;nsg=1 njo=nj nmo=nm ncl=0 nj'=nj-1
jst1=VEC(1,1)/nj' jen1=VEC(2,1)/nj xc1=0 yc1=0 zc1=0 nt'=nti+1
IF nti>0 ;xc1=VEC(0,gi)/nt' yc1=VEC(0,ri)/nt' zc1=VEC(0)*nt'
ENDIF ;ux1=VEC(0)*nj uy1=VEC(0)*nj uz1=VEC(0)*nj xc0=0 zc0=0
xc(nj+1)=xc(nj) zc(nj+1)=zc(nj) i=0 ;:3 ;i=i+1
lb=SQR((xc(i)-xc(i-1))^2+(zc(i)-zc(i-1))^2)/2
len=SQR((xc(i+1)-xc(i))^2+(zc(i+1)-zc(i))^2)/2+lb
yjl(i)=len*udl*wd ;IF i<njo GOTO 3 ;xjl1=VEC(0)*nj zjl1=VEC(0)*nj
xjm(1)=VEC(0)*njo yjm(1)=VEC(0)*njo zjm(1)=VEC(0)*njo ;#vmecs.ndf
fnm=$(vm520.stk) ;#vmres.ndf !Conclude results.
mjn=5 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```



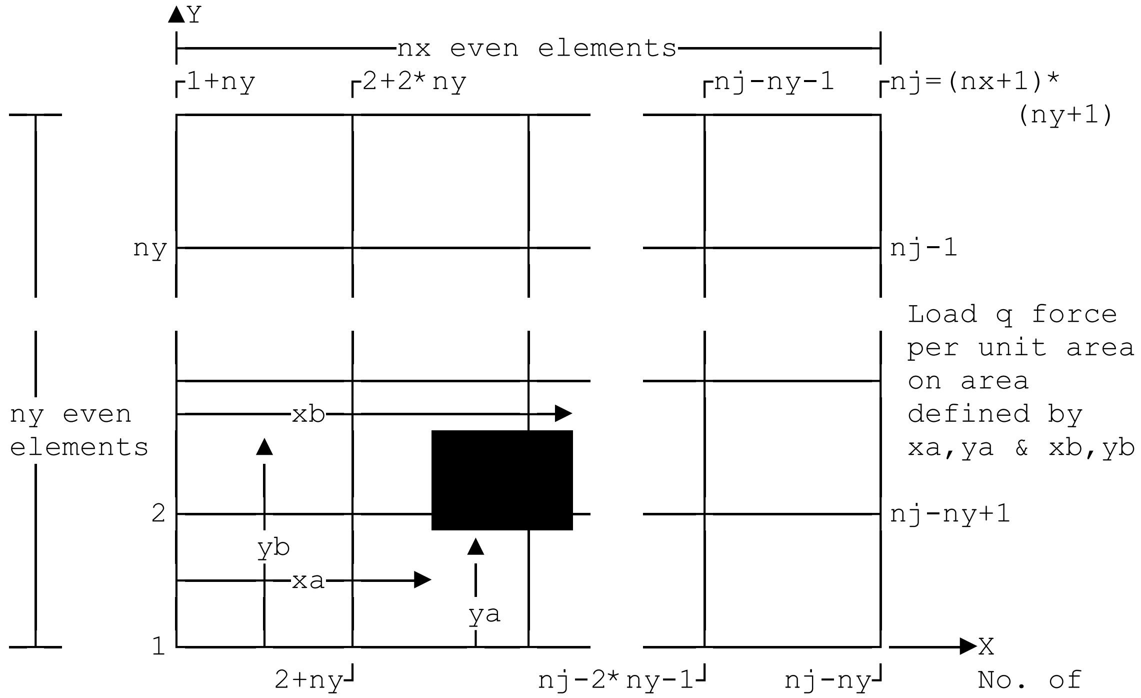
```
TITLE FLAT PLATE IN FLEXURE HAVING FOUR SIMPLY SUPPORTED
TITLE EDGES SUBJECTED TO UDL & CONCENTRATED LOADS.
TITLE RESULTS COMPARE THE MATRIX STIFFNESS METHOD WITH
TITLE NAVIER'S DOUBLE TRIGONOMETRIC SERIES SOLUTION.
MADEBY DWB ;PRINT DATA RESULTS FROM 1 ;DATE 26.01.05 ;REFNO VM601
METHOD ELASTIC JOINTS ;TYPE PLANE GRID run=0
TABULATE DISPLACEMENTS REACTIONS ;*/21
```




```
e=28E6/3 nu=.2 q=-22 ! Young's modulus, Poisson's ratio, slab udl.
xe=.5 ye=.5 t=.254   ! Element sizes in X & Y dirn. & thickness.
nx=16 ny=16 nl=2     ! No. of elems in X & Y; No. of conc. loads.
p(1)=VEC(-480)*nl    ! Magnitude of concentrated loads in order.
x(1)=VEC(2,6)        ! X ordinate of concentrated load/s in order.
y(1)=VEC(2,6)        ! Y ordinate of concentrated load/s in order.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nxj=nx+1 nyj=ny+1 nj=nxj*nyj nj
NUMBER OF MEMBERS nm=6*nx*ny nm ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 1
JOINT COORDINATES ;i=0 j=-ny x=-xe ;:10 ;i=i+1 j=j+ny+1 x=x+xe
j THRU j+ny X x Y 0 YL ny*ye ;IF i<nx+1 GOTO 10 ;JOINT RELEASES
fz=-1 mx=0 ;1 THRU 1+ny FORCE Z fz MOMENT X mx MOMENT Y mx
nj-ny THRU nj FORCE Z fz MOMENT X mx MOMENT Y mx
2+ny THRU nj-2*ny-1 STEP ny+1 FORCE Z fz MOMENT X mx MOMENT Y mx
2+2*ny THRU nj-ny-1 STEP ny+1 FORCE Z fz MOMENT X mx MOMENT Y mx
MEMBER INCIDENCES ;g=e/(2*(1+nu)) m=-nx*6 j=-1 ;:20 ;m=m+nx*6
j=j+1 ;m+1 THRU m+nx*6 ELEMENT j+ny+3,j+ny+2,j+1,j+2 BUMP ny+1
IF j<ny-1 GOTO 20 ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t
LOADING CASE 1 ;JOINT LOADS ;1 AREA nj FORCE Z q nx+ny ;IF nl>0
i=0 ;:25 ;i=i+1 ;0 FORCE Z p(i) x(i) y(i) ;IF i<nl GOTO 25
ENDIF ;SOLVE ;status=1
gtot=0 nur=0 a=xe*nx b=ye*ny d=e*t^3/(12*(1-nu^2)) j=INT(nj/2)+1
node=ARR(8,j,2) r=3*node dcf=ARR(6,r,1) x=ARR(8,node,3)
y=ARR(8,node,4) w=0 ;IF nl>0 ;i=0 ;:35 ;i=i+1 p=p(i) x'=x(i)
y'=y(i) pc=4*p/(PI^4*a*b*d) m=0 ;:40 ;m=m+1 n=0 ;:50 ;n=n+1
num=SIN(m*PI*x'/a)*SIN(n*PI*y'/b)*SIN(m*PI*x/a)*SIN(n*PI*y/b)
w=w+pc*num/(m^2/a^2+n^2/b^2)^2 ;IF n<11 GOTO 50 ;IF m<11 GOTO 40
IF i<nl GOTO 35 ;ENDIF ;i=0 j=-ny x'=-xe ;:52 ;i=i+1 j=j+ny+1
x'=x'+xe k=j-1 y'=-ye ;:53 ;k=k+1 y'=y'+ye p=q*xe*ye
IF i=1 OR i=nx+1 THEN p=p/2 ;IF k=j OR k=j+ny THEN p=p/2
pc=4*p/(PI^4*a*b*d) m=0 ;:54 ;m=m+1 n=0 ;:55 ;n=n+1
num=SIN(m*PI*x'/a)*SIN(n*PI*y'/b)*SIN(m*PI*x/a)*SIN(n*PI*y/b)
```

```
w=w+pc*num/(m^2/a^2+n^2/b^2)^2 ;IF n<11 GOTO 55 ;IF m<11 GOTO 54
IF k<j+ny GOTO 53 ;IF i<nx+1 GOTO 52 ;d1=dcf d2=w
#vmper.ndf !Compute percentage difference & any message.
*                  NL-STRESS           Navier               %age
*                  Analysis            Solution             diff.
* Defln at centre: +dcf                +w                   $ok
fnm=$(vm601.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
STRUCTURE FLAT PLATE IN FLEXURE HAVING FOUR SUPPORTED EDGES
STRUCTURE SUBJECTED TO UDL DEFINED BY AN AREA xa,ya & xb,yb.
STRUCTURE RESULTS COMPARE THE MATRIX STIFFNESS METHOD WITH
STRUCTURE NAVIER'S DOUBLE TRIGONOMETRIC SERIES SOLUTION.
MADEBY DWB ;DATE 27.08.05 ;PRINT DATA RESULTS FROM 1 ;REFNO VM602
TABULATE DISPLACEMENTS REACTIONS ;METHOD ELASTIC JOINTS
TYPE PLANE GRID run=0 ;*/21
```




```
xe=1  ye=1 t=.254  ! Element sizes in X & Y dirn. & thickness.
e=30E6/3   nu=.3   ! Young's modulus & Poisson's ratio.
nx=8  ny=8  q=-40  ! No. of elements in X & Y; udl on area, kN/m2.
xa=2   ya=2   xb=6   yb=6   ! Coordinates defining area loaded.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nxj=nx+1 nyj=ny+1 nj=nxj*nyj nj
NUMBER OF MEMBERS nm=6*nx*ny nm ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 1
JOINT COORDINATES ;i=0 j=-ny x=-xe ;:10 ;i=i+1 j=j+ny+1 x=x+xe
j THRU j+ny X x Y 0 YL ny*ye ;IF i<nx+1 GOTO 10 ;JOINT RELEASES
fz=-1 mx=0 ;1 THRU 1+ny FORCE Z fz MOMENT X mx MOMENT Y mx
nj-ny THRU nj FORCE Z fz MOMENT X mx MOMENT Y mx
2+ny THRU nj-2*ny-1 STEP ny+1 FORCE Z fz MOMENT X mx MOMENT Y mx
2+2*ny THRU nj-ny-1 STEP ny+1 FORCE Z fz MOMENT X mx MOMENT Y mx
MEMBER INCIDENCES
* Member incidences are generated by copying a row of elements
* and incrementing the member and joint numbers: 'm' and 'j'.
g=e/(2*(1+nu)) m=-nx*6 j=-1 ;:15 ;m=m+nx*6
j=j+1 ;m+1 THRU m+nx*6 ELEMENT j+ny+3,j+ny+2,j+1,j+2 BUMP ny+1
IF j<ny-1 GOTO 15 ;CONSTANTS E e ALL G g ALL
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t
LOADING CASE 1 ;JOINT LOADS ;xinc=(xb-xa)/10 yinc=(yb-ya)/10
p=xinc*yinc*q/100 i=0 nl=0 y=ya-yinc/2 ;:20 ;i=i+1 y=y+yinc m=0
x=xa-xinc/2 ;:22 ;m=m+1 nl=nl+1 x=x+xinc x(nl)=x y(nl)=y p(nl)=p
IF m<10 GOTO 22 ;IF i<10 GOTO 20 ;i=0 ;:25 ;i=i+1
0 FORCE Z p(i) x(i) y(i) ;IF i<nl GOTO 25 ;SOLVE ;status=1
gtot=0 nur=0 a=xe*nx b=ye*ny d=e*t^3/(12*(1-nu^2)) j=INT(nj/2)+1
node=ARR(8,j,2) r=3*node dcf=ARR(6,r,1) x=ARR(8,node,3)
y=ARR(8,node,4) ;i=0 w=0 ;:35 ;i=i+1 p=p(i) x'=x(i) y'=y(i)
pc=4*p/(PI^4*a*b*d) m=0 ;:40 ;m=m+1 n=0 ;:50 ;n=n+1
num=SIN(m*PI*x'/a)*SIN(n*PI*y'/b)*SIN(m*PI*x/a)*SIN(n*PI*y/b)
w=w+pc*num/(m^2/a^2+n^2/b^2)^2 ;IF n<11 GOTO 50 ;IF m<11 GOTO 40
IF i<nl GOTO 35 ;d1=dcf d2=w
```

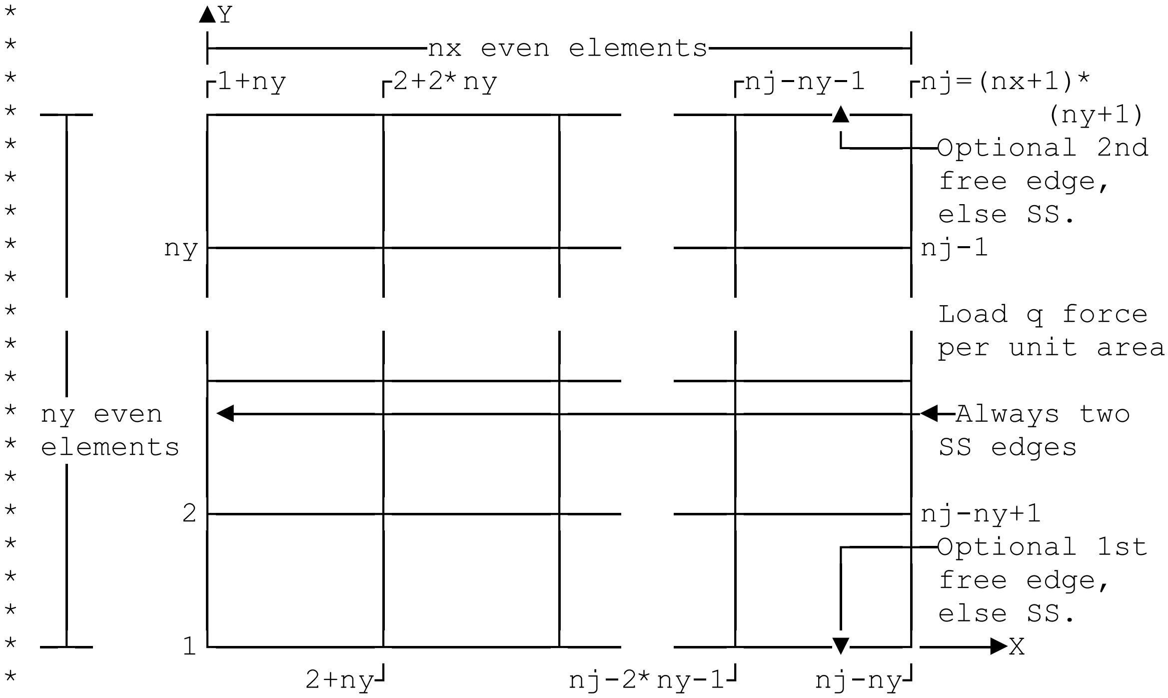
```
#vmper.ndf !Compute percentage difference & any message.
*                  NL-STRESS          Navier               %age
*                  Analysis           Solution             diff.
* Defln at centre: +dcf               +w                   $ok
fnm=$(vm602.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE FLOOR SLAB WITH SUBSTANTIAL HOLE IN ANY POSITION AND
TITLE IN ANY COMBINATION OF PINNED, FIXED OR FREE EDGES;
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 29.08.05 ;METHOD ELASTIC JOINTS ;REFNO VM605
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE GRID run=0
TABULATE DISPLACEMENTS REACTIONS ;*/17
*          │Y                                      dx─┼──┤
* ──┬─     ┌──────────────────────────────────────────────┐
*   │      │                   North                      │   dy
* nya      │                                         ┼───┼│   ─┬─
*   │      │                                         ┼───┼│   ─┴─
* ──┼─     │               ┌─────────────┐                │
*   │      │               │             │                │
* nyh      │West           │             │            East│
*   │      │               │             │                │
* ──┼─     │               └─────────────┘                │
*   │      ├──────nxb──────┼─────nxh─────┼──────nxa───────┤
* nyb      │                                              │
*   │      │                   South                      │
* ──┴─     └──────────────────────────────────────────────┘  ────X
nxb=2  nxh=2   nxa=2     ! No. elem X dir before, in & after hole.
nyb=2  nyh=2   nya=2     ! ........ Y ...........................
dx=0.220/2   dy=0.220/2      ! Size of elements in X & Y directions.
e=30E6/3   nu=0.2        ! Young's modulus; Poisson's ratio.
sn=-1 ss=-1 se=-1 sw=-1  ! Supports N,S,E,W. 0=free, -1=vert prop.
mn=-1 ms=-1 me=-1 mw=-1  ! Moment fixity N,S,E,W; 0=free -1=fixed.
t=0.265  q=-120 houdl=q  ! Slab thickness & udl; hole udl, 0=none.
#cc924.stk !Import verification data from cc924.stk if available.
nxe=nxb+nxh+nxa nye=nyb+nyh+nya neh=nxh*nyh njh=(nxh-1)*(nyh-1)
nxy=nxe+nye nyj=nye+1 nj=(nxe+1)*nyj-njh nm=6*(nxe*nye-neh) hole=1
IF nxh=1 AND nyh=1 AND houdl=q THEN nm=nm+6 hole=0 !Hole omit.
NUMBER OF JOINTS nj ;NUMBER OF MEMBERS nm ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 3
JOINT COORDINATES !Special if hole is 1 element in either dirn.
IF nxh=1 OR nyh=1 ;i=0 j=1-nyj x=-dx ;:10 ;i=i+1 j=j+nyj x=x+dx
j THRU j+nye X x Y 0 YL nye*dy ;IF i<=nxb+nxh+nxa GOTO 10 ;ENDIF
IF nxh>1 AND nyh>1 ;! Firstly coors to left of hole. ;i=0 j=1-nyj
x=-dx ;:20 ;i=i+1 j=j+nyj x=x+dx j THRU j+nye X x Y 0 YL nye*dy
IF i<nxb+1 GOTO 20 ;! Secondly coordinates below and above hole.
i=0 inc=nyj-(nyh-1) j=j+nye+1-inc ;:30 ;i=i+1 j=j+inc x=x+dx
j THRU j+nyb X x Y 0 YL nyb*dy
j+nyb+1 THRU j+nyb+1+nya X x Y (nyb+nyh)*dy YL nye*dy
IF i<nxh-1 GOTO 30 ;! Thirdly coordinates to right of hole.
i=0 j=j+nyb+1+nya+1-nyj ;:40 ;i=i+1 j=j+nyj x=x+dx
j THRU j+nye X x Y 0 YL nye*dy ;IF i<nxa+1 GOTO 40 ;ENDIF
JOINT RELEASES !Props around edge of floor slab.
! Fix corner supports propped/unpropped as special cases.
ssw=-1 sse=-1 snw=-1 sne=-1
IF sw=0 AND ss=0 THEN ssw=0 ;IF ss=0 AND se=0 THEN sse=0
IF se=0 AND sn=0 THEN sne=0 ;IF sn=0 AND sw=0 THEN snw=0
1 FORCE Z ssw ;nyj FORCE Z snw ;nj-nye FORCE Z sse ;nj FORCE Z sne
! Fix supports along Y axis and parallel to Y axis.
IF sw=-1 THEN 2 THRU nyj-1 FORCE Z sw
IF mw=-1 THEN 2 THRU nyj-1 MOMENT Y mw
IF mw=-1 THEN 1 THRU nyj STEP nye MOMENT Y mw
IF se=-1 THEN nj-nye+1 THRU nj-1 FORCE Z se
IF me=-1 THEN nj-nye+1 THRU nj-1 MOMENT Y me
IF me=-1 THEN nj-nye THRU nj STEP nye MOMENT Y me
! Special for all joints if hole is 1 element in either direction.
i=0 j=1 ;IF nxh=1 OR nyh=1 ;i=1 j=1 ;:50 ;i=i+1 j=j+nyj
IF ss=-1 THEN j FORCE Z ss ;IF ms=-1 THEN j MOMENT X ms
```

```
IF sn=-1 THEN j+nye FORCE Z sn ;IF mn=-1 THEN j+nye MOMENT X mn
IF i<nxb+nxh+nxa GOTO 50 ;ENDIF ;IF nxh>1 AND nyh>1
! Firstly supports to left of hole. ;i=1 j=1 ;:60 ;i=i+1 j=j+nyj
j FORCE Z ss MOMENT X ms ;j+nye FORCE Z sn MOMENT X mn
IF i<nxb+1 GOTO 60 ;! Secondly supports below and above hole.
i=0 inc=nyj-(nyh-1) j=j+nye+1-inc ;:70 ;i=i+1 j=j+inc
j FORCE Z ss MOMENT X ms ;j+nyb+1+nya FORCE Z sn MOMENT X mn
IF i<nxh-1 GOTO 70 ;! Thirdly supports to right of hole.
i=0 j=j+nyb+1+nya+1-nyj ;:80 ;i=i+1 j=j+nyj
j FORCE Z ss MOMENT X ms ;j+nye FORCE Z sn MOMENT X mn
IF i<nxa GOTO 80 ;ENDIF ;MEMBER INCIDENCES
! Firstly member incidences to left of hole. ;i=0 m=-nye*6 j=1-nyj
:90 ;i=i+1 m=m+nye*6 j=j+nyj
m+1 THRU m+nye*6 ELEMENT j+nyj+1,j+nyj,j,j+1 ;IF i<nxb GOTO 90
! Secondly member incidences below and above hole in figure. ;i=0
m'=6*(nye-nyh*hole) m=m+nye*6-m' inc=nyj-(nyh*hole-1)
s=nyj*(nxb+1)+1-inc ;:100 ;i=i+1 m=m+m' s=s+inc k=s-inc
IF i=1 THEN k=s-nyj ;m+1 THRU m+6*nyb ELEMENT s+1,s,k,k+1
ma=m+6*nyb s'=s+nyb+1 k'=s'-inc ;IF i=nxh THEN s'=s+nyb+nyh*hole
ma+1 THRU ma+6*(nya+1-hole) ELEMENT s'+1,s',k',k'+1
IF i<nxh GOTO 100 ;! Thirdly to right of hole. ;i=0 m=m+m'-nye*6
j=nj-(nxa+2)*nyj+1 ;:110 ;i=i+1 m=m+nye*6 j=j+nyj
m+1 THRU m+nye*6 ELEMENT j+nyj+1,j+nyj,j,j+1 ;IF i<nxa GOTO 110
CONSTANTS E e 1 THRU nm G g=e/(2*(1+nu)) g 1 THRU nm
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t
LOADING AREA LOAD ON SLAB ;nstr=nxe+nye ;IF nstr>64 THEN nstr=64
JOINT LOADS ;1 AREA nj FORCE Z q nstr ;! Next udl over hole.
j=nyj*nxb+nyb+1 k=nj-nyj*nxa-nya j AREA k FORCE Z houdl-q nstr
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn MOMENT X jn Y jn FORCE Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn MOMENT X jn' Y jn' FORCE Z jn' ;IF jn<nj GOTO 19 ;SOLVE
*/18
*           |Y
* ---       ┌────────────────────────────────────────────────┐
*  |        +nyj            +j+nyh+nya    +k+nya          +nj
* +nya*dy   |                                              |
*  |        |               +j+nyh        +k               |
* -+-       |               ┌─────────────┐                |
*  |        |               |             |                |
*  |        |               |             |                |
* +nyh*dy   |               |             |                |
*  |        |               |             |                |
* -+-       |               └─────────────┘                |
*  |        |               +j            +k-nyh           |
* +nyb*dy   |                                              |
*  |        1               +j-nyb        +k-nyh-nyb      +nj-nyj+1
* ---       └────────────────────────────────────────────────┘ ───X
*                +nxb*dx          +nxh*dx        +nxa*dx
*           |─────────────────|─────────────|──────────────|
*                   Key joint numbers and dimensions
va1=VEC(0)*nm vc1=VEC(0)*nm jld=q*dx*dy vl1=VEC(jld)*nj
nl9=j ch9=k dirn=3 ;a=(nxb+nxh+nxa)*dx !Length in X direction
b=(nyb+nyh+nya)*dy !Length in Y direction ;supfix=sn+ss+se+sw
njc=INT(nxe/2)*nyj+INT((nyj+1)/2) momfix=mn+ms+me+mw mns=mn+ms
mew=me+mw nn=ARR(8,njc,2) nr=nn*3 nl10=ARR(6,nr,1) ch10=0
IF nxh=1 AND nyh=1 AND a=b AND supfix=-4 AND momfix=-4
! For built-in square plate, defln for load q from Timoshenko
ch10=0.00126*q*a^4*12*(1-nu^2)/(e*t^3) ;ENDIF
IF nxh=1 AND nyh=1 AND a=b AND supfix=-4 AND momfix=0
! Simply supported square plate, defln for load q from Timoshenko
ch10=0.00406*q*a^4*12*(1-nu^2)/(e*t^3) ;ENDIF
```

```
status=0 ;IF mns=-2 AND mew=0 OR mns=0 AND mew=-2 THEN status=1
IF nxh=1 AND nyh=1 AND a=b AND supfix=-4 AND status=1
! For simply supported square plate, two opposite edges simply
! supp. and other two built-in, defln for load q from Timoshenko
ch10=0.00192*q*a^4*12*(1-nu^2)/(e*t^3) ;ENDIF
IF nxh=1 AND nyh=1 AND a=b AND supfix=-4 AND momfix=-1
! For simply supported square plate, three edges simply supp.
! and other edge built-in, defln for load q from Timoshenko
ch10=0.0028*q*a^4*12*(1-nu^2)/(e*t^3) ;ENDIF
#vmecg.ndf !Equilibrium/energy & compatibility checks for grid.
IF ch10<>0
* Central defln. cf. Timoshenko    +nl10        +ch10        $ok
ENDIF
fnm=$(vm605.stk) ;#vmres.ndf !Conclude results.
mjn=nyj+2 lcn=1 tot=3 drn=3 ;#vmtes.ndf
< ;FINISH
```

```
TITLE FLAT PLATE IN FLEXURE HAVING SIMPLY SUPPORTED EDGES
TITLE AND FROM ZERO TO TWO FREE EDGES, SUBJECTED TO UDL.
TITLE RESULTS COMPARE THE MATRIX STIFFNESS METHOD WITH
TITLE FINITE DIFFERENCE AND/OR EXACT SOLUTION ;MADEBY DWB
DATE 26.12.04 ;PRINT DATA RESULTS FROM 1 ;REFNO VM610
METHOD ELASTIC JOINTS ;TYPE PLANE GRID run=0
TABULATE DISPLACEMENTS REACTIONS ;*/21
```




```
xe=1  ye=1  t=.254  ! Element sizes in X & Y dirn. & thickness.
nx=8  ny=8  q=18    ! Number of elements in X & Y directions, UDL.
e=30E6/3    nu=.3   ! Young's modulus & Poisson's ratio.
nss=3               ! No. of SS edges (2-4); if not SS then free.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nxj=nx+1 nyj=ny+1 nj=nxj*nyj nj
NUMBER OF MEMBERS nm=6*nx*ny nm ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 1
JOINT COORDINATES ;i=0 j=-ny x=-xe ;:10 ;i=i+1 j=j+ny+1 x=x+xe
j THRU j+ny X x Y 0 YL ny*ye ;IF i<nx+1 GOTO 10 ;JOINT RELEASES
fz=-1 mx=0 ;1 THRU 1+ny FORCE Z fz MOMENT X mx MOMENT Y mx
nj-ny THRU nj FORCE Z fz MOMENT X mx MOMENT Y mx
IF nss=2 THEN fz=0 mx=0 !If nss=2 then second free edge.
2+2*ny THRU nj-ny-1 STEP ny+1 FORCE Z fz MOMENT X mx MOMENT Y mx
IF nss<4 THEN fz=0 mx=0 !If nss=2 or 3 then first free edge.
2+ny THRU nj-2*ny-1 STEP ny+1 FORCE Z fz MOMENT X mx MOMENT Y mx
MEMBER INCIDENCES
* Member incidences are generated by copying a row of elements
* and incrementing the member and joint numbers: 'm' and 'j'.
m=-nx*6 j=-1 ;:20 ;m=m+nx*6 j=j+1
m+1 THRU m+nx*6 ELEMENT j+ny+3,j+ny+2,j+1,j+2 BUMP ny+1
g=e/(2*(1+nu)) ;IF j<ny-1 GOTO 20
CONSTANTS E e ALL G g ALL ;MEMB PROPERTIES ;1 THRU nm ELEMENT T t
LOADING CASE 1 ;JOINT LOADS ;q=ABS(q) 1 AREA nj FORCE Z q nx
SOLVE ;status=1 gtot=0 nur=0 ;IF nss=4 GOTO 2000
IF nss=2 GOTO 1000 ;fmx=0.0374*q*(nx*xe)^2
fmy=0.0777*q*(nx*xe)^2 edc=.1404*q*(nx*xe)^4/(e*t^3)
emc=0.112*q*(nx*xe)^2 emx=0.039*q*(nx*xe)^2 emy=0.080*q*(nx*xe)^2
fdc=0.1429*q*(nx*xe)^4/(e*t^3) fmc=0.1084*q*(nx*xe)^2
jcfe=(ny+1)*nx/2+1 node=ARR(8,jcfe,2) r=3*node dcf=ARR(6,r,1)
jlt=jcfe-nyj node=ARR(8,jlt,2) r=3*node dlt=ARR(6,r,1)
jrt=jcfe+nyj node=ARR(8,jrt,2) r=3*node drt=ARR(6,r,1) ;*/10
```

```
* For free edge, compute moment about Y from M/I=E/R, using edge
* joint numbers +jlt +jcfe +jrt having corresponding deflections
* +dlt +dcf +drt and rearranging (2r'-del)*del=xe^2 where
* +del=dcf-dlt then +r'=(xe^2/del+del)/2 and
* corresponding moment is +mce=e*t^3/12/r' ;*/10
* Extract joint numbers at and around centre joint for points 1-9.
j5=(nj-1)/2+1 j4=j5-nyj j6=j5+nyj j1=j4+1 j2=j5+1 j3=j6+1 j7=j4-1
j8=j5-1 j9=j6-1 ;* Read displ w(1)-w(9) from arrays. ;i=0 ;:80
i=i+1 j=j(i) n=ARR(8,j,2) r=3*n w(i)=ARR(6,r,1) ;IF i<9 GOTO 80
* 1─────2─────3 ─┬─ +a=xe +b=xe +c=ye +d=ye
* │     │     │  d  Plate constant +k=e*t^3/(12*(1-nu^2))
* 4─────5─────6 ─┼─ Curvature δ²w/δx²
* Y     │     │  c  +dwx=2*(a*(w6-w5)+b*(w4-w5))/(a*b*(a+b))
* 7──X──8─────9 ─┴─ Curvature δ²w/δy²
* ├──a──┼──b──┤  +dwy=2*(c*(w2-w5)+d*(w8-w5))/(c*d*(c+d))
* Numerator      +n=(w3+w5-w2-w6)/b^2/d^2+(w2+w4-w1-w5)/a^2/d^2
*                +n=n+(w5+w7-w4-w8)/a^2/c^2+(w6+w8-w5-w9)/b^2/c^2
* Curvature δ²w/δxδy +dwxy=n/(1/b/d+1/a/d+1/a/c+1/b/c)
* Moments at point 5 +mx=k*(dwx+nu*dwy)
* +my=k*(dwy+nu*dwx) +mxy=k*(1-nu)*dwxy
* Principal moments  +m1=-((mx+my)/2+SQR(0.25*(mx-my)^2+mxy^2))
*                    +m2=-((mx+my)/2-SQR(0.25*(mx-my)^2+mxy^2))
IF ABS(my-mx)<1E-12 ;* At angles          +thet1=45 ;ENDIF
IF ABS(my-mx)>=1E-12
* At angles          +thet1=0.5*DEG(ATN(2*mxy/(my-mx))) ;ENDIF
*                    +thet2=thet1+90
* Principal stresses +f1=6*m1/t^2 +f2=6*m2/t^2 ;*/10
*                          NL-STRESS   Finite (1)  Exact (2)   %age
*                          stiffness   difference  solution    diff
d1=dcf d2=edc
#vmper.ndf !Compute percentage difference & any message.
* Defln centre free edge   +dcf        +fdc        +edc        $ok
d1=mce d2=emc
#vmper.ndf !Compute percentage difference & any message.
* Moment centre free edge  +mce        +fmc        +emc        $ok
d1=m1 d2=emx
#vmper.ndf !Compute percentage difference & any message.
* M. abt. X at plate cent. +m1         +fmx        +emx        $ok
d1=m2 d2=emy
#vmper.ndf !Compute percentage difference & any message.
* M. abt. Y at plate cent. +m2         +fmy        +emy        $ok
* (1) Ghali & Neville, Structural Analysis, Spon, London, 1997.
* (2) JM Gere, Moment Distribution, Van Nostrand, New York, 1963.
GOTO 2000 ;:1000 ;* Plate is 1 way spanning & simply supported.
* Cent. joint on X axis & other edge +jf=nyj*(nx/2)+1 +jl=jf+ny
* Read displ along centre from arrays. ;j=jf-1 sigd=0 ;:1080
j=j+1 n=ARR(8,j,2) r=3*n w=ARR(6,r,1) sigd=sigd+w
* Displacement at joint +j = +w ;IF j<jl GOTO 1080
* Average disp   +avd=sigd/nyj
* Bending defln  +dcb=5*q*(xe*nx)^4/(384*e*t^3/12)
* Shear defln    +dcs=(1/8)*(6/5)*q*(xe*nx)^2/(g*t*5/6)
* Total defln    +dct=dcb+dcs
d1=avd d2=dct
#vmper.ndf !Compute percentage difference & any message.
*                                 NL-STRESS  Simple beam theory
* For Poisson's ratio= +nu        stiffness  incl. shear defln
* Average defln along plate centre   +avd        +dct          $ok
:2000 ;fnm=$(vm610.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

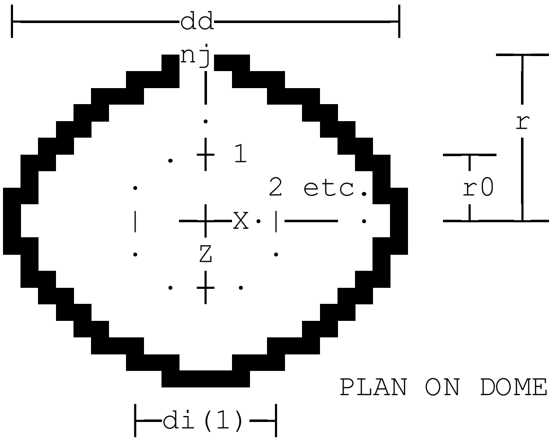
```
TITLE PLATE IN EXTENSION WITH HOLE FOR WINDOW WITH LINTEL
TITLE OVER MODELLING AN INFILL PANEL WITHIN A STEEL FRAME,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 02.09.05 ;METHOD ELASTIC JOINTS ;REFNO VM618
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE FRAME run=0
TABULATE DISPLACEMENTS REACTIONS ;*/15
*       ►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►►► udlx
*      Y│▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼ udly -ve
* ─┬─   ╟─────────────────────────────────────────────╢     down
*  │    ║      └beam         North            ─┬─ ┼──┼
* nya   ║                                     ─┼─ ┼──┼
*  │    ├column             ┌lintel          dy
* ─┼─   ║                 ═══════════════        dx┼──┤ ├column
*  │    ║                 │             │             ║
* nyh   ║West             │             │        East ║
*  │    ║                 │             │             ║  t= wall
* ─┼─   ║                 └─────────────┘             ║  thickness
*  │    ├──────nxb────────┼─────nxh─────┼──────nxa────┤
* nyb   ║                                             ║
*  │    ║      ┌beam         South                    ║
* ─┴─   ╟─────────────────────────────────────────────╢ ───X
nxb=1  nxh=3  nxa=2     ! No. elem. X dir before, in & after hole.
nyb=2  nyh=2  nya=2     ! ......... Y ...........................
dx=0.225 dy=0.220  t=0.265  ! Size of elements and wall thickness.
e=10E6    nu=0.2     ! Young's modulus & Poisson's ratio for wall.
es=205E6  nus=0.3    ! Young's & Poisson's ratio for frame/lintel.
udlx=200  udly=-200  ! Udl's along and normal to top of wall.
ab=32.0E-4 ayb=15.43E-4 ib=3420E-8  ! Area, shear area, I beams.
ac=38.3E-4 ayc=10.24E-4 ic=1750E-8  ! Area, shear area, I cols.
al=19.6E-4 ayl=12E-4  il=852.2E-8   ! Area, shear area, I lintel.
#cc924.stk !Import verification data from cc924.stk if available.
nxe=nxb+nxh+nxa nye=nyb+nyh+nya neh=nxh*nyh njh=(nxh-1)*(nyh-1)
nxy=nxe+nye nyj=nye+1 nj=(nxe+1)*nyj-njh ;NUMBER OF JOINTS nj
nm=6*(nxe*nye-neh) !Number of members in masonry.
nme=2*(nxe+nye)+nxh !Number of edge framing members.
NUMBER OF MEMBERS nmt=nm+nme nmt
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 3
JOINT COORDINATES !Special if hole is 1 element in either dirn.
IF nxh=1 OR nyh=1 ;i=0 j=1-nyj x=-dx ;:10 ;i=i+1 j=j+nyj x=x+dx
j THRU j+nye X x Y 0 YL nye*dy ;IF i<=nxb+nxh+nxa GOTO 10 ;ENDIF
IF nxh>1 AND nyh>1 ;! Firstly coors to left of hole. ;i=0 j=1-nyj
x=-dx ;:20 ;i=i+1 j=j+nyj x=x+dx j THRU j+nye X x Y 0 YL nye*dy
IF i<nxb+1 GOTO 20 ;! Secondly coordinates below and above hole.
i=0 inc=nyj-(nyh-1) j=j+nye+1-inc ;:30 ;i=i+1 j=j+inc x=x+dx
j THRU j+nyb X x Y 0 YL nyb*dy
j+nyb+1 THRU j+nyb+1+nya X x Y (nyb+nyh)*dy YL nye*dy
IF i<nxh-1 GOTO 30 ;! Thirdly coordinates to right of hole.
i=0 j=j+nyb+1+nya+1-nyj ;:40 ;i=i+1 j=j+nyj x=x+dx
j THRU j+nye X x Y 0 YL nye*dy ;IF i<nxa+1 GOTO 40 ;ENDIF
JOINT RELEASES !Props along bottom of wall. ;1 FORCE X -1 Y -1
nj-nye FORCE Y -1 ;MEMBER INCIDENCES ;! Firstly left of hole.
i=0 m=-nye*6 j=1-nyj ;:90 ;i=i+1 m=m+nye*6 j=j+nyj
m+1 THRU m+nye*6 ELEMENT j+nyj+1,j+nyj,j,j+1 ;IF i<nxb GOTO 90
! Secondly member incidences below and above hole in figure. ;i=0
m'=6*(nye-nyh) m=m+nye*6-m' inc=nyj-(nyh-1)
s=nyj*(nxb+1)+1-inc ;:100 ;i=i+1 m=m+m' s=s+inc k=s-inc
IF i=1 THEN k=s-nyj ;m+1 THRU m+6*nyb ELEMENT s+1,s,k,k+1
ma=m+6*nyb s'=s+nyb+1 k'=s'-inc ;IF i=nxh THEN s'=s+nyb+nyh
ma+1 THRU ma+6*nya ELEMENT s'+1,s',k',k'+1
IF i<nxh GOTO 100 ;! Thirdly to right of hole. ;i=0 m=m+m'-nye*6
j=nj-(nxa+2)*nyj+1 ;:110 ;i=i+1 m=m+nye*6 j=j+nyj
```

```
m+1 THRU m+nye*6 ELEMENT j+nyj+1,j+nyj,j,j+1 ;IF i<nxa GOTO 110
! First edge columns ;nm+1 THRU nm+nye RANGE 1,2 nyj-1,nyj
nm+nye+1 THRU nm+2*nye RANGE nj-nye,nj-nye+1 nj-1,nj
IF nxh=1 OR nyh=1 ;! Next bottom & top beams ;m=nm+2*nye j=nj-nye
m+1 THRU m+nxe RANGE 1,1+nyj j-nyj,j m=m+nxe
m+1 THRU m+nxe RANGE nyj,2*nyj nj-nyj,nj ;! Finally lintel
j=(nxb+1)*nyj-nya k=j+nyj*nxh m=m+nxe
m+1 THRU m+nxh RANGE j,j+nyj k-nyj,k ;ENDIF ;IF nxh>1 AND nyh>1
! Next bottom beam ;m=nm+2*nye j=1+(nxb+1)*nyj k=j+(nxh-1)*inc
m+1 THRU m+nxb+1 RANGE 1,1+nyj 1+nxb*nyj,j ;m=m+nxb+1
IF nxh>1 THEN m+1 THRU m+nxh-1 RANGE j,j+inc k-inc,k
m=m+nxh-1 j=nj-nye ;m+1 THRU m+nxa RANGE k,k+nyj j-nyj,j
! Next top beam ;m=nm+2*nye+nxe j=(nxb+1)*nyj
m+1 THRU m+nxb RANGE nyj,2*nyj j-nyj,j ;m=m+nxb k=j+inc*(nxh-1)
m+1 THRU m+nxh-1 RANGE j,j+inc k-inc,k ;m=m+nxh-1
m+1 THRU m+nxa+1 RANGE k,k+nyj nj-nyj,nj
! Next lintel. ;m=nm+2*(nxe+nye) j=nyj*(nxb+1)-nya k=j+(nxh-1)*inc
IF nxh>1 THEN m+1 THRU m+nxh-1 RANGE j,j+inc k-inc,k
IF nxh>1 THEN m+nxh k k+nyj ;IF nxh=1 THEN m+1 j j+nyj ;ENDIF
CONSTANTS E e 1 THRU nm G g=e/(2*(1+nu)) g 1 THRU nm
CONSTANTS E es ALL G gs=es/(2*(1+nus)) gs ALL
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t ;m=nm+2*nye j=m+2*nxe
nm+1 THRU m AX ac AY ayc IZ ic ;m+1 THRU j AX ab AY ayb IZ ib
j+1 THRU j+nxh AX al AY ayl IZ il
LOADING SHEAR AND VERTICAL DISTRIBUTED LOAD AT TOP OF WALL
MEMBER LOADS ;! On top beam ;m=nm+2*nye+nxe
m+1 THRU m+nxb FORCE X UNIFORM W udlx
m+1 THRU m+nxb FORCE Y UNIFORM W udly ;m=m+nxb
m+1 THRU m+nxh FORCE X UNIFORM W udlx
m+1 THRU m+nxh FORCE Y UNIFORM W udly ;m=m+nxh
m+1 THRU m+nxa FORCE X UNIFORM W udlx
m+1 THRU m+nxa FORCE Y UNIFORM W udly
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;jn=0 ;:18 ;jn=jn+1
jn FORCE X jn Y jn MOMENT Z jn ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn' Y jn' MOMENT Z jn' ;IF jn<nj GOTO 19 ;SOLVE
j=nyj*nxb+nyb+1 k=nj-nyj*nxa-nya ;*/18
*           |Y
*  ___      ______________________________________________
*   |      +nyj            +j+nyh+nya   +k+nya         +nj
*  +nya*dy |                                              |
*   |      |               +j+nyh       +k                |
*  _|_     |                                              |
*   |      |               ______________                 |
*   |      |              |              |                |
*  +nyh*dy |              |              |                |
*   |      |              |______________|                |
*  _|_     |                                              |
*   |      |               +j           +k-nyh            |
*  +nyb*dy |                                              |
*   |      1               +j-nyb       +k-nyh-nyb     +nj-nyj+1
*  _|_     |______________________________________________|  ____X
*                +nxb*dx      +nxh*dx        +nxa*dx
*          |----------------|--------------|----------------|
*                    Key joint numbers and dimensions
m=nm+2*nye+nxe va1=VEC(0)*m vc1=VEC(0)*m m=m+1 va(m)=VEC(udlx)*nxe
vc(m)=VEC(udly)*m m=m+nxe va(m)=VEC(0)*nxh vc(m)=VEC(0)*nxh
vjl1=VEC(0)*nj hjl1=VEC(0)*nj ch9=1 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm618.stk) ;#vmres.ndf !Conclude results.
mjn=nyj+2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CIRCULAR BALCONIES MODELLED AS CIRCULAR PLATE WITH
TITLE CIRCULAR HOLE AND SIMPLY SUPPORTED OUTSIDE EDGE.
TITLE RESULTS COMPARE THE MATRIX STIFFNESS METHOD WITH
TITLE CLASSICAL SOLUTION & ROARK'S FORMULA ;MADEBY DWB
DATE 01.01.05 ;METHOD ELASTIC ;REFNO VM620
TABULATE DISPLACEMENTS ;PRINT DATA, RESULTS FROM 1 ;*/14
type=1      ! 1=grid, 2=space frame         ├─d(1)─┼internal diam.
r1=5        ! Outside radius                    nj                ┬
r2=2        ! Inside radius                     │                 │
t=0.2       ! Plate thickness                   ·                 r1
nsg=24      ! No. of segments                 · ┼ 1          ┬    │
e=30E6/3    ! Young's modulus               ·   Y   ·        r2   │
nu=.2       ! Poisson's ratio               │  ─┼─X·│──  ·   ┴    ┴
q=-9.6      ! Distributed loading           ·       ·
* For accuracy, elements should               · ┼ ·
* be as square as possible, so                            PLAN ON
* make rings of joints wider as                            PLATE
* they approach the outside rim.
nrp=1 run=0  ! Initialise counters. ├──────────dp───────────┼diam.
#cc924.stk !Import set of parameters if available from cc924.stk.
dp=2*r1 d1=2*r2 g=e/(2*(1+nu))   !Derive 'g' from constants.
:20 ;es=PI*d(nrp)/nsg i=nrp nrp=nrp+1 d(nrp)=d(i)+2*es
IF d(nrp)<=dp GOTO 20
i=1 scale=(dp-d(1))/(d(nrp)-d(1))  !Scale down to fix overshoot.
:60 ;i=i+1 ;IF i<nrp THEN d(i)=d(1)+(d(i)-d(1))*scale
IF i>=nrp THEN d(i)=dp ;IF i<nrp GOTO 60
IF type=1 THEN TYPE PLANE GRID ;IF type=2 THEN TYPE SPACE FRAME
NUMBER OF JOINTS nj=nrp*nsg nj
NUMBER OF MEMBERS nel=nsg*(nrp-1) nm=nel*6 nm
NUMBER OF SUPPORTS 0    !Supports given in joint releases table.
NUMBER OF LOADINGS 1 ;JOINT COORDINATES ;*/5
*                                  C.│L.
* ╞═════════════════════                 ═══════════════════════╡
*   ELEVATION ON PLATE ├────────d(1)───────┼─────(dp-d(1))/2─────┤
* ├────────────────────────────dp────────────────────────────────┤
theta=PI/2              !Start position - radians
alpha=2*PI/nsg          !Angle subtended by segments - radians
nr=0 j=0                !Initialise counters and constants
REPEAT ;IF j-nsg*INT(j/nsg)=0 THEN np=nr nr=nr+1 ns=0
! ns is counter for number of segments ;ns=ns+1 j=j+1
x=0.5*d(nr)*COS(theta-ns*alpha) ;IF ABS(x)<1E-6 THEN x=0
y=-0.5*d(nr)*(-SIN(theta-ns*alpha)) ;IF ABS(y)<1E-6 THEN y=0
IF type=1 THEN j x y ;IF type=2 THEN j x 0 y ;UNTIL j=nj
ENDREPEAT
JOINT RELEASES ;IF type=1 THEN nj-nsg+1 THRU nj FORCE Z -1
IF type=2 THEN nj-nsg+1 THRU nj FORCE Y -1
IF type=2 THEN 1 FORCE X -1 Z -1 MOMENT Y -1  !Stop sliding/rotn.
MEMBER INCIDENCES ;nr=0 m=-5  !Init. counter for rings & members.
:100 ;nr=nr+1 i=0 ;:200 ;i=i+1 m=m+6
j1=nr*nsg+i j2=j1-nsg            !Set first & second joint Nos.
IF i=1 THEN j3=j1-1 j4=j1+nsg-1  !Set 3rd & 4th if 1st element.
IF i>1 THEN j3=j2-1 j4=j1-1      !Set 3rd & 4th if > 1st element.
m THRU m+5 ELEMENT j1,j2,j3,j4 ;IF i<nsg GOTO 200
IF nr<nrp-1 GOTO 100 ;CONSTANTS E e ALL G g ALL DENSITY -q/t ALL
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t
LOADING CASE 1 - UDL ON PLATE ;MEMBER SELF WEIGHTS ;1 THRU nm 1.0
SOLVE ;status=1 gtot=0 nur=0
* Classical solution ;* Plate stiffness +d=e*t^3/(12*(1-nu^2))
* Constants: +c1=-q*r2^2/(2*d) +cp=1+nu +cm=1-nu
* +c21=-q*r1^2*(3+nu)/(8*d*cp)-c1*LOG(r1) ;* +c22=-(c1/2)*cm/cp
* +c23=q*r2^4*(3+nu)/(8*d*r1^2*cp) ;* +c24=(r2^2/r1^2)*c1*LOG(r2)
* +c25=c1*r2^2*cm/(2*r1^2*cp) ;* +c26=1-r2^2/r1^2
```

```
* +c2=(c21+c22+c23+c24+c25)/c26 ;* +c31=(q*r2^4/(16*d))*(3+nu)/cm
* +c32=r2^2*(c1/2)*cp*LOG(r2)/cm
* +c33=c1*r2^2/4+(c2*r2^2/2)*(cp/cm)
* +c3=c31+c32+c33 ;* +c41=-q*r1^4/(64*d)-c1*r1^2*(LOG(r1)-1)/4
* +c42=-c2*r1^2/4-c3*LOG(r1) ;* +c4=c41+c42
* Substituting +r=r2 in general equation:
* +c=q*r^4/(64*d)+c1*r^2*(LOG(r)-1)/4+c2*r^2/4+c3*LOG(r)+c4
* Formulas for Stress and Strain fourth edition, Roark
* +m=1/nu +w=q +a=r1 +b=r2 ;* +con=3*w*(m^2-1)/(2*m^2*e*t^3)
* +k1=a^4*(5*m+1)/(8*(m+1)) ;* +k2=b^4*(7*m+3)/(8*(m+1))
* +k3=a^2*b^2*(3*m+1)/(2*(m+1))
* +k4=a^2*b^2*(3*m+1)*LOG(a/b)/(2*(m-1))
* +k5=2*a^2*b^4*(m+1)*(LOG(a/b))^2/((a^2-b^2)*(m-1))
* +rok=con*(k1+k2-k3+k4-k5) ;* Get joint 1 displ. from arrays.
IF type=1 THEN nls=ARR(6,3,1) ;IF type=2 THEN nls=ARR(6,2,1)
d1=nls d2=c
#vmper.ndf !Compute percentage difference & any message.
*                NL-STRESS     Classical     Roark's
*                stiffness     solution      formula
* Max. defln.    +nls          +c            +rok          $ok
fnm=$(vm620.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE SPHERICAL DOME WITH CONCENTRATED LOAD AT POLE.
TITLE VERTICAL SUPPORT, EDGE NEITHER HELD NOR FIXED.
TITLE RESULTS COMPARE THE MATRIX STIFFNESS METHOD WITH
TITLE ROARK'S FORMULAS FOR STRESS AND STRAIN, FIFTH EDTN.
MADEBY DWB ;DATE 01.03.05 ;TYPE SPACE FRAME run=0 ;REFNO VM630
METHOD ELASTIC ;TABULATE DISPLACEMENTS REACTIONS
PRINT DATA, RESULTS FROM 1
```




```
hd=4.8   ! Height of dome.
r2=6.4   ! External rad. of sphere.
t=0.2    ! Dome thickness.
r0=0.1   ! Radius of pole ring.
nsg=32   ! No. of segments.
e=28E6   ! Young's modulus.
nu=.3    ! Poisson's ratio.
p=-100   ! Total load on pole.
udl=-10  ! UDL on dome.
*
*
*
* Supports given in joint releases.
#cc924.stk !Import verification data from cc924.stk if available.
r=SQR((2*r2-hd)*hd) nrd=1 d=2*r2 dd=2*r di(1)=2*r0 g=e/(2*(1+nu))
:20 ;es=PI*di(nrd)/nsg i=nrd nrd=nrd+1
di(nrd)=d*SIN(ASN(di(i)/d)+es/r2)
IF di(nrd)<di(i) THEN Failure. ;IF di(nrd)<=dd GOTO 20
i=1 scale=(dd-di(1))/(di(nrd)-di(1))  !Scale to fix over shoot.
:60 ;i=i+1 ;IF i<nrd THEN di(i)=di(1)+(di(i)-di(1))*scale
IF i>=nrd THEN di(i)=dd ;IF i<nrd GOTO 60 ;nj=nrd*nsg
NUMBER OF JOINTS nj
NUMBER OF MEMBERS nel=nsg*(nrd-1) nm=nel*6 nm+nsg/2
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 2
JOINT COORDINATES ;theta=PI/2 alpha=2*PI/nsg nr=0 j=0 ;:80
IF j-nsg*INT(j/nsg)=0 ;np=nr nr=nr+1 ns=0
y=hd-(d-SQR(d^2-di(nr)^2))/2 ;ENDIF ;ns=ns+1 j=j+1
x=0.5*di(nr)*COS(theta-ns*alpha) ;IF ABS(x)<1E-6 THEN x=0
z=-0.5*di(nr)*(-SIN(theta-ns*alpha)) ;IF ABS(z)<1E-6 THEN z=0
j x y z ;IF j<nj GOTO 80 ;JOINT RELEASES
nj-nsg+1 THRU nj FORCE Y -1 ;1 FORCE X -1 Z -1 MOMENT Y -1
MEMBER INCIDENCES ;nr=0 m=-5 ;:100 ;nr=nr+1 i=0 ;:200 ;i=i+1 m=m+6
j1=nr*nsg+i j2=j1-nsg ;IF i=1 THEN j3=j1-1 j4=j1+nsg-1
IF i>1 THEN j3=j2-1 j4=j1-1 ;m THRU m+5 ELEMENT j1,j2,j3,j4
IF i<nsg GOTO 200 ;IF nr<nrd-1 GOTO 100
nm+1 THRU nm+nsg/2 RANGE 1,1+nsg/2 nsg/2,nsg
CONSTANTS E e ALL G g ALL DENSITY -udl/t ALL
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t
nm+1 THRU nm+nsg/2 RECTANGLE DY t DZ 2*PI*r0/nsg/2 !Infill hole.
LOADING CASE 1 - UDL ON PLATE
MEMBER SELF WEIGHTS ;1 THRU nm 1.0
LOADING CASE 2 - LOAD p DISTRIBUTED TO POLE RING
JOINT LOADS ;1 THRU nsg FORCE Y p/nsg ;SOLVE
*/10 ;* Surface area of dome +s=2*PI*r2*hd-PI*r0^2
* Self weight of dome  +sw=-udl*s
* ▪ Formulas for Stress and Strain, 4th Edition by Roark
* Roark in Table XIII case 20, gives a deflection coefficient
* dependent on +mu=r0*(12*(1-nu^2)/(r2^2*t^2))^0.25
r0'=SQR(1.6*r0^2+t^2)-0.675*t ;IF r0'>0.5*t THEN r0'=r0
mu1=VEC(0,0.1,0.2,0.4,0.6,0.8,1.0,1.2,1.4) n=9 i=0
a1=VEC(0.433,0.431,0.425,0.408,0.386,0.362,0.337,0.311,0.286)
IF mu<mu(1) OR mu>mu(n) THEN Out of range for interpolation for a.
:400 ;i=i+1 ;IF mu>mu(i) GOTO 400
```

```
* +a=a(i-1)+(a(i)-a(i-1))*(mu-mu(i-1))/(mu(i)-mu(i-1))
* Maximum deflection +ymax=a*p*r2*SQR(1-nu^2)/(e*t^2) ;*
* Central joint No. +cj=1
* Node number at centre of roof +nc=ARR(8,cj,2) thus to extract
* centre defln from ARRay 8, 'look up' row +rc=(nc-1)*6+2
* NL-STRESS central joint deflection +dc=ARR(6,rc,2) ;*/4
* Position      Deflection      Deflection by     %age
* ref. No.      by NL-STRESS    Roark's Formula   diff.
status=1 gtot=0 nur=0 d1=dc d2=ymax
#vmper.ndf !Compute percentage difference & any message.
*   +nc         +dc             +ymax             $ok
fnm=$(vm630.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE I-SECTION SUBJECTED TO A TORQUE (TWISTING MOMENT)
TITLE COMPARISON OF RESULTS WITH CLASSICAL ANALYSIS
TITLE GIVEN BY ROARK BASED ON TIMOSHENKO'S 'STRENGTH OF
TITLE MATERIALS PART II', 1930, PUBLISHED D VAN NOSTRAND.
MADEBY DWB ;DATE 06.01.05 ;METHOD ELASTIC JOINTS ;REFNO VM640
TABULATE DISPLACEMENTS  ;TYPE SPACE FRAME run=0
PRINT DATA, RESULTS FROM 1
a=.10        ! Width of flange.
b=.01        ! Thickness of flange.           Y
c=.14        ! Depth between flanges.                          b
d=.01        ! Thickness of web.
lx=0.6       ! Length of Isection.       c          X          Z
t=1          ! End torque.                            into medium
e=205E6      ! Young's modulus.                      d<=b
nu=.3        ! Poisson's ratio.
na=4         ! Even No. of elements in 'a'.
nc=6         ! No. of elements in 'c'.           a             b
warp=0       ! Warping prevented at free end (1=yes,0=no)
#cc924.stk  !Import set of parameters if available from cc924.stk.
esa=a/na        esc=(c+b)/nc       !  Element sizes along a & c
eav=(esa+esc)/2 nx=INT(lx/eav+.5)  !  & number along the length.
njx=nx+1        njz=na+1       !No. of joints in X & Z directions.
NUMBER OF JOINTS nj=(nc+1+2*na)*njx nj ;nst=2*na+nc nel=nst*nx
nm=nel*6+nst ;NUMBER OF MEMBERS nm ;NUMBER OF SUPPORTS 0
! Supports given in joint releases table. ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;* First joints in top and bottom flanges.
nr=0 j=0 xinc=lx/nx x=-xinc njs=nj/njx  !Initialise values. ;:10
nr=nr+1 x=x+xinc j=njs*(nr-1) j+1 THRU j+njz X x Y c+b Z 0 ZL a
j+1+njz THRU j+2*njz X x Y 0 Z 0 ZL a ;IF nr<njx GOTO 10
* Next joints in the web.
nr=0 j=0 xinc=lx/nx x=-xinc  !Initialise counters and constants.
:30 ;nr=nr+1 x=x+xinc yinc=(c+b)/nc y=c+b j=njs*(nr-1)
j+2*njz+1 THRU j+2*njz+nc-1 X x Z a/2 Y y-yinc YL yinc
IF nr<njx GOTO 30 ;JOINT RELEASES
1 THRU 2*njz+nc-1 FORCE X -1 Y -1 Z -1 MOMENT X -1 Y -1 Z -1
j'=nj-njs+na/2+1 h=na/2 p'=h+1 q=p'+nc r'=njz+1 s'=r'+h u=nj-njs+1
v=j'+h+1 f=v+h ;*/13
*                          +a
*  +b               1     +p'   +njz          +u    +j'   +j'+h
*
*
*
*
*  +c
*
*
*
*
*  +b               +r'   +s'   +2*njz        +v    +f    +f+h
*               START JOINT NUMBERS        END JOINT NUMBERS ;*
MEMBER INCIDENCES
nr=0 m=-6*(2*na+nc) j=-njs  !Initialise rows, members & joints.
:100 ;nr=nr+1 m=m+6*(2*na+nc) j=j+njs jz=j+njz
* First set member incidences in top and bottom flanges.
m+1 THRU m+6*na ELEMENT j+njs+1,j+njs+2,j+2,j+1
m+6*na+1 THRU m+12*na ELEMENT jz+njs+1,jz+njs+2,jz+2,jz+1
! Next set member incidences in the web.
i=0  !Initialise members & element counter. ;:120
i=i+1 r=j+2*njz+i s=r-1 p=s+njs q=r+njs m'=m+12*na+6*(i-1)
IF i=1 THEN s=j+1+na/2 p=s+njs
IF i=nc THEN r=j+njz+na/2+1 q=r+njs
m'+1 THRU m'+6 ELEMENT p,q,r,s ;IF i<nc GOTO 120
```

```
IF nr<nx GOTO 100 ;! Next set member incidences for end stiffener.
nm-nst+1 THRU nm-nst+na RANGE u,u+1 j'+h-1,j'+h
nm-nst+na+1 THRU nm-nst+2*na RANGE v,v+1 f+h-1,f+h
nm-nc+1 j' nj-nc+2 ;nm nj f
IF nc>2 THEN nm-nc+2 THRU nm-1 RANGE nj-nc+2,nj-nc+3 nj-1,nj
! For all FEM analyses CONSTANTS must come before MEMBER PROPS.
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
! First member properties in top and bottom flanges.
nr=0 m=-6*(2*na+nc)  !Initialise rows & members. ;:200
nr=nr+1 m=m+6*(2*na+nc) m+1 THRU m+12*na ELEMENT T b
! Next member properties in the web.
m+12*na+1 THRU m+12*na+6*nc ELEMENT T d ;IF nr<nx GOTO 200
nm-nst+1 THRU nm CONIC D a*warp+1E-6 !Warping restraint on/off.
LOADING CASE 1 - END TORQUE ;JOINT LOADS
* Apply torque as equal & opposite shears at flanges of Isection.
j=nj-njs+na/2+1
j FORCE Z t/(c+b) ;j+njz FORCE Z -t/(c+b) ;SOLVE
* Rotation computed by NL-STRESS: ;status=1 gtot=0 nur=0
* For +row=(j-1)*6+3 joint +j displ +jjd=ARR(6,row,1)
* For +row=(f-1)*6+3 joint +f displ +jfd=ARR(6,row,1)
* NL-STRESS rotation at end +nr=(jjd-jfd)/(c+b) rad.
* Torsion constant from Formulas for Stress and Strain by Roark.
* Dia. inscribed circle at root +d'=(d+b)/(1+COS(RAD(45)))
* Roark's factor +alpha=0.15*d/b for zero root radius.
* Constant +k1=a*b^3*(1/3-0.21*b/a*(1-b^4/(12*a^4)))
*          +k2=c*d^3/3
* Torsion constant +k=2*k1+k2+2*alpha*d'^4
* Inertia about Y  +iy=2*b*a^3/12+c*d^3/12
* Constant         +a'=(c+b)/2*SQR(iy*e/(k*g))
* Rotation computed by Roark's theory based on Timoshenko.
* Roark's end rotn +rr=t*(lx-a'*TNH(lx/a'))/(k*g) rad.
d1=nr d2=rr
#vmper.ndf !Compute percentage difference & any message.
*                NL-STRESS     Roark's       %age
*                stiffness     analysis      diff.
* Max. rotation  +nr           +rr           $ok
fnm=$(vm640.stk) ;#vmres.ndf !Conclude results.
mjn=16 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE RECTANGULAR HOLLOW SECTION CANTILEVER BEAM
TITLE SUBJECTED TO BIAXIAL BENDING AND/OR TORQUE
TITLE RESULTS COMPARE THE MATRIX STIFFNESS METHOD WITH
TITLE ROARK'S FORMULAS FOR STRESS AND STRAIN, FOURTH EDTN.
MADEBY DWB ;DATE 02.01.06 ;REFNO VM641
TABULATE DISPLACEMENTS ;TYPE SPACE FRAME run=0
a=.100       ! Breadth of section.
b=.008       ! Thickness of flange.                             b
c=.150       ! Depth of section.                      Y
d=.008       ! Thickness of webs.
lx=1.2       ! Length of RHS.                  c      X in       Z
e=205E6      ! Young's modulus.
nu=.3        ! Poisson's ratio.                    d            d
na=4         ! Even No. of elems in 'a'.
nc=6         ! Even No. of elems in 'c'.
nli=1        ! No. of load incs. 1=elastic.             a       b
warp=1       ! Warping prevented at free end (1=yes,0=no)
fy=1    fz=1    mx=0.1   ! End forces: Y & Z directions & about X.
#cc924.stk !Import set of parameters if available from cc924.stk.
PRINT DATA, RESULTS FROM 1 ;NUMBER OF INCREMENTS nli ;IF nli=1
METHOD ELASTIC JOINTS ;ENDIF ;IF nli>1 THEN METHOD SWAY JOINTS
esa=(a-d)/na    esc=(c-b)/nc       !┐ Element sizes along a & c
eav=(esa+esc)/2 nx=INT(lx/eav+.5)  !┘ & number along the length.
njx=nx+1 njz=na+1 a'=(a-d)/2 c'=(c-b)/2 nst=(na+nc)*2 nel=nst*nx
IF nel<=5300 GOTO 8 !Adjust to keep mesh within sensible limits.
na=na+1 ;:5 ;na=na-1 nc=INT(na*(c-b)/(a-d)+.5) esa=(a-d)/na
esc=(c-b)/nc eav=(esa+esc)/2 nx=INT(lx/eav+.5) njx=nx+1 njz=na+1
a'=(a-d)/2 c'=(c-b)/2 nst=(na+nc)*2 nel=nst*nx
IF na>2 AND nel>5300 GOTO 5 ;:8
NUMBER OF JOINTS nj=(na+nc)*2*njx nj
NUMBER OF MEMBERS nm=nel*6+nst nm ;NUMBER OF SUPPORTS 0
! Supports given in joint releases table. ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;* First joints in top and bottom flanges.
nr=0 j=0 xinc=lx/nx x=-xinc njs=nj/njx  !Initialise values. ;:10
nr=nr+1 x=x+xinc j=njs*(nr-1) j+1 THRU j+njz X x Y c' Z -a' ZL a'
j+nc+njz THRU j+nc+njz+na X x Y -c' Z a' ZL -a' ;IF nr<njx GOTO 10
* Next joints in the webs. ;nr=0 j=0 x=-xinc y'=c'-(c-b)/nc
! Initialise counters & constants. ;:30 ;nr=nr+1 x=x+xinc
j=njs*(nr-1) j+njz+1 THRU j+njz+nc-1 X x Z a' Y y' YL -y'
j+njs-nc+2 THRU j+njs X x Z -a' Y -y' YL y' ;IF nr<njx GOTO 30
JOINT RELEASES
1 THRU njs FORCE X -1 Y -1 Z -1 MOMENT X -1 Y -1 Z -1
j'=nj-njs+na+1 r'=njz+nc+na
u=nj-njs+1 v=nj-nc+1 f=v-na/2 ;*/13
*                       +a
*      +b          1           +njz          +u         +j'
*
*
*
*
*  +c                                +d
*
*
*
*
*      +b          +r'         +r'-na        +v         +v-na
*               START JOINT NUMBERS       END JOINT NUMBERS ;*
MEMBER INCIDENCES
nr=0 nes=2*(na+nc) m=-nes*6 j=-njs ;:100 ;nr=nr+1 m=m+6*nes
j=j+njs m+1 THRU m+6*(nes-1) ELEMENT j+njs+1,j+njs+2,j+2,j+1
m+6*(nes-1)+1 THRU m+6*(nes-1)+6 ELEMENT j+2*njs,j+njs+1,j+1,j+njs
IF nr<nx GOTO 100
```

```
nm-nst+1 THRU nm-1 RANGE nj-njs+1 nj-njs+2 nj-1 nj ;nm nj nj-njs+1
! For all FEM analyses CONSTANTS must come before MEMBER PROPS.
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
nr=0 nes=2*(na+nc) m=-nes*6 j=-njs ;:200 ;nr=nr+1 m=m+6*nes
m+1 THRU m+6*na ELEMENT T b
m+6*na+1 THRU m+6*(na+nc) ELEMENT T d
m+6*(na+nc)+1 THRU m+6*(2*na+nc) ELEMENT T b
m+6*(2*na+nc)+1 THRU m+6*(2*na+2*nc) ELEMENT T d
IF nr<nx GOTO 200 ;! Next set member incidences for end stiffener.
nm-nst+1 THRU nm CONIC D a/4*warp+1E-9 !Warping restraint on/off.
LOADING CASE 1 ;JOINT LOADS ;nj-njs+1 THRU nj FORCE Y fy/njs
nj-njs+1 THRU nj FORCE Z fz/njs !Forces to apply torque.
zmx=0.5*mx/(na+1)/(c-b) ymx=0.5*mx/(nc+1)/(a-d)
j=nj-njs+1 j FORCE Y ymx ;j THRU j+na FORCE Z zmx
j=j+na j THRU j+nc FORCE Y -ymx ;j=j+nc j THRU j+na FORCE Z -zmx
j=j+na j THRU nj FORCE Y ymx ;SOLVE ;status=1 gtot=0 nur=0
! NL-STRESS end displacements in Y & Z dirn & rotation about X.
j=nj-njs sigx=0 sigy=0 sigz=0 ;:250 ;j=j+1 row=(j-1)*6+4
disp=ARR(6,row,nli) sigx=sigx+disp row=(j-1)*6+2
disp=ARR(6,row,nli) sigy=sigy+disp row=(j-1)*6+3
disp=ARR(6,row,nli) sigz=sigz+disp ;IF j<nj GOTO 250
nxr=(sigx/njs)*360/(2*PI) nyd=(sigy/njs) nzd=(sigz/njs)
! Roark's formulas for end displacements & rotation in degrees:
iz=a*c^3/12-(a-2*d)*(c-2*b)^3/12 ax=a*c-(a-2*d)*(c-2*b)
d1=c/2-b d2=c/2 t1=2*d t2=a
frk=(1+3*(d2^2-d1^2)*d1/(2*d2^3)*(t2/t1-1))*4*d2^2*ax/(10*iz)
ryd=fy*lx^3/(3*e*iz)+frk*fy*lx/(ax*g)
iy=c*a^3/12-(c-2*b)*(a-2*d)^3/12 d1=a/2-d d2=a/2 t1=2*b t2=c
frk=(1+3*(d2^2-d1^2)*d1/(2*d2^3)*(t2/t1-1))*4*d2^2*ax/(10*iy)
rzd=fz*lx^3/(3*e*iy)+frk*fz*lx/(ax*g)
ix=2*b*d*(a-d)^2*(c-b)^2/(a*d+c*b-d^2-b^2)
rxr=mx*lx/(ix*g)*360/(2*PI) ;*/6
*                               NL-STRESS     Roark's       %age
*                               stiffness     formulae      diff.
d1=nyd d2=ryd
#vmper.ndf !Compute percentage difference & any message.
* End displacement in Y dirn.   +nyd          +ryd          $ok
d1=nzd d2=rzd
#vmper.ndf !Compute percentage difference & any message.
* End displacement in Z dirn.   +nzd          +rzd          $ok
d1=nxr d2=rxr
#vmper.ndf !Compute percentage difference & any message.
* End rotation° about X axis    +nxr          +rxr          $ok
fnm=$(vm641.stk) ;#vmres.ndf !Conclude results.
mjn=nj lcn=nli tot=3 drn=6 ;#vmtes.ndf
< ;FINISH
```

```
TITLE T-SECTION CANTILEVER BEAM SUBJECTED TO BENDING ABOUT
TITLE THE Z AXIS AND/OR TORQUE FROM END LOADING.  RESULTS
TITLE COMPARE THE MATRIX STIFFNESS METHOD WITH ROARK'S
TITLE FORMULAS FOR STRESS AND STRAIN, SEVENTH EDITION.
MADEBY DWB ;DATE 06.01.06 ;REFNO VM642
TABULATE DISPLACEMENTS  ;TYPE SPACE FRAME run=0
a=.100       ! Breadth of flange.                         |Y
b=.01        ! Thickness of flange.                      |-|d
c=.150       ! Depth of web.
d=.01        ! Thickness of web.
lx=1.2       ! Length of T-section.                                       b
e=205E6      ! Young's modulus.
nu=.3        ! Poisson's ratio.            c             X            ---Z
na=4         ! Even No. of elems in 'a'.                 |--into medium
nc=6         ! Even No. of elems in 'c'.
nli=1        ! No. of load incs. 1=elast.
warp=0       ! Warp=1 for warping restraint.   |-----a-----|
fy=10  mx=0.2 ! End forces: Y direction & about X.
#cc924.stk !Import set of parameters if available from cc924.stk.
PRINT DATA, RESULTS FROM 1 ;NUMBER OF INCREMENTS nli ;IF nli=1
METHOD ELASTIC JOINTS ;ENDIF ;IF nli>1 THEN METHOD SWAY JOINTS
esa=a/na        esc=(c+b/2)/nc    !  Element sizes along a & c.
eav=(esa+esc)/2 nx=INT(lx/eav+.5) !  & No. of elements in X dirn.
njx=nx+1 njz=na+1 ;NUMBER OF JOINTS nj=(nc+1+na)*njx nj
NUMBER OF MEMBERS nel=(na+nc)*nx nm=nel*6+na+nc nm
NUMBER OF SUPPORTS 0 !Supports given in joint releases table.
NUMBER OF LOADINGS 1 ;JOINT COORDINATES ;! First top flange.
nr=0 j=0 xinc=lx/nx x=-xinc njs=nj/njx ;:10 ;nr=nr+1 x=x+xinc
j=njs*(nr-1) ;j+1 THRU j+njz X x Y c+b/2 Z 0 ZL a
IF nr<njx GOTO 10 ;! Next joints in the web.
nr=0 j=0 xinc=lx/nx x=-xinc ;:30 ;nr=nr+1 x=x+xinc yinc=(c+b/2)/nc
y=c+b j=njs*(nr-1) j+njz+1 THRU j+njz+nc X x Z a/2 Y y-yinc YL 0
IF nr<njx GOTO 30 ;JOINT RELEASES ;IF warp=1
1 THRU njs FORCE X -1 Y -1 Z -1 MOMENT X -1 Y -1 Z -1 ;ENDIF
IF warp=0 ;1 THRU njs FORCE X -1 Z -1 MOMENT Z -1
na/2+1 FORCE Y -1
njz+1 THRU njs FORCE Y -1 ;ENDIF ;h=na/2 v=nj-njs+1 ;*/12
*                  |---- +a  ----|
*  +b              1     +h+1  +njz          +v    +v+h  +v+na
*
*
*
*
*  +c
*
*
*
*                        +njs                      +nj
*              START JOINT NUMBERS        END JOINT NUMBERS ;*
MEMBER INCIDENCES ;nr=0 m=-6*(na+nc) j=-njs ;:100
nr=nr+1 m=m+6*(na+nc) j=j+njs
m+1 THRU m+6*na ELEMENT j+njs+1,j+njs+2,j+2,j+1 ;! Top flange.
! Next web. ;i=0 ;:120 ;i=i+1 r=j+njz+i s=r-1 p=s+njs q=r+njs
m'=m+6*na+6*(i-1) ;IF i=1 THEN s=j+1+na/2 p=s+njs
m'+1 THRU m'+6 ELEMENT p,q,r,s ;IF i<nc GOTO 120
IF nr<nx GOTO 100 ;! Next member incidences for end stiffener.
j=nj-njs nel*6+1 THRU nel*6+na RANGE j+1,j+2 j+na,j+njz
nel*6+na+1 j+na/2 j+na+2
nel*6+na+2 THRU nm RANGE j+na+2,j+na+3 nj-1,nj
! For all FEM analyses CONSTANTS must come before MEMBER PROPS.
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
```

```
! First top flange. ;nr=0 m=-6*(na+nc) ;:200 ;nr=nr+1
m=m+6*(na+nc) ;m+1 THRU m+6*na ELEMENT T b ;! Web properties.
m+6*na+1 THRU m+6*na+6*nc ELEMENT T d ;IF nr<nx GOTO 200
nel*6+1 THRU nm CONIC D a/4*warp+1E-9 !Warping restraint on/off.
LOADING CASE 1 - END LOAD IN Y DIRECTION & TORQUE ;JOINT LOADS
! Share end load & torque to each end joint of section.
nj-njs+1 THRU nj FORCE Y fy/njs ;nj-njs+1 THRU nj MOMENT X mx/njs
SOLVE ;status=1 gtot=0 nur=0 ;! NL-STRESS end displacements.
j=nj-njs sigx=0 sigy=0 ;:250 ;j=j+1 row=(j-1)*6+4
disp=ARR(6,row,nli) sigx=sigx+disp row=(j-1)*6+2
disp=ARR(6,row,nli) sigy=sigy+disp ;IF j<nj GOTO 250
nxr=(sigx/njs)*360/(2*PI) nyd=(sigy/njs) ;! Roark displacements.
yc=(a*b^2+d*c*(2*b+c))/(2*(a*b+c*d))
iz=a/3*(b+c)^3-c^3/3*(a-d)-(a*b+c*d)*(b+c-yc)^2
ryd=fy*lx^3/(3*e*iz)+6/5*(fy*lx)/((c+b)*d*g)
d'=(d+b)/(1+COS(RAD(45))) alpha=0.15*d/b
k1=a*b^3*(1/3-0.21*b/a*(1-b^4/(12*a^4)))
k2=c*d^3*(1/3-0.105*d/c*(1-d^4/(192*c^4))) ix=k1+k2+alpha*d'^4
rxr=mx*lx/(ix*g)*360/(2*PI) ;*/6
*                               NL-STRESS     Roark's       %age
*                               stiffness     formulae      diff.
d1=nyd d2=ryd
#vmper.ndf !Compute percentage difference & any message.
* End displacement in Y dirn.   +nyd          +ryd          $ok
d1=nxr d2=rxr
#vmper.ndf !Compute percentage difference & any message.
* End rotation° about X axis    +nxr          +rxr          $ok
fnm=$(vm642.stk) ;#vmres.ndf !Conclude results.
mjn=nj lcn=nli tot=3 drn=6 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CHANNEL CANTILEVER BEAM SUBJECTED TO BENDING ABOUT
TITLE THE Y AXIS AND/OR TORQUE FROM END LOADING.  RESULTS
TITLE COMPARE THE MATRIX STIFFNESS METHOD WITH ROARK'S
TITLE FORMULAS FOR STRESS AND STRAIN, SEVENTH EDITION.
MADEBY DWB ;DATE 09.01.06 ;REFNO VM643
TABULATE DISPLACEMENTS ;TYPE SPACE FRAME run=0
a=.100      ! Breadth of flange.              |Y
b=.01       ! Thickness of flange.           ├─┼d
c=.150      ! Depth of web.              ──  ┌──────────┐  ──
d=.01       ! Thickness of web.           |  │ ┌────────┘  ─┼─
lx=1.2      ! Length of channel.          |  │ │             └b
e=205E6     ! Young's modulus.            |  │ │
nu=.3       ! Poisson's ratio.            c  │X│          ───Z
na=4        ! Even No. of elems in 'a'.   |  │└┴─into medium
nc=6        ! Even No. of elems in 'c'.   |  │ │
nli=1       ! No. of load incs. 1=elastic.|  │ └────────┐  ──
warp=0      ! Warp=1 =warping restraint. ──  └──────────┘  ─┼─
fz=10  mx=0.2 ! End forces: Z dirn & about X. ├────a─────┤  └b
#cc924.stk !Import set of parameters if available from cc924.stk.
PRINT DATA, RESULTS FROM 1 ;NUMBER OF INCREMENTS nli ;IF nli=1
METHOD ELASTIC NODES ;ENDIF ;IF nli>1 THEN METHOD SWAY NODES
esa=(a-d/2)/na esc=(c-b)/nc       !┐ Element sizes along a & c.
eav=(esa+esc)/2 nx=INT(lx/eav+.5) !┘ & No. of elements in X dirn.
NUMBER OF JOINTS njs=nc+1+2*na nj=njs*(nx+1) nj+1 nel=(2*na+nc)*nx
NUMBER OF MEMBERS nm=nel*6+2*na+nc nm+2 ;NUMBER OF SUPPORTS 0
! Supports given in joint releases table. ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;nr=0 j=0 xinc=lx/nx x=-xinc ;:10 ;nr=nr+1 i=0
x=x+xinc c'=c-b ;:20 ;i=i+1 j=j+1 y=0 z=0
IF i<=na THEN y=c'/2 z=(na-i+1)*esa
IF i>na+1+nc THEN y=-c'/2 z=(i-na-nc-1)*esa
IF z=0 THEN y=c'/2-(i-na-1)*esc ;j x y z
IF i<nc+1+2*na GOTO 20 ;IF nr<nx+1 GOTO 10 ;nj+1 0 0 a-d/2
JOINT RELEASES
! Fix all joints in X direction and hold top root against movement
! in Z direction, and bottom
na+1 FORCE Z -1
na+1+nc FORCE Y -1 Z -1
na+1 THRU na+1+nc FORCE X -1
! Support rocker to permit warping. ;nj+1 FORCE X -1 Y -1 Z -1
p=na+1 q=p+nc r=q+na u=nj-2*na-nc v=u+na w=v+nc ;*/15
* ─┬─    ─┬─       1─────────┬─────────┬─────────┬──────   ───┬─────────┬─────────── +u
*  +na     |        |         |         |         |            |         |           |
*  |      +esa*na   |         |         |         |            |         |           |
* ─┼─    ─┼─       +p  ──────┼─Fold────┼─line────┼──────   ───┼─────────┼─────────── +v
*  +nc     |        |         |         |         |            |         |           |__X
*  |      +esc*nc   |         |         |         |            |         |           |
* ─┼─    ─┼─       +q  ──────┼─Fold────┼─line────┼──────   ───┼─────────┼─────────── +w
*  +na     |        |         |         |         |            |         |           |
*  |      +esc*na   |         |         |         |            |         |           |
* ─┴─    ─┴─       +r  ──────┴─────────┴─────────┴──────   ───┴─────────┴─────────── +nj
* No.    Size of
* elems  legs       ├──────────────────── +nx  elements─────────────┤
*                   ├──────────────────── +lx  ─────────────────────┤
*       KEY JOINT NUMBERS, NUMBER OF ELEMENTS & SIZES ;*
MEMBER INCIDENCES ;nr=0 m=-6*(2*na+nc) j=-1-2*na-nc ;:100
nr=nr+1 m=m+6*(2*na+nc) j=j+1+2*na+nc
m+1 THRU m+6*(2*na+nc) ELEMENT j+2*na+nc+2,j+2*na+nc+3,j+2,j+1
IF nr<nx GOTO 100 ;! Next member incidences for end stiffener.
j=nj-njs nel*6+1 THRU nm RANGE j+1,j+2 nj-1,nj
nm+1 1 nj+1 ;nm+2 nj+1 njs
MEMBER RELEASES ;nm+1 START MOMENT X Y Z ;nm+2 END MOMENT Y Z
! For all FEM analyses CONSTANTS must come before MEMBER PROPS.
```

```
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
nr=0 m=-6*(2*na+nc) ;:120 ;nr=nr+1 m=m+6*(2*na+nc)
m+1 THRU m+6*na ELEMENT T b ;m+6*na+1 THRU m+6*(na+nc) ELEMENT T d
m+6*(na+nc)+1 THRU m+6*(2*na+nc) ELEMENT T b ;IF nr<nx GOTO 120
nel*6+1 THRU nm CONIC D a/4*warp+1E-9 !Warping restraint on/off.
! Make rocker very sturdy. ;nm+1 THRU nm+2 CONIC D 10*a
LOADING CASE 1 - END LOAD IN Z DIRECTION & TORQUE ;JOINT LOADS
! Share end load & torque to each end joint of section.
nj-njs+1 THRU nj FORCE Z fz/njs ;nj-njs+1 THRU nj MOMENT X mx/njs
SOLVE ;status=1 gtot=0 nur=0 ;! NL-STRESS end displacements.
j=nj-njs sigx=0 sigz=0 ;:250 ;j=j+1 n=ARR(8,j,2) row=(n-1)*6+4
disp=ARR(6,row,nli) sigx=sigx+disp row=(n-1)*6+3
disp=ARR(6,row,nli) sigz=sigz+disp ;IF j<nj GOTO 250
nxr=(sigx/njs)*360/(2*PI) nzd=(sigz/njs) ;! Roark displacements.
a'=a-d yc=(c*d^2+2*b*a'*(2*d+a'))/(2*(d*c+2*b*a'))
iy=c/3*a^3-((a')^3)/3*(c-2*b)-(d*c+2*b*a')*(a-yc)^2
rzd=fz*lx^3/(3*e*iy)+6/5*(fz*lx)/(2*a*b*g)
d'=(d+b)/(1+COS(RAD(45))) c'=c-2*b alpha=0.07*d/b
k1=a*b^3*(1/3-0.21*b/a*(1-b^4/(12*a^4)))
k2=c'*d^3*(1/3-0.105*d/c'*(1-d^4/(192*c'^4)))
ix=2*k1+k2+2*alpha*d'^4 rxr=mx*lx/(ix*g)*360/(2*PI) ;*/6
*                               NL-STRESS     Roark's       %age
*                               stiffness     formulae      diff.
d1=nzd d2=rzd
#vmper.ndf !Compute percentage difference & any message.
* End displacement in Z dirn.   +nzd          +rzd          $ok
d1=nxr d2=rxr
#vmper.ndf !Compute percentage difference & any message.
* End rotation° about X axis    +nxr          +rxr          $ok
fnm=$(vm643.stk) ;#vmres.ndf !Conclude results.
mjn=nj lcn=nli tot=3 drn=6 ;#vmtes.ndf
< ;FINISH
```

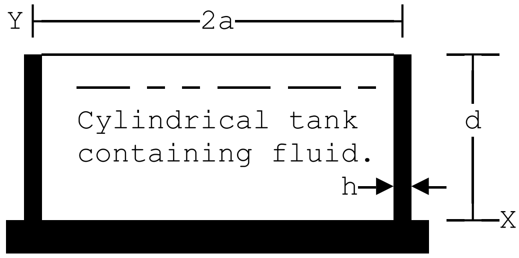
```
TITLE ANGLE AS CANTILEVER BEAM, SUBJECTED TO TORQUE
TITLE (TWISTING MOMENT) ABOUT THE X AXIS.  RESULTS
TITLE COMPARE THE MATRIX STIFFNESS METHOD WITH ROARK'S
TITLE FORMULAS FOR STRESS AND STRAIN, SEVENTH EDITION.
MADEBY DWB ;DATE 11.01.06 ;REFNO VM644
TABULATE DISPLACEMENTS ;TYPE SPACE FRAME run=0
a=.055 c=.080 ! Dimensions of angle.       │Y
b=.01  d=.01  ! Thicknesses of angle b>=d. ├─┤b
lx=0.6        ! Length of angle.       ───  ┌─┐
e=205E6       ! Young's modulus.        │   │ │
nu=.3         ! Poisson's ratio.l.      a   │ │
na=4          ! Even No. elems in 'a'.  │   │ │            d
nc=6          ! Even No. elems in 'c'.  │   │ └─────────┐ ─┼─  __Z
nli=1         ! No. of load incs.      ───  └───────────┘ ─┴─
warp=0        ! Warp=1 =warping restraint. ├─────c──────┤
mx=1          ! Twisting moment.
#cc924.stk !Import set of parameters if available from cc924.stk.
PRINT DATA, RESULTS FROM 1 ;NUMBER OF INCREMENTS nli ;IF nli=1
METHOD ELASTIC NODES ;ENDIF ;IF nli>1 THEN METHOD SWAY NODES
esa=(a-d/2)/na esc=(c-b/2)/nc     !┐ Element sizes along a & c.
eav=(esa+esc)/2 nx=INT(lx/eav+.5) !┘ & No. of elements in X dirn.
NUMBER OF JOINTS njs=1+na+nc nj=njs*(nx+1) nj nel=(na+nc)*nx
NUMBER OF MEMBERS nms=njs-1 nm=nel*6 nm+nms ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 1 ;JOINT COORDINATES ;nr=0 j=0 xinc=lx/nx
x=-xinc ;:10 ;nr=nr+1 i=0 x=x+xinc ;:20 ;i=i+1 j=j+1 y=0 z=0
IF i<=na THEN y=(na+1-i)*esa z=0 ;IF i>na THEN y=0 z=(i-na-1)*esc
j x y z ;IF i<na+nc+1 GOTO 20 ;IF nr<nx+1 GOTO 10 ;JOINT RELEASES
IF warp=1 ;1 THRU njs FORCE X -1 Y -1 Z -1 MOMENT X -1 Y -1 Z -1
ENDIF ;IF warp=0 ;na+1 FORCE X -1 Y -1 Z -1 MOMENT X -1 Y -1 Z -1
1 THRU na FORCE X -1 Z -1 ;na+2 THRU na+nc+1 FORCE X -1 Y -1
ENDIF ;p=na+1 q=na+nc+1 u=nj-nc v=u-na ;*/12
* ─┬─    ─┬─        1───────────────────────  ──────────────── +v
*  +na    │         │      │       │       │     │       │     │
*  │      +esa*na   │      │       │       │     │       │     │
* ─┼─    ─┼─        +p  ─────┼Fold─┼─line─┼──  ─────┼───────┼── +u
*  +nc    │         │      │       │       │     │       │     │
*  │      +esc*nc   │      │       │       │     │       │     │
* ─┴─    ─┴─        +q  ──────────────────────  ──────────────── +nj
* No.    Size of
* elems  legs            ├──────────────────── +nx  elements──────────┤
*                        ├──────────────────── +lx  ──────────────────┤
*      KEY JOINT NUMBERS, NUMBER OF ELEMENTS & SIZES ;*
MEMBER INCIDENCES ;nr=0 m=-6*(na+nc) j=-1-na-nc ;:100
nr=nr+1 m=m+6*(na+nc) j=j+1+na+nc
m+1 THRU m+6*(na+nc) ELEMENT j+na+nc+2,j+na+nc+3,j+2,j+1 BUMP 1
IF nr<nx GOTO 100 ;nm+1 THRU nm+nms RANGE nj-njs+1 nj-njs+2 nj-1 nj

! For all FEM analyses CONSTANTS must come before MEMBER PROPS.
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU nm ELEMENT T b
nm+1 THRU nm+nms CONIC D a/4*warp+1E-9 !Warping restraint on/off.
LOADING CASE 1 - END TORQUE ;JOINT LOADS ;nj-nc MOMENT X mx
SOLVE ;status=1 gtot=0 nur=0 ;!NL-STRESS end rotation.
j=nj-njs sigx=0 sigz=0 ;:250 ;j=j+1 n=ARR(8,j,2) row=(n-1)*6+4
disp=ARR(6,row,nli) sigx=sigx+disp ;IF j<nj GOTO 250
nxr=(sigx/njs)*360/(2*PI) nzd=(sigz/njs) ;!Roark end rotation.
d'=(d+b)/(1+COS(RAD(45))) c'=c-b alpha=0.07*d/b
k1=a*b^3*(1/3-0.21*b/a*(1-b^4/(12*a^4)))
k2=c'*d^3*(1/3-0.105*d/c'*(1-d^4/(192*c'^4)))
ix=k1+k2+alpha*d'^4 rxr=mx*lx/(ix*g)*360/(2*PI) ;*/6
```

```
*                                NL-STRESS     Roark's       %age
*                                stiffness     formulae      diff.
d1=nxr d2=rxr
#vmper.ndf !Compute percentage difference & any message.
* End rotation° about X axis    +nxr          +rxr          $ok
fnm=$(vm644.stk) ;#vmres.ndf !Conclude results.
mjn=nj lcn=nli tot=3 drn=6 ;#vmtes.ndf
< ;FINISH
```

```
TITLE RECTANGULAR HOLLOW SECTION SIMPLY SUPPORTED BEAM
TITLE SUBJECTED TO AXIAL LOAD, BIAXIAL BENDING, TORQUE
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 17.06.06 ;REFNO VM645 ;TYPE SPACE FRAME run=0
a=.100      ! Breadth of section.       ──┬──  ┌───────────┐  ──┬──
b=.008      ! Thickness of flange.        │    │ ┌───────┐ │  ──┼──
c=.150      ! Depth of section.           │    │ │   Y   │ │    └b
d=.008      ! Thickness of webs.          │    │ │   │   │ │
lx=1.2      ! Length of RHS.              c    │ │   X in│ │  ───Z
e=205E6     ! Young's modulus.            │    │ │       │ │
nu=.3       ! Poisson's ratio.            │   d┼─┤       ├─┼d
na=4        ! Even No. of elems in 'a'.   │    │ └───────┘ │  ──┬──
nc=6        ! Even No. of elems in 'c'. ──┴──  └───────────┘  ──┼──
nli=1       ! No. of load incs. 1=elastic.     ├─────a─────┤    └b
warp=1 yield=220E3 ! Warping prevented at ends (1=yes, 0=no).
fx=-80  wy=-10  wz=6  ! Axial force & udls in Y  & Z directions.
ecy=.025  ecz=.025 ! Eccentricities for wy & wz loads (+ or -).
tlr=lx  blr=lx   ! Top & bottom lateral restraint spacing, 0=none.
trp=0   brp=0    ! T & b rotn. on plan restraint spacing, 0=none.
#cc924.stk !Import set of parameters if available from cc924.stk.
TABULATE ALL ;PRINT DATA, RESULTS FROM 1 LENGTH gen
NUMBER OF INCREMENTS nli ;IF nli=1 ;METHOD ELASTIC JOINTS ;ENDIF
IF nli>1 THEN METHOD SWAY JOINTS
esa=(a-d)/na    esc=(c-b)/nc       !┐ Element sizes along a & c
eav=(esa+esc)/2 nx=INT(lx/eav+.5)  !┘ & number along the length.
njx=nx+1 njz=na+1 a'=(a-d)/2 c'=(c-b)/2 nst=(na+nc)*2 nel=nst*nx
IF nel<=5300 GOTO 8 !Adjust to keep mesh within sensible limits.
na=na+2 ;:5 ;na=na-2 nc=INT(na*(c-b)/(a-d)+.5) esa=(a-d)/na
esc=(c-b)/nc eav=(esa+esc)/2 nx=INT(lx/eav+.5) njx=nx+1 njz=na+1
a'=(a-d)/2 c'=(c-b)/2 nst=(na+nc)*2 nel=nst*nx
IF na>=4 AND nel>5300 GOTO 5 ;:8
NUMBER OF JOINTS nj=(na+nc)*2*njx nj
NUMBER OF MEMBERS nm=nel*6+2*nst nm ;NUMBER OF SUPPORTS 0
! Supports given in joint releases table. ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;* First joints in top and bottom flanges.
nr=0 j=0 xinc=lx/nx x=-xinc njs=nj/njx  !Initialise values. ;:10
nr=nr+1 x=x+xinc j=njs*(nr-1) j+1 THRU j+njz X x Y c' Z -a' ZL a'
j+nc+njz THRU j+nc+njz+na X x Y -c' Z a' ZL -a' ;IF nr<njx GOTO 10
* Next joints in the webs. ;nr=0 j=0 x=-xinc y'=c'-(c-b)/nc
! Initialise counters & constants. ;:30 ;nr=nr+1 x=x+xinc
j=njs*(nr-1) j+njz+1 THRU j+njz+nc-1 X x Z a' Y y' YL -y'
j+njs-nc+2 THRU j+njs X x Z -a' Y -y' YL y' ;IF nr<njx GOTO 30
JOINT RELEASES ;s=nj-njs+na+1 r=njz+nc+na u=nj-njs+1 v=nj-nc+1
! First fixities as for s.s. beam. ;r-na/2 FORCE X -1 FORCE Z -1
v-na/2 FORCE Z -1 ;r-na THRU r FORCE Y -1 ;v-na THRU v FORCE Y -1
! Next additional: t & b lateral & rotational on plan restraints.
IF tlr=0 GOTO 45 ;x=-tlr ;:40 ;x=x+tlr eln=INT(nx*x/lx+0.5)
j=1+na/2+eln*nst j FORCE Z -1 ;IF x<lx GOTO 40 ;:45
IF blr=0 GOTO 55 ;x=-blr ;:50 ;x=x+blr eln=INT(nx*x/lx+0.5)
j=r-na/2+eln*nst ;IF j<>r-na/2 AND j<>v-na/2 THEN j FORCE Z -1
IF x<lx GOTO 50 ;:55 ;IF trp=0 GOTO 65 ;x=-trp ;:60 ;x=x+trp
eln=INT(nx*x/lx+0.5) 1+eln*nst THRU njz+eln*nst MOMENT Y -1
IF x<lx GOTO 60 ;:65 ;IF brp=0 GOTO 75 ;x=-brp ;:70 ;x=x+brp
eln=INT(nx*x/lx+0.5) r-na+eln*nst THRU r+eln*nst MOMENT Y -1
IF x<lx GOTO 70 ;:75 ;*/13
```

```
*                    +a
*       +b          1              +njz          +u             +s
*
*
*
*
*    +c                                 +d
*
*
*
*
*       +b          +r             +r-na         +v             +v-na
*                START JOINT NUMBERS        END JOINT NUMBERS ;*
MEMBER INCIDENCES
nr=0 nes=2*(na+nc) m=-nes*6 j=-njs ;:100 ;nr=nr+1 m=m+6*nes
j=j+njs m+1 THRU m+6*(nes-1) ELEMENT j+njs+1,j+njs+2,j+2,j+1
m+6*(nes-1)+1 THRU m+6*(nes-1)+6 ELEMENT j+2*njs,j+njs+1,j+1,j+njs
IF nr<nx GOTO 100 ;nm-2*nst+1 THRU nm-nst-1 RANGE 1 2 nst-1 nst
nm-nst nst 1 ;nm-nst+1 THRU nm-1 RANGE nj-njs+1 nj-njs+2 nj-1 nj
nm nj nj-njs+1 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;nr=0 nes=2*(na+nc) m=-nes*6 j=-njs ;:200
nr=nr+1 m=m+6*nes m+1 THRU m+6*na ELEMENT T b
m+6*na+1 THRU m+6*(na+nc) ELEMENT T d
m+6*(na+nc)+1 THRU m+6*(2*na+nc) ELEMENT T b
m+6*(2*na+nc)+1 THRU m+6*(2*na+2*nc) ELEMENT T d
IF nr<nx GOTO 200 ;! Next end stiffeners of 4 times average thick.
nm-2*nst+1 THRU nm-nst RECTANGLE D 4*(d+b)/2*warp+1E-9
nm-nst+1 THRU nm RECTANGLE D 4*(d+b)/2*warp+1E-9 !Warp is on/off.
LOADING CASE 1 ;JOINT LOADS ;! Vert. loads with eccentricity 'ec'.
w=wy*xinc/2 1 THRU u STEP nst FORCE Y w-w*ecy/((a-d)/2)
1 THRU u STEP u-1 FORCE Y -(w-w*ecy/((a-d)/2))/2
njz THRU s STEP nst FORCE Y w+w*ecy/((a-d)/2)
njz THRU s STEP s-njz FORCE Y -(w+w*ecy/((a-d)/2))/2
w=wz*xinc/2 1 THRU u STEP nst FORCE Z w+w*ecz/((c-b)/2)
1 THRU u STEP u-1 FORCE Z -(w+w*ecz/((c-b)/2))/2
r THRU v STEP nst FORCE Z w-w*ecz/((c-b)/2)
r THRU v STEP v-r FORCE Z -(w-w*ecz/((c-b)/2))/2 ;SOLVE
nli=ARR(12,4,1) lli=ARR(12,4,2) ;IF lli<nli THEN nli=lli ;m=0
maxs=0 mem=0 ;:300 ;m=m+1 ;rn=(nli-1)*nm+m ;fax=ARR(13,rn,13)
fby=ARR(13,rn,17) fbz=ARR(13,rn,18) fab=ABS(fax)+ABS(fby)+ABS(fbz)
IF fab>maxs THEN maxs=fab mem=m ;IF m<nm GOTO 300 ;n=ARR(1,mem,2)
diss=ARR(8,n,3) j=0 maxd=0 jnt=0 ;:310 ;j=j+1 n=ARR(8,j,2)
row=6*(n-1)+1 dx=ARR(6,row,nli) row=row+1 dy=ARR(6,row,nli)
row=row+1 dz=ARR(6,row,nli) dab=SQR(dx^2+dy^2+dz^2)
IF dab>maxd THEN maxd=dab jnt=j ;IF j<nj GOTO 310 ;n=ARR(8,jnt,2)
disd=ARR(8,n,3) ;! '*>' pipes summary to 'public.stk'. ;*/5
* Maximum combined axial & bend. stress +maxs         member +mem
* Distance from start to maximum stress +diss
* Maximum combined displacements       +maxd          joint  +jnt
* Distance from start to maximum displ. +disd ;IF maxs>yield
* Combined stress= +maxs exceeds yield stress= +yield ;ENDIF
IF run>0
*>Run No. +run      Max. comb. axial & bend. stress +maxs kN/m2
*>Self check at     Dist. from start to max. stress +diss m
*>working load.     Maximum combined displacement   +maxd m
*>                  Dist. from start to max. displ. +disd m
ENDIF ;IF run>0 AND maxs>yield
*>*** Combined stress= +maxs exceeds yield stress= +yield
ENDIF ;< ;FINISH
```

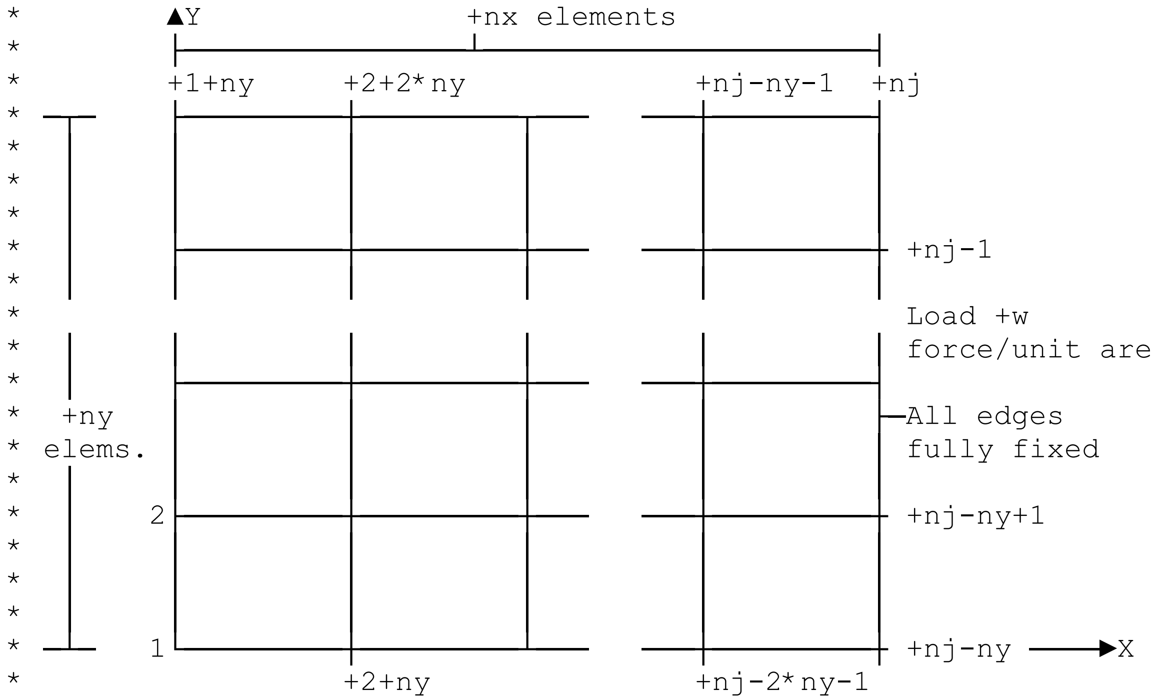
```
TITLE CIRCULAR TANK WITH RIGID BASE SUBJECTED TO INTERNAL
TITLE FLUID PRESSURE.  RESULTS COMPARE THE MATRIX STIFFNESS
TITLE METHOD WITH TIMOSHENKO & WOINOWSKY-KRIEGER, THEORY
TITLE OF PLATES & SHELLS, SECOND EDITION. ;METHOD ELASTIC
MADEBY DWB ;DATE 08.03.05 ;TYPE SPACE FRAME run=0 ;REFNO VM650
PRINT DATA, RESULTS FROM 1 ;TABULATE DISPLACEMENTS
a=5           ! Radius of tank.
d=2           ! Height of tank.
h=0.105       ! Thickness of wall.
gam=10        ! Weight/unit volume.
nsg=48        ! No. segments/ring
              ! divisible by 4.
e=30E6/3      ! Young's modulus.
nu=0.2        ! Poisson's ratio
```




```
#cc924.stk !Import verification data from cc924.stk if available.
dc=2*a nrj=INT(d/(PI*dc/nsg)+0.5)+1 de=d/(nrj-1) nel=nsg*(nrj-1)
NUMBER OF JOINTS nj=nrj*nsg nj ;NUMBER OF MEMBERS nme=nel*6 nme
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS nls=1 nls
JOINT COORDINATES ;theta=PI/2 alpha=2*PI/nsg nr=0 j=0 y=-de ;:5
IF j-nsg*INT(j/nsg)=0 THEN nr=nr+1 ns=0 y=y+de ;ns=ns+1 j=j+1
x=0.5*dc*COS(theta-ns*alpha) ;IF ABS(x)<1E-6 THEN x=0
z=0.5*dc*(-SIN(theta-ns*alpha)) ;IF ABS(z)<1E-6 THEN z=0 ;j x y z
IF j<nj GOTO 5 ;JOINT RELEASES
1 THRU nsg FORCE X -1 Y -1 Z -1 MOMENT X -1 Y -1 Z -1
MEMBER INCIDENCES ;nr=0 m=-5 ;:10 ;nr=nr+1 i=0 ;:20 ;i=i+1 m=m+6
j1=nr*nsg+i j2=j1-nsg ;IF i=1 THEN j3=j1-1 j4=j1+nsg-1
IF i>1 THEN j3=j2-1 j4=j1-1 ;m THRU m+5 ELEMENT j1,j2,j3,j4
IF i<nsg GOTO 20 ;IF nr<nrj-1 GOTO 10 ;g=e/(2*(1+nu))
CONSTANTS E e ALL G g ALL ;MEMBER PROPS ;1 THRU nme ELEMENT T h
LOADING CASE 1: INTERNAL FLUID PRESSURE ;MEMBER LOADS ;nr=0 m=-5
gam'=gam/2 p=gam'*de/2 wel=gam'*PI*a/nsg
m=-5 db=d+de nr=0 ;:30 ;nr=nr+1 db=db-de i=0 ;:40 ;i=i+1
m=m+6 m FORCE Z LINEAR WA +(-wel*(db-de)) WB +(-wel*db)
m+2 FORCE Z LINEAR WA -wel*db WB +(-wel*(db-de))
m+1 FORCE Z UNIFORM W -p*db ;m+3 FORCE Z UNIFORM W -p*(db-de)
IF i<nsg GOTO 40 ;IF nr<nrj-1 GOTO 30 ;SOLVE
* Position      Deflection      Deflection by         %age
* ref. No.      by NL-STRESS    Timoshenko et al.     diff.
status=1 gtot=0 nur=0 b=(3*(1-nu^2)/(a^2*h^2))^0.25 x=0-de j=0
:50 ;j=j+nsg x=x+de bx=b*x brac=d*COS(bx)+(d-1/b)*SIN(bx)
tim=-gam*a^2/(e*h)*(d-x-2.71828^(-bx)*brac)
nc=ARR(8,j,2) rc=(nc-1)*6+3 dnls=ARR(6,rc,1) d1=dnls d2=tim
#vmper.ndf !Compute percentage difference & any message.
IF j=nj THEN gtot=gtot-per nur=nur-1 ok=$(Ignored) !Length ∞.
*   +j          +dnls           +tim              $ok
IF j<nj GOTO 50 ;fnm=$(vm650.stk) ;#vmres.ndf !Conclude results.
mjn=49 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

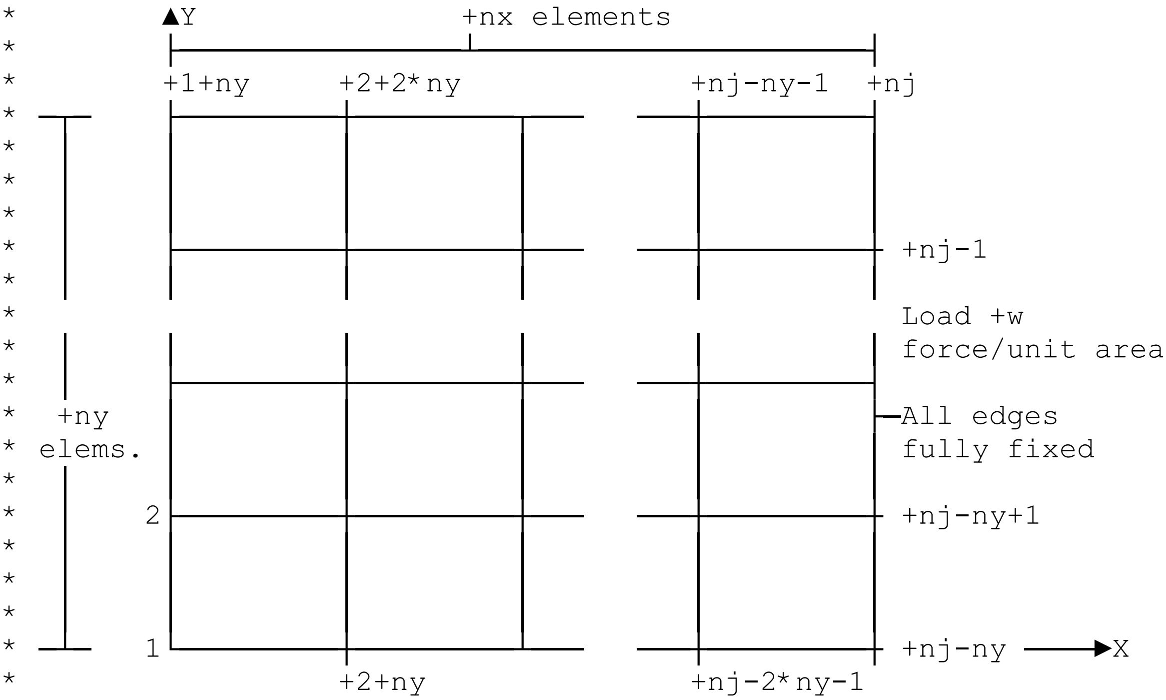
```
TITLE DYNAMICAL BEHAVIOUR OF CANTILEVER OR MULTI-STOREY
TITLE FRAME WITH LUMPED MASSES.  RESULTS COMPARE MATRIX
TITLE STIFFNESS METHOD AND RAYLEIGH, WITH FLEXIBILITY
TITLE AND LATENT ROOT TO GIVE NATURAL FREQUENCY ;MADEBY DWB
DATE 21.11.04 ;PRINT DATA RESULTS FROM 1 ;REFNO VM710
TABULATE DISPLACEMENTS FORCES ;TYPE PLANE FRAME run=0
*/5
* \1                w(2)              w(3)              w(nj)
* \╪════════════════█═════════════════█══   ══════════════█
* \│1               2                 3                  nj
*  ├───────s(1)─────┼──────s(2)───────┼─   ───s(n)───────┤
nj=4   g=386               ! No. of joints; gravitational acceln.
w1=VEC(0,2.5,2.33,0.42)  ! Masses at joints 1-nj.
s(1)=VEC(120,180,100)    ! Spans left to right for 'nj-1' spans.
ax(1)=VEC(36,36,36)      ! Section areas, left to right.
as(1)=VEC(30,30,30)      ! Shear areas, left to right.
iz(1)=VEC(1E3,1E3,1E3)   ! Moments of inertia, left to right.
e=1E4    nu=0.3          ! Young's modulus & Poisson's ratio.
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj       ;NUMBER OF MEMBERS nm=nj-1 nm
NUMBER OF SUPPORTS 1      ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;1 0 0 SUPPORT ;j=1 x=0 ;:10 ;j=j+1 x=x+s(j-1)
j x 0 ;IF j<nj GOTO 10 ;MEMBER INCIDENCES
1 THRU nm RANGE 1,2 nj-1,nj
CONSTANTS E e ALL G sm=e/(2*(1+nu)) sm ALL ;MEMBER PROPERTIES
n=0 ;:30 ;n=n+1 ax=ax(n) ay=as(n) ;IF nu=1E-12 THEN ay=0
n AX ax AY ay IZ iz(n) ;IF n<nm GOTO 30
LOADING DYNAMIC G g ;JOINT LOADS ;j=0 ;:40 ;j=j+1 j FORCE Y w(j)
IF j<nj GOTO 40 ;SOLVE ;status=1 gtot=0 nur=0
a=nm f=nm m=nm y(1)=VEC(1)*nm ;i=0 l=0 ;:70 ;i=i+1 j=0 l=l+s(i)
k=0 del(i)=0 th(i)=0 l'=l ;! Deflection & rotation by area moment
:80 ;k=k+1 l'=l'-s(k) ar=l'*s(k) lr=l'+s(k)/2 at=s(k)^2/2
lt=l'+2*s(k)/3 th(i)=th(i)+(ar+at)/(e*iz(k))
del(i)=del(i)+(ar*lr+at*lt)/(e*iz(k)) ;IF k<i GOTO 80 ;:85 ;j=j+1
IF j<i THEN f(i,j)=f(j,i) ;IF j=i THEN f(i,j)=del(i)
IF j>i THEN j'=j-1 f(i,j)=f(i,j')+s(j)*th(i) ;IF j<a GOTO 85
IF i<a GOTO 70 ;i=0 ;:90 ;i=i+1 j=0 ;:95 ;j=j+1 m(i,j)=0
IF i=j THEN m(i,j)=w(i+1) ;IF j<a GOTO 95 ;IF i<a GOTO 90
i=0 ;:100 ;i=i+1 j=0 ;:105 ;j=j+1 k=0 a(i,j)=0 ;:107 ;k=k+1
a(i,j)=a(i,j)+f(i,k)*m(j,k) ;IF k<a GOTO 107 ;IF j<a GOTO 105
IF i<a GOTO 100 ;k=0 lprev=0 ;:110 ;k=k+1 i=0 ;:120
i=i+1 j=0 ay(i)=0 ;:130 ;j=j+1 ;ay(i)=ay(i)+a(i,j)*y(j)
IF j<a GOTO 130 ;IF i<a GOTO 120 ;i=0 ;:140 ;i=i+1
IF i=1 THEN lambda=ay(1) y(1)=1 ;IF i>1 THEN y(i)=ay(i)/lambda
IF i<a GOTO 140 ;diff=ABS(lambda-lprev) lprev=lambda
IF k<16 AND diff>1E-6 GOTO 110 ;! The period T=2π√(lambda/g).
nf=1/(2*PI*SQR(lambda/g)) nfnls=ARR(12,9,82) d1=nfnls d2=nf ;*/3
#vmper.ndf !Compute percentage difference & any message.
* Natural frequency:  Analysis by       Flexibility &       %age
* Hertz (cycles/sec)  NL-STRESS         latent root         diff.
*                     +nfnls            +nf                 $ok
fnm=$(vm710.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE DYNAMICAL BEHAVIOUR OF FLAT PLATE IN FLEXURE WITH
TITLE WITH BUILT-IN EDGES; RESULTS COMPARE THE MATRIX
TITLE STIFFNESS METHOD AND RAYLEIGH WITH THE FORMULAE
TITLE GIVEN BY ROARK AND WARBURTON. ;MADEBY DWB
DATE 31.01.05 ;PRINT DATA RESULTS FROM 1 ;REFNO VM718
METHOD ELASTIC JOINTS ;TYPE PLANE GRID run=0 ;*/21
! Number of bays (nx,ny) in the x & y directions must be even.
nx=6  ny=6  w=10    ! Number of elements in X & Y directions, UDL.
xe=2  ye=2  t=.25   ! Element sizes in X & Y dirn. & thickness.
e=30E6  nu=.3  g=9.80665  ! Young's mod. Poisson rat., Grav. acc.
#cc924.stk !Import set of parameters if available from cc924.stk.
nj=(nx+1)*(ny+1) ;*/21
```




```
NUMBER OF JOINTS nj ;NUMBER OF MEMBERS nm=6*nx*ny nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1 ;TABULATE DISPLACEMENTS
JOINT COORDINATES ;i=0 j=-ny x=-xe ;:10 ;i=i+1 j=j+ny+1 x=x+xe
j THRU j+ny X x Y 0 YL ny*ye ;IF i<nx+1 GOTO 10 ;JOINT RELEASES
fx=-1 ;1 THRU 1+ny FORCE Z fx MOMENT X fx MOMENT Y fx
nj-ny THRU nj FORCE Z fx MOMENT X fx MOMENT Y fx
2+ny THRU nj-2*ny-1 STEP ny+1 FORCE Z fx MOMENT X fx MOMENT Y fx
2+2*ny THRU nj-ny-1 STEP ny+1 FORCE Z fx MOMENT X fx MOMENT Y fx
* Member incidences are generated by copying a row of elements and
* incrementing the member and joint numbers: 'm' and 'j'.
MEMBER INCIDENCES ;m=-nx*6 j=-1 ;:20 ;m=m+nx*6 j=j+1
m+1 THRU m+nx*6 ELEMENT j+ny+3,j+ny+2,j+1,j+2 BUMP ny+1
IF j<ny-1 GOTO 20 ;CONSTANTS E e ALL G e/(2*(1+nu)) ALL
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t
LOADING DYNAMIC G g ;JOINT LOADS ;nst=nx ;IF ny>nx THEN nst=ny
w=ABS(w) 1 AREA nj FORCE Z w nst
SOLVE ;status=1 ;lx=nx*xe ly=ny*ye nfn=ARR(12,9,83) ;IF lx<>ly
* Verification omitted; formulae only for built-in square slabs.
STOP ;ENDIF ;nfw=36/(2*PI*SQR((w*lx^4*12*(1-nu^2))/(e*t^3*g)))
nfr=34.01*SQR(e*t^3*(g/386)/(w*lx^4)) d1=nfn d2=nfw gtot=0 nur=0
*/5
#vmper.ndf !Compute percentage difference & any message.
*                  NL-STRESS     Roark's **    Warburton's   %age
*                  Analysis      formula       formula       diff.
* Nat.freq. Hertz  +nfn          +nfr          +nfw          $ok
* ;** Roark's formula omits Poisson's ratio, therefore ignored.
fnm=$(vm718.stk) ;#vmres.ndf !Conclude results.
mjn=9 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE DYNAMICAL BEHAVIOUR OF SIMPLY SUPPORTED RECTANGULAR
TITLE FLAT PLATE IN FLEXURE.  RESULTS COMPARE THE MATRIX
TITLE STIFFNESS METHOD WITH THOSE BY NAVIER, FLEXIBILITY
TITLE AND LATENT ROOT TO GIVE NATURAL FREQUENCY ;MADEBY DWB
DATE 01.02.05 ;PRINT DATA RESULTS FROM 1 ;REFNO VM720
METHOD ELASTIC JOINTS ;TYPE PLANE GRID run=0
! Number of bays (nx,ny) in the x & y directions must be even.
nx=6  ny=6  w=10    ! Number of elements in X & Y directions, UDL.
xe=2  ye=2  t=.25   ! Element sizes in X & Y dirn. & thickness.
e=30E6  nu=.2  g=9.80665  ! Young's mod. Poisson rat., Grav. acc.
#cc924.stk ! Import set of parameters if available from cc924.stk.
nj=(nx+1)*(ny+1) ;*/21
```




```
NUMBER OF JOINTS nj  ;NUMBER OF MEMBERS nm=6*nx*ny nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1 ;TABULATE DISPLACEMENTS
JOINT COORDINATES ;i=0 j=-ny x=-xe ;:10 ;i=i+1 j=j+ny+1 x=x+xe
j THRU j+ny X x Y 0 YL ny*ye ;IF i<nx+1 GOTO 10 ;JOINT RELEASES
fx=-1 ;1 THRU 1+ny FORCE Z fx MOMENT X fx
nj-ny THRU nj FORCE Z fx MOMENT X fx
2+ny THRU nj-2*ny-1 STEP ny+1 FORCE Z fx MOMENT Y fx
2+2*ny THRU nj-ny-1 STEP ny+1 FORCE Z fx MOMENT Y fx
1 MOMENT Y fx ;nj-ny MOMENT Y fx ;1+ny MOMENT Y fx ;nj MOMENT Y fx
* Member incidences are generated by copying a row of elements and
* incrementing the member and joint numbers: 'm' and 'j'.
MEMBER INCIDENCES ;m=-nx*6 j=-1 ;:20 ;m=m+nx*6 j=j+1
m+1 THRU m+nx*6 ELEMENT j+ny+3,j+ny+2,j+1,j+2 BUMP ny+1
IF j<ny-1 GOTO 20 ;CONSTANTS E e ALL G e/(2*(1+nu)) ALL
MEMBER PROPERTIES ;1 THRU nm ELEMENT T t
LOADING DYNAMIC G g ;JOINT LOADS ;w=ABS(w) nst=nx
IF ny>nx THEN nst=ny ;1 AREA nj FORCE Z w nst
SOLVE ;status=1 gtot=0 nur=0 nfn=ARR(12,9,83) ! Set constants
a=xe*nx b=ye*ny d=e*t^3/(12*(1-nu^2)) vb=(nx-1)*(ny-1)
p=1 pc=4*p/(PI^4*a*b*d) k=0 i=0 ;:25 ;i=i+3 f'=ARR(6,i,1)
IF f'<>0 THEN k=k+1 j=i/3 vc(k)=ARR(8,j,3) vd(k)=ARR(8,j,4)
IF k<vb GOTO 25 ;i=0 ;:30 ;i=i+1 j=0 ;:35 ;j=j+1
IF j<i THEN vb(i,j)=vb(j,i) ;IF j>=i ;del=0 x=vc(i) y=vd(i)
x'=vc(j) y'=vd(j) m=0 ;:40 ;m=m+1 n=0 ;:45 ;n=n+1
num=SIN(m*PI*x'/a)*SIN(n*PI*y'/b)*SIN(m*PI*x/a)*SIN(n*PI*y/b)
del=del+pc*num/(m^2/a^2+n^2/b^2)^2 ;IF n<11 GOTO 45
IF m<11 GOTO 40 ;vb(i,j)=del ;ENDIF ;IF j<vb GOTO 35
IF i<vb GOTO 30 ;vc=vb va=vb y(1)=VEC(1)*vb i=0 ;:90 ;i=i+1 j=0
:95 ;j=j+1 va(i,j)=0 ;IF i=j THEN va(i,j)=w*xe*ye ;IF j<vc GOTO 95
```

```
IF i<vc GOTO 90 ;i=0 ;:100 ;i=i+1 j=0 ;:105 ;j=j+1 k=0 vc(i,j)=0
:107 ;k=k+1 vc(i,j)=vc(i,j)+vb(i,k)*va(j,k) ;IF k<vc GOTO 107
IF j<vc GOTO 105 ;IF i<vc GOTO 100 ;k=0 lprev=0 ;:110 ;k=k+1 i=0
:120 ;i=i+1 j=0 ay(i)=0 ;:130 ;j=j+1 ;ay(i)=ay(i)+vc(i,j)*y(j)
IF j<vc GOTO 130 ;IF i<vc GOTO 120 ;i=0 ;:140 ;i=i+1
IF i=1 THEN lambda=ay(1) y(1)=1 ;IF i>1 THEN y(i)=ay(i)/lambda
IF i<vc GOTO 140 ;diff=ABS(lambda-lprev) lprev=lambda
IF k<16 AND diff>1E-12 GOTO 110 ;! The period T=2π√(lambda/g).
nff=1/(2*PI*SQR(lambda/g)) d1=nfn d2=nff ;*/3
#vmper.ndf !Compute percentage difference & any message.
*                  NL-STRESS          Navier                %age
*                  Analysis           Solution              diff.
* Nat.freq. Hertz  +nfn               +nff                  $ok
fnm=$(vm720.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CANTILEVER WITH LARGE DISPLACEMENTS SUBJECTED TO
TITLE UNIFORMLY DISTRIBUTED LOADING & END VERTICAL LOADS,
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 14.11.05 ;TYPE PLANE FRAME run=0 ;REFNO VM802
PRINT DATA, RESULTS, FROM 1 ;METHOD SWAY ;NUMBER OF JOINTS 3
NUMBER OF MEMBERS 2 ;NUMBER OF SUPPORTS 1 ;NUMBER OF LOADINGS 1
*/8
sp=1.91  nsg=4   ! Span.                Span is divided into 2*nsg
ax=38.2E-4       ! Area of section.   segments of length sp/2/nsg.
ay=0             ! Shear area.            Y +ve               ▲
iz=558E-8        ! Moment of inertia.     │        ┌w         │v
e=206E6          ! Young's modulus.      /│▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
nu=0.3           ! Poisson's ratio.      /╞═════════════════════  ──►h
nli=100          ! Number of increments. /│1         2           3
w=0              ! Load/unit length.      ├──────────sp──────────┤
v=1     h=-1000  ! Vert. & horizontal loads at end of cantilever.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF INCREMENTS nli 0.1 200 ;NUMBER OF SEGMENTS nsg TRACE
JOINT COORDINATES ;1 0 0 SUPPORT ;2 sp/2 0 ;3 sp 0
MEMBER INCIDENCES ;1 1 2 ;2 2 3
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;1 THRU 2 AX ax AY ay IZ iz
LOADING CASE 1 ;JOINT LOADS ;3 FORCE X h FORCE Y v
IF w<>0 ;MEMBER LOADS ;1 THRU 2 FORCE Y UNIFORM W w ;ENDIF
SOLVE ;va1=VEC(0)*2 vc1=VEC(w)*2 hjl1=VEC(0,0,h) vjl1=VEC(0,0,v)
ch9=0 nn=ARR(8,3,2) nr=(nn-1)*3+2 nl10=ARR(6,nr,nli) ch10=0
#vmecp.ndf !Equilibrium, compat. & energy checks.
fnm=$(vm802.stk) ;#vmres.ndf !Conclude results.
mjn=3 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE SWAY & WITHIN-MEMBER STABILITY OF COLUMNS SUBJECTED
TITLE TO AXIAL LOAD, FOR VARIOUS SUPPORT CONDITIONS.
TITLE BUCKLING LOAD FOR VARIOUS SUPPORT CONDITIONS COMPARED
TITLE TO FORMULAS FOR STRESS & STRAIN BY ROARK.
MADEBY DWB ;DATE 12.01.05 ;TYPE PLANE FRAME run=0 ;REFNO VM810
METHOD SWAY ;TABULATE DISPL REACTIONS ;PRINT DATA, RESULTS FROM 1
! Notes follow FINISH. ;*/7
l=18 nseg=16  bow=0.5  ! Length, No. of segments, bow percent. |p
rb=-1    rt=-1         ! Rotn. stiffness bot & top, -1=fixed.  ▼
ht=-1                  ! Horiz. stiffness top, -1=fixed.    ——2
e=205E6   nu=0.3       ! Young's modulus & shear modulus.    |  ||
st=2   ! st=sectn type: 1=rect 2=Isect 3=Hsect 4=rhs 5=chs.  |l ||ei
dy=.6 dz=.6 ty=.02 tz=.02  ! Section dims & thicknesses.     |  ||
n=100                  ! No. of load increments, 50-500.    ——1——X
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS 2 ;NUMBER OF MEMBERS 1 ;NUMBER OF SUPPORTS 0
! See JOINT RELEASES table for supports. ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS n !Add TRACE to list increments to collapse.
NUMBER OF SEGMENTS 8 bow
JOINT COORDINATES ;1 0 0 ;2 0 l ;JOINT RELEASES ;!Fix instability.
IF ht=0 AND rt=0 AND rb=0 THEN rb=-1 ;tfix=-(rb+rt+2*(-1+ht))
k=.25 ;IF tfix>3 THEN k=1 ;IF tfix>4 THEN k=2 ;IF tfix>5 THEN k=4
1 FORCE X -1 FORCE Y -1 MOMENT Z rb ;2 FORCE X ht MOMENT Z rt
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER INCIDENCES ;1 1 2
MEMBER PROPERTIES ;d1=dy-2*ty d2=dz-2*tz i'=2*tz*dy^3/12
ir=dz*dy^3/12 ;IF st=1 THEN 1 RECT DY dy DZ dz i=ir
IF st=2 THEN 1 ISECT DY dy DZ dz TY ty TZ tz i=ir-(dz-tz)*d1^3/12
IF st=3 THEN 1 HSECT DY dy DZ dz TY ty TZ tz i=i'+d2*ty^3/12
IF st=4 THEN 1 RECT DY dy DZ dz TY ty TZ tz i=ir-d2*d1^3/12
IF st=5 THEN 1 CONIC D dy T ty i=PI*(dy^4-d1^4)/64
p=-1.2*k*PI^2*e*i/l^2
LOADING CASE 1 ;JOINT LOADS ;2 FORCE Y p X p/1E6 ;SOLVE
incn=ARR(12,4,2) fy1=ARR(14,2,incn) !Inc.No. & supp. reaction.
status=1 gtot=0 nur=0 ;IF rb=-1 AND rt=0 AND ht=0 ;*/11
*         |p'          Uniform straight bar under end load
*         ▼            ___________________________________
* ——      o            Top end free, bottom end fixed.
*  |     /
*  |    |
*  | l  |ei            Buckling load
*  |    |
*  |    |              +p'=PI^2*e*i/(4*l^2)
* ——   ——
*      ///             cf. +fy1 from NL-STRESS ;ENDIF
IF rb=0 AND rt=-1 AND ht=0 ;*/11
*         |p'          Uniform straight bar under end load
*         ▼            ___________________________________
*        ///           Top end fixed rotationally but
* ——     -|-           not restrained against sway.
*  |      |
*  |      |            Buckling load
*  | l    |ei
*  |      |            +p'=PI^2*e*i/(4*l^2)
*  |     /
* ——   -o-
*      ///             cf. +fy1 from NL-STRESS ;ENDIF
IF rb=-1 AND rt=-1 AND ht=-1 ;*/11
*         |p'          Uniform straight bar under end load
*         ▼            ___________________________________
```

```
*        ///            Both ends fixed.
* ─┬─    ─┼─
*  │      /
*  │     │              Buckling load
*  │ l   │ei
*  │     │              +p'=4*PI^2*e*i/l^2
*  │      \
* ─┴─    ─┼─            cf. +fy1 from NL-STRESS
*        /// ;ENDIF
IF rb=0 AND rt=0 AND ht=-1 ;*/10
*         │p'           Uniform straight bar under end load
*         ▼             ───────────────────────────────────
* ─┬─    ─o─            Both ends pinned.
*  │      /
*  │     │
*  │ l   │ei            Buckling load
*  │     │
*  │      \             +p'=PI^2*e*i/l^2
* ─┴─    ─o─
*        ///            cf. +fy1 from NL-STRESS ;ENDIF
IF rb=-1 AND rt=-1 AND ht=0 ;*/11
*           │p'         Uniform straight bar under end load
*           ▼           ───────────────────────────────────
*          ///          Both ends fixed, but sway at top
* ─┬─      ─┼─          not prevented.
*  │        /
*  │       │
*  │ l     │ei          Buckling load
*  │       │
*  │      /             +p'=PI^2*e*i/l^2
* ─┴─  ─┼─
*      ///              cf. +fy1 from NL-STRESS ;ENDIF
IF rb=-1 AND rt=0 AND ht=-1 ;*/10
*         │p'           Uniform straight bar under end load
*         ▼             ───────────────────────────────────
* ─┬─     o             One end fixed, other end hinged and
*  │      /             horizontally constrained at fixed
*  │     │              end.
*  │ l   │ei
*  │      \             Buckling load
*  │      │             +p'=PI^2*e*i/(0.7*l)^2
* ─┴─    ─┴─
*        ///            cf. +fy1 from NL-STRESS ;ENDIF
IF rb=0 AND rt=-1 AND ht=-1 ;*/10
*         │p'           Uniform straight bar under end load
*         ▼             ───────────────────────────────────
* ─┬─    ─┼─            One end fixed, other end hinged and
*  │      │             horizontally constrained at fixed
*  │      /             end.
*  │ l   │ei
*  │     │              Buckling load
*  │      \             +p'=PI^2*e*i/(0.7*l)^2
* ─┴─    ─o─
*        ///            cf. +fy1 from NL-STRESS ;ENDIF
d1=fy1 d2=p'
#vmper.ndf
*                NL-STRESS     Classical     %age
*                stiffness     formula       diff.
* Critical load  +fy1          +p'           $ok
fnm=$(vm810.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=81 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE WITHIN-MEMBER STABILITY OF CIRCULAR RING OR PIPE
TITLE SUBJECTED TO UNIFORM EXTERNAL RADIAL PRESSURE.
TITLE BUCKLING LOAD COMPARED TO THAT GIVEN BY ROARK
TITLE IN FORMULAS FOR STRESS & STRAIN, FOURTH EDITION.
MADEBY DWB ;DATE 12.03.05 ;TYPE PLANE FRAME run=0 ;REFNO VM830
METHOD SWAY ;TABULATE DISPL REACTIONS ;PRINT DATA, RESULTS FROM 1
n=200     ! Number of loading increments.      _|_ ┌Joints are
r=0.5     ! Radius of pipe.                /    |Y 1 \  clockwise
t=0.025   ! Thickness of pipe.                  |          from 1
l=0.05    ! Length of pipe Z direction.  ─┼─────Z─────┼─X  to nsg.
nsg=32    ! No. elements in ring (even).        |out
e=28E6    ! Young's modulus.               \   _|_   /
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nsg ;NUMBER OF MEMBERS nsg ;NUMBER OF SUPPORTS 0
NUMBER OF LOADINGS 1 ;NUMBER OF INCREMENTS n ;JOINT COORDINATES
j=0 ainc=RAD(360/nsg) a=PI/2 ;:40 ;j=j+1 a=a-ainc
j r*COS(a) r*SIN(a) ;IF j<nsg GOTO 40 ;MEMBER INCIDENCES
1 THRU nsg-1 RANGE 1,2 nsg-1,nsg ;nsg nsg 1 ;JOINT RELEASES
ax=t*l iz=l*t^3/12 p'=3*e*iz/r^3 p=p'*2 ;nsg/2 FORCE X -1 Y -1
nsg FORCE X p ;CONSTANTS E e ALL
MEMBER PROPERTIES ;1 THRU nsg AX ax IZ iz ;rf=p*2*PI*r/nsg
LOADING ;JOINT LOADS ;j=0 ainc=RAD(360/nsg) a=PI/2 ;:70 ;j=j+1
a=a-ainc j FORCE X -rf*COS(a) Y -rf*SIN(a) ;IF j<nsg GOTO 70
SOLVE ;status=1 gtot=0 nur=0 incn=ARR(12,4,2) d1=p*incn/n d2=p'
#vmper.ndf
*                  NL-STRESS     Classical     %age
*                  stiffness     formula       diff.
* Buckling load    +p*incn/n     +p'           $ok
fnm=$(vm830.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=105 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CANTILEVER BEAM SUBJECTED TO UNIFORMLY DISTRIBUTED
TITLE LOADING & END POINT LOADS.  STABILITY ANALYSIS
TITLE CHECKED BY AUDIT OF STRAIN ENERGY & EXTERNAL WORK,
TITLE EQUILIBRIUM AND COMPATIBILITY BY CLASSICAL THEORY.
MADEBY DWB ;DATE 24.12.04 ;TYPE PLANE FRAME run=0 ;REFNO VM850
PRINT DATA, RESULTS, FROM 1 ;*/8
sp=3.0          ! Span of cantilever.  Span is divided into 2*nsg
nsg=16          ! No. of segments.     segments of length sp/2/nsg.
dy=0.36         ! Depth of beam.          Y +ve           |v
dz=0.3          ! Breadth of beam.        |     ┌w         ▼
e=28E6/3        ! Young's modulus.      / |▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
nu=0.2          ! Poisson's ratio.      /═════════════════  ──► h
w=-36           ! Load/unit length.     / |1      2       3
v=-80  h=-50    ! End point loads. .      ├───────sp──────┤
nli=10          ! No. of load increments, typically 10-40.
#cc924.stk !Import verification data from cc924.stk if available.
IF nli>1 THEN METHOD SWAY ;NUMBER OF JOINTS nj=3 nj
NUMBER OF MEMBERS 2 ;NUMBER OF SUPPORTS 1 ;NUMBER OF LOADINGS 3
IF nli>1 THEN NUMBER OF INCREMENTS nli
NUMBER OF SEGMENTS nsg ;JOINT COORDINATES ;1 0 0 SUPPORT ;2 sp/2 0
3 sp 0 ;MEMBER INCIDENCES ;1 1 2 ;2 2 3
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
1 THRU 2 RECTANGLE DY dy DZ dz ay=dz*dy*5/6 iz=dz*dy^3/12
LOADING CASE 1 ;MEMBER LOADS ;1 THRU 2 FORCE Y UNIFORM W w
JOINT LOADS ;3 FORCE Y v FORCE X h
LOADING CASE 2 ;TABULATE ;JOINT LOADS ;ch9=1 a=1 nj=3 jn=0 ;:18
jn=jn+1 jn FORCE X jn*a Y jn*a MOMENT Z jn*a ;IF jn<nj GOTO 18
LOADING CASE 3 ;TABULATE ;JOINT LOADS ;jn=0 ;:19 ;jn'=nj-jn
jn=jn+1 jn FORCE X jn'*a Y jn'*a MOMENT Z jn'*a ;IF jn<nj GOTO 19
SOLVE ;va1=VEC(0)*2 vc1=VEC(w)*2 hjl1=VEC(0)*3 vjl1=VEC(0,0,v)
ch10=0 ;IF nli=1 ;nn=ARR(8,3,2) nr=(nn-1)*3+2 nl10=ARR(6,nr,1)
ch10=w*sp^4/(8*e*iz)+w*sp/(2*ay*g)+v*sp^3/(3*e*iz)+v*sp/(ay*g)
dir=$(Y) ;ENDIF ;IF nli>1 AND w=0 AND h=0
nn=ARR(8,3,2) nr=(nn-1)*3+1 nl10=ARR(6,nr,nli)
ch10=v^2*sp^5*(1-2*v^2*sp^4/(63*e^2*iz^2))/(15*e^2*iz^2)
ch10=v^2*sp^3/(3*e^2*iz*dy*dz)-ch10 dir=$(X) ;ENDIF
#vmecp.ndf !Equilibrium, compatibility & energy checks.
IF ch10<>0
* Displacement at joint 3 in $dir  +nl10        +ch10        $ok
ENDIF ;fnm=$(vm850.stk) ;#vmres.ndf !Conclude results.
mjn=2 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE NON-LINEAR ELASTIC ANALYSIS OF MULTI-STOREY FRAME
TITLE SUBJECTED TO UDL & VERTICAL & HORIZONTAL POINT LOADS
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 12.11.05 ;TYPE PLANE FRAME run=0 ;REFNO VM852
PRINT DATA, RESULTS FROM 1 ;METHOD SWAY
nb=1   ns=2   ! No. of bays & storeys.        ╔══════════╦═════   ═══╗
s=3.82 h=3.82 ! Bay span & storey height.     ║          ║nj-1    nj║
axb=38.2E-4   ! X-sect. area of beams.     ▲
ayb=0         ! Shear area of beams.       │  ║          ║          ║
izb=558E-8    ! Mom. of inert. of beams. ──┼─ ╠══════════╬═════   ═══╣
dyb=0.1529    ! Depth of beams.            │  ║          ║          ║
axc=38.2E-4   ! X-sect. area of cols.      h  ║Y         ║          ║
ayc=0         ! Shear area of columns.     │  ║          ║          ║
izc=558E-8    ! Mom. of inert. of cols. ───┼─ ╠══════════╬═════   ═══╣
dyc=0.1529    ! Depth of columns.          │  ║nb+2      ║nb+3 etc. ║
e=206E6       ! Young's modulus.           h  ║ └joint numbers.     ║
nu=0.3        ! Poisson's ratio.           │  ║          ║          ║
fix=-1        ! Bases, 0=pin, -1=fixed. ───┴─ 1╨ ───X    2╨      nb+1╨
nsg=8         ! Number of segments.           ├─────s────┼►►nb bays.
nli=100       ! No. of load increments.
hjl1=VEC(0,0,5,0,5,0) ! Hor. joint loads, l. to r., bot. to top.
vjl1=VEC(0,0,-500,-500,-500,-500) ! Vir. loads, l to r, b to t.
udl=0     nc=0  ! Udl on all beams; No. of conc. loads on members.
! Members are numbered left to right ground floor columns, left to
! right first floor beams, left to right 1st floor cols and so on.
IF nc>0
nc(1)=VEC(3,6)              !  ├─cs(n)─▲cn(n)   ┌n'th load occurs
cs(1)=VEC(1.91)*2           !  ╪═══════════════════╪ on member nc(n).
cn(1)=VEC(-80)*2            !  lower or left      higher or right.
ENDIF
#cc924.stk !Import set of parameters if available from cc924.stk.
NUMBER OF JOINTS nj=(nb+1)*(ns+1) nj ;nm=ns*(2*nb+1)
NUMBER OF MEMBERS nm ;NUMBER OF LOADINGS 1 ;NUMBER OF SUPPORTS 0
NUMBER OF INCREMENTS nli ;NUMBER OF SEGMENTS nsg
JOINT COORDINATES ;n=0 j=-nb ;:40 ;n=n+1 j=j+nb+1
j THRU j+nb X 0 Y h*(n-1) XL nb*s ;IF n<ns+1 GOTO 40
JOINT RELEASES ;1 THRU nb+1 FORCE X -1 FORCE Y -1 MOMENT Z fix
MEMBER INCIDENCES
n=0 j=-nb m=-2*nb ;:50 ;n=n+1 j=j+nb+1 m=m+2*nb+1
m+nb+1 THRU m+2*nb RANGE j+nb+1 j+nb+2 j+2*nb j+2*nb+1
m THRU m+nb RANGE j j+nb+1 j+nb j+2*nb+1 ;IF n<ns GOTO 50
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;n=0 m=-2*nb ;:70 ;n=n+1  m=m+2*nb+1
m+nb+1 THRU m+2*nb AX axb AY ayb IZ izb CY dyb/2
m THRU m+nb AX axc AY ayc IZ izc CY dyc/2 ;IF n<ns GOTO 70
LOADING CASE 1 ;MEMBER LOADS ;n=0 m=-2*nb ;:48 ;n=n+1  m=m+2*nb+1
m+nb+1 THRU m+2*nb FORCE Y UNIFORM W udl
IF n<ns GOTO 48 ;i=0 ;:52 ;i=i+1 ;IF i>nc GOTO 53
nc(i) FORCE Y CONCENTRATED P cn(i) L cs(i) ;GOTO 52 ;:53
JOINT LOADS ;i=0 ;:56 ;i=i+1 ;IF hjl(i)<>0 THEN i FORCE X hjl(i)
IF vjl(i)<>0 THEN i FORCE Y vjl(i) ;IF i<nj GOTO 56
SOLVE ;status=1E-36 !Tells vmecp point loads on members
nb'=nb+1 n=0 m=-2*nb ;:67 ;n=n+1 m=m+2*nb+1 m'=m+nb+1
s(m')=VEC(s)*nb ct(m')=VEC(0)*nb va(m')=VEC(0)*nb
vc(m')=VEC(udl)*nb s(m)=VEC(h)*nb' ct(m)=VEC(0)*nb'
va(m)=VEC(0)*nb vc(m)=VEC(0)*nb ;IF n<ns GOTO 67 ;ch9=0 ch10=0
#vmecp.ndf !Equilibrium, compatibility & energy checks.
fnm=$(vm852.stk) ;#vmres.ndf !Conclude results.
mjn=nb+2 lcn=1 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE CABLE OF NEGLIGIBLE WEIGHT, CARRIES BY SUSPENSION
TITLE RODS, A FLEXIBLE PLATFORM UPON WHICH THERE IS A UDL.
TITLE COMPARED WITH THE ANALYSIS OF ENGINEERING STRUCTURES,
TITLE BY PIPPARD & BAKER, THIRD EDITION, 1957. ;MADEBY DWB
METHOD SWAY ;DATE 26.03.05 ;TYPE PLANE FRAME run=0 ;REFNO VM950
PRINT DATA, RESULTS FROM 1 ;*/10
*  ┌nsg+2                                                   2*nsg+2┐
*  o  _                             ──┬──                              _  o──┬──
*  │    │  _                          │d                        _  │    │   │
*  │    │    │ ‾  _                   │                   _  ‾ │    │    │   │
*  │    │    │       │  _   ┌─   _  ──┴──  _   ─┐  _   │       │    │    │   │h
*  │ ┌w │    │       │     │     │     │     │      │       │    │ ┌w │   │
*  │▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│   │
*  ■════■════■════■════■════■════■════■════■════■════■════■════■───┴──
*  1    2    3..    └Flexible platform.    ├────┼segment     nsg+1
*  ├──────────────────────────── l ────────────────────────────┤
l=106.68  d=10.668  h= 15  ! Span, centre dip d, overall height h.
dc=0.5 dsr=0.097 w=-32.6903  ! Diameter of cable & hangers; & udl.
e=205E6 ec=205E6   ! Young's modulus for platform/hangers, cable.
ax=1 iz=1 nsg=12   ! Platform area, inertia; Even No. of segments.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*(1+nsg) nj ;NUMBER OF MEMBERS nm=3*nsg+1 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS nli=10 nli
JOINT COORDINATES ;1 THRU 1+nsg X 0 Y 0 XL l ;j=1+nsg inc=l/nsg
x=-inc ;:10 ;j=j+1 x=x+inc y=h-4*d*x/l^2*(l-x) j x y
IF j<nj GOTO 10 ;JOINT RELEASES ;1 nsg+1 INCLUSIVE MOMENT Z -1
nsg/2+1 FORCE X -1 ;nsg+2 2*nsg+2 INCLUSIVE FORCE X -1 Y -1
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nsg,nsg+1
nsg+1 THRU 2*nsg RANGE nsg+2,nsg+3 2*nsg+1,2*nsg+2
2*nsg+1 THRU nm RANGE 1,nsg+2 1+nsg,2*nsg+2
MEMBER RELEASES ;1 THRU nsg START MOMENT Z END MOMENT Z
2*nsg+1 THRU nm END MOMENT Z ;CONSTANTS E ec nsg+1 THRU 2*nsg
CONSTANTS E e ALL ;MEMBER PROPERTIES ;1 THRU nsg AX ax IZ iz
nsg+1 THRU 2*nsg AX PI*dc^2/4 IZ PI*dc^4/64
2*nsg+1 THRU nm AX PI*dsr^2/4 IZ PI*dsr^4/64
LOADING CASE 1 ;MEMBER LOADS ;1 THRU nsg FORCE Y UNIFORM W w
SOLVE ;status=1 gtot=0 nur=0 j=3*(nsg+1)+1 nl1=ARR(14,j,nli) j=j+1
nl2=ARR(14,j,nli) m=nm*(nli-1)+nsg+1 nl3=ARR(13,m,1)
pb1=w*l^2/(8*d) pb2=-w*l/2 pb3=pb1*SQR(1+16*d^2/l^2)
*/6
* Location                      NL-STRESS     Classical       %age
*                               analysis      analysis        diff.
d1=nl1 d2=pb1
#vmper.ndf !Compute percentage difference & any message.
* Horiz. force at supports      +nl1          +pb1            $ok
d1=nl2 d2=pb2
#vmper.ndf
* Vert.  force at supports      +nl2          +pb2            $ok
d1=nl3 d2=pb3
#vmper.ndf
* Tens. in cable at supports    +nl3          +pb3            $ok
fnm=$(vm950.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=10 tot=3 drn=1 ;#vmtes.ndf
< ;FINISH
```

```
TITLE SUSPENSION BRIDGE WITH THREE-PINNED STIFFENING
TITLE GIRDER.  RESULTS COMPARE THE MATRIX STIFFNESS
TITLE METHOD WITH THE ANALYSIS OF ENGINEERING STRUCTURES,
TITLE PIPPARD & BAKER, THIRD EDITION, 1957. ;MADEBY DWB
METHOD SWAY ;DATE 26.03.05 ;TYPE PLANE FRAME run=0 ;REFNO VM951
PRINT DATA, RESULTS FROM 1 ;*/10
* ┌nsg+2                                                2*nsg+2┐
* o  _    ┌cable                 ─┬─                         _ o─┬─
* │   │   ┴                       │d                   _  │  │ │
* │   │    ┬ _                    │                  ┬  │  │ │
* │   │    │   │ _   ┌─  _       ─┴─ _    ─┐ _   _  │  │  │ │ h
* │ ┌w │   │   │   │   │   │   │   │   │   │   │  ┌w │ │
* │▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│ │
* o═══════════════════════════════o═══════════════════════════o─┴─
* 1    2    3..    └Three pinned stiffening girder          nsg+1
* ├──────────────────────────── l ──────────────────────────────┤
l=106.68  d=10.668  h= 15  ! Span, centre dip d, overall height h.
dc=0.5 dsr=0.097 w=-32.6903  ! Dia. cable & hangers; w=udl.
e=205E6 ec=205E6   ! Young's modulus for platform/hangers, cable.
ax=1 iz=2.08 nsg=12  ! Girder area, inertia; Even No. of segments.
njl=2 jn(1)=VEC(4,8)  ! No. of joint loads & loaded joint numbers.
jl(1)=VEC(-3487.404)*2  ! Magnitude of njl loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*(1+nsg) nj ;NUMBER OF MEMBERS nm=3*nsg+1 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS nli=10 nli
JOINT COORDINATES ;1 THRU 1+nsg X 0 Y 0 XL l ;j=1+nsg inc=l/nsg
x=-inc ;:10 ;j=j+1 x=x+inc y=h-4*d*x/l^2*(l-x) j x y
IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1
nsg+2 2*nsg+2 INCLUSIVE FORCE X -1 Y -1
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nsg,nsg+1
nsg+1 THRU 2*nsg RANGE nsg+2,nsg+3 2*nsg+1,2*nsg+2
2*nsg+1 THRU nm RANGE 1,nsg+2 1+nsg,2*nsg+2 ;MEMBER RELEASES
nsg/2 END MOMENT Z ;2*nsg+1 THRU nm START MOMENT Z END MOMENT Z
CONSTANTS E ec nsg+1 THRU 2*nsg ;CONSTANTS E e ALL
MEMBER PROPERTIES ;1 THRU nsg AX ax IZ iz
nsg+1 THRU 2*nsg AX PI*dc^2/4 IZ PI*dc^4/64/2
2*nsg+1 THRU nm AX PI*dsr^2/4 IZ PI*dsr^4/64
LOADING CASE 1 ;MEMBER LOADS
IF w<>0 THEN 1 THRU nsg FORCE Y UNIFORM W w
IF njl=0 GOTO 20 ;JOINT LOADS ;i=0 ;:15 ;i=i+1 jn(i) FORCE Y jl(i)
IF i<njl GOTO 15 ;:20 ;SOLVE ;status=1 gtot=0 nur=0 j=3*(nsg+1)+1
nl1=ARR(14,j,nli) j=j+1 nl2=ARR(14,j,nli) m=nm*(nli-1)+nsg+1
nl3=ARR(13,m,1) we=w va=w*l/2 ;IF njl=0 GOTO 40 ;i=0 ;:30 ;i=i+1
x=(jn(i)-1)*l/nsg jl=jl(i) va'=jl-jl*x/l ;IF x>l/2 THEN x=l-x
we=we+4*jl*x/l^2 va=va+va' ;IF i<njl GOTO 30 ;:40
pb1=we*l^2/(8*d) pb2=-va pb3=pb1*SQR(1+16*d^2/l^2) ;*/6
* Location                        NL-STRESS     Classical       %age
*                                 analysis      analysis        diff.
d1=nl1 d2=pb1
#vmper.ndf !Compute percentage difference & any message.
* Horiz. force at supports        +nl1          +pb1            $ok
d1=nl2 d2=pb2
#vmper.ndf
* Vert. force at left support     +nl2          +pb2            $ok
d1=nl3 d2=pb3
#vmper.ndf
* Cable tens. at left support     +nl3          +pb3            $ok
fnm=$(vm951.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=10 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

```
TITLE SUSPENSION BRIDGE WITH TWO-PINNED STIFFENING
TITLE GIRDER.  RESULTS COMPARE THE MATRIX STIFFNESS
TITLE METHOD WITH THE ANALYSIS OF ENGINEERING STRUCTURES,
TITLE PIPPARD & BAKER, THIRD EDITION, 1957. ;MADEBY DWB
METHOD SWAY ;DATE 01.04.05 ;TYPE PLANE FRAME run=0 ;REFNO VM952
PRINT DATA, RESULTS FROM 1 ;*/10
*  ┌nsg+2                                               2*nsg+2┐
*  o  _    ┌cable              ───                            _  o───
*  │    │  ┴                    │d                       _    │  │ │
*  │    │    │ ─                │                  ─   │    │  │ │
*  │    │    │    │   _         │            _   │     │    │  │ │
*  │    │    │    │   ┌──   _  ─┴─  _   ──┐    │     │    │    │  │ │h
*  │ ┌w │    │    │   │    │    │    │    │    │     │    │ ┌w │  │ │
*  │▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│▼▼▼▼│  │ │
*  o═══════════════════════════════════════════════════════════o───
*  1    2    3..    └Two pinned stiffening girder             nsg+1
*  ├─────────────────────────── l ─────────────────────────────┤
l=106.68  d=10.668  h= 15  ! Span, centre dip d, overall height h.
dc=0.5 dsr=0.097 w=-32.6903  ! Dia. cable & hangers; w=udl.
e=205E6 ec=205E6   ! Young's modulus for platform/hangers, cable.
ax=1 iz=2.08 nsg=12  ! Girder area, inertia; Even No. of segments.
njl=2 jn(1)=VEC(4,8)  ! No. of joint loads & loaded joint numbers.
jl(1)=VEC(-3487.404)*2  ! Magnitude of njl loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj=2*(1+nsg) nj ;NUMBER OF MEMBERS nm=3*nsg+1 nm
NUMBER OF SUPPORTS 0 ;NUMBER OF LOADINGS 1
NUMBER OF INCREMENTS nli=10 nli
JOINT COORDINATES ;1 THRU 1+nsg X 0 Y 0 XL l ;j=1+nsg inc=l/nsg
x=-inc ;:10 ;j=j+1 x=x+inc y=h-4*d*x/l^2*(l-x) j x y
IF j<nj GOTO 10 ;JOINT RELEASES ;1 FORCE X -1
nsg+2 2*nsg+2 INCLUSIVE FORCE X -1 Y -1
MEMBER INCIDENCES ;1 THRU nsg RANGE 1,2 nsg,nsg+1
nsg+1 THRU 2*nsg RANGE nsg+2,nsg+3 2*nsg+1,2*nsg+2
2*nsg+1 THRU nm RANGE 1,nsg+2 1+nsg,2*nsg+2
MEMBER RELEASES ;2*nsg+1 THRU nm START MOMENT Z END MOMENT Z
CONSTANTS E ec nsg+1 THRU 2*nsg ;CONSTANTS E e ALL
MEMBER PROPERTIES ;1 THRU nsg AX ax IZ iz
nsg+1 THRU 2*nsg AX ac=PI*dc^2/4 ac IZ izc=PI*dc^4/64 izc
2*nsg+1 THRU nm AX ar=PI*dsr^2/4 ar IZ PI*dsr^4/64
LOADING CASE 1
MEMBER LOADS ;IF w<>0 THEN 1 THRU nsg FORCE Y UNIFORM W w
IF njl=0 GOTO 20 ;JOINT LOADS ;i=0 ;:15 ;i=i+1 jn(i) FORCE Y jl(i)
IF i<njl GOTO 15 ;:20 ;SOLVE ;status=1 gtot=0 nur=0 j=3*(nsg+1)+1
nl1=ARR(14,j,nli) j=j+1 nl2=ARR(14,j,nli) m=nm*(nli-1)+nsg+1
nl3=ARR(13,m,1) ha=0 k1=0.25*(2.5+16*d^2/l^2)*SQR(1+16*d^2/l^2)
k2=3*l/(32*d)*LOG(4*d/l+SQR(1+16*d^2/l^2)) k=l*(k1+k2)
IF w=0 GOTO 40 ;i=0 ;:30 ;i=i+1 x1=(i-1)*l/nsg ;jl=w*l/nsg
IF i=1 OR i=nj THEN jl=0.5*w*l/nsg x1=0.25*l/nsg
IF i=nj THEN x1=l-x1 ;num=jl*x1*d/(3*e*iz)*(l-x1^2/l^2*(2*l-x1))
den=8*l*d^2/(15*e*iz)+k/(ac*ec)+64*d^2*(h-d+d/3)/(nsg*ar*e*l^2)
ha=ha+num/den ;IF i<nsg+1 GOTO 30 ;:40 ;va=l*w/2 ;IF njl=0 GOTO 70
i=0 ;:60 ;i=i+1 x1=(jn(i)-1)*l/nsg jl=jl(i) va=va+jl*(l-x1)/l
num=jl*x1*d/(3*e*iz)*(l-x1^2/l^2*(2*l-x1))
den=8*l*d^2/(15*e*iz)+k/(ac*ec)+64*d^2*(h-d+d/3)/(nsg*ar*e*l^2)
ha=ha+num/den ;IF i<njl GOTO 60 ;:70 ;pb1=ha pb2=-va
pb3=pb1*SQR(1+16*d^2/l^2) ;*/6
```

```
* Location                          NL-STRESS     Classical       %age
*                                   analysis      analysis        diff.
d1=nl1 d2=pb1
#vmper.ndf !Compute percentage difference & any message.
* Horiz. force at supports      +nl1          +pb1            $ok
d1=nl2 d2=pb2
#vmper.ndf
* Vert. force at left support   +nl2          +pb2            $ok
d1=nl3 d2=pb3
#vmper.ndf
* Cable tens. at left support   +nl3          +pb3            $ok
fnm=$(vm952.stk) ;#vmres.ndf !Conclude results.
mjn=1 lcn=10 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

APPENDIX B - NL-STRESS

Appendix B commences with a summary of the NL-STRESS language for those who are unfamiliar with STRESS (1964). The NL-STRESS summary is followed by a formal definition of the arrays file. It is the ability to select values from the arrays file, which allows the comparison of NL-STRESS results with those of the self check, and permits reporting of percentage differences to be included in each set of results.

The definition of the arrays file is followed by a full listing of a typical Verified Model.

NL-STRESS was chosen for this research into the subject of verification because it contains a language which can be extended. The following extensions were necessary for this research:

- parametric data, so that spans, loads etc. may be changed by changing the current value
- functions SIN COS etc. to compute coordinates etc., for example for a spiral staircase
- logic e.g. looping to apply different patterns of imposed loading to a continuous beam
- post-processing, to enable incorporation of the self check which follows SOLVE
- segmenting of members, to enable strain energy to be integrated along each member
- parameter specification table, to enable automatic generation of sets of data
- val=ARR(n,row,column) where val takes the value read from row & column of array n
- import of data e.g. #cc924.stk imports data which overwrites the default values
- calling external procedures e.g. procedure vmres.ndf concludes the post processing
- piping data to named files e.g. saving percentages from each self check for averaging
- ability to interface with SCALE; both SCALE & NL-STRESS can call each other.

An illustrated summary for the NL-STRESS language follows, which in turn is followed by a full listing of a typical NL-STRESS model, just one of 108 models which have been written and verified as part of this research.

Much of the NL-STRESS language is explained here by reference to an illustrated example for a propped beam. Lines in the data should be kept to a maximum length of 80 characters (see the User's Manual for spreading data to more than one line). The data may be prepared using the NL-STRESS editor or other text editors such as Notepad.

Page headings follow STRUCTURE or TITLE (up to 4 such lines) and the MADEBY, DATE and REFNO commands. A member always joins two joints. The number of supported joints is that for those with any type of external support. NUMBER OF LOADINGS is the number of basic + combined load cases.

The exclamation mark signifies that the rest of the line (including the exclamation mark) is to be ignored.

Sketches can be made from any of the characters on the keyboard and the 'graphics characters' which are accessed by holding down the Alt key and typing in an ASCII value in the range 0128 to 0255 on the numeric keypad and then releasing the Alt key. Sketch lines start with a *.

```
STRUCTURE PROPPED BEAM !7
MADEBY DWB
DATE 07-07-05
REFNO 05123
PRINT DATA RESULTS FROM 1
TYPE PLANE FRAME
METHOD ELASTIC
NUMBER OF JOINTS 2
NUMBER OF MEMBERS 1
NUMBER OF SUPPORTS 2
NUMBER OF LOADINGS 1
* Units are kN & m
*     ┌1kN/m  A┐    |—.1m
* /|▼▼▼▼▼▼▼▼▼▼▼▼▼   ┌─┐ ─┬─
* /|═════════════   │ │  │.3
*             A┘ ▪  └─┘ ─┴─
*  ├─────3m─────┤   A-A
*
JOINT COORDINATES
1 0 0 SUPPORT
2 3 0 SUPPORT
JOINT RELEASES
2 FORCE X MOMENT Z
MEMBER INCIDENCES
1 1 2
CONSTANTS E 28E6 ALL
CONSTANTS G 79E6 ALL
MEMBER PROPERTIES
1 AX .1*.3 IZ .1*.3^3/12
LOADING CASE 1
MEMBER LOADS
1 FORCE Y UNIFORM W -1
SOLVE ;FINISH
```

A semi-colon separates statements.

Every joint number (from 1 to the total number of joints) must be given in the JOINT COORDINATES table followed by X & Y co-ordinates; if the joint has any type of external support then the keyword SUPPORT should follow. In the example shown, joint 1 is rigidly supported, joint 2 is supported by a vertical prop only.

Every member number (from 1 to the total number of members) must be given in the MEMBER INCIDENCES table followed by the start joint No. and the end joint No. of the member. Local x axis goes from start to end.

Every member number (from 1 to the total number of members) must be given in the MEMBER PROPERTIES table followed by section properties. In the example AX is the cross-sectional area, IZ is the inertia about the Z axis. Any item of data may be written as an expression e.g. the area is .1*.3. SOLVE ensures a solution, omission checks the data.

Joints tagged with the word SUPPORT in the joint coordinates table are deemed to be built-in unless releases are given in the JOINT RELEASES table. Joint number 2 has a FORCE release in the X direction and a moment release about the Z axis (the Z axis comes out of the picture). Joint 1 is built-in and therefore has no releases.

Each load case starts with the command LOADING followed by some description e.g. LOADING CASE 1. The number of load cases must be the same as that declared in the NUMBER OF LOADINGS command. Basic load cases come firstly followed by load case combinations.

MEMBER LOADS are applied in the direction of the local (member) axes unless you request otherwise. As the local axes for member 1 point in the same direction as the global axes then the keyword GLOBAL may be omitted in this example. To apply 'plan loads' to rafters, use the word PROJECTED in place of GLOBAL.

FORMAL DEFINITION OF THE NL-STRESS ARRAY'S FILE

```
NJORG  No-of-joints-originally   after NUMBER OF JOINTS command
NMORG  No-of-members-originally  after NUMBER OF MEMBERS ......
NSPM   No-of-segments-per-member ............... SEGMENTS .....
NLSORG No-of-loading-systems     ............... LOADINGS .....
NINC   No-of-loading-increments  ............... INCREMENTS ...
NLSB   No-of-loading-systems-basic, excluding combinations
NLSC   No-of-loading-systems-combinations =NL-NLSB
NM     No-of-members-in-analysis    =NMORG*NSPM
NJ     No-of-joints-in-analysis     =NJORG+NMORG*(NSPM-1)
NLS    No-of-loadings-in-analysis   =NLSORG*NINC
NDJ    No-of-displacements/joint    =3 for 2D, =6 for 3D
NDM    No-of-displacements/member   =2*NDJ
NPD    No-of-possible-displacements =NDJ*NJ
MML    Maximum-member-loads on any member

Array
No. Description
 1  For each of NM rows, columns:
    1=end node start, 2=end, 3=member release No., 4=member loads
    counter, 5=order for posting to stiffness matrix,
    plane frame: 6-7 for MZ springs at start and end nodes
    plane grid:  6-9 for MX,MY springs at start and end nodes
    space grid: 6-11 for MX,MY,MZ springs at start and end nodes.

 2  For each of NM rows, columns:
    1-9=rotation matrix, 10=new member length, 11=original member
    length, 12=axial correcting load, 13=member results
    required (1=Yes, 0=No).

 3  For each of NM rows, columns:
    1-13 for upper triangle member stiffn for plane truss/frames
    1-26 .................................. space ............

 4  For NPD rows & NLS columns: artificial actions at joints

 5  .................................. actions at joints

 6  .................................. joint displacements

 7  For each of NPD rows, 3 columns elastic, (9 non-linear) give:
    restraint list, cumulative restraint list, current combined
    joint load vector (col4=predicted displacement for last but 2,
    5=previous 6=current incr. for non-linear case, 7=old current,
    8=previous array 4 to the current incr., 9=previous array 5 to
    the current increment for possible increment repeating case).

 8  For each of NJ rows, 6 cols hold: joint & node Nos. 3 coords
    column 6 holds 1 if joint results reqd, else =0.  For 2d
    plastic, z coordinate used for: number of members attached,
    then for plane frame plastic holds 0 if not exactly 2 members
    & free moment spring; or 1.0D150 if 2 members & free moment
    spring; or plastic spring value - used to save adding hinges
    on either side of a midspan joint, especially for pseudo
    mechanism problem before unloaded joint is detected.
```

```
 9  For each on NJ or NM rows (whichever is the maximum) for non-
    linear, columns 1-15 variously hold:
    Members: 1=FXP 2=MXP 3=MYP 4=MZP 5=ALPHA1 6=ALPHA2 7=formula
             FXH MXH MYH MZH (start end) FXH MXH MYH MZH (end end).
    Joints: Lowest unity factor at joint, member No. with lowest
             unity factor, number of plastic hinges this increment.

10  Each loading increment & each member, NLS*NM rows: columns 1=
    FXC 2=MXC 3=MYC 4=MZC (start), 5=FXC 6=MXC 7=MYC 8=MZC (end).

11  Each of NM rows, 22 columns hold: 1=AX 2=AY 3=AZ 4=IX 5=IY 6=
    IZ 7=CX 8=CY 9=CZ 10=BETA 11=E 12=G 13=CTE 14=DENS 15=YIELD
    16=SYIELD 17=DIRECTION 18=sectn ref 19=DY 20=DZ 21=TY 22=TZ.

12  4 lines of common +
    4 line page heading+load case titles & tabulate requirements.
    Value corresponds to order reqd.——\
    Line 8+loading number, column 1 >0 if FORCES tabulated, else 0
                                  2 ..... STRESSES................
                                  3 ..... DISPLACEMENTS...........
                                  4 ..... REACTIONS...............
    Array 12 holds four sets of COMMON in the first four records.
    These may be accessed from SCALE etc. e.g. +nm=ARR(12,1,2)
    will extract the number of members for the current problem &
    store it in the variable 'nm'.  The contents of the first four
                                        records of array 12 follow.
RECORD 1:
    ┌── Column number 'n' in ARR(12,1,n)
    1   NJ   No. of joints.
    2   NM   No. of members.
    3   NSUP No. of supports.
    4   NLS  No. of loading systems.
    5   ITAB(1) Default for tabulate forces   =1 if reqd, else=0.
    6       (2) .................... stresses  ..................
    7       (3) .................... displac.  ..................
    8       (4) .................... reactions ..................
    9   IUBW   Upper band width (approx. NDJ * (joint diff+1)).
   10   ISOLVE =1 if solve reqd, else =0.
   11   NLSB   No. of basic loading systems.
   12   NLSC   No. of combined loading systems.
   13   MML    Max No. of loads on any member for structure.
   14   LCASE  Current load case No.
   15   LINOL  Line No. in data file for start of loading data.
   16   ISTYP  Structure type 1 -> 5 for plane truss, plane frame
               grid, space truss, space frame.
   17   NDJ    No. of displacements / joint (3 for 2D, 6 for 3D)
   18   NDM    ...................... member (2*NDJ).
   19   NCOOR  No. of coordinates (2 for 2D, 3 for 3D).
   20   IPRNT(1)   1 if data reqd,  else =0.
   21        2     1 if trace reqd, else =0.
   22        3     1 if collection, else =0.
   23   LCSTA   Start load case number.
   24   LCEND   End   load case number.
   25   NLSORG  Number of loading systems originally specified.
   26   NERR    No. of errors.
27/28   NPD  No. of possible displacements (NJ*NDJ).
29/30   N    No. of degrees of freedom (NPD - No. of restraints).
```

RECORD 2:

```
    ┌── Column number 'n' in ARR(12,2,n)
    1   LINO    Line No. in output file.
    2   LUSED   No. of lines to be used on a page (typically 58).
    3   NPFLG   New page flag =1 if new page reqd., else =0.
    4   LPP     Lines per page (typically 66).
    5   IPNUM   Current page number in results file.
 6-65   IHEAD() Default screen & firm heading.
```

RECORD 3:

```
    ┌── Column number 'n' in ARR(12,3,n)
    1    .dat filename without the .dat.
  ->75
76->80   File handling switches.
```

RECORD 4:

```
    ┌── Column number 'n' in ARR(12,4,n)
    1    NINC     Number of increments in which load is applied.
    2    INCN     Current increment number.
    3    ITRACI   Trace of increments if =1.
    4    NSPM     Number of segments per member.
    5    ITRACS   Trace of segments if <>0.
    6    NMORG    Number of members originally set.
    7    NJORG    ......... joints  ...............
    8    NJC      Number of joints currently set.
    9    METH     Method  1=elastic  2=sway  3=plastic.
10-15  SIGA(1:6)  Applied forces from /CMPRO    ┐ See PWRREA
16-21  SIGR(1:6)  Applied reactions from /CMPRO ┘ GWRREA & SWRREA.
```

13 For each loading increment & each member, NLS*NM rows: columns 1-NDM member end forces, cols NDM+1 to 2*NDM member end stresses, cols 2*NDM+1 to 3*NDM member end rotations.

14 For each of NPD rows & NLS columns: support reactions.

If node renumbering required, temporary arrays 15-17 are built then after checking load data, re-initialised in RDLCD as follows.

15 For NLSB*NM rows, columns give member load information in 5*MML columns i.e. 5 columns for each member load set giving load type and start & end positions & magnitudes.

16 For each of NLSC rows of combined loadings: column 1 holds the combination type (COMBINE MAXOF MINOF ABSOF), and data for up to 10 lines of the command at 25 numbers/line.

17 For each of NLS*NM rows: cols 1-NDM hold member end reactions + scratch array for buildup of array in loadcase combinations.

18 The structure stiffness matrix, initialised in BLDSM, has N or 2*N rows and IUBW columns where:

```
N      Number of degrees of freedom
IUBW   Upper band width of structure stiffness matrix
```

If there is only one loading, or the engineer has set the /M switch, then the number of rows =N; else the number of rows=2*N for computational efficiency.

TYPICAL NL-STRESS VERIFIED MODEL

```
! This verified model is parametric.  General notes follow FINISH.
! A semi-colon preceded by a space is used to separate statements.
! Parameters precede '!' help which follows, is copied to results.
! VEC (short for VECtor, a rank-one array), is used for multiple
! assignments: a(7)=VEC(3.2,b,-5.7) ≡ a(7)=3.2, a(8)=b, a(9)=-5.7;
! cs(1)=VEC(12,2.8)*2 ≡ cs(1)=12, cs(2)=2.8, cs(3)=12, cs(4)=2.8.

TITLE GROUND BEAM ON ELASTIC FOUNDATION SUBJECTED
TITLE TO TRAIN OF MOVING LOADS; CHECKING OF RESULTS
TITLE AGAINST HETENYI'S CLASSICAL SOLUTION, UNIVERSITY
TITLE OF MICHIGAN PRESS; AND BY BUILDERS' ARITHMETIC.
MADEBY DWB ;DATE 08.02.05 ;REFNO VM130
PRINT DATA, RESULTS FROM 1 ;TYPE PLANE FRAME run=0 ;*/7
* ├──────────────────────a(nl)─────────────────┤
* ├────────a(2)─────┤          nl=number       │p(nl)
* ├a(1)─┤p(1)       │p(2)      of loads        │
*       ▼           ▼                          ▼
* █████████████████████████████████████████████████████
* //\\//\\//\\//\\//\\//\\//\\//\\//\\//\\//\\//\\//\\/
* ├────────────────────────l───────────────────────────┤
l=11 br=1 d=0.5  ! Length breadth & depth of ground beam.
nj=45            ! No. of joints each modelling lumped stiffness.
k'=1E3           ! Modulus of subgrade reaction kN/m3.
nl=3             ! Number of load points in train of loads.
nls=3   spc=1    ! nls=number of loadings; spc=step spacing.
e=28E6/3 nu=.2   ! Young's mod; nu=Poisson, 0 to ignore shear def.
a(1)=VEC(1.5,5.5,9.5)          ! Distances to loads.
p(1)=VEC(-500)*nl              ! Magnitude of loads.
#cc924.stk !Import verification data from cc924.stk if available.
NUMBER OF JOINTS nj          ;NUMBER OF MEMBERS nm=nj-1 nm nsg=1
NUMBER OF SUPPORTS nj        ;NUMBER OF LOADINGS nls
JOINT COORDINATES ;crs=l/(nj-1) ;1 THRU nj X 0 Y 0 XL l SUPPORT
JOINT RELEASES ;1 FORCE Y k'*br*crs/2 MOMENT Z 0
nj FORCE X 0 FORCE Y k'*br*crs/2 MOMENT Z 0
IF nj>2 THEN 2 THRU nj-1 FORCE X 0 Y k'*br*crs MOMENT Z 0
MEMBER INCIDENCES ;1 THRU nm RANGE 1,2 nj-1,nj
CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL ;MEMBER PROPERTIES
ay=br*d*5/6 ;IF nu=1E-12 THEN ay=0
1 THRU nm AX br*d AY ay IZ br*d^3/12
n=0 ;REPEAT ;n=n+1 m=0 som=-crs
LOADING TRAIN OF MOVING POINT LOADS (DOWN IS NEGATIVE)    POSITION
MEMBER LOADS ;:200 ;som=som+crs lc=0 m=m+1 !som is start of memb.
:300 ;lc=lc+1 x=a(lc)+(n-1)*spc
IF x=0 THEN x=1E-6  !Fix to get left end load on the span.
eom=som+crs l'=x-som ;IF m=nm THEN eom=l l'=l-som
IF x>som AND x<=eom THEN m FORCE Y CONCENTR P p(lc) L l'
IF lc<nl GOTO 300 ;IF m<nm GOTO 200 ;UNTIL n=nls ;ENDREPEAT ;SOLVE
* BUILDERS' ARITHMETIC.  Find: centre of loading, pressures at
* each end (P/A±M.y/I), then moments & shears at load positions.
* Load  Dist.   Pressure      Shear to      Shear to      Moment @
*  No.  m       kN/m2         left kN       right kN      load kNm
lcase=0 inc=-spc ;:310 ;lcase=lcase+1 inc=inc+spc
i=0 p=0 m=0 a(0)=0 ;:320 ;i=i+1 ai=a(i)+inc p=p-p(i) m=m-p(i)*ai
IF i<nl AND a(i+1)+inc<=l GOTO 320 ;IF m/p<l/3 OR m/p>2*l/3
* Centre of loading outside middle third; lift off will occur.
STOP ;ENDIF ;ec=l/2-m/p pa=p/(l*br) p(0)=pa+p*ec/(br*l^2/6)
pr=pa-p*ec/(br*l^2/6) pd=pr-p(0)
i=0 df(0)=0 mc(0)=0 ma(0)=0 sr(0)=0
```

```
:340 ;i=i+1 ai=a(i)+inc ps(i)=p(0)+pd*ai/l
u(i)=0.5*(p(0)+ps(i))*ai*br sl(i)=u(i)-df(i-1) df(i)=df(i-1)-p(i)
sr(i)=u(i)-df(i) mc(i)=p(0)*br*ai^2/2+(ps(i)-p(0))*br*ai^2/6
ma(i)=ma(i-1)+df(i-1)*(a(i)-a(i-1)) m(i)=mc(i)-ma(i)
* +i  +ai       +ps(i)        +sl(i)        +sr(i)        +m(i)
IF i<nl AND a(i+1)+inc<=l GOTO 340 ;IF lcase<nls GOTO 310
*/10 ;* M.HETENYI in 'Beams on elastic foundations', University
* of Michigan Press, gives formulae for deflection, slope,
* bending moment & shearing force for a concentrated force
* at an arbitrary point as used below.
* Position    Loading     Deflection      Deflection        %age
* ref. No.    case No.    by NL-STRESS    by Hetenyi        diff.
status=1 gtot=0 nur=0 com=a(1) ext=a(1)+(nls-1)*spc
st=e*br*d^3/12 k=k'*br s(1)=0 i=1 ;:400 ;i=i+1 s(i)=a(i)-a(1)
IF i<nl GOTO 400 ;i=0 ax(0)=-crs ;:500 ;i=i+1 ax(i)=ax(i-1)+crs
IF i<nj GOTO 500 ;lam=(k/(4.*st))^0.25 sinhLL=SNH(lam*l)
fp=1./(sinhLL^2-(SIN(lam*l))^2) c=0 ;:600 ;c=c+1 x=ax(c)
sinhlx=SNH(lam*x) coshlx=CSH(lam*x)
sp1=coshlx*SIN(lam*x)+sinhlx*COS(lam*x)
fip1=coshlx*SIN(lam*x)-sinhlx*COS(lam*x)
z=l-x sinhlz=SNH(lam*z) coshlz=CSH(lam*z)
sp2=coshlz*SIN(lam*z)+sinhlz*COS(lam*z)
fip2=coshlz*SIN(lam*z)-sinhlz*COS(lam*z) d'=com-spc lcase=0
:700 ;d'=d'+spc lcase=lcase+1 toty=0 i=0
:800 ;i=i+1 p=p(i) a=d'+s(i) ;IF a>l GOTO 799
b=l-a sinhla=SNH(lam*a) sinhlb=SNH(lam*b) coshla=CSH(lam*a)
coshlb=CSH(lam*b) tp1=SIN(lam*a)*coshlb-COS(lam*a)*sinhlb
fp1=sinhla*COS(lam*b)-coshla*SIN(lam*b)
sip1=sinhLL*COS(lam*a)*coshlb-SIN(lam*l)*coshla*COS(lam*b)
sip2=sinhLL*COS(lam*b)*coshla-SIN(lam*l)*coshlb*COS(lam*a)
IF x<a
y=(2.*coshlx*COS(lam*x)*sip1+sp1*(sinhLL*tp1+SIN(lam*l)*fp1))
y=(p*lam*fp/k)*y ;ENDIF
IF x=a ;sinh2a=SNH(2.*lam*a) sinh2b=SNH(2.*lam*b)
c1=(coshlb^2+(COS(lam*b))^2)*(sinhla*coshla-SIN(lam*a)*COS(lam*a))
c2=(coshla^2+(COS(lam*a))^2)*(sinhlb*coshlb-SIN(lam*b)*COS(lam*b))
y=(p*lam*fp/k)*(c1+c2) ;ENDIF ;IF x>a
c1=2.*coshlz*COS(lam*z)*sip2+sp2*(sinhLL*(-fp1)+SIN(lam*l)*(-tp1))
y=(p*lam*fp/k)*c1 ;ENDIF ;toty=toty+y ;:799
IF i<nl GOTO 800 ;row=3*c-1 dy=ARR(6,row,lcase) d1=dy d2=toty
#vmper.ndf !Compute percentage difference & any message.
*   +c          +lcase    +dy             +toty             $ok
IF lcase<nls GOTO 700 ;IF c<nj GOTO 600 ;fnm=$(vm130.stk)
#vmres.ndf !Conclude results.
mjn=1 lcn=1 tot=3 drn=2 ;#vmtes.ndf
< ;FINISH
```

■ 1.11 GENERAL NOTES

PARAMETER

No.	Name	Start	End	Type	Dependency conditions
1	l	5	100	0	
2	br	1	5	0	
3	d	0.3	1	0	
4	nj	20	800	1	=INT(4*l/d)
5	k'	5000	30000	1	
6	nl	5	10	1	No. of load points in train.
7	nls	2	10	1	No. of load cases.
8	spc	0.1	50	0	=l/(2*(nls-1))
9	e	14E6	28E6	1	
10	nu	0.1	0.3	3	
11	a0	0	0	0	
12 21	a	0.5	50	0	=zva(zp'-1)+0.5*l/nl
22 31	p	-50	-500	0	

For the benefit of engineers who wish to develop their own Verification Models (VMs), an overview on how the PARAMETER table is compiled into a program is given in VM120.NDF.

Engineers will be concerned with 'engineering' practical dependency conditions; generally these are straightforward; such as those in the following line extracted from the PARAMETER table:

```
 4  nj     20     800     1       =INT(4*l/d)
```

which specifies that the number of joints nj must be an integer in the range 20 to 800, which is to be made equal to INT(4*l/d) where l is the length of the beam in metres, and d is the depth of the beam in metres, e.g. if the beam is 10m long, and d is 0.25m deep then the number of joints will be INT(4*10/.25)=160, OK as in the range 20 to 800.

As described in VM120.NDF, the parameter names listed in the parameters table are converted to variables zva(1) to zva(znp), where znp is the total number of parameters. It is these variables which are evaluated in accordance with their type, and dependency conditions. The variables zva(1) to zva(znp) are computed in order, thus when zva(n) is being computed, values for zva(1) to zva(n-1) are known for the current set of data being generated. Other known values include:

zp'	The current parameter number, in the range 1-znp, where znp is the total number of parameters.
zva(zp')	The value of the current parameter, being computed.
zva(zp'-1)	The value of the previous parameter, already computed.
pz'	Direction of parameter value, 1=start to end, 0=end to start; 2*pz'-1 behaves as a flip-flop, values 1 or -1.
zns'	The pattern number in the range 1 to 6.

It is normal, when specifying a train of loads, to dimension the location of each load in order from a reference point, either at, or to the left of the first load. The subscripted variable a() is used for storing the load location positions for the train of loads. The first line of the tabular couplet below, initialises a(0), the parameter prior to a(1).

```
11  a0     0       0       0
12 21 a    0.5     50      0       =zva(zp'-1)+0.5*l/nl
```

The second of these two lines specifies that the minimum value of a() is 0.5, and the maximum is 50, but it is the expression =zva(zp'-1)+0.5*l/nl which sets values a(1) to a(10).

The zva(zp'-1) is the dimension from the left end to the previous load position to which is added 0.5*l/nl which increments the previous value by half the beam length divided by the number of loads.

PERCENTAGE DIFFERENCES

The comparison of results produced by the matrix stiffness method, with those produced by a classical method e.g. Hetenyi, requires that a comparison be done for ≈1000 runs for each Verified Model. As the results of each run is typically 25 pages, then 25,000 pages have to be checked i.e. a pile of A4 from floor to ceiling for each model. Checking on this scale is not a task for humans, thus 'yet more software' is needed to compare deflections/forces for each run:

- highlighting percentage differences exceeding 1%
- computing the average percentage difference for each run
- tabulating the average percentage differences for all runs (contained in the file public.stk) for the engineer to spot anomalies, and investigate further.

Even with the best computing facilities & reporting, many engineering hours are needed investigating anomalies. In a thousand runs there are usually a few rogues which need to be explained and the Verification Data dependencies and/or model adjusted accordingly. A typical procedure:

- note the run number in the file public.stk which has the highest percentage difference (after analysis, each set of data is held in the file scrtcha.001 to scrtcha.996 where the filename extension is the run number)
- run NL-STRESS and click Edit, press F3 and enter the name of the file to be imported file e.g. scrtcha.664 and press Enter
- click the cursor in the line above which the import is reqd
- press F2 to import the data
- edit the imported data, changing just one parameter at a time, run and investigate the results, repeat as necessary.

The line commencing #vmper.ndf invokes the logic block 'vmper.ndf' to compute percentage difference between two values d1 & d2, and compose a text message. The composed text will be: blank for percentage differences less than 1%, an integer percentage for percentages <100%, and the message NOT OK for percentages greater than 99%. The variable 'nur' in the logic block is incremented by 1 each time the block is called; the variable 'gtot' cumulates the percentage differences, at the end of each run the invoking procedure prints the average percentage difference i.e. gtot/nur. The logic block treats d1 & d2 as equal if their absolute values are approximately equal; this is to exclude percentages computed using arithmetic roundoff values.

All checking is based on percentage differences and averages of percentage differences. The calculation of percentage difference between two numbers is ambiguous. Engineers like worked examples, those which follow should be read in conjunction with the logic block in 'vmper.ndf' to see how the rules work.

```
Ex.1: d1=88 d2=77, r=88/77=1.1428 per=INT(ABS(100-114.28)+.5)=14.
Ex.2: d1=77 d2=88, r=0.875 r=1/r=1.1428 per=14 as Ex.1.
Ex.2: d1=-8 d2=-7, r=-8/-7=1.1428 per=14 as Ex.1.
Ex.4: d1=0.1 d2=0, r=2 per=100.   Thus when one value is zero, the
Ex.5: d1=0 d2=0.1, r=2 per=100.   other not, then 100% returned.
Ex.6: d1=2  d2=-1, r=-2 r=2 per=100.   Thus when different signs
Ex.7: d1=-2 d2=1,  r=-2 r=2 per=100.   then 100% always returned.
Ex.8: d1=-1 d2=2, r=-.5 r=2 per=100.
Ex.9: d1=-1E-14 d2=1E-14 r=1 per=0.
Ex.10: d1=0.123E-4 d2=0.567E-7 r=216.93 r=2 per=100.
Ex.11: d1=0.567E-7 d1=0.123E-4 r=0.0046097 r=216.93 r=2 per=100.
Ex.12: d1=0.567E-14 d2=-0.123E-11 r=-0.0046097 r=2 per=100.
```

NOTES ON THIS VERIFIED MODEL

The verification is with the classical solution by M Hetenyi, Beams on elastic foundations, University of Michigan Press, which gives formulae for deflection, slope, bending moment & shearing force for a concentrated force at an arbitrary point. Engineers' arithmetic check is also included, in which it is assumed that the beam is infinitely stiff; thus from the centroid of the loads a linear pressure beneath the beam is given by P/A±MY/I and from this linear pressure the bending moment at each load position may be calculated.

COLLECTION collects all loadings for each joint/member.
Add the keyword COLLECTION after RESULTS, if required.

From Terzaghi the engineer assesses a soil stiffness (units kN/m3 i.e. pressure to give the soil unit deflection) by means of charts and tables, taking due account of the foundation size and the distribution of loads. The coefficient of subgrade reaction is then multiplied by the area (assumed lumped at a spring support) and the resulting spring stiffness used in the data. If the soil is of poor quality, the value of k' can be increased by: compacting soil if possible; stabilizing the soil with cement or lime; applying a well compacted subbase of sufficient thickness; removing the poor quality layer and replacing it with well compacted sand or crushed stone, stabilised sand or lean concrete.

The k' value cannot be used as a measure of settlement. The settlement must be calculated on the basis of the results of a geotechnical study. A.A.Alexandrou, formerly of the University of Greenwich has provided the following table of moduli of subgrade reactions (k').

Table by AA Alexandrou University of Greenwich	Modulus of subgrade reaction k'	
	k/ft3	kN/m3
Humus soil or peat	30 – 100	5000 – 15000
Recent embankment	60 – 125	10000 - 20000
Slightly compacted sand	90 – 200	15000 - 30000
Well compacted sand	300 – 650	50000 - 100000
Very well compacted sand	650 – 900	100000 - 150000
Loam or clay (moist)	200 – 400	30000 - 60000
Loam or clay (dry)	500 – 650	80000 - 100000
Clay with sand	500 – 650	80000 - 100000
Crushed stone with sand	650 – 900	100000 - 150000
Coarse crushed stone	1300 – 1600	200000 - 250000
Compacted crushed stone	1300 – 2000	200000 - 300000

The above table gives elastic spring stiffnesses for a unit area. Soils do not behave in a linear elastic manner in the long term, they settle due to pore water dissipation and other effects which compact the soil, such as vibrations. Engineers measure the void ratio of the soil to estimate the amount of consolidation expected in the long term. It is normal to assume that the self weight of the ground beam is supported directly by the supporting soil, therefore the self weight is omitted from the this model.

The REPEAT-UNTIL-ENDREPEAT, used by NL-STRESS for looping, is treated as a special case when it encloses a loading. For such cases, the loading title which follows the keyword LOADING would cause confusion if the same title were to be printed in the results for all the loadings within the loop. NL-STRESS recognises this situation and automatically adds the loading number near the end of the line.

Before launching a thousand runs - to simulate the mixture of parameter variation likely in general usage - it is prudent to carry out several single runs varying just one parameter at a time. For this problem, keeping l, br, d, k', nls, spc, e, nu, a(), p() constant and varying the number of joints to simulate a uniformly distributed soil stiffness:

Centres of joints	Number of joints	Average %age diff. for k'=10000 kN/m2
1	12	4.67
0.5	23	0.88
0.25	45	0.21
0.125	89	0.25
0.0625	177	0.27

From the above, 45 or more joints give an average percentage difference between NL-STRESS & Hetenyi of less than 0.3%. Next keep the number of joints constant at 45, and vary the beam depth.

Beam depth	Centres of joints	Average %age diff. for k'=10000 kN/m2
0.25	0.25	1.13
0.5	0.25	0.21
0.75	0.25	0.16
1.0	0.25	0.14

From the above, good accuracy is obtained when the beam depth is greater than twice the joint centres. From an engineering viewpoint, a point load may be assumed to be spread at 45° from the top of the beam to the neutral axis. The vertical distance from the top of the beam to the neutral axis is half the beam depth, i.e. equal to the suggested centres for the joints, so far. Next vary the modulus of subgrade reaction.

Modulus of subgrade reaction kN/m3	Average %age diff.
2500	0.0
10000	0.21
40000	1.07
160000	3.19

Consulting AA Alexandrou's table entitled Modulus of subgrade reaction k', when the soil stiffness is increased from humus soil to crushed stone with sand, the average percentage difference between NL-STRESS & Hetenyi, increases from 0.21% to 3.19%. Next keep the soil stiffness at 160000 kN/m2 and vary the number of joints again.

Centres of joints	Number of joints	Average %age diff. for k'=160000 kN/m2
1	12	6.20
0.5	23	3.67
0.25	45	3.19
0.125	89	3.39
0.0625	177	3.36

Once again, making the joint centres equal to half the beam depth, gives best results. High soil stiffnesses cause concentrated loads to be carried by the soil directly beneath, only residual bending moments ripple between the joints which are not beneath the concentrated loads.

Following the comparison between Hetenyi and the stiffness method, comes the traditional method of analysis in use before computers were generally available. In this analysis entitled Engineers' Arithmetic, the centre of loading is first found, then pressures at each end of the ground beam computed from P/A±M.y/I assuming a linear pressure distribution beneath the ground beam, then moments & shears at load positions are calculated. The bending moments and shears computed by Engineers' Arithmetic do not agree with those computed by NL-STRESS/Hetenyi. NL-STRESS/Hetenyi take the soil stiffness into account, Engineers' Arithmetic does not; however, Engineers' Arithmetic and NL-STRESS/Hetenyi can be reconciled by reducing the modulus of subgrade reaction to a very low value, thereby making the beam so stiff by comparison with the soil, that the pressure distribution beneath the beam is linear.

Modulus of subgrade reaction kN/m3	Bending moment at the first load, Case 1. Nl-STRESS	Engineers' Arithmetic
10000	158.346	153.409
100	154.039	153.409
1	153.416	153.409
1E-3	153.409	153.409 Q.E.D.

The modulus of subgrade reaction =1E-3 kN/m3, is 1 kN/m2 (weight of a 16 stone man) spread over an area of 1 m2 and resulting in a deflection of 1000m =1km. Inspection of the NL-STRESS results for the default values but with k'=1E-3, shows a resulting deflection =163 km beneath the first load, Case 1.

After all this preliminary testing has been done to gauge the structural behaviour of the model, modifying the dependency conditions in the Verification Data in accordance with that structural behaviour, then 996 runs in a batch are carried out under the control of the Verification Data. Inspection of the text file 'public.stk', shows the average percentage difference for each run in the batch. A good way to proceed is to note down the first significant 'percentage difference' with its run number, then look for one higher and note that, and so on to the last. From the first 996 runs, the following was obtained.

Run No.	30	506	530	828
Average %age diff.	2.57	2.95	5.34	15.55

Obviously run 828 needs investigation, the data for which (held in the file scrtch.828).
Importing this data to VM130.NDF (this data file) and running it directly, confirms the 15.55% average percentage difference.
Further runs, again varying just one parameter at a time, shows that the number of joints modelling the subgrade reaction (nj), is insufficient. The dependency conditions were adjusted in the light of the above to be =INT(10*l/d).

Using this new dependency condition, a study was made of the results of 996 runs with shear deformation suppressed, the average percentage difference was 0.399%. Inspection of the averages for each run showed that the most of the runs had an average percentage difference near zero, whereas one or two runs had appreciable differences, the largest being in run 832 which had a difference of 34.55%, accordingly this set of data was studied.
The first obvious item of data to be considered in run 832, was the high number of joints =183, the previous dependency condition was =INT(2*l/d) which had been changed to INT(10*l/d), for the beam length=5.5758, breadth=4.9758, depth=0.30424, thus:
number of joints =INT(10*5.5758/0.30424) =183. Changing the:

Number of joints	366	183	91	45
Average %age diff.	0.001	34.55	0.049	0.50

The mystery deepens, closing in:

Number of joints	181	183	185	187	189
Average %age diff.	0.022	34.55	34.54	34.55	0.040

Inspection of the structure with the loading superimposed, gives a clue to what is happening. There are two sets of loads, each set having 5 loads, when the percentage difference is very low both sets come on the beam, when the percentage difference is high, only 4 out of the five loads in the second set come on the beam; thus the problem is due to roundoff. Placing load on the member was controlled by the following which was copied from the data.

```
IF x>som AND x<=som+crs THEN m FORCE Y CONCENTR P p(lc) L x-som
```

The logic for this is robust, for all spans except the last one. Any load which occurs almost at a joint position will either be considered on the current span, or left for the next one, but for the last span there is a possibility that sometimes a load will be fluked onto the end of the beam, and sometimes will not. The problem could be solved by a special case for the last span.

```
IF x>som AND x<=l THEN m FORCE Y CONCENTR P p(lc) L l-som
```

More elegantly both cases are covered by:

```
eom=som+crs l'=x-som ;IF m=nm THEN eom=l l'=l-som
IF x>som AND x<=eom THEN m FORCE Y CONCENTRATED P p(lc) L l'
```

The dependency conditions were adjusted in the light of the above to be =INT(4*l/d).

CONCLUSIONS

The engineer who devises models such as this, in addition to devising robust logic, must also be aware of roundoff, especially when a set of load data is incremented. The problem of roundoff only came to light when 996 runs in a batch was completed and the results studied.

If shear deformation is suppressed by setting Poisson's ratio =1E-12, results from NL-STRESS and Hetenyi agree to an average accuracy of 0.0430% for all 996 sets of data generated from the PARAMETER table. When shear deformation in not suppressed but Poisson's ratio varied from 0.1 to 0.3, the average difference between both methods for 996 sets of data generated from the PARAMETER table =0.0425%, with the largest individual result being for run 502 when the average difference was 2.037%. Shear deformation was insignificant for the range of values tested.

It is remarkable that such close agreement was achieved, for Hetenyi solved the governing differential equations and provided a solution which includes many trigonometric and hyperbolic functions, as will be evident by perusal of the expressions between the SOLVE and FINISH commands in the data; whereas NL-STRESS uses the matrix stiffness method, modelling the soil by lumped stiffness at the joints.

APPENDIX C - PRAXIS

Verification of models for the structural analysis of a framework includes: a self check near the end of the model and an embedded table which provided a parametric description of the model. The structure of the parametric description developed in this research is common to both the structural analysis of a framework and the design of a structural component; for the former the model is written in the STRESS (1964) language with extensions; for the latter, the model is written in a notation called Praxis. Praxis (1990) is used for composing proforma calculations for the design of structural components such as beams, slabs & columns; each proforma calculation may be thought of as an interactive (question & answer) program which both prompts for data and writes the calculations using the layout of the proforma itself.

Appendix C commences with an illustrated example of a very simple proforma calculation and a tabular summary of the Praxis notation. It will be seen that Praxis is English with embedded logic. The is followed by proforma calculation sc075.pro for the design of flanged reinforced concrete beam sections, which in turn is followed by sc385.pro for the design of stainless steel structural hollow sections; just two of more than seven hundred proforma calculations, written over the past twenty years, for the design and detailing of structural components in steel, concrete, masonry and timber. Proforma sc075.pro has its self check written in Praxis, whereas proforma sc385.pro has its self check written in the NL-STRESS with extensions developed as part of this research.

Experience with supporting proforma calculations over twenty years is that a few engineers just use the proformas, a few engineers become expert at writing their own proformas, the great majority have some knowledge of Praxis, the distribution being similar to the engine knowledge of car drivers. Much of the Praxis notation is explained by reference to the illustrated example for checking that the bending stress on a floor joist is within permissible limits. The layout of the calculation can conform to any set of recommendations. Lines of the proforma should be kept to a length of 70 characters to make the resulting calculation fit the width of A4 or quarto paper and leave a binding margin. The illustrated example is not to any particular code of practice, the two proforma calculations which follow are authentic.

When the proforma is run by SCALE, lines after the word START are copied one after the other. These copied lines become the calculations. Items of data are substituted for question marks, values looked up in tables, arithmetic performed as appropriate, until the word STOP or FINISH is found.

A percentage sign at the start of a line tells SCALE to copy that line to the screen but not to the calculations file. In other words the percentage sign introduces a line to help the person running the proforma.

If you make a mistake when entering data, click 'Go back', so directing the calculation back to the nearest previous > in the proforma. If there is no previous > in the proforma, the calculation restarts.

If the computed stress 'si' does not exceed the permissible stress 'sd', then deal with the lines between IF and ELSE, ignoring the alternative lines to ENDIF. If the computed stress is not less than the permissible, ignore the lines before ELSE, dealing only with those between ELSE and ENDIF

/5 says throw a page unless there is enough room on the current page to print 5 lines. / on its own says start a new page regardless.

Lines starting with a full stop appear in a summary. A summary is produced when clicked, and lists input data (i.e. lines originally containing ????) and lines starting with a full stop.

Tables precede the word START. The table number and size follow the word STORE. In the example, the first number following STORE is the table number, the second says the table has 6 rows, the third says it has 1 column.

```
STORE 1 6 1 !Permissible
1  3.1      !stresses for
2  4.5      !six classes
3  5.5      !of timber.
4  8.3
5 11.0
6 12.5
START
Floor joist ref: ????

  ┌w        A┐        ├──┼─b
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼      ┌─┐ ─┬─
═══════════════      │ │  │ h
▲           A┘  ▲    └─┘ ─┴─
├──────L────────┤     A─A

%An arithmetic expression
%may be given in response
%to any prompt e.g. 5*3.
Effective span  +L=????
Dead+live load  +w=????
Strength class  +SC=????
>
Depth of sectn  +h=????
IF h>300
'h' out of table range.
STOP
ENDIF
Section width  +b=????
Stress   +sd=TABLE(1,SC)
Bend. moment  +M=w*L^2/8
Sectn modulus +Z=b*h^2/6
Bending stress +si=M/z
IF si<=sd
As si<=sd ( +si <= +sd )
bending stress is OK.
ELSE
As si>sd ( +si > +sd )
bending stress exceeds
permissible.
STOP
ENDIF
/5
.SUMMARY
.────────
.Bending stress      +sd
.Permiss bend stress +si
.Bending stress OK.
FINISH
```

The exclamation mark signifies that the rest of the line (including the exclamation mark) is to be ignored.

Sketches can be made from any of the characters on the keyboard and the graphics characters which are accessed by holding down the Alt key while typing in a value between 0128 to 0255 on the numeric keypad.

If you respond by entering 3, the 3 is assigned to the 'variable' named L. A variable may be thought of as a named box; in this case a box named L. The plus sign in front of the assignment tells SCALE to put the value 3 into the box named L for subsequent use.

This assignment causes SCALE to look in TABLE 1 using the value in variable SC as the search key. Thus with a class (the value in SC) of 3 the variable named sd would be assigned the value 5.5, extracted from the row, which starts with 3, of table 1. In general, tables have several rows and several columns with headings to each.

The plus sign causes SCALE to work out: $w.L^2$ over 8, and assign the value to the variable M. In general an assignment will be followed by units, omitted for reason of space.

FINISH marks the end of a proforma.

THE STRUCTURE OF A PRAXIS PROFORMA CALCULATION

SUMMARY of Praxis — sheet 1 of 2

@ as the first character of a line, before the keyword START, followed by a filename containing tables and other data, includes them in the proforma. @ at the end of a proforma, followed by a file—name containing procedures causes the procedures to be included in the proforma.

! means do not copy the rest of the line (or the !) to the calculations file.

% at the start of a line means send the line to the screen but not to the calcs file.

/ at the start of a line means throw a new page. / followed by an integer number or expression (e.g. /5), means start a new page unless 5 or more lines remain available on the current page.

> at the start of a line means return to this line whenever Go back is clicked in response to a subsequent prompt.

? conventionally ???? means display the line and wait. Your response then replaces the ????

+ says the word which follows is the name of a variable. This name may stand alone or be part of an expression or assignment. Examples are:
+a   +a*b   +I=b*d^3/12
+e=TABLE(25,Grade)

. at the start of a line means include the line when a summary is requested.

EXPRESSIONS
Expressions comprise terms. Each term may be a number, the name of a variable, or function. The terms are bound together with operators ^ * / + — shown in order of precedence. You can override precedence with brackets: 2*3+4 =10, 2*(3+4) =14

FUNCTIONS

ABS	Absolute value
INT	Integral part
SGN	1, 0, —1 if positive, zero negative, respectively
LOG	Natural log
EXP	Natural exponent
SQR	Square root
DEG	Degrees from radians
RAD	Radians from degrees
SIN	Sine of angle — radians
ACS	Arcsine
COS	Cosine of angle — radians
ACS	Arccosine
TAN	Tangent of angle — radians
ATN	Arctangent
SNH	Hyperbolic sine
CSH	Hyperbolic cosine
TNH	Hyperbolic tangent

ORDER
A proforma has the following essential structure:

```
Overall settings (if any)
Tables (if any)
START
lines of proforma
FINISH
```

The overall settings may include:

```
PAGELENGTH  n  m
```

n is the page length expressed as a number of lines
m is the number of lines to print on such page

PAGELENGTH 1000 means omit headings but leave binding margin in the calculations; PAGELENGTH 2000 means omit both headings and margin.

```
DIGITS  n
```

n is the desired number of significant digits in printed results (1—16). Five is implied if omitted. Arithmetic is to 15+ digits whatever the setting.

CONTROL

The following keywords control SCALE when executing a proforma. The keywords are not copied to the calculations file.

Stop normally:

```
STOP
```

Process conditionally:

```
IF condition
lines
ELSE
alternative lines
ENDIF
```

First evaluate condition: true or false. If true, process 'lines', ignore 'alternative lines'. If false, ignore 'lines', deal only with 'alternative lines'. You may omit 'ELSE alternative lines' when nothing is to be done when condition evaluates as false.

Repeat lines:

```
REPEAT
lines
UNTIL condition
lines
ENDREPEAT
```

Process all lines between REPEAT and ENDREPEAT again and again. If, on evaluating condition, the result is true, leave loop and process line following ENDREPEAT.

Procedures:

```
DEFINE  name
lines
ENDDEFINE
```

Wherever 'name' appears at the start of a line of the proforma, substitute and process 'lines'. Procedures should be placed at the end of the proforma, between a STOP and FINISH. Use this facility if you would otherwise have to include the same set of lines in several places, or to simplify complicated proformas.

TABLES

You may include tables of values from which SCALE may be made to look up values for use in a calculation. Tables precede START. For each table, give a reference number and dimensions.

```
STORE reference rows columns
```

reference is the reference number for the table
rows is the number of rows
columns is the number of columns.

Follow with a line of column headings. (Omit if only one column.) Follow with rows of values, preceding each with a numerical heading. (Omit this heading if only one row.)

To make SCALE look up a value, include in an assignment:

TABLE(ref,row,col)

ref is the table reference number
row is the row heading
col is the column heading.

Instead of a number you may give the name of a variable that contains the relevant number.

For a table with one row or column, leave out the corresponding 1. For example: TABLE(1,SC) rather than TABLE(1,SC,1).
You do not have to match the row or column heading precisely; SCALE establishes the value by two—way linear interpolation.

SPECIALIST FACILITIES

See the Praxis manual for specialist facilities, including:

- setting and re—setting values for prompts
- special format operations
- external tables and files
- invoking external programs from within a proforma.

TYPICAL PROFORMA CALCULATIONS WITH SELF CHECKS

Proforma sc075.pro for the design of flanged beams in bending with optional shear, bar curtailment, lap length and span/effective depth check, follows, which in turn is followed by proforma sc385.pro for stainless steel hollow section design.

Both proformas were written by others. The self checks, which were developed by the writer as part of this research, are printed near the end of each proforma. The vertical rule on the right of each proforma indicates new work by the writer for the self checks, required as part of the verification process.

PROFORMA SC075.PRO

```
Proforma     No. 075                                      Revision: W
Title        Design of flanged beams in bending with optional shear, bar
             curtailment, lap length and span/eff.-depth checks
Devised by   Professor W. B. Cranston - 26 March 1990
Based on     BS8110: Part 1: 1997
Checked by   Jim Steedman - March 1993
Amendments   Cover and fire-resistance procedures added; miscellaneous
             minor and general cosmetic changes made       (a:jcs:03/93)
             Steel strength included as variable             (jcs:09/93)
             Link spacing criteria corrected     (b:jcs:10/93 and 12/93)
             Value of fyv in summary made variable         (c:jcs:02/94)
             Shear module modified to accommodate different
             main bar sizes to those used elsewhere       (d1:jcs:02/94)
             All defaults checked/modified if needed   (d2-3:jcs:3-4/94)
             Check-routine avoidance introduced            (e:jcs:05/94)
             Default-bar-size error in D1 Amnt corrected   (f:jcs:07/94)
             Automatic cover-setting routine improved      (g:jcs:07/94)
             Default-bar-size error in F Amnt corrected    (h:jcs:08/94)
             Link spacing criteria corrected               (i:jcs:01/95)
             Four sets of defaults provided, etc           (j:jcs:12/95)
             Metric/imperial version and QA upgrade        (ρ:jcs:10/96)
             Curly brackets corrected in deflection check  (k:jcs:01/97)
             Comp.bar size error in shear module corrected (l:jcs:04/97)
             Revised to meet BS8110:1997 requirements      (m:jcs:04/97)
             Modified to accept single bar in narrow rib   (n:jcs:06/98)
             Metric/imperial conversions removed           (p:jcs:08/98)
             Deflection check: tension-steel stress-limit
             corrected. BS8110(1997) now mandatory         (q:jcs:11/98)
             'Spring-cleaned'to remove extraneous material
             and span/eff.depth check corrected            (r:jcs:09/99)
             Option added to provide comp.steel solely to
             reduce deflection                             (s:jcs:12/99)
             Trap to ensure tension steel area OK added    (t:jcs:07/02)
             Cosmetic changes                              (u:dwb:12/03)
             Notes regarding shear close to support added  (w:jcs:04/06)
             Parameter table and self check added          (x:dwb:12/05)

PARAMETER   Start   End     Type    Dependency conditions
No. name    zst()   zen()   zty()   and notes.
 1  ans     0       0       0       Default values (1=Yes,0=No).
 2  user    1       0       2       More detailed description.
 3  Mbef    50      5000    0       Moment before redistribution.
 4  cont    0       0       2       Continuous or not.
 5  M       50      5000    0      =Mbef as beam not continuous.
```

```
 6  fcu    35      60      6      Char. concrete strength.
 7  hagg   10      60      6      Max aggregate size.
 8  fy     250     500     2      Char. strength of longitudinal.
 9  permn1 0.2     0.8     0      Minimum % when bw/b>=0.4.
10  permn2 0.2     0.8     0      Minimum % when bw/b<0.4.
11  dia    25      40     -3      Diameter of tension bars.
12  fyv    250     500     2      Char. strength of link steel.
13  dial   8       12      3      Diameter of link legs.
14  ccheck 1       0       2      Find cover (1=Yes,0=No).
15  d      250     2950    1     =(M*1E6*2/0.02/fy)^(1/3)
16  cover  20      80      1     >d*0.05 <d*0.1  Nominal cover.
17  h      300     3000    1     >d*1.1 <d*1.2  Overall depth.
18  b      300     3000    1     >d <2*d  Breadth of flange.
19  bw     200     600     1     >0.3*b <0.7*b  Breadth of rib.
20  hf     150     2950    1     >0.4*d <0.6*d  Thick. of flange.
21  diac   16      25     -3      Diameter of compression bars.
22  d'     40      100     7     >d/10 <d/6  Depth to compr.
23  nbart  2       20     19     =4*M/(0.75*d*fy*PI*dia^2)
24  ans0   1       1       0      Check span/eff.depth ratio.
25  btyp   2       2       1      Cant./ss./con-one-end/both-con.
26  ans5   1       0       2      Comp. bars to contrl. defln.
27  nbarc  2       10      1      No. of compr. bars provided.
28  span   1       6       0     >8*d/1E3 <15*d/1E3 Span of beam.
29  ans1   1       0       2      Should perm. be x10/span.
30  ans3   1       0       2      Find BM at bar curtail. points.
31  ans4   1       0       2      Data on anchorage & lap length.
32  Type   0       2       3      Plain, type-1&2 deformed round.
33  ans2   1       1       0      Undertake shear calculations.
34  V      1       10000   0     =4*M/span  Ultimate shear force.
35  av     1E-6    20000   0     =2*d Distance from support.
36  re1    0       0       2      Respecify main tension bars.
37  dias   16      25     -3      Diameter of tension bars.
38  nbars  2       20     19     =nbart  No. of bars effective.
39  re2    1       3       3      Options for comprn. bars.
40  diacs  16      25     -3      Diameter of comprn. bars.
41  nlegs  4       20      1     >nbars/2 <nbars   No. of legs.
42  flag1  1       2       2      Reduce spacing or links option.
43  dialr  8       12      3      Reduced dia. of link legs.
44  flag2  1       2       2      Adopt spacing or redesign optn.
45  sv'    50      3000    0     >d*.2 <d*.9 Chosen link spacing.
46  flag3  1       2       2      Options for incr. No. of legs.
47  ans6   0       0       2      Undertake another shear calc.
48  expos  1       5       5     <INT(fcu/12) Exposure condition.
49  mod    1       0       2      Systematic checking regime.
50  fire   0.5     2       4      Chosen fire resistance period.
51  expo   1       1       2      Indoor=1, outdoor=2.
52  aol    3       7       1      Age on loading 1 to 365 days.
53  Es     200E3   200E3   0      Young's modulus for steel.
54  Ec     28E3    28E3    0      Young's modulus for concrete.
55  euc    1       1       0      Eurocode check (1=Yes).

@scale.sta
STORE 3.10 4 3                   ! Table 3.9 BS 8110
        0  0.3  1.0
1     5.6  5.6  7.0
2    16.0 16.0 20.0
3    18.4 18.4 23.0           !this line interpolated from lines 2 & 4
4    20.8 20.8 26.0
+ans=1 +cover=0 +ZZZZZ1=0 +M=0 +$1504=
+diac=12                         !Set default comp.bar size
```

```
START
munits
! +lwconc=0                          !Lightweight concrete not enabled
! +nsbflag=0                         !Initialize non-standard-bar flag
! +nbars=0 +nbarc=0 +d'=0            !Initialize number of bars provided
IF ZZZZZ1=0
%
%
%<H1>Bending in flanged beams with optional calculations for shear, lap
%
%<H1>lengths, bar curtailment and limiting span/effective-depth ratio
%
%The required areas of tension steel (As) and compression steel (A's)
%are determined using the design formulae in Clause 3.4.4.4 of BS8110,
%i.e. assuming a rectangular concrete stress-block and limiting the
%depth to the neutral axis to 0.5*d.  The expression given in Clause
%3.4.4.4 to determine A's (if required) is, however, appropriately
%modified when the stress in the comp.steel is less than fy/gammaS,
%where gammaS is the appropriate partial safety factor for steel.
%This proforma also contains options to determine bending-moment values
%corresponding to bar curtailment points, to evaluate bar lengths needed
%for anchorage bond and laps, to design link reinforcement to resist
%shearing forces, and to check for excessive deflection by means of the
%limiting span/effective-depth ratio (see Cl.3.4.6 of BS8110: Part 1).
%
%Would you like a set of default values to be provided ?  You can use
%these as references and type your own values beneath to replace them.
!Answer (1=Yes/0=No)                  +ans=????
//
! +maximum=4 +minimum=-4 +value=ans
maxmin
! +rf1=10                            !Metric rounding factor +ans=ABS(ans)
IF ans=1
! +$100=Example: small beam - high-yield steel - no compression steel
! +$101=0.25 m from support
! +user=1 +Mbef=12 +cont=0 +fcu=35 +fy=460 +dia=12 +fyv=460 +dial=6
! +ccheck=1 +expos=1 +mod=1 +fire=1 +cover=20 +h=275 +d=243 +b=400
! +bw=200 +hf=100 +ans0=1 +btyp=2 +span=7 +ans1=1 +ans3=1 +ans4=1
! +Type=2 +ans2=1 +V=100 +av=250 +nbars=2 +nlegs=2 +sv'=100 +ans6=0
! +expo=1 +aol=3 +Es=200E3 +Ec=28E3
ENDIF
IF ans=2
! +$100=Narrow deep beam - mild steel - no compression steel
! +$101=0.5 m from support
! +user=0 +Mbef=1000 +cont=1 +M=1000 +fcu=30 +fy=250 +dia=25 +fyv=250
! +dial=12 +ccheck=1 +expos=2 +mod=1 +fire=2 +cover=35 +h=950 +d=890.5
! +b=1500 +bw=300 +hf=150 +ans0=1 +btyp=4 +span=17 +ans1=1 +ans3=1
! +ans4=1 +Type=0 +ans2=1 +V=800 +av=500 +re1=1 +dias=25 +nbars=8
! +re2=2 +nlegs=2 +sv'=400 +ans6=0
ENDIF
IF ans=3
! +$100=Shallow beam - square twist - with compression steel
! +$101=100 mm from support
! +user=0 +Mbef=95 +cont=1 +M=76 +fcu=35 +fy=425 +permn1=0.14
! +permn2=0.2 +dia=20 +fyv=250 +dial=8 +ccheck=0 +cover=30 +h=175
! +d=127 +b=1000 +bw=300 +hf=75 +diac=16 +d'=46 +ans0=1 +ans3=1 +ans4=1
! +ans2=1 +btyp=4 +span=2.3 +Type=0 +V=50 +av=100 +re1=0 +nbars=6
! +re2=1 +nlegs=2 +flag1=1 +sv'=90 +flag3=1 +ans6=0
ENDIF
```

```
IF ans=4
! +$100=Large beam - high-yield steel - with compression steel
! +$101=0.459 m from support
! +user=0 +Mbef=9567 +cont=1 +M=9345 +fcu=50 +fy=460 +dia=40 +fyv=460
! +dial=12 +ccheck=1 +expos=3 +mod=1 +fire=4 +cover=50 +h=1000 +d=918
! +b=2000 +bw=500 +hf=175 +diac=20 +d'=76 +ans0=1 +btyp=3 +span=14
! +ans1=0 +ans3=1 +ans4=1 +Type=2 +ans2=1 +V=2210 +av=459 +re1=1
! +dias=40 +nbars=24 +re2=2 +diacs=20 +nlegs=4 +flag1=1 +dialr=8
! +sv'=240 +ans6=0
ENDIF
%
%Display a more-detailed description of the operation of this proforma.
%
!Answer (1=Yes/0=No)              +user=????
//
! +maximum=1 +minimum=0 +value=user
maxmin
IF user=1
notes
ENDIF
ENDIF
! +ZZZZZ1=1
.
Location: +$100=????
.
//
/14
.Bending in flanged beams with optional calculations for shear, lap
.__________________________________________________________________
.lengths, bar curtailment and limiting span/effective-depth ratio
.________________________________________________________________
.                                            d'        b
.
.Calculations are based on formulae          hf           (o)(o)
.in Clause 3.4.4.4 of BS8110: Part 1                        A's
.and thus assume the use of a simplified                                 d  h
.rectangular concrete stress-block, and                     As
.that the depth to the neutral axis is                     o  o
.restricted to 0.5*d.
.                                                           bw
.
.Design to BS8110(1997) with partial safety factor for steel +gammaS=1.0
>
Moment before redistribution      +Mbef=????  kNm
! +maximum=5000+1E-6 +minimum=1 +value=Mbef
maxmin
IF M=0 THEN M=1.0E39 M=Mbef ENDIF ! Set default value of M.
>
%
%If the beam containing the section being considered is supported at
%both ends and continuous at at least one, respond 'Yes' to next prompt.
%
!Is beam continuous (1=Yes/0=No)   +cont=????
! +maximum=1 +minimum=0 +value=cont
maxmin
>
IF cont=1
.Beam being analysed is considered as continuous.
Design moment (after redistrib.)  +M=????  kNm
! +maximum=5000+1E-6 +minimum=1 +value=M
maxmin
! +btyp=4
```

```
ELSE
.Beam being analysed is considered as non-continuous.
! +btyp=2 +M=1.0E39 +M=Mbef
ENDIF
//
>
! +hagg=15
EDIT /W 3
Characteristic concrete strength  +fcu=???? N/mm2
Max.aggregate size (for bar spc)  +hagg=???? mm
Char.strength of long'l bars      +fy=???? N/mm2
! +maximum=65+1E-6 +minimum=15 +value=fcu
maxmin
chkconc
! +maximum=60+1E-6 +minimum=5 +value=hagg
maxmin
! +error=1
IF ABS(fy-250)<1
.Longitudinal reinforcement is mild steel.
! +permn1=0.24 +permn2=0.32 +error=0
ENDIF
IF ABS(fy-460)<1
.Longitudinal reinforcement is high-yield steel.
! +permn1=0.13 +permn2=0.18 +error=0
ENDIF
IF error=1
! +error=0
//
%
%Characteristic steel strength is normally either +250 or +460 N/mm2
%
<<<
.Steel strength other than those given in Table 3.1 of BS8110.
! +permn1=1.0E39 +permn2=1.0E39
! +permn1=INT(.5+100*(77.9-.11*fy)/210)/100
! +permn2=INT(.5+100*(102.2-.14*fy)/210)/100
//
%
%As you have chosen a steel strength other than those given in Table 3.1
%of BS8110 the minimum percentages given in Table 3.25 (i.e. 0.24% and
%0.32% when fy= +250 N/mm2 and 0.13% and 0.18% when fy= +460 N/mm2 )
%are invalid and you must specify your own values.
%
EDIT /W 2
Min.percentage when bw/b>=0.4     +permn1=???? %
Min.percentage when bw/b<0.4      +permn2=???? %
//
ENDIF
>
EDIT /W 4
Diameter of tension bars          +dia=???? mm
Char.strength of link steel       +fyv=???? N/mm2
Diameter of link legs             +dial=???? mm
!Find cover, etc.(1=Yes/0=No)      +ccheck=????
! +di=dia
chkbar
! +dias=dia
! +error=1
IF ABS(fyv-250)<1
.Mild steel shear reinforcement.
! +error=0
ENDIF
```

```
IF ABS(fyv-460)<1
.High-yield steel shear reinforcement.
! +error=0
ENDIF
IF error=1
! +error=0
//
%
%Characteristic steel strength is normally either +250 or +460 N/mm2
<<<
.Shear steel strength other than those given in Table 3.1 of BS8110.
! +maximum=460+1E-6 +minimum=250 +value=fyv
maxmin
ENDIF
! +di=dial
chkbar
! +maximum=1 +minimum=0 +value=ccheck
maxmin
IF ccheck=1
chkcover                          !Min.cover set during this subroutine
! +mcover=mcover
ELSE
! +mcover=30                      !Set default value otherwise
ENDIF
IF dia-dial>mcover
! +mcover=dia-dial
According to Clause 3.3.1.2, nominal cover to main bars must be not
less than bar size, so increase minimum cover to dia-dial= +mcover mm
ENDIF
Minimum nominal cover to all steel is +mcover mm
IF cover<mcover THEN cover=1.0E39 cover=mcover ENDIF !Set cover default

>
EDIT /W 2
Nominal concrete cover            +cover=???? mm
Overall depth of section          +h=???? mm
! +maximum=80+1E-6 +minimum=mcover +value=cover
maxmin
! +ccover=cover
! +maximum=3000+1E-6 +minimum=100 +value=h
maxmin
>
! +dia2=dia/2 +d=1.0E39 +d=h-cover-dial-dia2 +vs=hagg/1.5
IF dia>vs THEN vs=dia ENDIF
! +d2=h-cover-dial-dia-vs/2
//
%
%Max.effective depth of section is overall depth of +h mm minus
%nominal cover of +cover mm minus link size of +dial mm minus one-
%half of main bar diameter of +dia2 mm .  Thus max.d is +d mm but
%is reduced to +d2 mm if two layers are needed to accommodate steel.
%
EDIT /W 4
Effective depth of section        +d=???? mm
Breadth of flange                 +b=???? mm
Breadth of rib                    +bw=???? mm
Thickness of flange               +hf=???? mm
! +maximum=h-cover-dial-dia2+0.1 +minimum=h-200
IF h>400 THEN minimum=h/2
! +value=d
maxmin
```

```
IF d>maximum
Depth to tension steel cannot exceed +maximum mm
STOP
ENDIF
! +maximum=3000+1E-6 +minimum=75 +value=b
maxmin
! +maximum=b+1E-6 +minimum=75 +value=bw
maxmin
IF ccheck=1
chksize
IF bw<cover2-1E-3
Breadth insufficient for fire regulations (Fig.3.2).  Please increase.
<<<
STOP
ELSE
Breadth provided complies with fire regulations (Figure 3.2 of Code).
ENDIF
ENDIF
! +maximum=d+1E-6 +minimum=75 +value=hf
maxmin
!
! +MnSpct=hagg+5 Determine max.number of tension bars in a single layer
IF dia>MnSpct THEN MnSpct=dia ENDIF
! +layer=INT((b-2*(cover+1.1*dial)+MnSpct)/(1.1*dia+MnSpct))
/10
.
.Longitudinal reinforcement
.
! +xmax=0.5*d +K'=0.156 +As'=0 +As'pr=0
IF Mbef<M
As Mbef is less than M,
thus redistribution ratio +betab=1
xmax=0.5*d= +xmax mm and K'=0.156 [Condition (b) in Clause 3.2.2.1].
ENDIF
IF Mbef=M
As Mbef=M, no redistribution has occurred.
Thus redistribution ratio +betab=1
xmax=0.5*d= +xmax mm and K'=0.156 [Condition (b) in Clause 3.2.2.1].
ELSE
! +perct=100*(Mbef-M)/Mbef
IF perct<=10
Percentage redistribution +perct=100*(Mbef-M)/Mbef % (i.e. <= 10%).
Thus redistribution ratio +betab=M/Mbef
with xmax=0.5*d= +xmax mm and K'=0.156.
ELSE
IF perct>30
The proposed reduction in bending moment due to redistribution
exceeds 30%, which is not permitted.
STOP
ENDIF
Percentage redistribution    +perct=100*(Mbef-M)/Mbef % (exceeds 10%).
Thus redistribution ratio    +betab=M/Mbef
                             +xmax=(betab-0.4)*d mm  (see Cl.3.2.2.1),
and                          +K'=0.402*(betab-0.4)-0.18*(betab-0.4)^2
ENDIF
ENDIF
! +dpmax=0.9*xmax
>
IF dpmax<=hf
Max.depth of conc.stress-block    +dpmax=0.9*xmax mm
As maximum depth of stress-block of +dpmax mm cannot exceed flange
thickness of +hf mm , design section as rectangular beam of breadth b.
```

```
! +K=M*1000000/(b*d^2*fcu)     !i.e. K assuming beam is rectangular bxd
IF K<=K'
DSSRRECTBEAM
ELSE
Applied-moment factor             +K=M*1000*1000/(b*d^2*fcu)
/3
As applied-moment factor K= +K exceeds resistance-moment factor
of concrete alone K'= +K' , compression steel is required, so that
lever arm                         +z=d*(0.5+SQR(0.25-K'/0.9)) mm
and depth to neutral axis         +x=(d-z)/0.45 mm
!Diameter of compression bars      +diac=???? mm
! +di=diac
chkbar
Diameter of compression bars      diac= +diac mm
! +diacs=diac
IF diac-dial>ccover
! +ccover=diac-dial
According to Clause 3.3.1.2, nominal cover to main bars must be not less
than bar size, so increase cover to comp.bars to diac-dial= +ccover mm
ENDIF
! +diac2=diac/2 +d'=1.0E39 +d'=ccover+dial+diac2 +mind'=d'
<<//
%
%Minimum depth to main compression bars is nominal cover of +ccover mm
%plus size of link of +dial mm plus one-half of size of compression
%bar of +diac2 mm .  Thus minimum depth is +d' mm .
%
Depth to compression bars         +d'=???? mm
! +maximum=d/2+1E-6
IF x<maximum THEN maximum=x+1E-6
! +minimum=mind'-1E-3 +value=d'
maxmin
IF d'<minimum
Depth to compression steel cannot be less than +minimum mm
STOP
ENDIF
IF d'>maximum
Depth to compression steel cannot exceed +maximum mm
STOP
ENDIF
COMPF'S
Area of compression steel reqd.   +As'=(K-K')*fcu*b*d^2/(f's*(d-d')) mm2
! +vsc=hagg/1.5
IF diac>vsc THEN vsc=diac
! +MnSpcc=hagg+5 Determine max.number of comp.bars in single layer.
IF diac>hagg+5 THEN MnSpcc=diac ENDIF
! +layerc=INT((b-2*(cover+1.1*dial)+MnSpcc)/(1.1*diac+MnSpcc))
IF f's=fy/gammaS
Area of tension steel required   +As=K'*fcu*b*d^2/(z*fy/gammaS)+As' mm2
ELSE
Area of tension steel required
                 +As=K'*fcu*b*d^2/(z*fy/gammaS)+As'*f's/(fy/gammaS) mm2
ENDIF
ENDIF
ENDIF
>
IF dpmax>hf
Max.depth of conc.stress-block    +dpmax=0.9*xmax mm
As max.depth of stress-block of +dpmax mm exceeds flange thickness of
+hf mm , stress-block may extend below flange: check resistance moment.
! +Mfl=b*hf*0.45*fcu*(d-hf/2)/1000000
IF M<=Mfl
```

```
Moment capacity of flange alone +Mfl=b*hf*0.45*fcu*(d-hf/2)/1000000 kNm
As this exceeds M, lower edge of stress-block does not fall below
underside of flange and section may be designed as a rectangular beam.
DSSRRECTBEAM
ELSE

Moment capacity of flange alone +Mfl=b*hf*0.45*fcu*(d-hf/2)/1000000 kNm
As this is less than M, stress block extends below underside of flange
(and compression steel may be required).
Moment capacity of flange arms
                          +Mof=0.45*fcu*(b-bw)*hf*(d-hf/2)/1000000 kNm
Area of tension steel required
to resist this moment          +Asof=0.45*fcu*(b-bw)*hf/(fy/gammaS) mm2
/6
Now the web section of the beam must be designed to resist the
remaining moment                  +Mweb=M-Mof kNm
! +K=Mweb*1000000/(bw*d^2*fcu)
IF K<=K'
Evaluate applied-moment factor    +K=Mweb*1000*1000/(bw*d^2*fcu)
As this applied-moment factor ( +K ) is less the resistance-moment
factor of the web ( +K' ), no compression steel is required,
and lever arm                     +z=d*(0.5+SQR(0.25-K/0.9)) mm
! 0.95*d cannot govern
Area of tension steel required +Asweb=Mweb*1000*1000/(z*fy/gammaS) mm2
ELSE
Evaluate applied-moment factor    +K=Mweb*1000*1000/(bw*d^2*fcu)
As this applied-moment factor ( +K ) exceeds the resistance-moment
factor of the web ( +K' ), compression steel is required, so that
lever arm                         +z=d*(0.5+SQR(0.25-K'/0.9)) mm
and depth to neutral axis         +x=xmax mm (see above).
!Diameter of compression bars      +diac=???? mm
! +di=diac
chkbar
Diameter of compression bars      diac= +diac mm
! +diacs=diac
IF diac-dial>ccover
! +ccover=diac-dial
According to Clause 3.3.1.2, nominal cover to main bars must be not less
than bar size, so increase cover to comp.bars to diac-dial= +ccover mm
ENDIF
! +diac2=diac/2 +d'=1.0E39 +d'=ccover+dial+diac2 +mind'=d'
<<//
%
%Minimum depth to main compression bars is nominal cover of +ccover mm
%plus size of link of +dial mm plus one-half of size of compression
%bar of +diac2 mm .  Thus minimum depth is +d' mm .
%
Depth to compression bars         +d'=???? mm
! +maximum=d/2+1E-6
IF x<maximum THEN maximum=x+1E-6
! +minimum=mind'-1E-3 +value=d'
maxmin
IF d'<minimum
Depth to compression steel cannot be less than +minimum mm
STOP
ENDIF
IF d'>maximum
Depth to compression steel cannot exceed +maximum mm
STOP
ENDIF
COMPF'S
/10  'throws' to next page
```

```
Area of compression steel reqd. +As'=(K-K')*fcu*bw*d^2/(f's*(d-d')) mm2
!
! +vsc=hagg/1.5
IF diac>vsc THEN vsc=diac
! +MnSpcc=hagg+5 Determine max.number of comp.bars in single layer.
IF diac>hagg+5 THEN MnSpcc=diac ENDIF
! +layerc=INT((b-2*(cover+1.1*dial)+MnSpcc)/(1.1*diac+MnSpcc))
IF f's=fy/gammaS
Area of tension steel reqd +Asweb=K'*fcu*bw*d^2/(z*fy/gammaS)+As' mm2
ELSE
Area of tension steel required
           +Asweb=K'*fcu*bw*d^2/(z*fy/gammaS)+As'*f's/(fy/gammaS) mm2
ENDIF
ENDIF
Total area of tension steel reqd. +As=Asof+Asweb mm2
ENDIF
ENDIF
/10 (throws to next page)
IF As'<>0
! +As'mn=0.4*b*hf/100
IF As'mn>As'
Minimum permissible amount of compression steel (Table 3.25) is 0.4%;
i.e.                               +As'mn=0.4*b*hf/100 mm2
As this exceeds calculated area of As' reqd., make As'= +As'mn mm2
! +As'=As'mn
ENDIF
! +nbarc=INT(As'/(PI*diac^2/4))+1
IF nbarc<2
Minimum number of comp.bars reqd. +nbarc=2
ELSE
Minimum number of comp.bars reqd. +nbarc=INT(As'/(PI*diac^2/4))+1
ENDIF
! +As'pr=nbarc*PI*diac^2/4
Comp.steel area provided As'pr= +As'pr mm2 ( +nbarc No. +diac mm bars).
! +Asmax=0.04*(b*hf+bw*(h-hf)) mm2
IF As'pr>Asmax
Percentage of comp.steel provided +per'=100*As'pr/(b*hf+bw*(h-hf)) %
As this exceeds 4% (see Table 3.25), design changes must be made.
Increase either the concrete strength or the section dimensions.
STOP
ELSE
Percentages of compression steel provided:
  Of flange area: +per'=100*As'pr/(b*hf) % (Code limit 0.4% min.)
  Of gross area : +per''=100*As'pr/(b*hf+bw*(h-hf)) % (Code limit 4%)
As this falls within Code limits, this is satisfactory.
ENDIF
! +per'0=per' +per'=100*As'pr/(bw*h) %
ENDIF
IF As'>0
! +scr=12*diac
Spacing of links containing compression bars (see Clause 3.12.7.1 of
BS8110) must not exceed           12*diac= +scr mm
! +di=diac
minlink
Minimum diameter of these links is one-quarter of that of compression
bars (Clause 3.12.7.1), i.e.      +dialm mm
IF dialm>dial
//
Min.permissible size of links containing compression bars exceeds
link size previously assumed.
<<<
STOP
```

```
ENDIF
ENDIF
IF bw/b<0.4
! +Asmin=permn2*bw*h/100
IF As<Asmin
As ratio bw/b is less than 0.4, thus from specified input,
Minimum permissible steel area    +Asmin=permn2*bw*h/100 mm2
As this exceeds calculated steel area, make As= +Asmin mm2
! +As=Asmin
ENDIF
ELSE
! +Asmin=permn1*bw*h/100
IF As<Asmin
As ratio bw/b is not less than 0.4, thus from specified input,
Minimum permissible steel area    +Asmin=permn1*bw*h/100 mm2
As this exceeds calculated steel area, make As= +Asmin mm2
! +As=Asmin
ENDIF
ENDIF
! +nbrtmn=INT(As/(PI*dia^2/4))+1
IF bw>=2*(cover+1.1*(dia+dial))+MnSpct
IF nbrtmn<2 THEN nbrtmn=2
ENDIF
Min.number of tension bars reqd.  nbrtmn= +nbrtmn
! +nbart=nbrtmn
IF nbars>nbart THEN nbart=nbars
REPEAT
! +loop=1
Number of tension bars provided   +nbart=????
! +Aspr=nbart*PI*dia^2/4
IF nbart<nbrtmn
%
%Number of tension bars provided cannot be less than +nbrtmn
%Please reconsider.
%
! +loop=0
ENDIF
UNTIL loop=1
ENDREPEAT
! +Aspr=nbart*PI*dia^2/4 +Asprx=Aspr
Area of tension steel provided Aspr= +Aspr mm2 ( +nbart No. +dia mm )
! +Asmax=0.04*(b*hf+bw*(h-hf))
/10                               !'throws' to next page
IF Aspr>Asmax
Tension steel prov. +per0=100*Aspr/(b*hf+bw*(h-hf)) % of gross section.
As this exceeds 4% (see Clause 3.12.6.1), design changes must be made.
Increase either the concrete strength or the section dimensions.
STOP
ELSE
Tension steel prov. +per0=100*Aspr/(b*hf+bw*(h-hf)) % of gross section.
As this does not exceed 4% (see Clause 3.12.6.1), the design is O.K.
ENDIF
IF bw/b>=0.4
Percentage of tension steel prov. +per=100*Aspr/(bw*h) % of web area.
As this is not less than minimum of +permn1 %, the design is O.K.
ELSE
Percentage of tension steel prov. +per=100*Aspr/(bw*h) % of web area.
As this is not less than minimum of +permn2 %, the design is O.K.
ENDIF
IF nbart>1
! +cdist=47000*3*Aspr*betab/(2*fy*As)
IF cdist>300
/9
```

```
Maximum clear distance allowed between tension bars according to
Clause 3.12.11.2.4 is given by
                           +cdist=47000*3*Aspr*betab/(2*fy*As) mm
but as this exceeds +300 mm , take +cdist=300 mm
ELSE
/8
Maximum clear distance allowed between tension bars according to
Clause 3.12.11.2.4 is given by
                           +cdist=47000*3*Aspr*betab/(2*fy*As) mm
which is satisfactory as it is less than +300 mm
! +cdist=INT(cdist)              !Rounds cdist down to nearest mm
! +nmin=(bw-2*(cover+dial)-dia)/cdist +nmin=INT(nmin)+2
IF nmin<2 THEN nmin=2
Min.number of tension bars needed to meet bar-spacing req.= +nmin
IF nmin>nbart
<<//
/5
    Additional tension bars must be provided to satisfy bar-spacing
    requirements.  Please rerun proforma, selecting an increased
    number of tension bars to meet the above requirement.
STOP
ENDIF
ENDIF
ENDIF
! +WtT=.00785*Aspr +WtC=.00785*As'pr
IF layer<1 THEN layer=1
! +tlayer=INT(nbart/layer-1E-3)+1
/15
.
.   TENSION                         Characteristic strength  +fy N/mm2
.   REINFORCEMENT                   Diameter of bars         +dia mm
.   SUMMARY                         Number of bars           +nbart
IF tlayer=1
.                                   arranged in a single layer
ELSE
.                                   arranged in +tlayer layers
ENDIF
.                                   Cover to all steel       +cover mm
.                                   Area of steel required   +As mm2
.                                   Area of steel provided   +Aspr mm2
.                                   Percentage provided:
.                                     of gross section       +per0 %
.                                     of web area            +per %
.                                   Weight of steel provided +WtT kg/m
IF nbart>1
.                                   Max.permissible spacing  +cdist mm
ENDIF
.                                   Link size assumed        +dial mm
.
IF As'>0
/14
! +clayer=INT(nbarc/layerc-1E-3)+1
.   COMPRESSION                     Characteristic strength  +fy N/mm2
.   REINFORCEMENT                   Diameter of bars         +diac mm
.   SUMMARY                         Number of bars           +nbarc
IF clayer=1
.                                   arranged in a single layer
ELSE
.                                   arranged in +clayer layers
ENDIF
```

```
.                                   Cover to all steel       +ccover mm
.                                   Area of steel required   +As' mm2
.                                   Area of steel provided   +As'pr mm2
.                                   Percentage provided:
.                                     of gross section       +per'' %
.                                     of web area            +per' %
.                                     of flange area         +per'0 %
.                                   Weight of steel provided +WtC kg/m
.                                   Minimum poss.link size   +dialm mm
.                                   Maximum spacing of links +scr mm
.
ENDIF
IF tlayer>1
IF d>d2
WARNING: As more than one layer of tension bars is required, you should
determine a revised effective depth corresponding to the position of
the centroid of this reinforcement, and rerun this proforma.
STOP
ENDIF
ENDIF
IF As'>0
IF clayer>1
IF d'+(clayer-1)*(1.1*diac+vsc)>x
.WARNING: Lowest layer of compression reinforcement is located below
.          neutral axis, which is impossible. Please redesign section.
.
STOP
ENDIF
WARNING: As more than one layer of compression bars is required, you
should determine a revised depth of comp.steel corresponding to the
position of the centroid of this steel, and rerun this proforma.
STOP
ENDIF
ENDIF
IF h>750
.As overall depth h exceeds +750 mm , bars to control cracking must be
.provided on side faces of beam (see Code Cl.3.12.5.4 and 3.12.11.2.9).
.
ENDIF
>
<<//
%
%Do you wish to check the span/effective-depth ratio ?
%Note: This is only relevant if the section being designed is:
%      at the support of a cantilever beam;               or
%      at midspan of a simply-supported or a continuous beam.
%
!Check ratio (1=Yes/0=No)          +ans0=????
//
! +maximum=1 +minimum=0 +value=ans0
maxmin
>
IF ans0=1
/10
//
.
.Check on span/effective-depth ratio
.___________________________________
IF cont=0
%
```

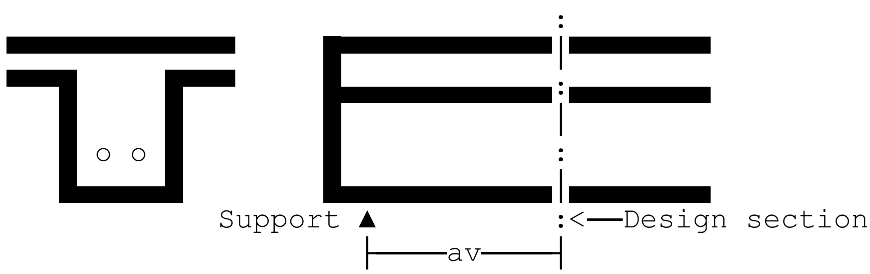
```
%The following types of beam may be considered:
%  1. Cantilever beam.
%  2. Simply-supported beam.
%
!Which (1-2)                       +btyp=????
//
! +maximum=2 +minimum=1 +value=btyp
maxmin
ELSE
%
%The following types of beam may be considered:
%  3. Beam continuous at one end and simply supported at the other.
%  4. Beam continuous at both ends.
%
!Which (3-4)                       +btyp=????
//
! +maximum=4 +minimum=3 +value=btyp
maxmin
ENDIF
Calculate redistribution ratio    +btab=M/Mbef
Ratio of web width/flange width   +bw'b=bw/b
IF btyp=1
! +bs'd=TABLE(3.10,1,bw'b)
.Interpolating from Table 3.9 with bw/b= +bw'b
.Basic ratio for cantilever span   bs'd= +bs'd
IF btab<>1
As redistribution has been carried out at the section just designed
the beam must be continuous so enter Beam Type 3 or 4 instead.
<<<
STOP
ENDIF
ENDIF
IF btyp=2
! +bs'd=TABLE(3.10,2,bw'b)
.Interpolating from Table 3.9 with bw/b= +bw'b
.Basic ratio for simply-sup.span   bs'd= +bs'd
IF btab<>1
As redistribution has been carried out at the section just designed
the beam must be continuous, so enter Beam Type 3 or 4 instead.
<<<
STOP
ENDIF
ENDIF
IF btyp=3
! +bs'd=TABLE(3.10,3,bw'b)
.Interpolating from Table 3.9 with bw/b= +bw'b
.Basic ratio for beam
.continuous at one end        bs'd= +bs'd
ENDIF
IF btyp=4
! +bs'd=TABLE(3.10,4,bw'b)
.Interpolating from Table 3.9 with bw/b= +bw'b
.Basic ratio for continuous span   bs'd= +bs'd
ENDIF
Area of tension steel provided    Aspr= +Asprx mm2
Service stress in this steel      +fs=2*fy*As/(3*Asprx*btab) N/mm2
IF fs>460/1.5
As fs exceeds 306.7 N/mm2 Table 3.10 is invalid (see Clause 3.4.6.5).
Either the steel yield stress is too high or the moment redistribution
from the midspan section is excessive.
STOP
ENDIF
```

```
.As applied-moment factor          +M'bd2=M*1000*1000/(b*d^2) N/mm2
.Mod.factor for tension steel from
.equation 7 (Table 3.10) +modf1=0.55+(477-fs)/(120*(.9+M'bd2))
IF modf1>2
.but this cannot exceed 2, so      +modf1=2
ENDIF
IF As'pr>0
.Area of comp.steel provided       As'pr= +As'pr mm2
.Percentage of compression steel   +per'n=100*As'pr/(b*d) %
.From Equation 9 of BS8110, with percentage of comp.steel= +per'n %,
.Mod.factor for compression steel  +modf2=1+per'n/(3+per'n)
IF modf2>1.5
. but this cannot exceed 1.5, so   +modf2=1.5
ENDIF
ELSE
! +modf2=1 +ans5=0
!Provide comp.bars to help control deflection (1=Yes/0=No) +ans5=????
! +maximum=1 +minimum=0 +value=ans5
maxmin
IF ans5=1
%
%Remember that these bars must be secured with links that meet the
%requirements set out in Clauses 3.12.7.1 and 3.12.7.2 of BS8110.
%
EDIT /W 2
Number of comp.bars provided      +nbarc=????
Diameter of compression bars      +diac=???? mm
//
! +maximum=10 +minimum=2 +value=nbarc
maxmin
! +di=diac
chkbar
! +As'pr=nbarc*PI*diac^2/4
.Area of comp.steel provided       As'pr= +As'pr mm2
.Percentage of compression steel   +per'n=100*As'pr/(b*d) %
.From Equation 9 of BS8110, with percentage of comp.steel= +per'n %,
.Mod.factor for compression steel  +modf2=1+per'n/(3+per'n)
IF modf2>1.5
. but this cannot exceed 1.5, so   +modf2=1.5
ENDIF
ELSE
.Mod.factor for no comp.steel      +modf2=1
ENDIF
ENDIF
.Maximum permissible
.span/effective-depth ratio        +ps'd=bs'd*modf1*modf2
Span of beam (see Cls.3.4.1.2-4)  +span=???? m
! +maximum=20+1E-6 +minimum=1 +value=span
maxmin
IF span>10
IF btyp=1
As span exceeds +10 m and member is a cantilever, the method of
satisfying deflection requirements using span/effective-depth ratios
does not apply — see Clause 3.4.6.4 of BS8110: Part 1.
STOP
ELSE
%
%As span exceeds +10 m should the permissible ratios given in Table
%3.9 of BS8110 be multiplied by 10/span (in m) — see Cl.3.4.6.4. (This
%is required if partitions and/or finishes are sensitive to deflection.)
%
!Answer (1=Yes/0=No)               +ans1=????
```

```
//
! +maximum=1 +minimum=0 +value=ans1
maxmin
IF ans1=1
.As span exceeds +10 m and partitions/finishes are sensitive to
.deflection multiply basic ratio by 10/span (in m) +ps'd=ps'd*10/span
ELSE
.Although span exceeds +10 m partitions/finishes are not considered
.sensitive to deflection so modification of basic ratio is unnecessary.
ENDIF
ENDIF
ENDIF
! +as'd=1000*span/d
IF as'd>ps'd
.Actual span/effective-depth ratio +as'd=1000*span/d
.As this exceeds                        +ps'd , this is Unacceptable.
STOP
ELSE
.Actual span/effective-depth ratio +as'd=1000*span/d
.As this does not exceed                +ps'd , this is Acceptable.
ENDIF
ENDIF
>
%
%Find bending-moment values corresponding to bar-curtailment points ?
%
!Answer (1=Yes/0=No)                +ans3=????
! +maximum=1 +minimum=0 +value=ans3
maxmin
IF ans3=1
/14
.
.Bending-moment capacities at steel curtailment points
._____________________________________________________
IF nbart<3
IF nbart=2
.As there are only two bars in tension, no curtailment is possible.
ELSE
.As there is only one bar in tension, no curtailment is possible.
ENDIF
ELSE
IF As'pr<>0
! +As'rd=As'pr — sets initial value for As'red
! +nbcr=nbarc — sets initial value for nbcr
ENDIF
IF nbart-2*INT(nbart/2)=1
.The number of tension bars is odd, so reduce by a single bar initially
.and then in pairs.
! +nbtr=nbart-1
ELSE
.The number of tension bars is even, so reduce in pairs.
! +nbtr=nbart-2
ENDIF
! +flag0=0                        !Initialise comp.bar message flag
REPEAT
/10
.
REDUCE
UNTIL nbtr=0
ENDREPEAT
.
```

```
.No further curtailment is possible. However, bars must extend beyond
.the theoretical curtailment points as required below.
.
! +scr=12*dia
IF d>=12*dia
.Min.extension to tension bars equal to eff.depth d of +d mm
.as this is not less than 12*tension bar diameter, i.e. +scr mm .
ELSE
.Min.extension to tension bars equal to 12*bar diameter, i.e. +scr mm
.as this is not less than the effective depth d of +d mm .
ENDIF
IF As'<>0
.
! +scr=12*diac
IF d>=12*diac
.Min.extension to comp.bars equal to effective depth d of +d mm
.as this is not less than 12*comp.bar diameter, i.e. +scr mm .
ELSE
.Min.extension to comp.bars equal to 12*bar diameter, i.e. +scr mm
.as this is not less than the effective depth d of +d mm .
ENDIF
ENDIF
ENDIF
ENDIF
>
<<//
%
%Is data on anchorage and lap lengths required ?
%
!Answer (1=Yes/0=No)               +ans4=????
//
! +maximum=1 +minimum=0 +value=ans4
maxmin
>
IF ans4=1
/16
.
.Anchorage and lap lengths (Clause 3.12.8 of BS8110: Part 1)
.___________________________________________________________
//
%
%Types of bar (see Table 3.26 in Code) are as follows:
%  0. Plain bars
%  1. Type 1 deformed bars
%  2. Type 2 deformed bars
%
!Which (0-2)                       +Type=????
//
! +maximum=2 +minimum=0 +value=Type
maxmin
IF Type=0
.Plain bars: bond coefficient      +beta=0.28 (see Table 3.26)
ELSE
IF Type=1
.Type 1 deformed bars: bond coefficient +beta=0.40 (see Table 3.26)
ELSE
.Type 2 deformed bars: bond coefficient +beta=0.50 (see Table 3.26)
ENDIF
ENDIF
```

```
.Tension reinforcement:
.Force in tension bar at yield     +Ft=fy/gammaS*PI*dia^2/4 N
.Ultimate anchorage bond stress    +fbu=beta*SQR(fcu) N/mm2
.Anchorage and tension lap length  +lap=Ft/(PI*dia*fbu) mm
! +lap14=rf1*INT(1.4*lap/rf1)+rf1 +lap2=rf1*INT(2*lap/rf1)+rf1
! +lap=rf1*INT(lap/rf1)+rf1
.                                     = +lap mm (rounded)
.1.4 * tension lap length             = +lap14 mm (rounded)
.2 * tension lap length               = +lap2 mm (rounded)
IF As'<>0
/10
.Compression reinforcement:
IF Type=0
.Plain bars: bond coefficient      +beta=0.35 (see Table 3.26)
ELSE
IF Type=1
.Type 1 deformed bars: bond coefficient +beta=0.50 (see Table 3.26)
ELSE
.Type 2 deformed bars: bond coefficient +beta=0.63 (see Table 3.26)
ENDIF
ENDIF
.Force in comp.bar at yield        +Fsc=fy/gammaS*PI*diac^2/4 N
.Ultimate anchorage bond stress    +fbu=beta*SQR(fcu) N/mm2
.Anchorage length in compression   +lac=Fsc/(PI*diac*fbu) mm
! +lap=rf1*INT(1.25*lac/rf1)+rf1 +lac=rf1*INT(lac/rf1)+rf1
.                                     = +lac mm (rounded)
.Compression lap length (1.25*lac)    = +lap mm (rounded)
ENDIF
ENDIF
>
<<//
%
%Do you want to undertake shear calculations ?
%
!Answer (1=Yes/0=No)               +ans2=????
//
! +maximum=1 +minimum=0 +value=ans2
maxmin
>
IF ans2=1
REPEAT       !This allows several sections to be designed if required.
! +dialr=dial
/10
.
Location along beam for shear calculation: +$101=????
.
.Shear calculations
.are in accordance
.with Clauses 3.4.5
.of Code.
.
.
.
.
```




```
Effective breadth for shear        +bv=bw mm
Shear force due to ultimate load   +V=???? kN
! +maximum=1000+1E-6 +minimum=1 +value=V
maxmin
! +d2=d/2
%
%For several reasons (see Code Handbook), there is seldom any point in
%examining sections closer to the support than d/2 (i.e. +d2 mm ).
%
```

```
Distance from support             +av=???? mm
! +maximum=20000+1E-6 +minimum=1E-6 +value=av
maxmin
//
%
%For detailed requirements concerning 'effective' tension bars for data
%input below, see Clause 3.4.5.4 of BS8110.  Main bars must extend a
%distance of at least effective depth beyond section being considered.
%Near supports, all tension bars may be included, provided that rules
%in Clause 3.12.8 regarding bar curtailment and anchorage are satisfied.
%
IF btyp>2
%
%At those supports where the beam is continuous, if the location of the
%section that is being considered is between the support and the point
%of contraflexure the tension steel is now that near the top of the beam
%and you may need to respecify the number and size of these bars.
%
!Respecify main tension bars (1=Yes/0=No)  +re1=????
! +maximum=1 +minimum=0 +value=re1
maxmin
IF re1=1
Diameter of tension bars          +dias=???? mm
! +di=dias
chkbar
ENDIF
ENDIF
! +nbars=1.0E39 +nbars=nbart
No.of tension bars effective at section +nbars=????
! +maximum=nbart +minimum=2 +value=nbars
maxmin
IF btyp>2
//
%
%Compression reinforcement may now be required near the bottom of the
%section.  This requirement may be satisfied by utilising the principal
%tension bars from midspan, although it may be difficult to provide the
%full compression-bond lap needed.  Alternatively, separate compression
%bars may be employed.
%
%Options are as follows:
%  1. No compression bars required
%  2. Size of compression bars unchanged
%  3. Size of compression bars changed
%
!Which (1-3)                       +re2=????
//
! +maximum=3 +minimum=1 +value=re2
maxmin
IF re2=1 THEN As'=0
IF re2=3
Diameter of compression bars      +diacs=???? mm
! +di=diacs
chkbar
minlink
! +As'=1
ENDIF
ENDIF
Area of eff.longitudinal bars     +As=nbars*PI*dias^2/4 mm2
Percentage of longitudinal steel  +per=100*As/(bv*d) %
```

```
IF fcu<39.0625
Clause 3.4.5.2 limits the maximum shearing stress to 0.8√fcu N/mm2
and thus limiting shear stress    +vlim=0.8*SQR(fcu) N/mm2
ELSE
Clause 3.4.5.2 limits the maximum shearing stress to +5 N/mm2
and thus limiting shear stress    +vlim=5 N/mm2
ENDIF
! +pcnt=per
IF pcnt>3                         !Set upper percentage for shear
! +pcnt=3
As steel percentage exceeds 3, take upper limit of 3% in shear calcs.
ENDIF
! +f00d=400/d
IF f00d<1                         !Set depth factor for shear
! +f00d=1
As d exceeds 400 mm take value of 400/d as unity in shear calculations.
ENDIF
Now using the formula given in Note 2 below Table 3.8, with f00d=400/d
and pcnt=100*As/(bv*d),
Design shear stress in concrete +vc=0.79*pcnt^(1/3)*f00d^.25/1.25 N/mm2
IF fcu>25
As characteristic concrete strength exceeds +25 N/mm2
therefore increase vc according to footnote in Table 3.8.
IF fcu<40
Modified design shear stress      +vc=vc*(fcu/25)^(1/3) N/mm2
ELSE
Modified design shear stress      +vc=vc*(40/25)^(1/3) N/mm2
ENDIF
ENDIF
IF av<2*d
As av is less than 2*d,
enhanced value of               +vc=vc*2*d/av N/mm2 (see Cl.3.4.5.8).
IF vc>vlim
but this cannot exceed the limiting shear stress vlim, so that
                                  +vc=vlim N/mm2
ENDIF
ELSE
As av is not less than 2*d, there is no enhancement to vc.
ENDIF
IF av<.6*d
newnote
ENDIF
Design shear stress               +v=V*1000/(bv*d) N/mm2
IF v>5
The design shear stress exceeds the absolute limit of +5 N/mm2
specified in Clause 3.4.5.2, and you must employ a larger section.
STOP
ENDIF
IF v>0.8*SQR(fcu)
The design shear stress exceeds the absolute limit of 0.8√fcu specified
in Clause 3.4.5.2. Increase concrete strength or employ larger section.
STOP
ENDIF
IF bv<=350
//
%
IF bv>d+50
%Transverse spacing of link leg is governed by eff.depth d. As this is
%less than breadth bv, more than 2 legs may be needed (see Cl.3.4.5.5).
ELSE                              !i.e. bv<=350 and d>=300
%Transverse leg spacing (see Clause 3.4.5.5)
%__________________________________________
```

```
IF bv<250                          !i.e. bv<250 and d>=300
%As the breadth bv is less than +250 mm two link legs
%should be OK, unless the shearing stress is very high.
ELSE                               !i.e. bv>250 and d>=300
%As breadth bv of +bv mm is not less than +250 mm with an eff.depth
%d of +d mm two link legs should be OK unless shear is very high.
ENDIF
%
ENDIF
ELSE                               !i.e. bv>350
IF bv<d+50                         !i.e. d>400
.As bv exceeds +350 mm note the condition in Clause 3.4.5.5 that
.no longitudinal bar should be more than +150 mm from a vertical leg.
%
%More than two link legs may be necessary.
%
ELSE                               !i.e. bv>350 and bv>d+50
.As bv exceeds +350 mm note the conditions in Clause 3.4.5.5:
.  i) that no longitudinal bar should be more than +150 mm from a
.     vertical leg, and
. ii) (because bv exceeds d), that the transverse spacing of the
.     legs must not exceed the effective depth d (i.e. +d mm ).
%
%More than two link legs may be necessary.
%
ENDIF
ENDIF
! +nlegs=2 +plegs=2 +dialr=dial
REPEAT
! +flag3=1    !Gets set to zero below if transverse spacing too wide
REPEAT
REPEAT
! +flag1=1    !Gets set to zero below if longitudinal spacing too large
! +flag2=1    !Gets set to zero below if longitudinal spacing too small
Number of legs to be provided     +nlegs=????
//
! +maximum=12 +minimum=2 +value=nlegs
maxmin
Total leg area                     +Asv=nlegs*(PI*dialr^2/4) mm2
! +vc04=vc+0.4
IF v<=vc04
As v ( +v N/mm2 ) does not exceed (vc+0.4) ( +vc04 N/mm2 ),
minimum spacing of links          +sv=Asv*fyv/(gammaS*0.4*bv) mm
ELSE
As v ( +v N/mm2 ) exceeds (vc+0.4) ( +vc04 N/mm2 ),
minimum spacing of links          +sv=Asv*fyv/(gammaS*bv*(v-vc)) mm
ENDIF
! +svx=sv
ROUND
                                     = +sv mm (rounded)
! +scr=.75*d +flag4=0
IF As'>0
! +scr1=12*diacs
IF scr>scr1
IF sv>scr1
.Proposed link spacing sv exceeds twelve times diameter of compression
.bars, i.e. 12* +diacs = +scr1 mm (which is less than 0.75*d=
.+scr mm ), and is thus excessive (see Clause 3.12.7.1).
! +cscr=scr1 +flag4=1
ENDIF
ELSE                               !i.e.scr<=scr1
```

```
IF sv>scr
.Proposed link spacing sv exceeds three-quarters of effective depth d,
.i.e. 0.75* +d = +scr mm (which is less than 12*diacs= +scr1 mm ),
.and is thus excessive (see Clause 3.12.7.1).
! +cscr=scr +flag4=2
ENDIF
ENDIF
ELSE                            !Compression steel nominal
IF sv>scr
.Proposed spacing sv exceeds 0.75*d= 0.75* +d = +scr mm
! +cscr=scr +flag4=2
ENDIF
ENDIF
IF flag4>0
%
%You may either:
IF flag4=2
%  1. Reduce spacing to +cscr mm (i.e. 0.75 * effective depth).
ELSE
%  1. Reduce spacing to +cscr mm (i.e. 12 * size of compression bars).
ENDIF
%  2. Redesign by reducing the number of legs and/or the size of link.
%
!Which (1-2)                      +flag1=????
//
! +maximum=2 +minimum=1 +value=flag1
maxmin
IF flag1=1
! +sv=cscr
ROUND
.Reset link spacing to limiting value of +sv mm (rounded).
ELSE
/12
IF nlegs>2
Number of legs and/or link size must be reduced.
ELSE
Link diameter to be reduced.
ENDIF
IF flag1=2
IF dialr>6
IF nlegs=plegs
! +index=TABLE(903,dialr) +index=INT(index)-1 +dummy=TABLE(904,index)
! +dialr=1.0E39 +dialr=dummy
ENDIF
Reduced diameter of link legs     +dialr=???? mm
! +di=dialr
chkbar
IF dialr<diacs/4
Diameter of link must be at least one-quarter of compression bar.
<<<
STOP
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
UNTIL flag1=1
! +plegs=nlegs
ENDREPEAT
IF sv<80
//
%
```

```
%Maximum longitudinal spacing of links is less than +80 mm . You may:
%  1. Adopt this spacing of +sv mm .
%  2. Redesign by increasing the number of legs and/or the size of link.
%
!Which (1-2)                      +flag2=????
//
! +maximum=2 +minimum=1 +value=flag2
maxmin
IF flag2<>1
/12
Number of legs and/or link size must be increased.
IF flag2=2
IF nlegs=plegs
! +index=TABLE(903,dialr) +index=INT(index)+1 +dummy=TABLE(904,index)
! +dialr=1.0E39 +dialr=dummy
ENDIF
Increased diameter of link legs   +dialr=???? mm
! +di=dialr
chkbar

IF dialr<diacs/4
Diameter of link must be at least one-quarter of compression bar.
<<<
STOP
ENDIF
ENDIF
ENDIF
ENDIF
UNTIL flag2=1
! +plegs=nlegs
ENDREPEAT
! +sv'=1.0E39 +sv'=sv
Chosen link spacing               +sv'=???? mm
! +maximum=svx+1E-6 +minimum=50 +value=sv'
maxmin
IF sv'>svx+1E-6
Maximum spacing of links cannot exceed +sv mm .
<<<
STOP
ENDIF
! +sv=sv'
IF nlegs=2
.
.Use +dialr mm links (two legs), spaced at +sv mm centres along beam.
ELSE
.
.Use +dialr mm links ( +nlegs legs), spaced at +sv mm ctrs.along beam.
ENDIF
! +flag99=1  Indicates that transverse spacing clearly inside limits
IF bv<=350
IF bv>d+50
IF nlegs=2
%
%Please check your decision to provide only two legs, as transverse
%spacing may exceed effective depth d (see Clause 3.4.5.5). You may:
%  1. Confirm that the use of only two legs is acceptable.
%  2. Repeat this analysis, increasing the number of link legs.
%
!Which (1-2)                      +flag3=????
//
! +maximum=2 +minimum=1 +value=flag3
maxmin
```

```
! +flag99=0
ENDIF
ENDIF
ELSE !Program below is entered if b>350 mm: note that d can be anything.
IF bv>d+50
IF nlegs=2
IF d<300
%
%Please check your decision to provide only two legs, as transverse
%spacing may exceed effective depth d (see Clause 3.4.5.5). You may:
%  1. Confirm that the use of only two legs is acceptable.
%  2. Repeat this analysis, increasing the number of link legs.
%
!Which (1-2)                       +flag3=????
//
! +maximum=2 +minimum=1 +value=flag3
maxmin
! +flag99=0
ENDIF
ENDIF
ENDIF
IF bv>400                          !Again d can be anything
!              Breadth>400mm so nlegs should exceed 2, so check needed.
IF flag3=1 !If flag3=0, 2 legs spotted above & action already triggered
IF nlegs=2
IF d<300
%
%Please check your decision to provide only two legs, as transverse
%spacing may exceed effective depth d (see Clause 3.4.5.5). You may:
%  1. Confirm that the use of only two legs is acceptable.
%  2. Repeat this analysis, increasing the number of link legs.
%
!Which (1-2)                       +flag3=????
//
! +maximum=2 +minimum=1 +value=flag3
maxmin
! +flag99=0
ELSE                              !i.e. d exceeds 300 mm
%
%Please check your decision to provide only two legs. Tension bars must
%then be kept close to the sides of the beam to meet the requirement in
%Cl.3.4.5.5 that no bar is more than +150 mm from a link leg. You may:
%  1. Confirm that the use of only two legs is acceptable.
%  2. Repeat this analysis, increasing the number of link legs.
%
!Which (1-2)                       +flag3=????
//
! +maximum=2 +minimum=1 +value=flag3
maxmin
! +flag99=0
ENDIF
ELSE
!                Check on spacing if interlocked links – worst case
IF bv/(nlegs-2)>d
%
%Please check your decision to provide +nlegs legs. The transverse
%spacing may exceed the effective depth d if the legs are not evenly
%spaced (see Clause 3.4.5.5). You may:
%  1. Confirm that the use of +nlegs legs is acceptable.
%  2. Repeat this analysis, increasing the number of link legs.
%
!Which (1-2)                       +flag3=????
```

```
//
! +maximum=2 +minimum=1 +value=flag3
maxmin
! +flag99=0
ELSE
IF bv/(nlegs-2)>300
%
%Please check your decision to provide +nlegs legs. The tension bars
%will have to be placed close to the legs in order to ensure that no
%bar is more than +150 mm from a leg (see Clause 3.4.5.5). You may:
%  1. Confirm that the use of +nlegs legs is acceptable.
%  2. Repeat this analysis, increasing the number of link legs.
%
!Which (1-2)                      +flag3=????
//
! +maximum=2 +minimum=1 +value=flag3
maxmin
! +flag99=0
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
IF flag3<>1
The number of legs must be increased (see Clause 3.4.5.5).
ENDIF
UNTIL flag3=1
ENDREPEAT
IF As'>0
IF dialr<dialm

Note: Size of link selected for shear is less than minimum needed to
      contain compression bars (see Clause 3.12.7.1) and must be
      increased over the length of beam requiring compression steel.
ENDIF
ENDIF
IF dialr>dial

Note: As link size has been increased from that assumed when
      undertaking calculations to design main reinforcement, those
      calculations should now be revised.
ENDIF
IF flag99=0 !Reset from 1 to 0 if transverse spacing possibly not OK
%
%Note: As complying with the rules governing the transverse spacing
%      of bars may be difficult, the following warning message is added:
%
.
.When detailing steel, watch carefully the requirements of Cl.3.4.5.5.
%You may add a message of not more than 50 characters for the detailer.
%If you wish to do so, type your message below (a maximum of 50
%characters is allowed).  Otherwise, simply press Enter.
%1       10        20        30        40        50
%|...+....|....+....|....+....|....+....|....+....|
????
ENDIF
!Link length based on assumption 2 leg sets needs 50% more width b than
!one set, 3 sets need 100% more, etc
! +llf=nlegs*h+(1+nlegs/2)*b      !Approx.length of each link set
! +WtL=.006165*dialr^2
! +WtL=WtL*((llf-8*(cover+dialr))+10*dialr)/sv kg/m
```

```
.
.   SHEAR                             Characteristic strength  +fyv N/mm2
.   REINFORCEMENT                     Diameter of links        +dialr mm
.   SUMMARY                           Number of legs           +nlegs
.                                     Spacing                  +sv mm
.                                     Approx.weight of links   +WtL kg/m
.
!Undertake another shear calculation (1=Yes/0=No) +ans6=????
//
! +maximum=1 +minimum=0 +value=ans6
maxmin
UNTIL ans6=0
ENDREPEAT                        !Returns to REPEAT at start of shear calc
ENDIF
eucsbt 3
IF btyp=2
selfch
ENDIF
mesage
STOP
!**********END OF MAIN PROGRAM : SUBROUTINES FOLLOW**********
!
DEFINE chkcover
STORE 3.4 5 7                           ! Table 3.3 BS8110
  25 30 35 40 45 50 100
1  0 25 20 20 20 20  20
2  0  0 35 30 25 20  20
3  0  0  0 40 30 25  25
4  0  0  0 50 40 30  30
5  0  0  0  0 60 50  50
!
STORE 3.98 5 7            ! Key to notes to Table 3.3 BS8110
  25 30 35 40 45 50 100
1  0  0  0  1  1  1   1
2  0  0  0  0  0  0   0
3  0  0  0  0  0  0   0
4  0  0  0  2  2  0   0
5  0  0  0  0  2  0   0
!
STORE 3.5 8 16                               ! Table 3.4 BS8110
     0  1  2  3  4  5  6   7   8   9  10  11  12  13  14  15
0.5 20 20 20 20 20 20 20 200 125  75 150 125 100 150 100  75
1   20 20 20 20 20 20 20 200 125  95 200 160 120 150 120  75
1.5 20 20 25 20 35 20 20 200 125 110 250 200 140 175 140 100
2   40 30 35 25 45 35 25 200 125 125 300 200 160   0 160 100
2.5 60 40 45 35 55 45 25 240 150 150 400 300 200   0 200 150
3   60 40 45 35 55 45 25 240 150 150 400 300 200   0 200 150
3.5 70 50 55 45 65 55 25 280 175 170 450 350 240   0 240 180
4   70 50 55 45 65 55 25 280 175 170 450 350 240   0 240 180
!
STORE 3.99 8 16 ! Key to notes to Table 3.4 BS8110
    0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
0.5 1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0
1   1  1  0  0  0  1  1  0  0  0  0  0  0  0  0  0
1.5 0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
2   0  0  0  0  2  0  0  0  0  0  0  0  0  3  0  0
2.5 2  0  2  0  2  2  0  0  0  0  0  0  0  3  0  0
3   2  0  2  0  2  2  0  0  0  0  0  0  0  3  0  0
3.5 2  2  2  2  2  2  0  0  0  0  0  0  0  3  0  0
4   2  2  2  2  2  2  0  0  0  0  0  0  0  3  0  0
%
```

```
%
%                  Designated condition of exposure
%
%  1 Mild:         Concrete surfaces protected against weather or
%                  aggressive conditions
%  2 Moderate:     Concrete surfaces sheltered from severe rain or
%                  freezing while wet
%                  Concrete subject to condensation
%                  Concrete surfaces continuously under water
%                  Concrete in contact with non-aggressive soil
%  3 Severe:       Concrete surfaces exposed to severe rain,
%                  alternate wetting and drying, or occasional
%                  freezing and severe condensation
%  4 Very severe:  Concrete surfaces exposed to sea water spray,
%                  de-icing salts (directly or indirectly) corrosive
%                  fumes or severe freezing conditions whilst wet
%  5 Extreme:      Concrete surfaces exposed to abrasive action,
%                  e.g. sea water carrying solids or flowing water
%                  with pH<=4.5 or machinery or vehicles
%
!Designated condition of exposure  +expos=????
//
! +maximum=5 +minimum=1 +value=expos
maxmin
! +fcexp=5*INT(fcu/5+1E-3)
IF expos=1
.Designated exposure condition is Mild.
ENDIF
IF expos=2
.Designated exposure condition is Moderate.
ENDIF
IF expos=3
.Designated exposure condition is Severe.
ENDIF
IF expos=4
.Designated exposure condition is Very Severe.
ENDIF
IF expos=5
.Designated exposure condition is Extreme.
ENDIF
! +mod=1
//
%
%According to Cl.3.3.5.2, if a systematic checking regime is established
%to ensure compliance with free-water/cement ratio and cement content
%limits, the minimum cover provided from Table 3.3 of BS8110 can be
%taken as that corresponding to a grade of 5N/mm2 above the true value.
%For example, the minimum grade of concrete to satisfy Moderate Exposure
%(with a minimum of 35 mm cover) can thus be reduced from C35 to C30.
%
%Is such a systematic checking regime to be established ?
%
!Answer (1=Yes/0=No)              +mod=????
//
! +maximum=1 +minimum=0 +value=mod
maxmin
IF mod=1
IF fcexp>=25 THEN fcexp=fcexp+5
It is assumed that a systematic checking regime will be established
to ensure compliance with free-water/cement ratio and cement content
limits (see Clause 3.3.5.2 of BS8110).  Thus modified concrete grade
to be used when reading from Table 3.3 is fc+5= +fcexp N/mm2.
ENDIF
```

```
! +cover1=TABLE(3.4,expos,fcexp) +note1=TABLE(3.98,expos,fcexp)
IF cover1=0
Concrete grade insufficient for degree of exposure required.
Please rerun program and reselect concrete grade or exposure.
STOP
ELSE
IF mod=0
From Table 3.3 of BS8110, minimum cover to all steel required for
fcu= +fcexp N/mm2 and exposure condition +expos is +cover1 mm
ELSE
From Table 3.3 of BS8110, min. cover to all steel for fcu= +fcexp N/mm2
with systematic checking and exposure condition +expos is +cover1 mm
ENDIF
ENDIF

Chosen fire resistance period     +fire=???? hours
! +maximum=4+1E-6 +minimum=.5 +value=fire
maxmin
IF fire<.5 THEN fire=.5
IF fire>4 THEN fire=4
! +fType=cont +fire=0.5*INT(.98+2*fire) +cvr2=TABLE(3.5,fire,fType)
! +note2=TABLE(3.99,fire,fType)
IF fType=0
From Table 3.4 of BS8110, minimum cover needed to all steel for simply—
supported beam with a fire period of +fire hour/s is +cvr2 mm
ELSE                              !fType=1
From Table 3.4 of BS8110, minimum cover needed to all steel for
continuous beam with a fire period of +fire hour/s is +cvr2 mm
ENDIF
! +mcover=cover1
IF cvr2>mcover THEN mcover=cvr2
IF note1=1
IF note2=1
Cover may be reduced to 15 mm provided that nominal maximum
size of aggregate does not exceed 15 mm
ENDIF
ENDIF
IF note1=2
Where concrete is subjected to freezing whilst wet, air—
entrainment should be used (see Clause 3.3.4.2 of BS8110).
ENDIF
IF note2=2
Additional measures are necessary to reduce the risks of spalling
(see Section 4 of BS8110: Part 2).
ENDIF
IF note2=3
A minimum of 0.4% of reinforcement must be provided for fire
resistance periods of 2 hours or more.
ENDIF
ENDDEFINE
!
DEFINE chksize
! +fType=7
! +cover2=TABLE(3.5,fire,fType)
From Figure 3.2 of BS8110, minimum possible beam width complying
with a fire period of +fire hour/s is +cover2 mm
ENDDEFINE
!
DEFINE minlink
! +dialm=6
IF di>24 THEN dialm=8
IF di>32 THEN dialm=10
```

```
IF di>40 THEN dialm=12
IF di>48 THEN dialm=16
ENDDEFINE
!
DEFINE ROUND
! +sv=5*INT(sv/5+1E-6) (round down to nearest 5mm)
IF sv<>75 AND sv<>125 AND sv<>175 AND sv<>225 AND sv<>275 AND sv<>325
IF sv<>375
! +sv=10*INT(sv/10+1E-6) (round down to nearest 10mm)
IF sv<>250
IF sv<>350
IF sv>400
! +sv=50*INT(sv/50+1E-6) (rounded to 50mm)
ELSE
IF sv>200
! +sv=20*INT(sv/20+1E-6) (round to 20mm, except for 250 & 350)
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDDEFINE
!
DEFINE REDUCE
!Reduction is taken as proportional to reduction in tensile steel area.
!This takes no account of any excess due to detailing, or to the fact
!that the lever arm usually increases as steel area reduces.
! +Asrd=nbtr*PI*dia^2/4 mm2
.With +nbtr tension bars, effective steel area Asrd= +Asrd mm2
.Reduced design moment             +Mred=M*nbtr/nbart kNm
IF As'<>0
! +As'rd=As'pr-(Aspr-Asrd)*fy/(gammaS*f's)
! +nbcr=INT(As'rd/(PI*diac^2/4))+1
IF As'rd<0
IF flag0=0                        !Ensures line below obeyed only once
.No compression steel is now required.
! +flag0=1                        !Set flag when line is displayed
ENDIF
ENDIF
IF As'rd>0
.Reduced area of comp.steel reqd.
             +As'rd=(As'pr-(Aspr-Asrd)*fy/(gammaS*f's)) mm2
! +nbcr=INT(As'rd/(PI*diac^2/4)-1E-6)+1
IF nbcr<2
! +nbcr=2
ENDIF
! +As'rd=nbcr*PI*diac^2/4 mm2
.Provide +nbcr compression bars, giving an area of As'rd= +As'rd mm2
ENDIF
ENDIF
! +nbtr=nbtr-2
ENDDEFINE
!
DEFINE COMPF'S
! +es'=0.0035*(x-d')/x +ey=fy/230000
Yield strain (see Figure 2.2)     ey=fy/(gammaS*200000)= +ey
IF es'>=ey
Strain in compression steel     +es'=0.0035*(x-d')/x (not less than ey).
Stress in compression steel     +f's=fy/gammaS N/mm2
ELSE
```

```
Strain in compression steel       +es'=0.0035*(x-d')/x (less than ey).
Stress in compression steel       +f's=es'*200000 N/mm2
ENDIF
ENDDEFINE
!
DEFINE DSSRRECTBEAM
Applied-moment factor             +K=M*1000*1000/(b*d^2*fcu)
As this does not exceed the resistance-moment factor K' ( +K' ),
no compression steel is required.
and lever arm                     +z=d*(0.5+SQR(0.25-K/0.9)) mm
! +scr=.95*d
IF scr<z
! +z=scr
ENDIF
but this must not exceed 0.95*d = 0.95* +d = +scr mm ,
so make                           z= +z mm
Area of tension steel required    +As=M*1000*1000/(z*fy/gammaS) mm2
! +diac=0                         !Set comp.bar size to zero
ENDDEFINE
!
DEFINE notes
//
%
%Explanatory notes on the application of BS 8110: Part 1: 1997
%
%BENDING:  The formulae in Clause 3.4.4.4 are used to determine the
%areas of steel required, these formulae being modified as appropriate
%when the stress in the compression steel is less than fy/gammaS.
%
%When determining the allowable clear distance between bars, the working
%stress fs in the tension steel is determined from equation 8 in Clause
%3.4.6.5 of the Code.
%
%When curtailing reinforcement, the reduction in design moment is taken
%as directly proportional to the reduction in the number of bars.  This
%assumption is conservative as no allowance is made for any detailing
%excess, or for the increase in the lever arm which occurs as steel area
%is reduced.
%
<<//
%
%SHEAR:  To cope with all possible shapes of beam and levels of shear
%the number of legs transversely across the beam may be greater than the
%usual two provided by single rectangular links.  The procedure adopted
%is to input first the number of legs to be provided transversely, and
%then the link diameter, from which the required longitudinal spacing sv
%is calculated.
%
%If this spacing is too large then you are given the option of repeating
%these calculations using a reduced link diameter.  If sv is too small a
%similar option is provided to increase the number of legs or the link
%size.
%
<<//
%
%Where the breadth of the beam exceeds the effective depth, and when it
%exceeds 350 mm, the number of link legs must be sufficient to satisfy
%the provisions of Clause 3.4.5.5.  In such cases you are provided with
%appropriate advice to assist in assessing the number of link legs in a
%transverse direction to be input.  If this advice is ignored and nlegs
%set at two (the usual value), then you are given cautionary notes and
%must either confirm your decision or go back and modify your choice.
%
```

```
%Finally, a message is included for the detailer, and an option given
%for you to append any further comments you may think are required.
%
<<//
ENDDEFINE
!
DEFINE newnote
//
%You have chosen to investigate a section closer to the support than
%0.6 times the effective depth. According to Clause 3.4.5.8 of BS8110
%the design concrete shear stress may be enhanced by multiplying the
%calculated value by the ratio d/av (provided that it never exceeds
%the lesser of either 0.8*SQR(fcu) or 5N/mm2). At locations extremely
%close to the support these requirements give rise to situations where,
%with an applied shearing force that is only very slightly less than
%that necessitating the redesign of the section because the maximum
%permissible shear stress has been exceeded, the theoretical area of
%shear reinforcement required may only be nominal.
%
%In such circumstances you may feel it prudent to either
%  1. Limit the enhancement factor to a maximum of 2
%  2. If links are needed, ignore the concrete contribution to shear
%     resistance altogether (as CP114 did and European Code EC2 does).
%
%You should also remember that in situations where the ratio of av/d
%is less than 0.6, the Code Handbook states that any vertical stirrups
%that are required are not very effective, and in such circumstances
%the provision of horizontal stirrups parallel to the main tension
%steel is recommended.
%
ENDDEFINE
DEFINE selfch
! Only called if btyp=2 i.e. simply supported.
STORE 7.1 6 3  ! 30 year creep coefficient for indoor exposure
       150   300   600  ! section thickness mm
1     4.50  3.70  3.50  ! 1 day age of loading
3     4.10  3.25  2.90  ! 3 day and so on
7     3.60  2.80  2.50
28    3.05  2.30  2.15
90    2.25  1.75  1.65
365.1 1.50  1.15  1.10  ! 365 day age of loading
STORE 7.2 6 3  ! 30 year creep coefficient for outdoor exposure
       150   300   600  ! section thickness mm
1     2.85  2.20  2.00  ! 1 day age of loading
3     2.50  1.95  1.75  ! 3 day and so on
7     2.15  1.65  1.50
28    1.75  1.30  1.25
90    1.30  1.10  0.95
365   0.85  0.63  0.60  ! 365 day age of loading
STORE 7.3 2 3  ! 30 year shrinkage indoor & outdoor x 10^6
       150   300   600  ! section thickness
1      415   365   300  ! indoor
3      125   110    95  ! outdoor

/9
Serviceability limit state
```

```
                                              ├────b────┤
                                        ┬  ┬   ──────────────   ┬   ┬
       ┌A   ┌Gk & Qk kN/m udl.          │  │hf      o o         │   ┼
    ▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼          │h ┴ ───┐       ┌───    │d  d'
    ══════════════════════════          │       │       │       ┴
   _▲_ └A                _▲_            │ SECTION│ o o  │
   ///                   ooo            ┴   AA   └──────┘      If no flange
    ├──────────L───────────┤                     ├─bw─┤       then b=bw.
```

```
pipcal
Span of simply supported beam     +L=span*1000 mm
IF hf=0
Breadth of flange                 +b=bw mm
ELSE
Breadth of flange                 +b=???? mm
ENDIF
Dead load                         +Gk=M*8/(3*span^2) kN/m or N/mm
Imposed load                      +Qk=Gk kN/m or N/mm
Service dead load bending moment  +Mg=Gk*L^2/8 Nmm
Service imposed load bending mmt  +Mq=Qk*L^2/8 Nmm
Total service load bending moment +Mt=Mg+Mq Nmm
Shear force for serviceability    +Vs=V/1.5*1000 N
Permanent load bending moment varies from typically Mg+Mq/2 to Mt.
! +Mp=1E39 +Mp=Mg+Mq/2
Permanent load bending moment     +Mp=???? Nmm
Young's modulus for steel         +Es=???? N/mm2
Young's modulus for concrete      +Ec=???? N/mm2
! +nt=1E39 +nt=nbart +dt=1E39 +dt=dia
Number of tension bars            +nt=????
Diameter of tension bars          +dt=???? mm
Number of compression bars        +nc=nbarc
Diameter of compression bars      +dc=diac mm
Area of tension steel             +As=nt*PI*dt^2/4 mm2
Area of compression steel         +As'=nc*PI*dc^2/4 mm2
Tension steel ratio               +rq=As/(b*d)
Compression steel ratio           +rq'=As'/(b*d)
Permissible steel stress          +fsp=fy/1.15 N/mm2
Permissible concrete stress       +fcp=0.67*fcu/1.5 N/mm2

/5
Short term
──────────
Ratio Es/Ec instantaneous         +ae=Es/Ec
naIc
Short term concrete tension str.  +fcs=1 N/mm2 BS 8110-2:1985 Cl. 3.6.
Moment due to concrete tension    +Mct=bw*(h-x)^3*fcs/(3*(d-x)) Nmm
Net moment for total load         +Mtnet=Mt-Mct Nmm
Net moment for permanent load     +Mpnet=Mp-Mct Nmm
Instantaneous curvature Mtnet     +curmtn=Mtnet/(Ec*Ic)
Instantaneous curvature Mpnet     +curmpn=Mpnet/(Ec*Ic)
Diff in instantaneous curvature   +curdit=curmtn-curmpn mm^-1
Diff in instantaneous curvature   +curdit=(Mt-Mp)/(Ec*Ic) mm^-1
Strain at tension steel level     +es=ae*Mtnet*(d-x)/(Es*Ic)
Stress at tension steel           +fs=Es*es N/mm2
IF nc>0
Strain at compression steel level +esc=ae*Mtnet*(x-d')/(Es*Ic)
Stress in compression steel       +fsc=Es*esc N/mm2
ENDIF
Concrete strain at top of section +ect=Mtnet*x/(Ec*Ic)
Concrete stress at top of section +fct=Ec*ect N/mm2
Concrete strain at tension face   +eh=Mtnet*(h-x)/(Ec*Ic)

/5
```

Shear reinforcement

```
Force in tension steel             +fts=fs*As N
Lever arm                          +a=Mtnet/fts mm
Number of legs                     +nlegs=????
Area of link legs                  +A=nlegs*PI*dial^2/4 mm2
Spacing of link legs               +sv'=???? mm
Permissible stress in links        +fvp=fyv/1.15 N/mm2
Shear resist. provided by links    +V'=A*fvp*a/sv' N
IF Vs>V'
+$924=*** Serviceability shear > link capacity, +Vs/1E3 > +V'/1E3 kN
%+$924
ENDIF
```

/5

Long term

```
Exposure 1=indoor, 2=outdoor       +expo=????
Age of loading (1 to 365 days)     +aol=???? days
! +tab=7+expo/10 +phi=TABLE(tab,aol,bw)
Thirty year creep coefficient      phi= +phi re. BS 8110-2:1985 Fig 7.1.
! +ecs=TABLE(7.3,expo,bw)
Concrete shrinkage strain          +ecs= +ecs/1E6
Long term modulus of concrete      +Elt=Ec/(1+phi) kN/mm2
Ratio Es/Elt long term             +ae=Es/Elt
naIc
Long term concrete tension stress +fct=0.55 N/mm2 BS 8110-2:1985 C.3.6.
Moment due to concrete tension     +Mct=bw*(h-x)^3*fct/(3*(d-x)) Nmm
First moment of steel              +Ss=As*(d-x) mm3
Curvature (1/rcs)                  +cursh=ecs*ae*Ss/Ic mm^-1
Net moment for permanent load      +Mpnet=Mp-Mct Nmm
Curvature total-permanent load     +curlt=Mpnet/(Elt*Ic) mm^-1
Total curvature short+long term    +curtot=curlt+curdit
From curvature-area theorems
Long term deflection +dlt=1/9.6*L^2*curtot+0.125*L^2*cursh mm
Maximum permissible deflection     +mpd=L/250 mm Clause 3.4.6.3.
IF dlt>mpd
+$924=*** Deflection exceeds permissible, +dlt > +mpd mm
%+$924
ELSE
Deflection within permissible ( +dlt <= +mpd )
ENDIF
IF nc>0
Strain at compression steel        +esc=ae*Mt*(x-d')/(Es*Ic)
Stress in compression steel        +fsc=Es*esc N/mm2
IF fsc>fsp
+$924=*** Stress in compression steel > permissible, +fsc > +fsp N/mm2
%+$924
ELSE
Stress in compression steel within permissible ( +fsc <=0 +fsp )
ENDIF
ENDIF
Strain at tension steel            +es=ae*Mt*(d-x)/(Es*Ic)
Stress in tension steel            +fs=Es*es N/mm2
IF fs>fsp
+$924=*** Stress in tension steel > permissible, +fs > +fsp N/mm2
%+$924
ELSE
Stress in tension steel within permissible ( +fs <= +fsp )
ENDIF
Concrete strain at top of sectn    +ec=Mt*x/(Ec*Ic)
Concrete stress at top of sectn.   +fc=ec*Ec N/mm2
```

```
IF fc>fcp
+$924=*** Stress in concrete > permissible, +fc > +fcp N/mm2
%+$924
ELSE
Stress in concrete within permissible ( +fc <= +fcp )
ENDIF
Tension face: midway between two bars
Concrete strain at tension face   +eh=es*(h-x)/(d-x)
Minimum cover soffit to bar       +cmin=h-d-dt/2 mm
IF nt>1
Gap between tension bars          +gap=(bw-2*cmin-nt*dt)/(nt-1) mm
Distance from point considered to surface of nearest longitudinal bar
                        +acr=SQR((gap+dt)^2/4+(cmin+dt/2)^2)-dt/2 mm
ELSE
Distance from point considered to surface of nearest longitudinal bar
                        +acr=SQR((bw/2)^2+(cmin+dt/2)^2)-dt/2 mm
ENDIF
Average strain at level considered
+em=eh-bw*(h-x)*(h-x)/(3*Es*As*(d-x))
Design surface crack width +cw=3*acr*em/(1+2*((acr-cmin)/(h-x))) mm
IF cw>0.3
+$924=*** Crack width exceeds permissible, +cw > 0.3 mm
%+$924
ELSE
Crack width within permissible ( +cw <= 0.3 )
ENDIF
FILE
ENDDEFINE
DEFINE naIc ! Neutral axis depth & second moment of area of concrete.
! +x=(-ae*(rq+rq')+SQR(ae^2*(rq+rq')^2+2*ae*(rq+rq'*d'/d)))*d +bw'=bw
IF hf>0 AND x<hf THEN bw'=b ENDIF
! Constants +c1=(ae-1)*rq' +c2=ae*rq +c3=c2+c1 +c4=c2+c1*d'/d
! +c5=1-bw'/b +c6=hf/d +c7=c3+c5*c6 +c8=2*(c4+0.5*c5*c6^2)*bw'/b
! +x=d*(SQR(c7^2+c8)-c7)*b/bw' mm  Reference RCDH Tenth Edition.
Depth to neutral axis             x= +x mm
! +c4=x/d +c6=b*d^3 +c7=c1*(c4-d'/d)^2 +c8=(c4-hf/d)^3
! +Ic=c6*((c4^3-c5*c8)/3+c2*(1-c4)^2+c7) mm4
Second moment of area             Ic= +Ic mm4
ENDDEFINE
DEFINE euccht ! Column design according to BCA.
!! remove  N=1000  Mx=1000  fcu=40  cover=30  h=800  b=800
!! remove  nbars=6  dia=25  As=0
! +cyl1=VEC(16,20,25,28,30,32,35,40,45,50)
! +cub1=VEC(20,25,30,35,37,40,45,50,55,60) +i=0
REPEAT
! +i=i+1 +fck=0
IF fcu=cub(i) THEN fck=cyl(i) ENDIF
UNTIL i=10 OR fcu=cub(i)
ENDREPEAT
IF fck=0
STOP Cube strength not recognised.
ENDIF
! +KLSB=0 +gamc=1.5 +gams=1.15 +alpcc=0.85
! +fyk=500 Chart factor multiplier +cfm=16
! Sets of bar diameters & numbers of bars with increasing areas.
! +brd1=VEC(20,20,25,20,25,20,32,20,25,32,25,40,25,32,40,32,32,40,40,40)
! +bno1=VEC(4,6,4,8,6,10,4,12,8,6,10,4,12,8,6,10,12,8,10,12) +bs=0
[ Factored axial load on column     +NEd=N kN
[ Factored bending moment on column +M=Mx kNm
REPEAT
! +bs=bs+1 +bardia=brd(bs) +barpf=bno(bs)/2 +lnkdia=6
```

```
IF bardia>24 THEN lnkdia=8 ENDIF
IF bardia>32 THEN lnkdia=10 ENDIF
[ For bardia= +bardia mm then link diameter lnkdia= +lnkdia mm
[ Steel percentage area +ascper=PI/4*bardia^2*barpf*2*100/h/b %
IF ascper<0.2
[ For steel percentage area <0.2% then +ascper=0.2 %
ENDIF
ec2cld
! Nbars= +2*barpf bardia= +bardia As= +b*h*ascper/100
! +i=0
REPEAT
! +i=i+1
[ Factored axial load resistance     +NEdr=rtn(i)*b*h*fck/1E3 kN
IF NEdr<NEd GOTO 100
! Now adequate for axial load, reject bar arrangement if
! inadequate for moment.
[ Factored bending moment resistance +Mr=rtm(i)*b*h^2*fck/1E6 kNm
IF Mr<M GOTO 110
IF i>1
[ Previous values
[ Factored axial load resistance     +NEdr1=rtn(i-1)*b*h*fck/1E3 kN
[ Factored bending moment resistance +Mr1=rtm(i-1)*b*h^2*fck/1E6 kNm
ENDIF
! +Ascp=PI*bardia^2/4*2*barpf
.Bar dia.= +bardia  No.bars= +2*barpf   Area= +Ascp    %age= +ascper
GOTO 120 ! Try next bar arrangement.
:100
UNTIL i=51*cfm+1 ! Originally 52
ENDREPEAT
:110
UNTIL bs=20
ENDREPEAT
:120
! +a=nbars +c=dia +f=2*barpf +g=bardia +ap=INT(Ascpro) +aq=INT(Ascp)
! +r=run +dA=ap +dB=aq
vmper ! Compute percentage difference 'per'.
/4 ! Following without the run number goes to .cal file.
.    Col dep  Factored    BS8110   EC2 No.  BS8110 EC2   Perc.
.    & bread  axial BM    No.bars  of bars  area   area  diff.
.    mm  mm   kN    kNm   & dia    & dia    mm2    mm2
.    +h  +b   +N    +M    +a  +c   +f  +g   +ap   +aq    +$777
IF run=1 ! Used for multiple run reporting, piped to cc924.res.
%Run Col dep  Factored    BS8110   EC2 No.  BS8110 EC2   Perc.
%No. & bread  axial BM    No.bars  of bars  area   area  diff.
%    mm  mm   kN    kNm   & dia    & dia    mm2    mm2
ENDIF
IF run>0
%+r  +h  +b   +N    +M    +a  +c   +f  +g   +ap   +aq    +$777
ENDIF
ENDDEFINE
DEFINE euccld ! Column design according to BCA.
! For a chart calcualte x at 24 intermediate points between 'N=0'
! and 'N bal' and a further 27 points between 'N bal' and x=2h
! and a final point for 'N uz'.  See spreadsheets to EC2 Reinforced
! Concrete Council.  The curve generated is for a given fraction
! As*fyk/(b*h*fck), each of the points corresponding to y axis
! value N/(b*h*fck) and x axis value of M/(b*h^2*fck).
[ Factor +lam=0.8-(fck-50)/400
IF lam>0.8
[ As lam>0.8 then +lam=0.8
ENDIF
[ +eta=1-(fck-50)/200
```

```
IF eta>1
[ As eta>1 then +eta=1
ENDIF
[ Factor +etafcd=alpcc* fck/gamc* eta
[ Design strength of steel +fyd=fyk/gams N/mm2
[ Net design strength +netfyd=fyd-etafcd N/mm2
[ Strain +epscu3=(2.6+0.9* ((90-fck)/40)^4)/1000
IF epscu3>3.5/1000
[ Since epscu3>3.5/1000 then +epscu3=3.5/1000
ENDIF
[ Strain +epsc3=1.75/1000
IF fck>50
[ Since fck>50 then +epsc3=1.75+0.55* ((fck-50)/40)/1000
ENDIF
[ Factor +epsh=(1-epsc3/epscu3)* h mm
[ Depth to reinforcement +d2=cover+lnkdia+bardia/2 mm
[ Area of reinforcement +asc=h* b* ascper/100 mm2
[ Effective depth +d=h-d2 mm
IF asc=0 THEN d=0 ENDIF
[ Ratio +ratd2h=d2/h
[ Ratio +ratash=asc* fyk/b/h/fck
[ First solve the quadratic to calculate x when N=0.
[ +quada=lam* etafcd* b
[ +quadb=asc/2* (200000* epscu3-etafcd-fyd)
[ +quadc=-100000* epscu3* d2* asc
[ Value x for zero N +xzN=(-quadb+SQR(quadb^2-4* quada* quadc))/2/quada
[ Value x at "N bal" +xNb=200000* epscu3/fyd* d/(1+200000* epscu3/fyd)
[ Value x at "N uz" +xNu=2* h
! +xinc=(xNb-xzN)/(24* cfm) +i=-1 +xinc'=(xNu-xNb)/(27* cfm) ! Orig 27
REPEAT
! +i=i+1 +x=xzN+i* xinc
IF i>24* cfm THEN x=xNb+(i-24* cfm)* xinc' ENDIF ! Originally 24
! +fc=h* b* etafcd
IF lam* x<h THEN fc=lam* x* b* etafcd ENDIF
!i= +i x= +x fc= +fc N
IF x>h
! +epscmx=epsc3
IF epsc3* x/(x-epsh)>epsc3 THEN epscmx=epsc3* x/(x-epsh) ENDIF
ELSE
! +epscmx=epscu3
ENDIF
![ Limiting strain i= +i epscmx= +epscmx
[ Steel compressive strain +epssc=epscmx* (x-d2)/x
[ Steel tensile strain +epsst=epscmx* (d-x)/x
IF fyd<=200000* epssc
[ Steel compressive stress +fsc=fyd-etafcd
ELSE
[ Steel compressive stress +fsc=200000* epssc-etafcd
ENDIF
! +temp1=0
IF d<lam* x THEN temp1=etafcd ENDIF
IF epsst>0
! +fst=fyd
IF 200000* epsst<fyd THEN fst=200000* epsst ENDIF
ELSE
! +fst=200000* epsst+temp1
IF fst<-netfyd THEN fst=-netfyd ENDIF
ENDIF
[ Steel tensile stress fst= +fst N/mm2
[ Steel compressive force +forsc=fsc* asc/2 N
[ Steel tensile force +forst=fst* asc/2 N
! +z=d-lam/2* x
```

```
IF d-h/2>z THEN z=d-h/2 ENDIF
[ Concrete lever arm z= +z mm
[ Fc+Fsc-Fst +n=(fc+forsc-forst)/1000
[ +m=(fc*z+forsc*(d-d2))/1E6-n*(d-h/2)/1000
[ N/(b*h*fck) +rtn(i)=n*1E3/(b*h*fck)
[ M/(b*h^2*fck) +rtm(i)=m*1E6/(b*h^2*fck)
UNTIL i>=(24+27)*cfm ! Originally 24+27
ENDREPEAT ! Special 'N uz' case, originally 52 & 51.
! +last=cfm*(24+27)+1
[ N/(b*h*fck) +rtn(last)=rtn(last-1)
[ M/(b*h^2*fck) +rtm(last)=0
ENDDEFINE
DEFINE vmper ! SCALE version of percentage difference.
! +zz=0 ! Compute %age diff. between dA & dB & write message.
IF ABS(dA)<1E-8 AND ABS(dB)<1E-8 OR dA=dB THEN zz=1 dA=0 dB=0 ENDIF
IF ABS(dA-dB)<1E-4 THEN zz=1 dA=0 dB=0 ENDIF
IF dA=0 AND dB<>0 THEN zz=2 ENDIF
IF dB=0 AND dA<>0 THEN zz=2 ENDIF
IF zz=0 THEN zz=dA/dB ENDIF
IF zz<0 OR zz>2 THEN zz=2 ENDIF
IF zz<1 THEN zz=1/zz ENDIF
IF zz>2 THEN zz=2 ENDIF
! +per=INT(ABS(100-100*zz)+.5) +dA=INT(per/10) +dB=per-dA*10
IF dA=0 THEN dA=-1 ENDIF
IF per<100 THEN +$777= +per
IF per>=100 THEN +$777=>99
ENDDEFINE
DEFINE pipcal ! Pipe calculations.  SCALE32 looks for cc924.stk,
! if found then extracts the run number and puts it on its stack.
! +euc=-1.0E39 Add to stack if not already on it.
IF euc=1.0E39 THEN euc=0 ENDIF
!Provide check to Eurocode (1=Yes,0=No) +euc=????
IF run>0 AND euc=1
! Clear string for holding leading zeros in name extension +$924=
IF run<100 THEN +$924=0
IF run<10 THEN +$924=00
! +$32000=cc924. +$924 +run
DOS del +$32000 ! Clear old file in range cc924.001 to cc924.996.
FILE +$32000 ! Set cc924.<run> as file for piped calculations.
ENDIF
ENDDEFINE
DEFINE eucsbt ! Eurocode 2.  euc1: 1=slab, 2=beam, 3=T-beam.
pipcal
! +d'=-1.0E39 +b=-1.0E39 +beff=b +fyk=fy +Mbef=-1.0E39
! +KLSB=0 for output, 0=none.  Add to stack if not on it
[ Eurocode 2.
IF d'<>1.0E39 THEN d2=d' ENDIF
IF b=1.0E39 THEN b=1000 ENDIF
IF Mbef=1.0E39 THEN Mbef=M ENDIF
[ Coeff for flexure & axial loads   +acc=0.85
[ Cylinder strength                 +fck=fcu*28/35 N/mm2
[ Factor lambda +lam=0.8-(fck-50)/400
IF lam>0.8
[ As lam>0.8 then +lam=0.8
ENDIF
[ Partial safety factor for conc.   +gamc=1.5
[ Design concrete compr. strength   +fcd=acc*fck/gamc N/mm2
[ Characteristic yield strength     +fyk= +fy N/mm2
[ Partial safety factor for steel   +gams=1.15
[ Design steel strength             +fyd=fyk/gams N/mm2
```

```
IF fck<25 OR fck>50
.Eurocode 2 fck out of range 25 to 50 N/mm2.
ENDIF
[ Redistribution ratio              +delta=M/Mbef
IF delta>1
[ Limit redistribution ratio        +delta=1
ENDIF
[ Factor                            +K=M*1E6/(b*d^2*fck)
[ Factor                            +K'=0.6*delta-0.18*delta^2-0.21
IF euc1=1 AND K>K'
.Eurocode 2  Compession steel not recommended by The Concrete Centre.
STOP
ENDIF
IF K>K'
[ Lever arm                         +z=d/2*(1+SQR(1-3.53*K')) mm
ELSE
[ No compression reinforcement required.
[ Lever arm                         +z=d/2*(1+SQR(1-3.53*K)) mm
ENDIF
IF euc1=1 AND z>0.95*d OR euc1=3 AND z>0.95*d
[ Reduce lever arm to               +z=0.95*d mm
ENDIF
[ Initialise area of comprn steel   +As2=0 mm2
[ Depth to neutal axis              +x=2.5*(d-z) mm
IF euc1=3
IF x<hf !1.25*hf
[ Neutral axis in flange.  Design as rectangular section and then
[ check longitudinal shear.  Fix for rectangular section +euc1=2
ELSE
[ Neutral axis in web.  Calculate moment capacity of flange from
[ +MRf=0.57*fck*(beff-bw)*hf*(d-hf/2) Nmm
[ +Kf=(M*1E6-MRf)/(fck*bw*d^2)
IF Kf>K'
.Eurocode 2 Compression steel awaiting worked example for T beam.
STOP
ELSE
[ Steel area +Asec=MRf/(fywd*(d-0.5*hf))+(M-MRf)/fywd mm2
eccls ! Check longitudinal shear.
ENDIF
ENDIF
ENDIF
IF euc1<3 ! Exclude compression steel from T beam calculations.
IF K>K'
[ Stress                            +fsc=700*(x-d2)/x N/mm2
IF fsc>fyd
[ Limit fsc to fyd                  +fsc=fyd N/mm2
ENDIF
[ Area of compression steel  +As2=(K-K')*fck*b*d^2/(fsc*(d-d2)) mm2
[ Tension reinforcement      +Asec=K'*fck*b*d^2/(fyd*z)+As2*fsc/fyd mm2
ENDIF
IF K<=K'
[ Tension reinforcement required    +Asec=M*1E6/(fyd*z) mm2
ENDIF
ENDIF ! Closes IF euc1<3.
[ Mean width of the tension zone    +bt=b mm
[ Mean value axial tensile strength +fctm=0.3*fck^(2/3) N/mm2
[ Minimum reinforcement required    +Asmin=0.26*fctm*bt*d/fyk mm2
IF Asec<Asmin
[ Area of tension reinforcement             +Asec=Asmin mm2
ENDIF
[ Maximum reinforcement permitted           +Asmax=0.04*b*d mm2
```

```
IF Asec+As2>Asmax
.Eurocode 2  Area of steel exceeds 4% of bd.
ENDIF
IF euc1=1 ! i.e. slabs e.g. sc080.
!Eurocode 2  Area of tension steel required +Asec mm2/m
ELSE ! i.e. beams e.g. sc072
!Eurocode 2  Area of tension steel required +Asec mm2
ENDIF
IF As2>0
!Eurocode 2  Area of compression steel required +As2 mm2
ENDIF
! +As=-1.0E39 +As'=-1.0E39
IF As=1.0E39 THEN As=0
IF As'=1.0E39 THEN As'=0
! +db=INT(d) +dA=As'+As +dB=As2+Asec +r=run
vmper ! Compute percentage difference 'per'.
/4 ! Following without the run number goes to .cal file.
.    Beam dpth Design  BS 8110  EC2     BS 8110  EC2 mm² Perc.
.    & breadth moment  comprn.  comprn. tension  tension diff.
.    mm   mm   kNm     mm²      mm²     area mm² area     %
.    +db  +b   +M      +As'     +As2    +As      +Asec    +$777
IF run=1 ! Used for multiple run reporting, piped to cc924.res.
%Run Beam dpth Design  BS 8110  EC2     BS 8110  EC2 mm² Perc.
%No. & breadth moment  comprn.  comprn. tension  tension diff.
%    mm   mm   kNm     mm²      mm²     area mm² area     %
ENDIF
IF run>0
%+r  +db  +b   +M      +As'     +As2    +As      +Asec    +$777
ENDIF
ENDDEFINE
DEFINE eccls ! T beam, check longitudinal shear.
pipcal
![ Eurocode 2 Longitudinal shear stress check between web & flange.
! +beff=b +fyk=fy +KLSB=0                   +ctheta=-1.0E39
IF ctheta=1.0E39 THEN ctheta=1 ENDIF ! Add to stack if not on it
[ Design steel strength              +fyd=fyk/gams N/mm2
[ For deltax=beff                    +deltax=beff mm
[ For Θf=1.25, see notes to (6.22)  +thetaf=1.25
[ For moment after redistribution   +M=Mbef*1E6 Nmm
[ Shear force dm/dx                  +v=M/deltax N  Worked ex. reqd.
ENDDEFINE
DEFINE eucvsh ! Vertical shear, Eurocode 2 Clause 6.2.3.
pipcal
! +bv=-1.0E39
IF bv=1.0E39 THEN bv=b ENDIF
! +bw=bv +VEd=V*1000 +s=sv +fyk=fyv +KLSB=0 +ctheta=-1.0E39
IF ctheta=1.0E39 THEN ctheta=1 ENDIF ! Add to stack if not on it
[ Eurocode 2.
[ Cot(Θ) 1-2.5 (See 6.7N EN 1992-1) +ctheta=???? (1 if tension).
[ Design shear force at section     +VEd= +V*1000 N
[ Characteristic yield strength     +fyk= +fyv N/mm2
[ Partial safety factor for steel   +gams=1.15
[ Design steel strength             +fywd=fyk/gams N/mm2
[ Depth tension to comress. steel   +z=0.9*d mm
[ X-sectional area of shear steel   +Asw=nlegs*PI*dial^2/4 mm2
[ Shear resistance                  +VRds=Asw/s*z*fywd*ctheta N
[ Cylinder strength                 +fck=fcu*28/35 N/mm2
[ Stress red. factor cracked conc.  +v1=0.6*(1-fck/250) N/mm2
[ Coefficient for comprn. chord     +acw=1 for no prestress.
[ Coeff. for other phenomena        +acc=1.0
[ Partial safety factor for conc.   +gamc=1.5
[ Design concrete compr. strength   +fcd=acc*fck/gamc N/mm2
```

```
[ Shear resistance         +VRdm=acw*bw*z*v1*fcd/(ctheta+1/ctheta) N
IF VRdm<VRds
[ VRdm<Vrds therefore smaller value  +VRd=VRdm N
ENDIF
IF VRds<VRdm
[ VRds<Vrdm therefore smaller value  +VRd=VRds N
ENDIF
!Eurocode 2 Shear resist. prov. +VRd/1E3 kN cf. design shear +V kN
! +V'=0.95*fyv*Asw*d/sv/1000+vc*bv*d/1000 +dA=V' +dB=VRd/1E3 +r=run
vmper ! Compute percentage difference 'per'.
/4 ! Following without the run number goes to .cal file.
.Run Beam depth Area of Link  Design  BS 8110  EC2 shear Perc.
.No. & breadth  links   crs.  shear   shear    resist.   diff.
.    mm   mm    mm²     mm    kN      resist.  prov. kN  %
.    +d   +bv   +Asw    +s    +V      +V'      +VRd/1E3  +$777
IF run=1 ! Used for multiple run reporting, piped to cc924.res.
%Run Beam depth Area of Link  Design  BS 8110  EC2 shear Perc.
%No. & breadth  links   crs.  shear   shear    resist.   diff.
%    mm   mm    mm²     mm    kN      resist.  prov. kN  %
ENDIF
IF run>0
%+r  +d   +bv   +Asw    +s    +V      +V'      +VRd/1E3  +$777
ENDIF
ENDDEFINE
@sc1071.pro                      !Load standard subroutines
FINISH
```

PROFORMA SC385.PRO

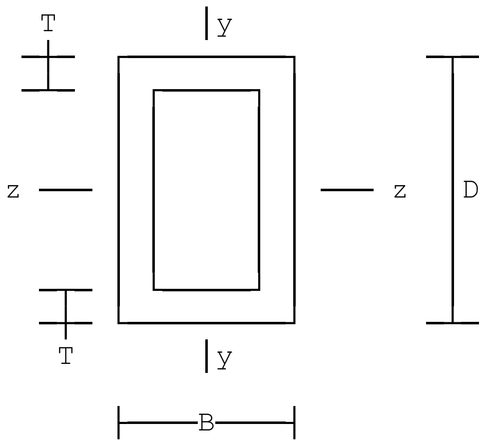
SCALE
Proforma No. 385
Title: Stainless Steel Hollow Section Design
Devised by Jim Dunbar May '03
Based on BS5950 and P291 'Structural Design of Stainless Steel'
Checked by B. Wadsworth June 2003
Amendments J.D. Aug'04 - cosmetic changes
J.D. Jan'05 - correction to logic statement
J.D. Feb'05 - unnecessary statement removed, reordering statements
D.Brown Dec'05 - spelling corrections, note "taken as the positive side" added to two figures. When prompted for My, figure now shows axes. Verification table added, tests show 'rtype' is not set.
D.Brown Jan'06 - parameter table and self check added.
J.Dunbar Apr'06 - reference to 'rtype' removed should be 'stype' NRESP=0 added after initial inputs.

PARAMETER No.	name	Start zst()	End zen()	Type zty()	Dependency conditions and notes.
1	L	0.3	6	0	Length of member in m.
2	stype	1	2	2	stype=1 is SHS, stype=2 is RHS
3	ssd1	11	11	0	Constant for SHS table number
4	sd11	40	400	1	>L*1000/24 <L*1000/8
5	sb11	40	400	1	=sd11 for SHS
6	st11	2	15	1	>sd11/66.667+1 <sd11/12+1
7	tri	3	58	1E40	
8	srd1	12	12	0	Constant for RHS table number
9	sd12	40	400	1	>L*1000/24 <L*1000/8
10	sb12	20	200	1	=sd12/2 say for RHS
11	st12	2	15	1	>sd12/66.667+1 <sd12/12+1
12	tri	3	72	1E40	Calls procedure tri
13	grade	1	5	5	
14	py	1	5	126	210 220 220 400 460
15	sd	40	400	0	=sd11*(2-stype)+sd12*(stype-1)
16	sb	20	400	0	=sb11*(2-stype)+sb12*(stype-1)
17	st	1.5	15	0	=st11*(2-stype)+st12*(stype-1)
18	zrn	4	0.667	1E40	Proc. creates zrn(1:4) Σ=0.667
19	Mz	0.1	1000	0	=sb*st*py*sd/1E6*zrn1 kNm
20	My	0.05	1000	0	=sd*st*py*sb/1E6*zrn2 kNm
21	Fv	1	1000	0	=4*Mz/L*zrn3 kN
22	F	1	1000	0	=(sd+sb)*2*st*py/1E3*zrn4 kN
23	Lz	300	6000	1000	>L*1000/6 <L*1000 mm
24	Ly	300	6000	1000	>L*1000/6 <L*1000 mm
25	Kz	0.7	1	3	
26	Ky	0.7	1	3	
27	Ae	5	221	0	<(sd+sb)*2*st/100 cm2
28	moment	1	0	2	
29	restz	1	0	2	
30	betaMz	0.1	1000	0	<Mz kNm
31	mz2	0.1	1000	0	<Mz*zrn1 Must write zrn1 etc.
32	mz3	0.1	1000	0	<Mz*zrn2 here and not write
33	mz4	0.1	1000	0	<Mz*zrn3 zrn(1) etc.
34	M24	0.1	1000	0	<Mz*zrn4
35	resty	1	0	2	
36	betaMy	0.05	1000	0	<My
37	my2	0.05	1000	0	<My*zrn1

```
38  my3    0.05    1000    0       <My* zrn2
39  my4    0.05    1000    0       <My* zrn3
40  My24   0.05    1000    0       <My* zrn4
41  LT     1       6       0       <L
42  refno  1       10      1
43  udl    1       2       2
44  betaM  0.01    1000    0       <Mz* zrn1
45  m2     0.01    1000    0       <Mz* zrn2
46  m3     0.01    1000    0       <Mz* zrn3
47  m4     0.01    1000    0       <Mz* zrn4
48  E      200     200     0       Young's modulus N/mm2
49  NRESP  0       0       0       Avoids importing from NL-STRESS
50  ans    0       0       0       ans=0 refuses default values

@scale.sta
!Proforma sc4100.pro imported.  Devised by J. Dunbar April 2004.
!Tables 18 and 26 BS 5950-1 Table 15 added 30/06/04
STORE 18 21 1
1.0    1.00
0.9    0.96
0.8    0.92
0.7    0.88
0.6    0.84
0.5    0.80
0.4    0.76
0.3    0.72
0.2    0.68
0.1    0.64
0      0.6
-0.1   0.56
-0.2   0.52
-0.3   0.48
-0.4   0.46
-0.5   0.44
-0.6   0.44
-0.7   0.44
-0.8   0.44
-0.9   0.44
-1.0   0.44

STORE 26 21 1
1.0    1.00
0.9    0.96
0.8    0.92
0.7    0.88
0.6    0.84
0.5    0.80
0.4    0.76
0.3    0.72
0.2    0.68
0.1    0.64
0      0.60
-0.1   0.58
-0.2   0.56
-0.3   0.54
-0.4   0.52
-0.5   0.50
-0.6   0.48
-0.7   0.46
-0.8   0.44
-0.9   0.42
-1.0   0.40
```

```
STORE 39 21 11           ! Appendix G.1 Table 39
       0   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  1.0
-1.0 1.00 0.76 0.61 0.51 0.44 0.39 0.35 0.31 0.28 0.26 0.24
-0.9 1.00 0.78 0.63 0.52 0.45 0.40 0.36 0.32 0.30 0.28 0.26
-0.8 1.00 0.80 0.64 0.53 0.46 0.41 0.37 0.34 0.32 0.30 0.28
-0.7 1.00 0.81 0.66 0.55 0.47 0.42 0.39 0.36 0.34 0.32 0.30
-0.6 1.00 0.83 0.67 0.56 0.49 0.44 0.40 0.38 0.36 0.34 0.33
-0.5 1.00 0.85 0.69 0.58 0.50 0.46 0.42 0.40 0.38 0.37 0.36
-0.4 1.00 0.86 0.70 0.59 0.52 0.48 0.45 0.43 0.41 0.40 0.39
-0.3 1.00 0.88 0.72 0.61 0.54 0.50 0.47 0.45 0.44 0.43 0.42
-0.2 1.00 0.89 0.74 0.63 0.57 0.53 0.50 0.48 0.47 0.46 0.45
-0.1 1.00 0.90 0.76 0.65 0.59 0.55 0.53 0.51 0.50 0.49 0.49
0    1.00 0.92 0.78 0.68 0.62 0.58 0.56 0.55 0.54 0.53 0.52
0.1  1.00 0.93 0.80 0.70 0.65 0.62 0.59 0.58 0.57 0.57 0.56
0.2  1.00 0.94 0.82 0.73 0.68 0.65 0.63 0.62 0.61 0.61 0.60
0.3  1.00 0.95 0.84 0.76 0.71 0.69 0.67 0.66 0.65 0.65 0.65
0.4  1.00 0.96 0.86 0.79 0.75 0.72 0.71 0.70 0.70 0.69 0.69
0.5  1.00 0.97 0.88 0.82 0.78 0.76 0.75 0.75 0.74 0.74 0.74
0.6  1.00 0.98 0.91 0.85 0.82 0.81 0.80 0.79 0.79 0.79 0.79
0.7  1.00 0.98 0.93 0.89 0.87 0.85 0.85 0.84 0.84 0.84 0.84
0.8  1.00 0.99 0.95 0.92 0.91 0.90 0.90 0.89 0.89 0.89 0.89
0.9  1.00 1.00 0.98 0.96 0.95 0.95 0.95 0.95 0.95 0.94 0.94
1.0  1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00

STORE 150 12 1 !Limiting value of Le/ry for RHS sections Table 15
1.25 770
1.33 670
1.40 580
1.44 550
1.50 515
1.67 435
1.75 410
1.80 395
2.00 340
2.50 275
3.00 225
4.00 170

+ZZZZZ1=0   Only read once as before start
+ans=1 +selfc=0 +ZZZZZ2=0  +E=200 +UNITS=0
Added Apr 06 +LT=0 +refno=0 +Ly=0
START
IF NRESP>0
! +ZZZZZ2=1 +ZZZZZ1=1
ENDIF
IF ZZZZZ1=0
%
%
%<H1>Stainless steel hollow section design
%Would you like a set of defaults to be provided which has been taken
%from Example 6 in 'Structural Design of Stainless Steel' ?
%You can use the default values as references and type your
%own values beneath to replace them.
!Answer ( 1 = Yes, 0 = No )       +ans=????
IF ans=1
! +$92=Example 6 'Structural Design of Stainless Steel'
! +Mz=2.74 +Fv=1 +F=19.6 +L=2.7 +My=0 +stype=2 +sd12=100
! +sb12=50 +st12=6 +grade=2 +E=200 +Lz=2700 +Ly=2700 +Kz=1 +Ky=1
! +moment=1 +udl=1 +betaM=0 +restz=0 +betaMz=0 +LT=2.7 +refno=5
ENDIF
```

```
IF ans=2
! +$92=Slender section bending about major axis only
! +Mz=77.4 +Fv=18 +F=1100 +L=4 +My=0 +stype=2 +sd12=400
! +sb12=200 +st12=6 +Lz=4000 +Ly=4000 +Kz=1 +Ky=1 +grade=5 +E=200
! +moment=0 +LT=4 +refno=9
ENDIF
IF ans=3
! +$92=Slender section bending about minor axis only
! +Mz=0 +Fv=18 +F=900 +L=4 +My=24 +stype=2 +sd12=400 +sb12=200
! +st12=6 +Lz=4000 +Ly=4000 +Kz=1 +Ky=1 +grade=5 +E=200 +moment=0
ENDIF
IF ans=4
! +$92=bi-axial bending - slender z-z axis
! +Mz=56 +Fv=8 +F=800 +L=4 +My=20 +stype=2 +sd12=400 +sb12=200
! +st12=6 +Lz=4000 +Ly=4000 +Kz=1 +Ky=1 +grade=5 +E=200 +moment=0
! +LT=4 +refno=9
ENDIF
IF ans=5
! +$92=Bi-axial bending - slender section both axes
! +Mz=110 +Fv=21 +F=20 +L=4 +My=100 +stype=2 +sd12=400 +sb12=200
! +st12=6 +Lz=4000 +Ly=4000 +Kz=0.7 +Ky=0.7 +grade=5 +E=200 +moment=1
! +udl=1 +betaM=-55 +restz=0 +betaMz=-55 +resty=0 +betaMy=-50 +LT=4
! +refno=9
ENDIF
ENDIF
IF ZZZZZ2=1
%
%
%<H1>Stainless steel hollow section design
%The following proforma is based on elastic design methods
%and assumes that primarily it will be interfaced with a plane
%frame program.
%
%It would be helpful to the user if the BM and SF diagrams were
%printed, enabling further input of information during the course
%of the calculations to be easily produced without having to stop
%periodically to display the diagrams as may be required.
ENDIF
! +ZZZZZ1=1 +ZZZZZ2=0
Location: +$92=????
```



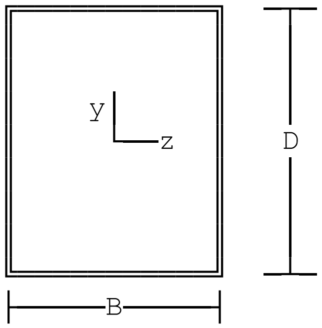
Stainless Steel Section
Structural Hollow Section Design

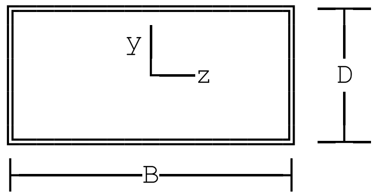
Calculations in accordance
with BS5950-1:2000 and SCI
publication 'Structural Design
of Stainless steel'

All loads and moments are
factored.

```
//
%The values of bending moment, shear force and axial load about
%to be input should be the FACTORED values resulting from
%the loads on the structure multiplied by the relevant load
%factors given in Table 2 of BS5950: Part 1:2000.
Factored bending moment axis zz   +Mz=???? kNm
Factored SF in y direction        +Fv=???? kN
```

```
Axial load ( comp. positive )      +F=???? kN
Length of member                   +L=???? m
%
%
%
%
%
%
%
%
%
%
%
```





Rectangular hollow sections
may be input with either side
vertical.

```
IF NRESP>0
!Depth of section                    +D'=???? m
!Width of section                    +B'=???? m
!Thickness of flange                 +T'=???? m
!Thickness of web                    +t'=???? m
! +D=D'*1000 +B=B'*1000 +T=T'*1000 +t=t'*1000
ENDIF
IF Mz<0 THEN Mz=ABS(Mz) ENDIF
IF Fv<0 THEN Fv=ABS(Fv) ENDIF
Factored bending moment axis y-y  +My=???? kNm
IF My<0 THEN My=ABS(My) ENDIF
IF F<0
.TENSILE LOAD APPLIED
ENDIF
! +Mtt=Mz+My

/10
Section properties
__________________
IF NRESP>0
IF D=B
! +stype=1 +ssd1=11 +tbn=11
! +sd(ssd1)=D +sb(ssd1)=B +st(ssd1)=T
ELSE
! +stype=2 +srd1=12 +tbn=12
! +sd(srd1)=D +sb(srd1)=B +st(srd1)=T
ENDIF
ELSE
%Section type available                                    Type
%______________________                                    ____
%Square Hollow section                                       1
%Rectangular Hollow section                                  2
!Section type (1=SHS 2=RHS)                                +stype=????
chkrng 2 1 stype 0
! +tbn=stype+10 +NRESP=0
ENDIF
IF stype=1
ssdp tbn
ENDIF
IF stype=2
srdp tbn
ENDIF
! +Iz=Ix +rz=rx +Zz=Zx +Sz=Sx +NRESP=0
IF F>0
//

/8
```

Effective lengths Clause 4.7.2

```
Length of member                    +L*1000 mm
%The length L for calculation purposes is taken as the distance
%between the points of effective restraint.
Length between restraints z axis  +Lz=???? mm
Length between restraints y axis  +Ly=???? mm
```

Table 22 Nominal effective length for a compression member		
Conditions of restraint at ends		Effective Length
Effectively held in position at both ends	Restrained in direction at both ends	K = 0.7
	Partially restrained in direction at both ends	K = 0.85
	Restrained in direction at one end	K = 0.85
	NOT restrained in direction at either end	K = 1.0

```
%
%Conditions of restraint factors
%
%restraint factor z-z axis        +Kz=????
%restraint factor y-y axis        +Ky=????
Effective length about z-z axis   +Lez=Kz*Lz mm
Effective length about y-y axis   +Ley=Ky*Ly mm
ENDIF

pyval
! +flag=1
IF Mz=0 THEN flag=0 ENDIF
IF My=0 THEN flag=0 ENDIF

/10
Classification Clause 3.5.2

Parameter (Table 3.1 Note 3)      +e=(275/py*E/205)^0.5
Classify internal element of compression flange:
Internal element                  +b=B-6*t mm
Internal element                  +d=D-6*t mm
IF flag=0
IF My<>0
For moment applied about the y-y axis only
Ratio - local buckling of flange  +b't=d/t
Ratio - local buckling of web     +d't=b/t
ELSE
For moment applied about the z-z axis only
Ratio - local buckling of flange  +b't=b/t
Ratio - local buckling of web     +d't=d/t
ENDIF
IF b't<=23*e
As b/t <= 23e ( +23*e ), internal element of compression
flange is classified as Class 1 plastic
! +classf=1
ELSE
```

```
IF b't<=25*e
As 23e < b/t <= 25e ( +25*e ), internal element of compression
flange is classified as Class 2 compact
! +classf=2
ELSE
IF b't<=28*e
As 25e < b/t <= 28e ( +28*e ), internal element of compression
flange is classified as Class 3 semi-compact
! +classf=3
ELSE
As b/t > 28e ( +28*e ), internal element of compression
flange is classified as Class 4 slender
! +classf=4
ENDIF
ENDIF
ENDIF
/5
Classify web of section:
IF Mz<>0
Ratio                             +r1=F*10^3/(2*d*t*py)
ELSE
Ratio                             +r1=F*10^3/(2*b*t*py)
ENDIF
IF r1>1
Value limited                     +r1=1
As the value of r > 1, section is overstressed
!Press < and RETURN to revise, RETURN to continue ????
ENDIF
IF r1<-1
Value limited                     +r1=-1
As the value of r < -1, section is overstressed
!Press < and RETURN to revise, RETURN to continue ????
ENDIF
Limiting local buckling ratio     +d'tlim=28*e
IF r1>=0
IF d't<52*e/(1+r1)
As d't < 52e/(1+r1) ( +52*e/(1+r1) ),  web is classified
as class 1 plastic
! +classw=1
ELSE
IF d't<54*e/(1+r1)
As d't < 54e/(1+r1) ( +54*e/(1+r1) ),  web is classified
as class 2 compact
! +classw=2
ELSE
IF Mz<>0
Maximum compressive stress        +sigma1=F*10/A+Mz*10^3/Zz N/mm2
Compressive/tensile stress        +sigma2=F*10/A-Mz*10^3/Zz N/mm2
ELSE
Maximum compressive stress        +sigma1=F*10/A+My*10^3/Zy N/mm2
Compressive/tensile stress        +sigma2=F*10/A-My*10^3/Zy N/mm2
ENDIF
Ratio of stresses                 +psi=sigma2/sigma1
IF psi<-2
Limit value                       +psi=-2
ENDIF
IF psi>1
Limit value                       +psi=1
ENDIF
```

```
IF psi<=-1
Factor                            +k=5.98*(1-psi)^2
ELSE
Factor                 +k=16/(((1+psi)^2+0.112*(1-psi)^2)^0.5+(1+psi))
ENDIF
Semi-compact limiting value        +d't1=14.1*e*k^0.5
IF d't<d'tl
As d't < d'tl,  web is classified as class 3 semi-compact
! +classw=3
ELSE
As d't > d'tl,  web is classified as class 4 slender
! +classw=4
ENDIF
ENDIF
ENDIF
ELSE
IF d't<572*e/(13*r1+11)
As d't < 572e/(13r1+11) ( +572*e/(13*r1+11) ),  web is classified
as class 1 plastic
! +classw=1
ELSE
IF d't<594*e/(13*r1+11)
As d't < 594e/(13r1+11) ( +594*e/(13*r1+11) ),  web is classified
as class 2 compact
! +classw=2
ELSE
IF My<>0
Tensile stress                    +sigma1=F*10/A-My*10^3/Zy N/mm2
Compressive/tensile stress        +sigma2=F*10/A+My*10^3/Zy N/mm2
ELSE
Tensile stress                    +sigma1=F*10/A-Mz*10^3/Zz N/mm2
Compressive/tensile stress        +sigma2=F*10/A+Mz*10^3/Zz N/mm2
ENDIF
Ratio of stresses                 +psi=sigma2/sigma1
IF psi>-2
Limit value                       +psi=-2
ENDIF
IF psi>1
Limit value                       +psi=1
ENDIF
IF psi<=1
Factor                            +k=5.98*(1-psi)^2
ELSE
Factor                +k=16/(((1*psi)^2+0.112*(1-psi)^2)^0.5+(1-psi))
ENDIF
Semi-compact limiting value        +d'tl=14.1*e*k^0.5
IF d't<d'tl
As d't < d'tl,  web is classified as class 3 semi-compact
! +classw=3
ELSE
As d't > d'tl,  web is classified as class 4 slender
! +classw=4
ENDIF
ENDIF
ENDIF
ENDIF
ELSE
As the section is subject to biaxial bending the classification
will be based on the comment in appendix H.3.1 for RHS sections
"the maximum axial force in each face should be determined taking
account of the moment about the axis parallel to the face"
```

/8

```
For the z-z axis

Axial force                        +Fc=F+2*My*10^3/(B-t) kN
Ratio - local buckling of web      +d't=d/t
! +webld=1
webcl
! +clasz=classw
IF classw>2 THEN psiz=psi ENDIF

/8
For the y-y axis

Axial force                        +Fc=F+2*Mz*10^3/(D-t) kN
Ratio - local buckling of web      +d't=b/t
! +webld=2
webcl
! +clasy=classw
IF classw>2 THEN psiy=psi ENDIF
IF clasy<clasz
! +class=clasz
ELSE
! +class=clasy
ENDIF
IF class=4
Section is classified slender

/8
Reduced section properties for slender sections

The effective modulus, Zeff, is determined using the arrangement
detailed in Table 3.2
IF clasz=4
Depth                              d= +d mm
Web width                          t= +t mm
IF F=0
Factor for effective web length    +rho=106/(d/(t*e)+37)
ELSE
IF psiz<0
Factor for effective web length    +rho=21.8*k^0.5/(d/(t*e)+7.7*k^0.5)
ELSE
Factor for effective web length    +rho=44/(d/(t*e)+16)
ENDIF
ENDIF
IF psiz<=0
The ratio 'psi' is based on the stress at either end of the web and
the gross sectional properties.
Ratio                              psiz= +psiz
Effective depth term               +beff=rho*d/(1-psiz) mm
Effective depth term               +de1=0.4*beff mm
Effective depth term               +de2=0.6*beff mm
Finding the non-effective zone of the web as shown in Table 3.2
requires an iterative procedure to balance the force equations
Considering the compressive stress as being equal to 'py'
the tensile stress is proportioned accordingly
!Take depth                         +d'=d/2
REPEAT
!+d'=d'+0.1
!Remainder                          +dr=d-d' mm
!Tensile load                       +FT=py*dr/d'*dr*t*0.5 N
!Compression load Terms
!                                   +fc1=py*de2/d'*de2*t*0.5 N
!                                   +pc1=py*(d'-de1)/d' N/mm2
```

```
!                                  +fc2=(py+pc1)/2*de1*t N
!                                  +FC=fc1+fc2 N
UNTIL FC>FT
ENDREPEAT
Consider
Depth of web in compression       d'= +d' mm
Depth of web in tension           +dr=d-d' mm
Tensile force                     +FT=1/2*py*dr/d'*dr*t/10^3 kN
Compression force term            +fc1=1/2*py*de2/d'*de2*t/10^3 kN
Compression force term     +fc2=(py+(py*(d'-de1)/d'))/2*de1*t/10^3 kN
Compressive force                 +FC=fc1+fc2 kN
Non-effective depth of web        +dn=d'-beff mm
ELSE
Effective depth term              +beff=rho*d mm
Non-effective depth of web        +dn=d-beff mm
ENDIF
Non-effective area                +an=dn*t*2 mm2
ENDIF
IF clasy=4
Depth                             b= +b mm
Web width                         t= +t mm
IF F=0
Factor for effective web length   +rho=106/(b/(t*e)+37)
ELSE
IF psiy<0
Factor for effective web length   +rho=21.8*k^0.5/(b/(t*e)+7.7*k^0.5)
ELSE
Factor for effective web length   +rho=44/(b/(t*e)+16)
ENDIF
ENDIF
IF psiy<=0
The ratio 'psi' is based on the stress at either end of the web and
the gross sectional properties.
Ratio                             psiy= +psiy
Effective depth term              +beff=rho*b/(1-psiy) mm
Effective depth term              +be1=0.4*beff mm
Effective depth term              +be2=0.6*beff mm
Finding the non-effective zone of the web as shown in Table 3.2
requires an iterative procedure to balance the force equations
Considering the compressive stress as being equal to 'py'
the tensile stress is proportioned accordingly
!Take depth                        +d'=b/2
REPEAT
!+d'=d'+0.1
!Remainder                         +dr=b-d' mm
!Tensile load                      +FT=py*dr/d'*dr*t*0.5 N
!Compression load Terms
!                                  +fc1=py*be2/d'*be2*t*0.5 N
!                                  +pc1=py*(d'-be1)/d' N/mm2
!                                  +fc2=(py+pc1)/2*be1*t N
!                                  +FC=fc1+fc2 N
UNTIL FC>FT
ENDREPEAT
Consider
Depth of web in compression       d'= +d' mm
Depth of web in tension           +dr=b-d' mm
Tensile force                     +FT=1/2*py*dr/d'*dr*t/10^3 kN
Compression force term            +fc1=1/2*py*be2/d'*be2*t/10^3 kN
Compression force term     +fc2=(py+(py*(d'-be1)/d'))/2*be1*t/10^3 kN
Compressive force                 +FC=fc1+fc2 kN
Non-effective depth of web        +bn=d'-beff mm
```

```
ELSE
Effective depth term              +beff=rho*b mm
Non-effective depth of web        +bn=b-beff mm
ENDIF
Non-effective area                +ar=bn*t*2 mm2
ENDIF
Area of section                   A= +A cm2
IF clasy<>4
Effective area                    +Ar=A*100-an mm2
ENDIF
IF clasz<>4
Effective area                    +Ar=A*100-ar mm2
ENDIF
IF clasz=4
IF clasy=4
Effective area                    +Ar=A*100-an-ar mm2
ENDIF
ENDIF
                                  +Aeff=Ar/100 cm2
IF clasy<>4
IF psiz<0
Distance to centre of 'dn'        +dm=3*t+de1+dn/2 mm
ELSE
As web is in compression throughout
Distance to centre of 'dn'        +dm=D/2 mm
ENDIF
Position of centroid              +yn=(A*10^2*D/2-an*dm)/Ar mm
Eccentricity                      +ez=yn-D/2 mm
Reduction in 'I' value (web) +iw=(2*t*dn^3/12+an*(yn-dm)^2)/10^4 cm4
Increase in 'I' value             +ia=Ar*(yn-D/2)^2/10^4 cm4
Effective elastic modulus         +Zeff=(Iz-iw+ia)/(yn/10) cm3
Reduction in 'I' value (web) +iwy=(2*dn*t^3/12+an*((B-t)/2)^2)/10^4 cm4
Effective elastic modulus         +Zyeff=(Iy-iwy)/(B/20) cm3
Eccentricity                      +ey=0 mm
ENDIF
IF clasz<>4
IF psiy<0
Distance to centre of 'bn'        +bm=3*t+be1+bn/2 mm
ELSE
As web is in compression throughout
Distance to centre of 'bn'        +bm=b/2 mm
ENDIF
Position of centroid              +yn=(A*10^2*D/2-ar*bm)/Ar mm
Eccentricity                      +ey=yn-B/2 mm
Reduction in 'I' value  (web) +iw=(2*t*bn^3/12+ar*(yn-bm)^2)/10^4 cm4
Increase in 'I' value             +ia=Ar*(yn-B/2)^2/10^4 cm4
Effective elastic modulus         +Zyeff=(Iy-iw+ia)/(yn/10) cm3
Reduction in 'I' value (web) +iwz=(2*bn*t^3/12+an*((D-t)/2)^2)/10^4 cm4
Effective elastic modulus         +Zeff=(Iz-iwz)/(D/20) cm3
Eccentricity                      +ez=0 mm
ENDIF
IF clasy=4
IF clasz=4
z-z axis:
IF psiz<0
Distance to centre of 'dn'        +dm=3*t+de1+dn/2 mm
ELSE
As web is in compression throughout
Distance to centre of 'dn'        +dm=d/2 mm
ENDIF
Area moment term for 'ar'         +amr=ar/2*t/2+ar/2*(D-t/2) mm3
Position of centroid              +yn=(A*10^2*D/2-an*dm-amr)/Ar mm
```

```
IF stype=1
For doubly symmetric slender sections
Eccentricity                      +ez=0 mm
ELSE
Eccentricity                      +ez=yn-D/2 mm
ENDIF
Reduction in 'I' value     +iw=(2*t*dn^3/12+an*(yn-dm)^2)/10^4 cm4
Reduction in 'I' value     +if1=bn*t*(yn-t/2)^2+bn*t*(D-yn-t/2)^2 mm4
Reduction in 'I' value            +if=(2*bn*t^3/12+if1)/10^4 cm4
Increase in 'I' value             +ia=Ar*(yn-D/2)^2/10^4 cm4
Effective elastic modulus         +Zeff=(Iz-if-iw+ia)/(yn/10) cm3
y-y axis:
IF psiy<0
Distance to centre of 'bn'        +bm=3*t+be1+bn/2 mm
ELSE
As web is in compression throughout
Distance to centre of 'bn'        +bm=B/2 mm
ENDIF
Area moment term for 'an'         +anr=an/2*t/2+an/2*(B-t/2) mm3
Position of centroid              +xn=(A*10^2*B/2-ar*bm-anr)/Ar mm
IF stype=1
For doubly symmetric slender sections
Eccentricity                      +ey=0 mm
ELSE
Eccentricity                      +ey=xn-B/2 mm
ENDIF
Reduction in 'I' value      +iw=(2*t*bn^3/12+ar*(xn-bm)^2)/10^4 cm4
Reduction in 'I' value      +if1=dn*t*(xn-t/2)^2+dn*t*(B-xn-t/2)^2 mm4
Reduction in 'I' value            +if=(2*dn*t^3/12+if1)/10^4 cm4
Increase in 'I' value             +ia=Ar*(xn-B/2)^2/10^4 cm4
Effective elastic modulus         +Zyeff=(Iy-if-iw+ia)/(xn/10) cm3
ENDIF
ENDIF
ENDIF class=4
ENDIF
IF flag=0
IF classf<3
IF classw<3 THEN class=1 ENDIF
ENDIF
IF classf=3 THEN class=3 ENDIF
IF classf<>3
IF classw=3 THEN class=3 ENDIF
ENDIF
! +slender=0
IF classf>3
! +slender=1 +class=4
ENDIF
IF slender<>1
IF classw>3 THEN class=4 ENDIF
ENDIF
IF slender=1
```

/8

Reduced section properties for slender sections

```
IF Mz<>0
The effective modulus, Zeff, is determined using the arrangement
detailed in Table 3.2
IF classf=4
Factor for effective flange width +rho=44/(b/(t*e)+16)
Effective flange width            +beff=rho*b mm
Reduction in width                +br=b-beff mm
```

```
Reduction in area                  +ar=br*t mm2
IF classw<>4
Effective area                     +Ar=A*10^2-ar mm2
                                   +Aeff=Ar/100 cm2
Position of centroid               +yn=(A*10^2*D/2-ar*t/2)/Ar mm
Reduction in 'I' value         +if=(br*t^3/12+ar*(yn-t/2)^2)/10^4 cm4
Increase in 'I' value              +ia=Ar*(yn-D/2)^2/10^4 cm4
Effective elastic modulus          +Zeff=(Iz-if+ia)/(yn/10) cm3
ENDIF
ENDIF
IF classw=4
%Please wait until the depth is calculated
Depth                              d= +d mm
Web width                          t= +t mm
IF F=0
Factor for effective web length    +rho=106/(d/(t*e)+37)
ELSE
IF psi<0
Factor for effective web length    +rho=21.8*k^0.5/(d/(t*e)+7.7*k^0.5)
ELSE
Factor for effective web length    +rho=44/(d/(t*e)+16)
ENDIF
ENDIF
IF psi<=0
The ratio 'psi' is based on the stress at either end of the web and
the gross sectional properties.
Ratio                              psi= +psi
Effective depth term               +beff=rho*d/(1-psi) mm
Effective depth term               +de1=0.4*beff mm
Effective depth term               +de2=0.6*beff mm
Finding the non-effective zone of the web as shown in Table 3.2
requires an iterative procedure to balance the force equations
Considering the compressive stress as being equal to 'py'
the tensile stress is proportioned accordingly
!Take depth                         +d'=d/2
REPEAT
!+d'=d'+0.1
!Remainder                          +dr=d-d' mm
!Tensile load                       +FT=py*dr/d'*dr*t*0.5 N
!Compression load Terms
!                                   +fc1=py*de2/d'*de2*t*0.5 N
!                                   +pc1=py*(d'-de1)/d' N/mm2
!                                   +fc2=(py+pc1)/2*de1*t N
!                                   +FC=fc1+fc2 N
UNTIL FC>FT
ENDREPEAT
Consider
Depth of web in compression        d'= +d' mm
Depth of web in tension            +dr=d-d' mm
Tensile force                      +FT=1/2*py*dr/d'*dr*t/10^3 kN
Compression force term             +fc1=1/2*py*de2/d'*de2*t/10^3 kN
Compression force term    +fc2=(py+(py*(d'-de1)/d'))/2*de1*t/10^3 kN
Compressive force                  +FC=fc1+fc2 kN
Non-effective depth of web         +dn=d'-beff mm
ELSE
Effective depth term               +beff=rho*d mm
Non-effective depth of web         +dn=d-beff mm
ENDIF
Non-effective area                 +an=dn*t*2 mm2
Area of section                    A= +A cm2
IF classf<>4
Effective area                     +Ar=A*100-an mm2
```

```
ELSE
Effective area                    +Ar=A*100-an-ar mm2
ENDIF
                                  +Aeff=Ar/100 cm2
IF psi<0
Distance to centre of 'dn'        +dm=3*t+de1+dn/2 mm
ELSE
Distance to centre of 'dn'        +dm=D/2 mm
ENDIF
IF classw<>4
Position of centroid              +yn=(A*10^2*D/2-an*dm)/Ar mm
Eccentricity                      +ez=yn-D/2 mm
Reduction in 'I' value (web)  +iw=(2*t*dn^3/12+an*(yn-dm)^2)/10^4 cm4
Increase in 'I' value             +ia=Ar*(yn-D/2)^2/10^4 cm4
Effective elastic modulus         +Zeff=(Iz-iw+ia)/(yn/10) cm3
ELSE
Position of centroid              +yn=(A*10^2*D/2-ar*t/2-an*dm)/Ar mm
Eccentricity                      +ez=yn-D/2 mm
Reduction in 'I' value (web)  +iw=(2*t*dn^3/12+an*(yn-dm)^2)/10^4 cm4
Reduction in 'I' value (flange) +if=(br*t^3/12+ar*(yn-t/2)^2)/10^4 cm4
Increase in 'I' value             +ia=Ar*(yn-D/2)^2/10^4 cm4
Effective elastic modulus         +Zeff=(Iz-if-iw+ia)/(yn/10) cm3
ENDIF
ENDIF
ENDIF
ENDIF
IF My<>0
The effective modulus, Zyeff, is determined using the arrangement
detailed in Table 3.2
IF classf=4
Factor for effective flange width +rho=44/(d/(t*e)+16)
Effective flange width            +beff=rho*d mm
Reduction in width                +br=d-beff mm
Reduction in area                 +ar=br*t mm2
IF classw<>4
Effective area                    +Ar=A*10^2-ar mm2
                                  +Aeff=Ar/100 cm2
Position of centroid              +yn=(A*10^2*B/2-ar*t/2)/Ar mm
Eccentricity                      +ey=yn-B/2 mm
Reduction in 'I' value        +if=(br*t^3/12+ar*(yn-t/2)^2)/10^4 cm4
Increase in 'I' value             +ia=Ar*(yn-B/2)^2/10^4 cm4
Effective elastic modulus         +Zyeff=(Iy-if+ia)/(yn/10) cm3
ENDIF
ENDIF
IF classw=4
%Please wait until the depth is calculated
Depth                             b= +b mm
Web width                         t= +t mm
IF F=0
Factor for effective web length   +rho=106/(b/(t*e)+37)
ELSE
IF psi<0
Factor for effective web length   +rho=21.8*k^0.5/(b/(t*e)+7.7*k^0.5)
ELSE
Factor for effective web length   +rho=44/(b/(t*e)+16)
ENDIF
ENDIF
IF psi<=0
The ratio 'psi' is based on the stress at either end of the web and
the gross sectional properties.
Ratio                             psi= +psi
Effective depth term              +beff=rho*b/(1-psi) mm
```

```
Effective depth term               +de1=0.4*beff mm
Effective depth term               +de2=0.6*beff mm
Finding the non-effective zone of the web as shown in Table 3.2
requires an iterative procedure to balance the force equations
Considering the compressive stress as being equal to 'py'
the tensile stress is proportioned accordingly
!Take depth                         +d'=b/2
REPEAT
!+d'=d'+0.1
!Remainder                          +dr=b-d' mm
!Tensile load                       +FT=py*dr/d'*dr*t*0.5 N
!Compression load Terms
!                                   +fc1=py*de2/d'*de2*t*0.5 N
!                                   +pc1=py*(d'-de1)/d' N/mm2
!                                   +fc2=(py+pc1)/2*de1*t N
!                                   +FC=fc1+fc2 N
UNTIL FC>FT
ENDREPEAT
Consider
Depth of web in compression        d'= +b' mm
Depth of web in tension            +dr=b-d' mm
Tensile force                      +FT=1/2*py*dr/d'*dr*t/10^3 kN
Compression force term             +fc1=1/2*py*de2/d'*de2*t/10^3 kN
Compression force term      +fc2=(py+(py*(d'-de1)/d'))/2*de1*t/10^3 kN
Compressive force                  +FC=fc1+fc2 kN
Non-effective depth of web         +dn=d'-beff mm
ELSE
Effective depth term               +beff=rho*b mm
Non-effective depth of web         +dn=b-beff mm
ENDIF
Non-effective area                 +an=dn*t*2 mm2
Area of section                    A= +A cm2
IF classf<>4
Effective area                     +Ar=A*100-an mm2
ELSE
Effective area                     +Ar=A*100-an-ar mm2
ENDIF
                                   +Aeff=Ar/100 cm2
IF psi<0
Distance to centre of 'dn'         +dm=3*t+de1+dn/2 mm
ELSE
Distance to centre of 'dn'         +dm=b/2 mm
ENDIF
IF classw<>4
Position of centroid               +yn=(A*10^2*D/2-an*dm)/Ar mm
Eccentricity                       +ey=yn-B/2 mm
Reduction in 'I' value (web) +iw=(2*t*dn^3/12+an*(yn-dm)^2)/10^4 cm4
Increase in 'I' value              +ia=Ar*(yn-D/2)^2/10^4 cm4
Effective elastic modulus          +Zyeff=(Iy-iw+ia)/(yn/10) cm3
ELSE
Position of centroid               +yn=(A*10^2*B/2-ar*t/2-an*dm)/Ar mm
Eccentricity                       +ey=yn-B/2 mm
Reduction in 'I' value (web)  +iw=(2*t*dn^3/12+an*(yn-dm)^2)/10^4 cm4
Reduction in 'I' value (flange) +if=(br*t^3/12+ar*(yn-t/2)^2)/10^4 cm4
Increase in 'I' value              +ia=Ar*(yn-B/2)^2/10^4 cm4
Effective elastic modulus          +Zyeff=(Iy-if-iw+ia)/(yn/10) cm3
ENDIF
ENDIF
ENDIF
ENDIF
IF Fv<>0

/10
```

Web shear

```
IF Mz<>0
Shear area                        +Av=(D/(D+B))*A*100 mm2
ELSE
Shear assumed to be acting in conjunction with moment about y-y axis
Shear area                        +Av=(B/(D+B))*A*100 mm2
ENDIF
Shear capacity                    +Pv=0.6*py*Av/10^3 kN
Shear force                       Fv= +Fv kN
IF Fv<=Pv
Since Fv <= Pv ( +Fv kN <= +Pv kN ) shear force within
shear capacity.
ELSE
Since Fv > Pv ( +Fv kN > +Pv kN ) shear force exceeds
shear capacity.
STOP
ENDIF
IF Fv>0.6*Pv
Since Fv > 0.6 Pv
Section unsuitable for design by this proforma
STOP
ENDIF
ENDIF
IF Mtt<>0

/10
```

Moment Capacities

```
IF Mz<>0
IF class<3
Since Fv < 0.6 Pv
Moment capacity for compact sec   +Mc=py*Sz/10^3 kNm
IF F=0
IF Mc>1.2*py*Zz/1000
Reduce Mc to limiting value       +Mc=1.2*py*Zz/10^3 kNm
ENDIF
ENDIF
IF F<>0
IF Mc>1.5*py*Zz/1000
Reduce Mc to limiting value       +Mc=1.5*py*Zz/10^3 kNm
ENDIF
ENDIF
ENDIF
IF class=3
Moment capacity based on elastic modulus of section
Since Fv < 0.6 Pv                 +Mc=py*Zz/10^3 kNm
ENDIF
IF class=4
Moment capacity based on effective elastic section
Since Fv < 0.6 Pv                 +Mc=py*Zeff/10^3 kNm
ENDIF
IF Mz>Mc
Since Mz > Mc ( +Mz kNm > +Mc kNm ) applied moment exceeds
moment capacity.
STOP
ELSE
Since Mz <= Mc ( +Mz kNm <= +Mc kNm ) applied moment within
moment capacity.
ENDIF
ENDIF
! +flag1=1
```

```
IF My<>0
IF Mz<>0
IF stype=1
For square sections              +Mcy=Mc kNm
! +flag1=0
ENDIF
ENDIF
IF flag1=1
IF class<3
Since Fv < 0.6 Pv
Moment capacity for compact sec  +Mcy=py*Sy/10^3 kNm
IF F=0
IF Mcy>1.2*py*Zy/1000
Reduce Mc to limiting value      +Mcy=1.2*py*Zy/10^3 kNm
ENDIF
ENDIF
IF F<>0
IF Mcy>1.5*py*Zy/1000
Reduce Mc to limiting value      +Mcy=1.5*py*Zy/10^3 kNm
ENDIF
ENDIF
ENDIF
IF class=3
Moment capacity based on elastic modulus of section
Since Fv < 0.6 Pv                +Mcy=py*Zy/10^3 kNm
ENDIF
IF class=4
Moment capacity based on effective elastic section
Since Fv < 0.6 Pv                +Mcy=py*Zyeff/10^3 kNm
ENDIF
ENDIF
IF My>Mcy
Since My > Mcy ( +My kNm > +Mcy kNm ) applied moment exceeds
moment capacity.
STOP
ELSE
Since My <= Mcy ( +My kNm <= +Mcy kNm ) applied moment within
moment capacity.
ENDIF
ENDIF
ENDIF
IF F<>0
IF Mtt<>0

/10
Local capacity check
____________________
IF F<0
%Gross area of section           +A cm2
%As the axial load is tensile the effective area Ae must be estimated.
%Ae = ( Gross area - area of holes ) * Ke      Ke= 1.2 for Grade 275
Estimated net area ( tension )   +Ae=???? cm2
IF Ae>A
Estimated net area must be less than gross area.
STOP
ENDIF
Area to be used for unity check  +A=Ae cm2
Tensile capacity                 +Pt=Ae*py/10 kN
IF ABS(F)>Pt
As the tensile load ( +ABS(F) kN ) > tensile capacity ( +Pt kN )
Section is not suitable.
STOP
```

ENDIF
ENDIF
IF F>0
IF class=4
For slender cross-sections

$$\frac{F}{Aeff.py} + \frac{Mz + Fc.ez}{Mc} + \frac{My + Fc.ey}{Mcy} \leq 1$$

IF Mz<>0
Applied moment +Mzz=Mz+F*ez/10^3 kNm
ENDIF
IF My<>0
Applied moment +Myy=My+F*ey/10^3 kNm
ENDIF
IF Mz<>0
IF My<>0
Unity factor +uf=F*10/(Aeff*py)+Mzz/Mc+Myy/Mcy
ENDIF
ENDIF
IF My=0
Minor axis moment is zero hence:
Unity factor +uf=F*10/(Aeff*py)+Mzz/Mc
ENDIF
IF Mz=0
Major axis moment is zero hence:
Unity factor +uf=F*10/(Aeff*py)+Myy/Mcy
ENDIF
ELSE
For plastic, compact and semi-compact cross-sections

$$\frac{F}{A.py} + \frac{Mz}{Mc} + \frac{My}{Mcy} \leq 1$$

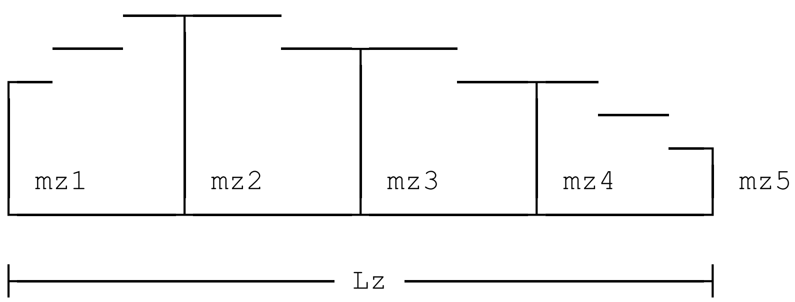
IF Mz<>0
IF My<>0
Unity factor +uf=F*10/(A*py)+Mz/Mc+My/Mcy
ENDIF
ENDIF
IF My=0
Minor axis moment is zero hence:
Unity factor +uf=F*10/(A*py)+Mz/Mc
ENDIF
IF Mz=0
Major axis moment is zero hence:
Unity factor +uf=F*10/(A*py)+My/Mcy
ENDIF
ENDIF
ENDIF
IF F<0

$$\frac{F}{A.py} + \frac{Mz}{Mc} + \frac{My}{Mcy} \leq 1$$

IF Mz<>0
IF My<>0
Unity factor +uf=ABS(F)/Pt+Mz/Mc+My/Mcy
ENDIF
ENDIF

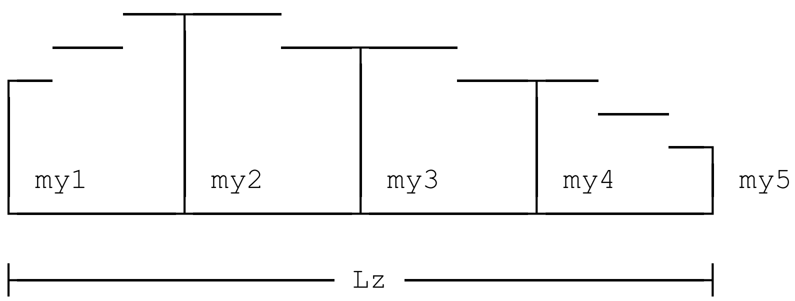
```
IF My=0
Minor axis moment is zero hence:
Unity factor                     +uf=ABS(F)/Pt+Mz/Mc
ENDIF
IF Mz=0
Major axis moment is zero hence:
Unity factor                     +uf=ABS(F)/Pt+My/Mcy
ENDIF
ENDIF
IF uf<=1
The interaction formula is satisfied.
IF F<0
adopt
STOP
ENDIF
ELSE
The interaction formula exceeds 1 and the section must be increased.
STOP
ENDIF
ENDIF
ENDIF
IF F>0
axload
IF Mtt=0
adopt
STOP
ENDIF
ENDIF
IF F>0
IF Mtt<>0
IF stype<>1

//
/8
Equivalent uniform moment factors 4.8.3.3.4

%When dealing with the overall buckling check the equivalent uniform
%moment factors are required.  The factors considered are:
%
IF Mz<>0
%mLT - for lateral torsional buckling
%mz  - for major axis flexural buckling
ENDIF
IF My<>0
%my  - for minor axis flexural buckling
ENDIF
%
%Conservatively or for the destabilising load condition
%these values may be taken as equal to 1.0.
%
%If a more exact approach is required and you have not already
%printed out the BM diagrams, at the next PROMPT press < and RETURN
%and go back to the start of the program selecting option 1 to print
%out BM diagrams for all those members you will be designing.
%Note : You will need the 'far end' bending moment and the direction
%       of the moment must also be considered, the maximum BM is
%       taken as the reference. If the other end moment is of
%       opposite sign to the maximum then they are considered as
%       giving a positive ratio.
%
!More exact approach (1=Yes 0=No) +moment=????
```

```
IF moment=0
For the overall buckling check the equivalent uniform moment factors
are conservatively taken as:
IF Mz<>0
For lateral torsional buckling    +mLT=1.0
For major axis flexural bucking   +mz=1.0
ENDIF
IF My<>0
For minor axis flexural bucking   +my=1.0
IF Mz=0
For lateral torsional buckling    +mLT=1.0
ENDIF
ENDIF
ELSE
%This proforma does not determine the lateral buckling resistance
%due to the moments in the hogging moment region (i.e. a rafter)
%using Appendix G.2.
mLTval
IF Mz<>0

/8
Major axis flexural buckling  Table 26

The factor 'mz' is over the segment length Lz governing Pcz
Length between restraints z axis  Lz= +Lz mm
%
%In a similar manner to that of lateral torsional buckling is
%the member loaded between restraints over the length Lz
%
!( 1 =Yes, 0 = No )     +restz=????
IF restz=0
The member is not loaded between restraints.
Maximum moment on segment         +Mz=????  kNm
Far end BM                        +betaMz=????  kNm
Table 26 beta factor              +beta=betaMz/Mz
IF ABS(beta)>1
%Table 26 value beta cannot exceed 1 or be less than -1.
%Revise input values as Me should be largest BM applied to the beam
!Press < and RETURN to revise, RETURN to stop ????
STOP
ENDIF
Equivalent uniform moment factor  +mz=TABLE(26,beta)
ENDIF
IF restz=1
The member is loaded between restraints and the equation given in the
notes to Table 26 is used to determine mz.
%
%The values m1 to m5 are the applied moments at the ends, quarter
%points and mid-length between the effective torsional restraints.
%
%
```



All values of mz shown are positive. If the values of mz2, mz3 and mz4 lie on the same side of the axis, they are taken as positive. If the values are on both sides, the side leading to the larger 'm' value is taken as the positive side.

```
%
Moments on member segment
Quarter point                    +mz2=???? kNm
Mid-span                         +mz3=???? kNm
Three quarter point              +mz4=???? kNm
Maximum Moment on segment        Mz= +Mz kNm
Maximum moment on central half   +M24=???? kNm
Equivalent uniform moment factor  +mz=0.2+(0.1*mz2+0.6*mz3+0.1*mz4)/Mz
Limiting value of m              +mzlim=0.8*M24/Mz
IF mz<mzlim
Revised moment factor            +mz=mzlim
ENDIF
ENDIF
ENDIF
IF My<>0

/8
Major axis flexural buckling  Table 26

The factor 'my' is over the segment length Ly governing Pcz
Length between restraints y axis  Ly= +Ly mm
%
%In a similar manner to that of lateral torsional buckling is
%the member loaded between restraints over the length Ly
%
!( 1 =Yes, 0 = No )     +resty=????
IF resty=0
The member is not loaded between restraints.
Maximum moment on segment        +My=???? kNm
Far end BM                       +betaMy=???? kNm
Table 26 beta factor             +beta=betaMy/My
IF ABS(beta)>1
%Table 26 value beta cannot exceed 1 or be less than -1.
%Revise input values as Me should be largest BM applied to the beam
!Press < and RETURN to revise, RETURN to stop ????
STOP
ENDIF
Equivalent uniform moment factor  +my=TABLE(26,beta)
ENDIF
IF resty=1
The member is loaded between restraints and the equation given in the
notes to Table 26 is used to determine my.
%
%The values m1 to m5 are the applied moments at the ends, quarter
%points and mid-length between the effective torsional restraints.
%
%
%
%
%
%
%
%
%
%
%
%
%
%
```



All values of mz shown are positive. If the values of my2, my3 and my4 lie on the same side of the axis, they are taken as positive. If the values are on both sides, the side leading to the larger 'm' value is taken as the positive side.

```
%
Moments on member segment
Quarter point                    +my2=???? kNm
```

```
Mid-span                           +my3=???? kNm
Three quarter point                +my4=???? kNm
Maximum Moment on segment          My= +My kNm
Maximum moment on central half     +My24=???? kNm
Equivalent uniform moment factor   +my=0.2+(0.1*my2+0.6*my3+0.1*my4)/My
Limiting value of m                +mylim=0.8*My24/My
IF my<mylim
Revised moment factor              +my=mylim
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
IF stype=1 THEN buck=0 ENDIF
IF stype=2
IF flag=1
! +buck=1
ELSE
IF Mz<>0
IF B<D
! +buck=1
ELSE
! +buck=0
ENDIF
ELSE
! +buck=0
ENDIF
ENDIF
ENDIF
IF buck=0

/3
In accordance with clause 4.3.6.1 square hollow sections and
sections with the moment applied about the minor axis need not be
checked separately for resistance to lateral-torsional buckling
! +rest=1 !!! added 22.1.06 +refno=11

ELSE
/8

Rectangular Hollow Section

Limiting value of slenderness Clause 4.3.6.1

The section slenderness will be compared with the limiting value
Table 15
Dist. betwn torsional restraints   +LT=???? m
Effective length  Cl. 4.3.5.1

/10
```

Table 13. Restraint conditions at SUPPORTS		Loading condition Normal	 Destabiliz.
Compression flange laterally restrained. Nominal torsional restraint against rotation about longtdl. axis, as specified in 4.2.2.	Both flanges fully restrained against rotation on plan	1	2
	Comp. flange fully restrained against rotation on plan	3	4
	Both flanges partially restrained against rotation on plan	5	6
	Comp. flange partially restrained against rotation on plan	7	8
	Both flanges free to rotate on plan.	9	10

```
!Reference from above table (1 to 10)  +refno=????
chkrng 10 1 refno
IF refno=1
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Both flanges fully restrained against rotation on plan.
Normal loading conditions therefore
Effective length (Table 13)       +Le=0.7*LT*1000 mm
ENDIF
IF refno=2
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Both flanges fully restrained against rotation on plan.
Destabilising loading conditions therefore
Effective length (Table 13)       +Le=0.85*LT*1000 mm
ENDIF
IF refno=3
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Compression flange fully restrained against rotation on plan.
Normal loading conditions therefore
Effective length (Table 13)       +Le=0.75*LT*1000 mm
ENDIF
IF refno=4
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Compression flange fully restrained against rotation on plan.
Destabilising loading conditions therefore
Effective length (Table 13)       +Le=0.9*LT*1000 mm
ENDIF
IF refno=5
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Both flanges partially restrained against rotation on plan.
Normal loading conditions therefore
Effective length (Table 13)       +Le=0.8*LT*1000 mm
ENDIF
IF refno=6
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Both flanges partially restrained against rotation on plan.
Destabilising loading conditions therefore
Effective length (Table 13)       +Le=0.95*LT*1000 mm
```

```
ENDIF
IF refno=7
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Compression flange partially restrained against rotation on plan.
Normal loading conditions therefore
Effective length (Table 13)       +Le=0.85*LT*1000 mm
ENDIF
IF refno=8
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Compression flange partially restrained against rotation on plan.
Destabilising loading conditions therefore
Effective length (Table 13)        +Le=1.0*LT*1000 mm
ENDIF
IF refno=9
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Both flanges free to rotate on plan.
Normal loading conditions therefore
Effective length (Table 13)       +Le=1.0*LT*1000 mm
ENDIF
IF refno=10
Compression flange laterally restrained.
Nominal torsional restraint about longitudinal axis.
Both flanges free to rotate on plan.
Destabilising loading conditions therefore
Effective length (Table 13)       +Le=1.2*LT*1000 mm
ENDIF
Slenderness (minor axis)          +lambda=Le/(ry*10)
Side ratio D/B ( Table 15 )       +D'B=D/B
IF D'B<1.25
.Side ratio must be at least 1.25, revise inputs
STOP
ENDIF
IF D'B>4.0
.Side ratio cannot exceed 4.0, revise inputs
STOP
ENDIF
!Ratio                             +d'b=TABLE(150,D'B)
Limiting slenderness              +laml=d'b*275/py
IF lambda<=laml
As actual slenderness is less than limiting slenderness
Lateral Torsional Buckling is not critical and no further
check is required.                +Mb=Mc kNm
! +rest=1
ELSE
Outside the scope of this proforma
!Press < and RETURN to revise, RETURN to stop ????
STOP
ENDIF

/10
Member buckling resistance

Overall buckling check

IF class<>4
```

$$\frac{F}{Pc} + \frac{mz.Mz}{py.Z} + \frac{my.My}{py.Zy} \leq 1$$

```
IF Mz<>0
Elastic moment capacity           +Mce=py*Zz/10^3 kNm
ENDIF
IF My<>0
Elastic moment capacity           +Mcye=py*Zy/10^3 kNm
ENDIF
IF My=0
Unity factor                      +unity=F/Pc+mz*Mz/Mce
ENDIF
IF Mz=0
Unity factor                      +unity=F/Pc+my*My/Mcye
ENDIF
IF Mz<>0
IF My<>0
Unity factor                      +unity=F/Pc+mz*Mz/Mce+my*My/Mcye
ENDIF
ENDIF
ELSE
```

$$\frac{F}{Pc} + \frac{mz.Mz + Fc.ez}{py.Zeff} + \frac{my.My + Fc.ey}{py.Zyeff} \leq 1$$

```
IF Mz<>0
Elastic moment capacity           +Mce=py*Zeff/10^3 kNm
ENDIF
IF My<>0
Elastic moment capacity           +Mcye=py*Zyeff/10^3 kNm
ENDIF
IF My=0
Unity factor                      +unity=F/Pc+mz*Mzz/Mce
ENDIF
IF Mz=0
Unity factor                      +unity=F/Pc+my*Myy/Mcye
ENDIF
IF Mz<>0
IF My<>0
Unity factor                      +unity=F/Pc+mz*Mzz/Mce+my*Myy/Mcye
ENDIF
ENDIF
ENDIF
IF unity<=1
The unity relationship is satisfied.
ELSE
The unity factor is greater than 1, thus the section must be
increased.
STOP
ENDIF
IF class<>4
/6
```

$$\frac{F}{Pcy} + \frac{mLT.Mz}{Mb} + \frac{mLT.My}{py.Zy} \leq 1$$

```
IF My=0
Unity factor                      +Unity=F/Pcy+mLT*Mz/Mb
ENDIF
IF Mz=0
Unity factor                      +Unity=F/Pcy+mLT*My*10^3/(py*Zy)
ENDIF
```

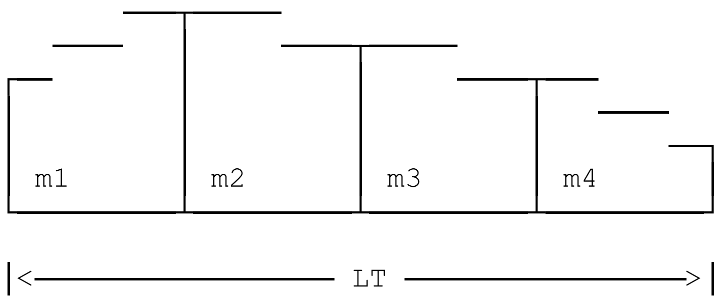
```
IF My<>0
IF Mz<>0
Unity factor            +Unity=F/Pcy+mLT*Mz/Mb+mLT*My*10^3/(py*Zy)
ENDIF
ENDIF
ELSE
/6
```

$$\frac{F}{Pcy} + \frac{mLT.Mz + F.ez}{Mb} + \frac{my.My + F.ey}{py.Zyeff} \leq 1$$

```
IF My=0
Unity factor                    +Unity=F/Pcy+mLT*Mzz/Mb
ENDIF
IF Mz=0
Unity factor                    +Unity=F/Pcy+my*Myy*10^3/(py*Zyeff)
ENDIF
IF My<>0
IF Mz<>0
Unity factor          +Unity=F/Pcy+mLT*Mzz/Mb+my*Myy*10^3/(py*Zyeff)
ENDIF
ENDIF
ENDIF
IF Unity<=1
The unity relationship is satisfied.
ELSE
The unity factor is greater than 1, thus the section must be
increased.
STOP
ENDIF
ENDIF
adopt
STOP
ENDIF
ENDIF
DEFINE adopt
/10
.
IF stype=2
.  SQUARE HOLLOW SECTION          +D x +B x +t SHS  Grade S +Grade
ENDIF
IF stype=3
.  RECTANGULAR HOLLOW SECTION     +D x +B x +t RHS  Grade S +Grade
ENDIF
.  SECTION                        Section is satisfactory for axial
.  SUMMARY                        load, and overall buckling check.
IF F<>0
.                                 Axial load              +F kN
IF F>0
.                                 Compression resistance  +Pc kN
ELSE
.                                 Axial capacity          +Pt kN
ENDIF
ENDIF
IF Mz>0
.                                 Maximum moment z axis   +Mz kNm
.                                 Moment capacity         +Mc kNm
ENDIF
IF My>0
.                                 Maximum moment y axis   +My kNm
.                                 Moment capacity         +Mcy kNm
ENDIF
```

```
IF F<0
.                                 Local capacity check    +uf < 1
ENDIF
IF F>0
.                                 Local capacity check    +uf < 1
IF buck=1
.                                 Overall buckling checks +unity < 1
.                                                         +Unity < 1
ENDIF
ENDIF
><
%%
!Self check, answer (1=Yes, 0=No) +selfc=????
IF EXIST ~self.stk OR selfc=1
selfch
ENDIF
mesage
! +NRESP=0
ENDDEFINE
DEFINE axload

/8
Compressive resistance
──────────────────────
Elastic buckling load             +Py=PI^2*E*Iy*10^4/(Ley^2) kN
Elastic buckling load             +Pz=PI^2*E*Iz*10^4/(Lez^2) kN
IF class<4
Factor                            +betac=1
ELSE
Factor                            +betac=Aeff/A
ENDIF
Non-dimensional slenderness       +Lam=(betac*A*10^2*py/(Py*10^3))^0.5
Factor for cold formed sections   +alpha=0.49
Factor                            +Lamo=0.4
Factor                            +phi=0.5*(1+alpha*(Lam-Lamo)+Lam^2)
Reduction factor for compression  +X=1/(phi+(phi^2-Lam^2)^0.5)
IF X>1.0
Reduction factor limited          +X=1.0
ENDIF
Compressive resistance            +Pcy=X*betac*A*py/10 kN
Non-dimensional slenderness       +Lam=(betac*A*10^2*py/(Pz*10^3))^0.5
Factor for cold formed sections   +alpha=0.49
Factor                            +Lamo=0.4
Factor                            +phi=0.5*(1+alpha*(Lam-Lamo)+Lam^2)
Reduction factor for compression  +X=1/(phi+(phi^2-Lam^2)^0.5)
IF X>1.0
Reduction factor limited          +X=1.0
ENDIF
Compressive resistance            +Pcz=X*betac*A*py/10 kN
IF Pcz<Pcy
Compressive resistance            +Pc=Pcz kN
ELSE
Compressive resistance            +Pc=Pcy kN
ENDIF
IF F<=Pc
Since F <= Pc ( +F kN <= +Pc kN ) design axial compressive
load does not exceed compression resistance, section suitable
in compression
ELSE
.Since F > Pc ( +F kN > +Pc kN ) design axial compressive
.load exceeds compression resistance and section size must
.be increased.
STOP
```

```
ENDIF
ENDDEFINE
DEFINE mLTval
/12
Clause 4.3.6.6 and Table 18
%When dealing with lateral torsional buckling in this proforma
%the assumption is that when the member has intermediate lateral
%restraints there are two possibilities
%                                                               Method
%The value of the "udl" is insignificant and the
%beam segment is assumed to have end moments only                1
%
%The value of the "udl" is such that the beam segment may
%be assumed to be loaded between intermediate restraints         2
%
!Which method is to be considered  (1 or 2)        +udl=????
IF udl=1
The member is not loaded between restraints.
Maximum moment on segment         +Mz= +Mz kNm
Far end BM                        +betaM=???? kNm
Table 18 beta factor              +beta=betaM/Mz
IF ABS(beta)>1
%Table 18 value beta cannot exceed 1 or be less than -1.
%Revise input values as Me should be largest BM applied to the beam
!Press < and RETURN to revise, RETURN to stop ????
STOP
ENDIF
Equivalent uniform moment factor  +mLT=TABLE(18,beta)
ENDIF
IF udl=2
The member is loaded between restraints and the equation given in the
notes to Table 18 is used to determine mLT.
%
%The values m1 to m5 are the applied moments at the ends, quarter
%points and mid-length between the effective torsional restraints.
%
%
```




```
%                                               All values of "m" shown
%                                               are positive in BMD
%                                               If values are below
%                                               line enter as positive
%                                               Mz is the maximum
%                                               moment on the segment
%
%                                               All moments are entered
%                                               as positive values
Moments on member segment
Quarter point                     +m2=???? kNm
Mid-span                          +m3=???? kNm
Three quarter point               +m4=???? kNm
Maximum Moment on segment         Mz= +Mz kNm
Equivalent uniform moment factor  +mLT=0.2+(0.15*m2+0.5*m3+0.15*m4)/Mz
IF mLT<0.44
Minimum moment factor             +mLT=0.44
ENDIF
ENDIF
ENDDEFINE
DEFINE webcl
```

```
IF webld=1
Ratio                             +r1=Fc*10^3/(2*d*t*py)
ELSE
Ratio                             +r1=Fc*10^3/(2*b*t*py)
ENDIF
IF r1>1
Value limited                     +r1=1
As the value of r > 1, section is overstressed
!Press < and RETURN to revise, RETURN to continue ????
ENDIF
IF r1<-1
Value limited                     +r1=-1
As the value of r < -1, section is overstressed
!Press < and RETURN to revise, RETURN to continue ????
ENDIF
Limiting local buckling ratio     +d't1im=28*e
IF r1>=0
IF d't<52*e/(1+r1)
As d't < 52e/(1+r1) ( +52*e/(1+r1) ),  web is classified
as class 1 plastic
! +classw=1
ELSE
IF d't<54*e/(1+r1)
As d't < 54e/(1+r1) ( +54*e/(1+r1) ),  web is classified
as class 2 compact
! +classw=2
ELSE
IF webld=1
Maximum compressive stress        +sigma1=F*10/A+Mz*10^3/Zz N/mm2
Compressive/tensile stress        +sigma2=F*10/A-Mz*10^3/Zz N/mm2
ELSE
Maximum compressive stress        +sigma1=F*10/A+My*10^3/Zy N/mm2
Compressive/tensile stress        +sigma2=F*10/A-My*10^3/Zy N/mm2
ENDIF
Ratio of stresses                 +psi=sigma2/sigma1
IF psi<-2
Limit value                       +psi=-2
ENDIF
IF psi>1
Limit value                       +psi=1
ENDIF
IF psi<=-1
Factor                            +k=5.98*(1-psi)^2
ELSE
Factor                 +k=16/(((1+psi)^2+0.112*(1-psi)^2)^0.5+(1+psi))
ENDIF
Semi-compact limiting value        +d'tl=14.1*e*k^0.5
IF d't<d'tl
As d't < d'tl,  web is classified as class 3 semi-compact
! +classw=3
ELSE
As d't > d'tl,  web is classified as class 4 slender
! +classw=4
ENDIF
ENDIF
ENDIF
ELSE
IF d't<572*e/(13*r1+11)
As d't < 572e/(13r1+11) ( +572*e/(13*r1+11) ),  web is classified
as class 1 plastic
! +classw=1
ELSE
```

```
IF d't<594*e/(13*r1+11)
As d't < 594e/(13r1+11) ( +594*e/(13*r1+11) ),  web is classified
as class 2 compact
! +classw=2
ELSE
IF webld=1
Tensile stress                    +sigma1=F*10/A+Mz*10^3/Zz N/mm2
Compressive/tensile stress        +sigma2=F*10/A-Mz*10^3/Zz N/mm2
ELSE
Tensile stress                    +sigma1=F*10/A+My*10^3/Zy N/mm2
Compressive/tensile stress        +sigma2=F*10/A-My*10^3/Zy N/mm2
ENDIF
Ratio of stresses                 +psi=sigma2/sigma1
IF psi<-2
Limit value                       +psi=-2
ENDIF
IF psi>1
Limit value                       +psi=1
ENDIF
IF psi<=1
Factor                            +k=5.98*(1-psi)^2
ELSE
Factor                +k=16/(((1*psi)^2+0.112*(1-psi)^2)^0.5+(1-psi))
ENDIF
Semi-compact limiting value        +d'tl=14.1*e*k^0.5
IF d't<d'tl
As d't < d'tl,  web is classified as class 3 semi-compact
! +classw=3
ELSE
As d't > d'tl,  web is classified as class 4 slender
! +classw=4
ENDIF
ENDIF
ENDIF
ENDIF
ENDDEFINE
!Proforma      No. 3800
!Title         Common subroutines used for stainless steel sections
!Devised by    Jim Dunbar Aug'04
DEFINE pyval
//
/10
```

Specified properties of steel BS EN 10088-2

Grade	Number	Type
Basic chromium-nickel austenitic	1.4301	1
Molybdenum-chromium-nickel austenitic	1.4401	2
Molybdenum-chromium-nickel austenitic	1.4404	3
Duplex Steel	1.4362	4
Duplex Steel	1.4462	5

```
!Steel type (1,2,3,4 or 5)                  +grade=????
chkrng 5 1 grade
The 0.2% proof stress is taken as 'py' the design strength
IF grade=1
Basic chromium-nickel austenitic grade 1.4301 (304)
Design strength                   +py=210 N/mm2
! +n1=6.5  +n2=8.5 +Grade=1.4301 +rhod=7900 +Us=520
ENDIF
```

```
IF grade=2
Molybdenum-chromium-nickel austenitic grade 1.4401 (316)
Design strength                      +py=220 N/mm2
! +n1=7.0  +n2=9.0 +Grade=1.4401 +rhod=8000 +Us=520
ENDIF
IF grade=3
Molybdenum-chromium-nickel austenitic grade 1.4404 (316L)
Design strength                      +py=220 N/mm2
! +n1=7.0  +n2=9.0 +Grade=1.4404  +rhod=8000 +Us=520
ENDIF
IF grade=4
Duplex steel grade 1.4362
Design strength                      +py=400 N/mm2
! +n1=5.0  +n2=5.0 +Grade=1.4362 +rhod=7800 +Us=600
ENDIF
IF grade=5
Duplex steel grade 1.4462
Design strength                      +py=460 N/mm2
! +n1=5.0  +n2=5.0 +Grade=1.4462 +rhod=7800 +Us=640
ENDIF
Young's Modulus                      +E=???? kN/mm2
ENDDEFINE
DEFINE mesage
POPUP 6 40 18 3

  To produce only a summary, type  CS
  and go through the calculation again.
  To reset back to normal operation
  after producing a summary, type  CN

ENDDEFINE

DEFINE chkrng ! Arguments 'max' & 'min' & 'value' ('quasi')
! +OK=1 +chk4=-1.0E39 +$27569=
! Above does not change chk4 if already set, but if it was unset,
! then chk4 is assigned the positive value 1.0E39.
IF chk3>chk1
Data +chk3 entered, exceeds maximum expected value +chk1
! +OK=0
ENDIF
IF chk3<chk2
Data +chk3 entered, is less than minimum expected value +chk2
! +OK=0
ENDIF
! If both minimum & maximum are integers, then check value is also.
IF INT(chk1)=chk1 AND INT(chk2)=chk2 AND INT(chk3)<>chk3
Data +chk3 entered, should be an integer.
! +OK=0
ENDIF
IF chk4<>1.0E39 ! If chk4 set, use it.  Check for one of two values.
IF chk4=0 AND chk3<>chk1 AND chk3<>chk2
Data +chk3 entered, should be equal to either +chk1 or +chk2
! +OK=0
ENDIF
IF chk4>0 AND chk4<32001 OR INT(chk4)<>chk4 ! Check for 1 of 3 values.
IF chk3<>chk1 AND chk3<>chk2 AND chk3<>chk4
Data +chk3 entered, should be one of: +chk1 +chk4 +chk2
! +OK=0
ENDIF
ENDIF
ENDIF
```

```
IF chk4<0 AND chk4<>INT(chk4) ! Check 'value' is in rng(1)-rng(-chk4)
! +chk=0
:2596
! +chk=chk+1
IF chk3=rng(chk) GOTO 2597
IF chk<ABS(chk4) GOTO 2596
Data +chk3 entered, should be in set: +rng(1) +rng(2) +rng(3) ...
! +OK=0
:2597
ENDIF
IF OK=0
IF chk4>32000 AND chk4<32768 AND chk4=INT(chk4)
Line number in dialogue box is +chk4-32000
ENDIF
IF chk4<0 AND chk4=INT(chk4) ! Print $(-chk4)
+$(-chk4)
ENDIF
!Press < and Enter to revise, or Enter to continue    +$27569=????
!Type * and Enter to continue, or press Enter to stop +$27569=????
IF $27569><*
! Reset chk4 to 'off' & clear text prompt. +chk4=1E39 +$27569=
STOP
ENDIF
ENDIF
! Reset chk4 to 'off' & clear text prompt. +chk4=1E39 +$27569=
ENDDEFINE

DEFINE ssdp ! Stainless steel SHS dimensions and properties
!Restructured sslist & sstab proformas
STORE ssd1 58 6
!Stainless steel Square Hollow Section properties
!              A     I    S     J     C
           1   2     3    4     5     6
40040002   1 2.87 6.659 3.994 11.33 5.187
40040003   2 4.05 8.689 5.407 15.61 6.924

50050002   3 3.67 13.71 6.488 22.77 8.469
50050003   4 5.25 18.48 8.994 32.15 11.62
50050004   5 6.67 21.97 11.02 39.88 14.07

60060002   6 4.47 24.51 9.583 40.07 12.55
60060003   7 6.45 33.71 13.48 57.34 17.50
60060004   8 8.27 41.01 16.80 72.41 21.62
60060005   9 9.93 46.53 19.56 85.01 24.92

80080002  10 6.07 60.58 17.57 96.99 23.11
80080003  11 8.85 85.32 25.15 140.9 32.88
80080004  12 11.5 106.5 31.95 181.2 41.49
80080005  13 13.9 124.4 37.98 217.6 48.99

100100003 14 11.3 173.1 40.43 280.6 53.04
100100004 15 14.7 219.4 51.91 364.4 67.75
100100005 16 17.9 260.1 62.41 442.4 81.04
100100006 17 21.0 295.6 71.95 514.4 92.93
100100008 18 26.7 351.6 88.20 638.1 112.6

125125003 19 14.3 348.4 64.58 556.3 85.00
125125004 20 18.7 446.3 83.60 727.1 109.6
125125005 21 22.9 535.5 101.4 889.6 132.3
125125006 22 27.0 616.2 117.9  1043 153.3
125125008 23 34.7 752.9 147.5  1321 190.0
```

```
150150003 24 17.3 613.9 94.36 970.5 124.5
150150004 25 22.7 792.1 122.8  1273 161.4
150150005 26 27.9 957.6 149.7  1565 196.1
150150006 27 33.0  1110 175.2  1844 228.7
150150008 28 42.7  1379 221.7  2364 287.4

175175004 29 26.7  1281 169.5  2040 223.2
175175005 30 32.9  1557 207.4  2515 272.4
175175006 31 39.0  1815 243.7  2974 319.0
175175008 32 50.7  2281 311.0  3842 404.7
175175010 33 61.7  2682 371.5  4637 480.6

200200004 34 30.7  1940 223.7  3067 295.0
200200005 35 37.9  2366 274.5  3788 361.2
200200006 36 45.0  2769 323.4  4490 424.3
200200008 37 58.7  3509 415.3  5829 542.0
200200010 38 71.7  4162 499.3  7078 648.3

250250005 39 47.9  4737 436.9  7488 576.2
250250006 40 57.0  5574 516.7  8901 680.0
250250008 41 74.7  7141 668.8 11630 876.6
250250010 42 91.7  8568 811.1 14230  1058
250250012 43  108  9859 943.6 16690  1226

300300005 44 57.9  8319 636.7 13040 841.2
300300006 45 69.0  9823 754.9 15530 995.6
300300008 46 90.7 12670 982.3 20380  1291
300300010 47  111 15320  1197 25040  1568
300300012 48  132 17700  1401 29510  1829

350350006 49 81.0 15820  1038 24820  1371
350350008 50  106 20510  1355 32660  1785
350350010 51  131 24920  1659 40250  2179
350350012 52  156 29050  1949 47600  2552
350350015 53  191 34750  2358 58120  3073

400400006 54 93.0 23850  1366 37230  1806
400400008 55  122 31050  1789 49070  2360
400400010 56  151 37870  2196 60620  2889
400400012 57  180 44320  2587 71840  3394
400400015 58  221 53330  3144 88050  4109
//
IF NRESP=0
%
% Size       |         Available thickness  t   mm
%
% 40 x  40   |2   3                                      Stainless
% 50 x  50   |2   3   4                                  Square
% 60 x  60   |2   3   4   5                              Hollow
% 80 x  80   |2   3   4   5                              Sections
%100 x 100   |    3   4   5   6
%125 x 125   |    3   4   5   6   8
%150 x 150   |    3   4   5   6   8
%175 x 175   |        4   5   6   8   10
%200 x 200   |        4   5   6   8   10
%250 x 250   |            5   6   8   10   12
%300 x 300   |            5   6   8   10   12
%350 x 350   |                6   8   10   12   15
%400 x 400   |                6   8   10   12   15
%
```

```
!Serial depth                     +sd(ssd1)=???? mm
!Serial breadth                   +sb(ssd1)=???? mm
!Thickness                        +st(ssd1)=???? mm
ENDIF
//
! +ru'=sd(ssd1)*10^6+sb(ssd1)*10^3+st(ssd1) +iu=0
IF ru'<40040002 OR ru'>400400015
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
REPEAT
! +iu=iu+1 +vu(iu)=TABLE(ssd1,ru',iu)
UNTIL iu=6
ENDREPEAT
IF vu(1)<>APR(INT(vu(1)))
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
.+sd(ssd1) x +sb(ssd1) x +st(ssd1) SHS
! +D=sd(ssd1) +B=sb(ssd1) +t=st(ssd1)
! +A=vu(2) +Ix=vu(3) +Sx=vu(4) +J=vu(5) +C=vu(6)
! +Iy=Ix +rx=(Ix/A)^0.5 +Zx=Ix/(D/20) +Sy=Sx +ry=rx +Zy=Zx
Properties (cm): A= +A ry= +ry Zx= +Zx Sx= +Sx Ix= +Ix J= +J C= +C
//
ENDDEFINE

DEFINE srdp ! Stainless steel RHS dimensions and properties
!Restructured sslist & sstab proformas
STORE srd1 72 8
!Stainless steel Rectangular Hollow Section properties
!                A     Ix    Sx     Iy    Sy     J     C
             1   2     3     4      5     6      7     8
50025001.5   1  2.06 6.408 3.235 2.192 2.005 5.562 3.108
50025002     2  2.67 7.946 4.088 2.699 2.528 7.063 3.875

60030002     3  3.27 14.42 6.103 4.919 3.780 12.62 5.837
60030003     4  4.65 19.08 8.351 6.441 5.150 17.33 7.800

80040002     5  4.47 36.24 11.33 12.44 7.034 31.06 10.96
80040003     6  6.45 49.73 15.91 16.92 9.847 43.95 15.13
80040004     7  8.27 60.30 19.79 20.36 12.21 54.77 18.49

100050002    8  5.67 73.25 18.16 25.24 11.29 61.97 17.68
100050003    9  8.25 102.5 25.88 35.07 16.04 88.99 24.86
100050004   10  10.7 127.1 32.71 43.19 20.22 113.0 31.00
100050005   11  12.9 147.2 38.66 49.70 23.84 133.7 36.11
100050006   12  15.0 162.9 43.75 54.72 26.91 150.9 40.23

150075003   13  12.8 370.8 61.29 127.8 38.10 313.7 59.68
150075004   14  16.7 472.3 78.99 161.9 49.02 406.5 76.24
150075005   15  20.4 563.2 95.35 192.2 59.06 492.8 91.21
150075006   16  24.0 643.6 110.4 218.6 68.26 572.1 104.6
150075008   17  30.7 774.2 136.5 260.9 84.13 708.0 126.9

150100003   18  14.3 451.8 72.31 243.7 54.98 510.5 81.26
150100004   19  18.7 578.9 93.59 311.6 71.11 666.1 104.6
150100005   20  22.9 694.6 113.5 373.0 86.16 813.5 126.1
150100006   21  27.0 799.2 132.0 428.4 100.2 952.2 145.9
150100008   22  34.7 976.0 164.9 521.3 125.0  1201 180.2
```

```
200100004   23  22.7  1171 145.3 403.8 90.31 991.5 141.5
200100005   24  27.9  1415 177.0 486.0 109.9  1213 171.3
200100006   25  33.0  1640 207.0 561.1 128.4  1423 198.9
200100008   26  42.7  2034 261.7 691.0 161.8  1808 248.0
200100010   27  51.7  2355 309.3 795.2 190.7  2139 288.9

200125004   28  24.7  1364 164.9 666.0 119.9  1449 179.8
200125005   29  30.4  1653 201.4 805.7 146.4  1781 218.7
200125006   30  36.0  1922 236.1 935.1 171.5  2099 255.2
200125008   31  46.7  2403 300.1  1164 217.7  2692 321.3
200125010   32  56.7  2807 356.8  1355 258.5  3223 378.4

250125006   33  42.0  3341 333.7  1147 207.2  2856 323.2
250125008   34  54.7  4211 426.8  1438 264.5  3673 409.0
250125010   35  66.7  4966 511.1  1686 316.0  4414 484.3
250125012   36  78.1  5607 586.6  1895 361.9  5072 549.5
250125015   37  93.8  6364 683.6  2137 420.5  5894 628.5

250150006   38  45.0  3787 370.3  1732 261.6  3911 394.5
250150008   39  58.7  4797 475.2  2187 335.3  5060 502.3
250150010   40  71.7  5686 571.1  2583 402.5  6121 598.9
250150012   41  84.1  6457 658.0  2925 463.3  7088 684.4
250150015   42   101  7400 771.7  3339 542.5  8346 792.4

300150006   43  51.0  5932 490.3  2044 304.8  5019 477.4
300150008   44  66.7  7557 631.9  2590 392.1  6504 609.9
300150010   45  81.7  9010 762.8  3074 472.5  7884 729.6
300150012   46  96.1 10300 883.1  3498 546.1  9153 836.9
300150015   47   116 11920  1043  4025 643.8 10830 974.9

300200006   48  57.0  7229 578.5  3899 439.8  8167 650.1
300200008   49  74.7  9262 748.7  4985 568.9 10660 836.8
300200010   50  91.7 11110 907.8  5968 689.3 13020  1009
300200012   51   108 12790  1055  6853 801.2 15240  1167
300200015   52   131 14970  1257  8003 953.4 18290  1378

350175006   53  60.0  9603 676.9  3315 421.1  8062 661.7
350175008   54  78.7 12320 877.1  4235 544.8 10500 850.8
350175010   55  96.7 14800  1064  5067 660.3 12800  1024
350175012   56   114 17050  1239  5814 767.7 14960  1184
350175015   57   138 20010  1479  6783 913.9 17900  1396

350200006   58  63.0 10490 728.5  4464 498.0 10130 763.0
350200008   59  82.7 13490 945.5  5722 645.7 13230 984.2
350200010   60   101 16250  1149  6872 784.3 16170  1189
350200012   61   120 18770  1341  7915 914.0 18960  1379
350200015   62   146 22110  1604  9289  1092 22820  1634

400200006   63  69.0 14540 893.6  5028 556.2 12140  875.9
400200008   64  90.7 18750  1162  6460 722.5 15860   1131
400200010   65   111 22650  1416  7775 879.3 19410   1370
400200012   66   132 26250  1656  8977  1026 22780   1591
400200015   67   161 31080  1989 10580  1230 27470   1891

400250006   68  75.0 16870  1011  8553 736.3 17680   1108
400250008   69  98.7 21820  1318 10660 959.2 23200   1438
400250010   70   121 26460  1611 12890  1171 28500   1749
400250012   71   144 30770  1889 14960  1371 33600   2041
400250015   72   176 36640  2278 17770  1652 40800   2445
IF NRESP=0
//
```

```
%
% Size          |        Available thickness  t    mm
%
% 50 x  25      |1.5   2                                   Stainless
% 60 x  30      |      2   3                               Rectangular
% 80 x  40      |      2   3   4                           Hollow
%100 x  50      |      2   3   4   5   6                   Sections
%150 x  75      |          3   4   5   6   8
%150 x 100      |          3   4   5   6   8
%200 x 100      |              4   5   6   8   10
%200 x 125      |              4   5   6   8   10
%250 x 125      |                      6   8   10   12   15
%250 x 150      |                      6   8   10   12   15
%300 x 150      |                      6   8   10   12   15
%300 x 200      |                      6   8   10   12   15
%350 x 175      |                      6   8   10   12   15
%350 x 200      |                      6   8   10   12   15
%400 x 200      |                      6   8   10   12   15
%400 x 250      |                      6   8   10   12   15
%
!Serial depth                         +sd(srd1)=???? mm
!Serial breadth                       +sb(srd1)=???? mm
!Thickness                            +st(srd1)=???? mm
ENDIF
//
IF sd(srd1)>sb(srd1)
! +ru'=sd(srd1)*10^6+sb(srd1)*10^3+st(srd1) +iu=0
ELSE
! +ru'=sb(srd1)*10^6+sd(srd1)*10^3+st(srd1) +iu=0
ENDIF
IF ru'<50025001.5 OR ru'>400250015
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
REPEAT
! +iu=iu+1 +vu(iu)=TABLE(srd1,ru',iu)
UNTIL iu=8
ENDREPEAT
IF vu(1)<>APR(INT(vu(1)))
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
.+sd(srd1) x +sb(srd1) x +st(srd1) RHS
! +D=sd(srd1) +B=sb(srd1) +t=st(srd1)
IF sd(srd1)>sb(srd1)
! +A=vu(2) +Ix=vu(3) +Sx=vu(4) +Iy=vu(5) +Sy=vu(6) +J=vu(7) +C=vu(8)
! +Zx=Ix/(D/20) +Zy=Iy/(B/20) +rx=(Ix/A)^0.5 +ry=(Iy/A)^0.5
Properties (cm): A= +A rx= +rx Zx= +Zx Sx= +Sx Ix= +Ix J= +J
                 C= +C Iy= +Iy Sy= +Sy Zy= +Zy ry= +ry
ELSE
! +A=vu(2) +Ix=vu(5) +Sx=vu(6) +Iy=vu(3) +Sy=vu(4) +J=vu(7) +C=vu(8)
! +Zx=Ix/(D/20) +Zy=Iy/(B/20) +rx=(Ix/A)^0.5 +ry=(Iy/A)^0.5
Properties (cm): A= +A rx= +rx Zx= +Zx Sx= +Sx Ix= +Ix
                 Iy= +Iy Sy= +Sy Zy= +Zy ry= +ry
ENDIF
//
ENDDEFINE
```

```
DEFINE ccdp ! Stainless steel CHS dimensions
//
IF NRESP=0
%------------------------------------------------------------------------------
% Diameter  |         Available thickness  t   mm
%------------------------------------------------------------------------------
%    21.3   |1.0  1.2  1.6  2.0  2.3                    Stainless
%    33.7   |1.0       1.6  2.0    2.5      3.2         Circular
%    42.4   |1.0       1.6  2.0      2.6    3.2         Hollow
%    48.3   |1.0       1.6  2.0      2.6    3.2         Sections
%    60.3   |1.0       1.6  2.0      2.6    3.2  4.0  5.0
%    76.1   |1.0       1.6  2.0      2.6    3.2  4.0  5.0
%    88.9   |1.0       1.6  2.0      2.6    3.2  4.0  5.0
%   101.6   |1.0       1.6  2.0      2.6    3.2  4.0  5.0
%   114.3   |     1.2  1.6  2.0      2.6    3.2  4.0  5.0
%   139.7   |     1.2  1.6  2.0      2.6    3.2  4.0  5.0
%   168.3   |          1.6  2.0      2.6    3.2  4.0  5.0
%   219.1   |               2.0      2.6    3.2  4.0  5.0
%   273     |                        2.6    3.2  4.0  5.0
!Serial depth                      +D=???? mm
!Thickness                         +t=???? mm
ENDIF
//
ENDDEFINE

DEFINE scdp ! Stainless steel channel dimensions and properties
!Restructured sslist & sstab proformas
STORE scd1 44 10                ! Stainless steel channels
!              A     Ix    Iy     Sx     Sy     J       H       xo  cy
            1  2      3     4      5      6     7       8        9  10
400150015   1 95.7 21100  1921   1283   312.7  71.75  418100  8.39 3.96
400150012   2 78.0 17640  1593   1060   254.2  37.45  367500  8.41 3.80
400150010   3 65.9 15130  1360   903.2  214.1  21.95  325500  8.41 3.70
400150008   4 53.3 12450  1114   737.9  173.1  11.38  276300  8.42 3.60

350125015   5 80.7 13160  1089   928.0  215.0  60.50  173100  6.82 3.35
350125012   6 66.0 11100  910.2  771.9  175.2  31.69  155200  6.84 3.19
350125010   7 55.9  9569  780.3  659.8  147.8  18.62  139100  6.86 3.09
350125008   8 45.3  7914  642.0  541.1  119.6  9.674  119500  6.87 2.99

300100015   9 65.7  7486  541.3  628.8  136.1  49.25   58820  5.25 2.74
300100012  10 54.0  6393  456.6  528.0  111.2  25.93   54340  5.29 2.59
300100010  11 45.9  5557  394.0  453.9  93.95  15.28   49610  5.31 2.48
300100008  12 37.3  4631  326.3  374.4  76.15  7.967   43340  5.32 2.38

250100012  13 48.0  4079  429.2  400.4  108.3  23.05   33690  5.73 2.83
250100010  14 40.9  3563  371.5  345.5  91.94  13.62   31090  5.75 2.73
250100008  15 33.3  2984  308.4  286.0  74.88  7.114   27430  5.76 2.62
250100006  16 25.5  2340  239.9  221.7  57.14  3.061   22600  5.77 2.52

225075012  17 39.0  2470  179.5  278.3  60.81  18.73   10620  3.93 2.10
225075010  18 33.4  2184  156.8  242.3  51.65  11.12   10090  3.96 1.99
225075008  19 27.3  1849  131.4  202.2  42.09  5.834    9139  3.97 1.89
225075006  20 21.0  1465  103.2  157.9  32.13  2.521    7713  3.99 1.79

200075010  21 30.9  1629  150.6  202.1  50.66  10.28    7379  4.17 2.11
200075008  22 25.3  1385  126.5  169.2  41.46  5.407    6746  4.19 2.01
200075006  23 19.5  1102  99.62  132.6  31.78  2.341    5742  4.20 1.90
200075005  24 16.5 945.6  85.02  112.9  26.76  1.372    5085  4.21 1.85
```

```
175060010  25 25.4 966.5  74.91  140.4  32.08  8.451    2594  3.18 1.72
175060008  26 20.9 834.0  63.62  118.8  26.32  4.469    2455  3.21 1.62
175060006  27 16.2 672.2  50.62  93.96  20.22  1.945    2153  3.22 1.51
175060005  28 13.7 580.0  43.43  80.35  17.05  1.143    1934  3.23 1.46

150060008  29 18.9 568.8  60.25  93.82  25.66  4.042    1627  3.43 1.74
150060006  30 14.7 461.9  48.14  74.63  19.86  1.765    1450  3.45 1.64
150060005  31 12.5 399.9  41.38  63.99  16.80  1.039    1311  3.45 1.58
150060004  32 10.1 332.1  34.13  52.64  13.64  0.5406   1136  3.46 1.53

125050006  33 12.0 255.0  26.83  50.05  13.54  1.441    526.5 2.86 1.42
125050005  34 10.2 222.7  23.22  43.19  11.49  0.8511   485.8 2.87 1.36
125050004  35 8.34 186.5  19.28  35.75  9.360  0.4446   428.6 2.88 1.31
125050003  36 6.38 146.3  15.00  27.72  7.143  0.1913   353.2 2.88 1.26

100050005  37 8.96 130.1  21.43  31.21  11.17  0.7470   272.8 3.13 1.52
100050004  38 7.34 109.7  17.86  25.95  9.159  0.3913   244.0 3.14 1.46
100050003  39 5.63 86.56  13.94  20.21  7.033  0.1688   203.6 3.14 1.41

75035005   40 6.21 47.00  6.901  15.53  5.254  0.5178   42.81 2.12 1.11
75035004   41 5.14 40.59  5.860  13.15  4.351  0.2740   40.52 2.13 1.06
75035003   42 3.98 32.76  4.656  10.41  3.371  0.1193   35.54 2.14 1.00

50025003   43 2.63 9.239  1.540  4.497  1.633 0.07881   4.550 1.56 0.788
50025002   44 1.83 6.855  1.116  3.244  1.145 0.02446   3.813 1.57 0.731
//
%
% Size        |         Available thickness  t   mm
%
% 50 x  25    |2   3                                   Stainless
% 75 x  35    |    3   4   5                           Steel
%100 x  50    |    3   4   5                           Channels
%125 x  50    |    3   4   5   6
%150 x  60    |        4   5   6   8
%175 x  60    |            5   6   8   10
%200 x  75    |            5   6   8   10
%225 x  75    |                6   8   10   12
%250 x 100    |                6   8   10   12
%300 x 100    |                    8   10   12   15
%350 x 125    |                    8   10   12   15
%400 x 150    |                    8   10   12   15
%
!Serial depth                     +sd(scd1)=???? mm
!Serial breadth                   +sb(scd1)=???? mm
!Thickness                        +st(scd1)=???? mm
//
! +ru'=sd(scd1)*10^6+sb(scd1)*10^3+st(scd1) +iu=0
IF ru'<50025002 OR ru'>400150015
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
REPEAT
! +iu=iu+1 +vu(iu)=TABLE(scd1,ru',iu)
UNTIL iu=10
ENDREPEAT
IF vu(1)<>APR(INT(vu(1)))
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
.+sd(scd1) x +sb(scd1) x +st(scd1) Channel
! +D=sd(scd1) +B=sb(scd1) +t=st(scd1)
! +A=vu(2) +Ix=vu(3) +Iy=vu(4) +Sx=vu(5) +Sy=vu(6) +J=vu(7) +H=vu(8)
```

```
! +xo=vu(9) +cy=vu(10)
Properties (cm): A= +A Sx= +Sx Ix= +Ix Iy= +Iy Sy= +Sy J= +J
                 H= +H xo= +xo cy= +cy
//
ENDDEFINE

DEFINE eadp !Stainless steel equal angle
STORE ead1 24 9  ! Table 6 stainless steel equal angle
!              A   Ix    Iu    Iv     J     H     c   uo
           1   2    3     4     5     6     7     8   9
50040005   1 4.48 10.71 17.86 3.563 0.374 0.576 1.52 1.65
50040006   2 5.25 12.32 20.75 3.896 0.630 0.904 1.58 1.59
50040008   3 6.67 15.04 25.83 4.241 1.424 1.733 1.70 1.46
50040010   4 7.93 17.11 29.97 4.243 2.642 2.644 1.82 1.29

75075006   5 8.25 45.21 74.68 15.74 0.990 3.676 2.19 2.54
75075008   6 10.7 57.05 95.43 18.66 2.277 7.714 2.31 2.45
75075010   7 12.9 67.33 114.1 20.53 4.309 13.20 2.42 2.33
75075012   8 15.0 76.13 130.8 21.47 7.207 19.74 2.54 2.19

100100008  9 14.7 142.9 236.0 49.76 3.130 20.66 2.92 3.38
100100010 10 17.9 171.4 285.8 57.00 5.976 36.85 3.04 3.29
100100012 11 21.0 197.2 332.0 62.33 10.09 57.85 3.15 3.19
100100015 12 25.3 230.7 394.4 67.06 19.00 96.66 3.33 2.99

120120008 13 17.9 253.6 416.4 90.84 3.813 37.82 3.42 4.12
120120010 14 21.9 306.6 507.1 106.0 7.309 68.70 3.53 4.04
120120012 15 25.8 355.4 592.6 118.2 12.39 110.0 3.64 3.95
120120015 16 31.3 421.3 711.1 131.4 23.50 190.4 3.82 3.79

150150008 17 22.7 508.6 830.2 187.0 4.837 78.09 4.17 5.21
150150010 18 27.9 619.2  1016 221.8 9.309 144.3 4.28 5.15
150150012 19 33.0 723.4  1194 251.9 15.85 235.3 4.39 5.08
150150015 20 40.3 867.7  1446 288.6 30.25 419.8 4.56 4.94

200200008 21 30.7  1237  2008 465.9 6.544 195.4 5.42 7.00
200200010 22 37.9  1516  2472 561.1 12.64 366.5 5.52 6.96
200200012 23 45.0  1784  2921 647.8 21.61 607.7 5.63 6.90
200200015 24 55.3  2165  3568 762.2 41.50  1113 5.79 6.81
//
%
%
% Size     |         Available thickness  t   mm
%
% 50 x  50 |    5   6   8   10                        Stainless
% 75 x  75 |        6   8   10    12                  Equal Angle
%100 x 100 |            8   10    12    15            Section
%120 x 120 |            8   10    12    15
%150 x 150 |            8   10    12    15
%200 x 200 |            8   10    12    15
%
!Serial size                        +sd(ead1)=???? mm
!Thickness                          +st(ead1)=???? mm
//
! +ru'=sd(ead1)*10^6+sd(ead1)*10^3+st(ead1) +iu=0
IF ru'<50050005 OR ru'>200200015
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
```

```
REPEAT
! +iu=iu+1 +vu(iu)=TABLE(ead1,ru',iu)
UNTIL iu=9
ENDREPEAT
IF vu(1)<>APR(INT(vu(1)))
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
.+sd(ead1) x +sd(ead1) x +st(ead1) equal angle
! +D=sd(ead1) +B=sd(ead1) +t=st(ead1)
! +A=vu(2) +Ix=vu(3) +Iu=vu(4) +Iv=vu(5) +J=vu(6) +H=vu(7) +c=vu(8)
! +uo=vu(9)
Properties (cm): A= +A Ix= +Ix Iu= +Iu Iv= +Iv J= +J H= +H
                 c= +c uo= +uo
//
ENDDEFINE
DEFINE dadp ! Stainless steel back to back double angles
STORE dad1 24 9
!              A   Ix    Iy     J     H     cy   xo rv
           1   2    3     4     5     6     7    8   9
100050005  1 8.96 42.10 21.43 0.747 1.152 1.52 1.16  0.892
100050006  2 10.5 50.75 24.64 1.261 1.808 1.58 1.13  0.861
100050008  3 13.3 68.46 30.07 2.847 3.467 1.70 1.03  0.797
100050010  4 15.9 86.88 34.21 5.285 5.288 1.82 0.912 0.732

150075006  5 16.5 169.9 90.42 1.981 7.353 2.19 1.79 1.38
150075008  6 21.3 227.6 114.1 4.554 15.43 2.31 1.73 1.32
150075010  7 25.9 286.5 134.7 8.618 26.41 2.42 1.65 1.26
150075012  8 30.0 346.6 152.3 14.41 39.49 2.54 1.55 1.20

200100008  9 29.3 536.8 285.8 6.261 41.31 2.92 2.39 1.84
200100010 10 35.9 673.5 342.8 11.95 73.70 3.04 2.33 1.78
200100012 11 42.0 812.0 394.3 20.17 115.7 3.15 2.25 1.72
200100015 12 50.7  1023 461.4 38.00 193.3 3.33 2.11 1.63

240120008 13 35.7 925.8 507.3 7.626 75.63 3.42 2.91 2.25
240120010 14 43.9  1160 613.1 14.62 137.4 3.53 2.86 2.20
240120012 15 51.6  1396 710.8 24.78 220.1 3.64 2.80 2.14
240120015 16 62.7  1756 842.6 47.00 380.9 3.82 2.68 2.05

300150008 17 45.3  1805  1017 9.674 156.2 4.17 3.68 2.87
300150010 18 55.9  2260  1238 18.62 288.5 4.28 3.64 2.82
300150012 19 66.0  2717  1446 31.69 470.6 4.39 3.59 2.76
300150015 20 80.7  3409  1735 60.50 839.5 4.56 3.49 2.67

400200008 21 61.3  4273  2474 13.09 390.8 5.42 4.95 3.90
400200010 22 75.9  5346  3033 25.28 733.0 5.52 4.92 3.85
400200012 23 90.0  6423  3569 43.21  1215 5.63 4.88 3.79
400200015 24  110  8046  4330 83.00  2226 5.79 4.81 3.71
//
%
% Size      |         Available thickness  t   mm
%
%100 x  50  |    5   6   8   10                      Stainless
%150 x  75  |        6   8   10    12                Double Angle
%200 x 100  |            8   10    12    15          Back to Back
%240 x 120  |            8   10    12    15          Section
%300 x 150  |            8   10    12    15
%400 x 200  |            8   10    12    15
!Serial width                     +sd(dad1)=???? mm
!Serial depth                     +sb(dad1)=???? mm
!Thickness                        +st(dad1)=???? mm
```

```
//
! +ru'=sd(dad1)*10^6+sb(dad1)*10^3+st(dad1) +iu=0
IF ru'<100050005 OR ru'>400200015
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
REPEAT
! +iu=iu+1 +vu(iu)=TABLE(dad1,ru',iu)
UNTIL iu=9
ENDREPEAT
IF vu(1)<>APR(INT(vu(1)))
!Section unknown; press < and Enter to revise, Enter to stop +$27569=?
STOP
ENDIF
.+sd(dad1) x +sb(dad1) x +st(dad1) double angle
! +D=sd(dad1) +B=sb(dad1) +t=st(dad1)
! +A=vu(2) +Ix=vu(3) +Iy=vu(4) +J=vu(5) +H=vu(6) +cy=vu(7) +xo=vu(8)
! +rv=vu(9)
Properties (cm): A= +A Ix= +Ix Iy= +Iy J= +J H= +H cy= +cy xo= +xo
                 rv= +rv (minimum radius of gyration of single angle)
//
ENDDEFINE
DEFINE selfch
! +run=0
IF EXIST cc924.stk
#cc924.stk ! Imports run No. which refers to set of parameters.
ENDIF
! Clear string for holding leading zeros in name extension +$924=
IF run<100 THEN +$923=0
IF run<10 THEN +$923=00
! +$32000=nl924. +$923 +run
! Save filename nl924.<run> in +$924= +$32000
! +$32000=public. +$923 +run
! Save filename public.<run> in +$925= +$32000
DOS del +$924 ! Clear old file in range nl924.001 to nl924.996.
DOS del +$925 ! Clear old file in range public.001 to public.996.
DOS del nl924.stk  ! Clear file used for piping data to NL-STRESS.
DOS del public.stk ! Clear file used for importing summary of check.
! +$701=plastic. +$702=compact. +$703=semi-compact. +$704=slender.
                 N.B. Section is classed as +$(700+class)
FILE +$924 ! Set nl924.001 etc. as file for piped calculations.
%run= +run
! +$94=wy= +8*Mz/L^2/1.5 wz= +8*My/L^2/1.5 fx= +-F/1.5 lx= +L
%+$94
! +r=B/D
IF sd(tbn)>=sb(tbn)
! +na=4
! +$94=c= +D/1E3 a= +B/1E3 t= +t/1E3 e= +E*1E6 nu=0.3
ELSE
! +na=6
! +$94=c= +B/1E3 a= +D/1E3 t= +t/1E3 e= +E*1E6 nu=0.3
ENDIF
%+$94
! +nc=2*INT(0.5*na/r+0.5)
! +$94=yield= +py*1E3 nli=1 na= +na nc= +nc warp=0
%+$94
! +$94=ecy=.025 ecz=.025
%+$94
! Default is top & bottom flanges laterally restrained at supports
! which provides nominal torsional restraint at ends, Table 13.
IF refno>0 AND refno<11
! +$94=tlr= +L blr= +L
ENDIF
```

```
IF refno=11
! +$94=tlr=0 blr=0
ENDIF
%+$94
IF refno=1 OR refno=2
! +$94=trp= +L brp= +L
ENDIF
IF refno=3 OR refno=4
! +$94=trp= +L brp=0
ENDIF
IF refno>4 AND refno<9
! +$94=trp= +L brp=0
ENDIF
IF refno=9 OR refno=10
! +$94=trp=0 brp=0
ENDIF
%+$94
%| 1 run       ! Add to end to display run number at top of screen.
FILE fil.nam   ! Copy the filename sc385.dat to the file fil.nam;
%sc385.dat/b   ! switch /b causes file to be run in batch mode.
FILE           ! Closes the current 'piped' file.
DOS copy +$924 nl924.stk ! Copy nl924.001 etc. for import by NLS.
DOS copy fil.nam fil.sav ! Saves current page headings file.
WIN nls32.exe  ! Run NL-STRESS which writes summary to public.stk.
#public.stk    ! Imports summary written by NL-STRESS.
DOS copy fil.sav fil.nam  ! Reinstates page heading file.
ENDDEFINE
FINISH
```

Calls NL-STRESS which runs the model sc385.dat in batch mode.
The model sc385.dat follows, in which

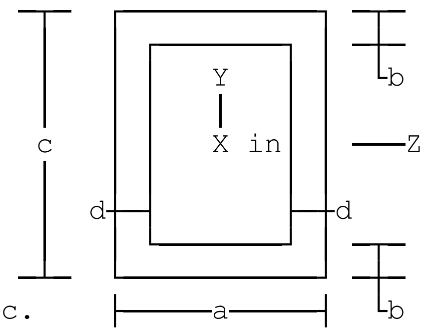
- #nl924.stk imports sets of parameters from nl924.stk which have been generated by the procedure or proforma calculation called sc924.pro when running the proforma calculation sc385.pro which
- invokes NL-STRESS to carry out the self check modelling the rectangular hollow section by finite elements
- sc385.dat below pipes the following four lines to public.stk

```
Run +run        Max. comb. axial & bend. stress +maxs kN/m2
Self check at   Dist. from start to max. stress +diss m
working load.   Maximum combined displacement   +maxd m
                Dist. from start to max. displ. +disd m
```

- when maxs>yield then pipes the following warning to public.stk
  *** Combined stress= +maxs exceeds yield stress= +yield
- the file public.stk is imported by proforma calculation sc385.pro and included in the calculations for the engineer to consider.

```
! This verified model is parametric.  General notes follow FINISH.
! A semi-colon preceded by a space is used to separate statements.
! Parameters precede '!' help which follows, is copied to results.
! VEC (short for VECtor, a rank-one array), is used for multiple
! assignments: a(7)=VEC(3.2,b,-5.7) ≡ a(7)=3.2, a(8)=b, a(9)=-5.7;
! cs(1)=VEC(12,2.8)*2 ≡ cs(1)=12, cs(2)=2.8, cs(3)=12, cs(4)=2.8.

TITLE RECTANGULAR HOLLOW SECTION SIMPLY SUPPORTED BEAM
TITLE SUBJECTED TO AXIAL LOAD, BIAXIAL BENDING, TORQUE
TITLE INCLUDING CHECKS FOR: COMPATIBILITY, LOCAL & OVERALL
TITLE EQUILIBRIUM, & THAT STRAIN ENERGY EQUALS WORK DONE.
MADEBY DWB ;DATE 17.06.06 ;REFNO VM645 ;TYPE SPACE FRAME run=0
```

```
a=.100       ! Breadth of section.
b=.008       ! Thickness of flange.
c=.150       ! Depth of section.
d=.008       ! Thickness of webs.
lx=1.2       ! Length of RHS.
e=205E6      ! Young's modulus.
nu=.3        ! Poisson's ratio.
na=4         ! Even No. of elems in 'a'.
nc=6         ! Even No. of elems in 'c'.
nli=1        ! No. of load incs. 1=elastic.
warp=1 yield=220E3 ! Warping prevented at ends (1=yes, 0=no).
fx=-80  wy=-10  wz=6  ! Axial force & udls in Y  & Z directions.
ecy=.025  ecz=.025 ! Eccentricities for wy & wz loads (+ or -).
tlr=lx  blr=lx   ! Top & bottom lateral restraint spacing, 0=none.
trp=0   brp=0    ! T & b rotn. on plan restraint spacing, 0=none.
#cc924.stk !Import set of parameters if available from cc924.stk.
TABULATE ALL ;PRINT DATA, RESULTS FROM 1 LENGTH gen
NUMBER OF INCREMENTS nli ;IF nli=1 ;METHOD ELASTIC JOINTS ;ENDIF
IF nli>1 THEN METHOD SWAY JOINTS
esa=(a-d)/na    esc=(c-b)/nc       !┐ Element sizes along a & c
eav=(esa+esc)/2 nx=INT(lx/eav+.5)  !┘ & number along the length.
njx=nx+1 njz=na+1 a'=(a-d)/2 c'=(c-b)/2 nst=(na+nc)*2 nel=nst*nx
IF nel<=5300 GOTO 8 !Adjust to keep mesh within sensible limits.
na=na+2 ;:5 ;na=na-2 nc=INT(na*(c-b)/(a-d)+.5) esa=(a-d)/na
esc=(c-b)/nc eav=(esa+esc)/2 nx=INT(lx/eav+.5) njx=nx+1 njz=na+1
a'=(a-d)/2 c'=(c-b)/2 nst=(na+nc)*2 nel=nst*nx
IF na>=4 AND nel>5300 GOTO 5 ;:8
NUMBER OF JOINTS nj=(na+nc)*2*njx nj
NUMBER OF MEMBERS nm=nel*6+2*nst nm ;NUMBER OF SUPPORTS 0
! Supports given in joint releases table. ;NUMBER OF LOADINGS 1
JOINT COORDINATES ;* First joints in top and bottom flanges.
nr=0 j=0 xinc=lx/nx x=-xinc njs=nj/njx  !Initialise values. ;:10
nr=nr+1 x=x+xinc j=njs*(nr-1) j+1 THRU j+njz X x Y c' Z -a' ZL a'
j+nc+njz THRU j+nc+njz+na X x Y -c' Z a' ZL -a' ;IF nr<njx GOTO 10
* Next joints in the webs. ;nr=0 j=0 x=-xinc y'=c'-(c-b)/nc
! Initialise counters & constants. ;:30 ;nr=nr+1 x=x+xinc
j=njs*(nr-1) j+njz+1 THRU j+njz+nc-1 X x Z a' Y y' YL -y'
j+njs-nc+2 THRU j+njs X x Z -a' Y -y' YL y' ;IF nr<njx GOTO 30
JOINT RELEASES ;s=nj-njs+na+1 r=njz+nc+na u=nj-njs+1 v=nj-nc+1
! First fixities as for s.s. beam. ;r-na/2 FORCE X -1 FORCE Z -1
v-na/2 FORCE Z -1 ;r-na THRU r FORCE Y -1 ;v-na THRU v FORCE Y -1
! Next additional: t & b lateral & rotational on plan restraints.
IF tlr=0 GOTO 45 ;x=-tlr ;:40 ;x=x+tlr eln=INT(nx*x/lx+0.5)
j=1+na/2+eln*nst j FORCE Z -1 ;IF x<lx GOTO 40 ;:45
IF blr=0 GOTO 55 ;x=-blr ;:50 ;x=x+blr eln=INT(nx*x/lx+0.5)
j=r-na/2+eln*nst ;IF j<>r-na/2 AND j<>v-na/2 THEN j FORCE Z -1
IF x<lx GOTO 50 ;:55 ;IF trp=0 GOTO 65 ;x=-trp ;:60 ;x=x+trp
eln=INT(nx*x/lx+0.5) 1+eln*nst THRU njz+eln*nst MOMENT Y -1
IF x<lx GOTO 60 ;:65 ;IF brp=0 GOTO 75 ;x=-brp ;:70 ;x=x+brp
eln=INT(nx*x/lx+0.5) r-na+eln*nst THRU r+eln*nst MOMENT Y -1
IF x<lx GOTO 70 ;:75 ;*/13
```



```
*                      ├──── +a ────┤
*      +b              1            +njz          +u          +s
*
*
*
*
*  +c                                  +d
*
*
*
*
*      +b              +r           +r-na         +v          +v-na
*               START JOINT NUMBERS        END JOINT NUMBERS ;*
MEMBER INCIDENCES
nr=0 nes=2*(na+nc) m=-nes*6 j=-njs ;:100 ;nr=nr+1 m=m+6*nes
j=j+njs m+1 THRU m+6*(nes-1) ELEMENT j+njs+1,j+njs+2,j+2,j+1
m+6*(nes-1)+1 THRU m+6*(nes-1)+6 ELEMENT j+2*njs,j+njs+1,j+1,j+njs
IF nr<nx GOTO 100 ;nm-2*nst+1 THRU nm-nst-1 RANGE 1 2 nst-1 nst
nm-nst nst 1 ;nm-nst+1 THRU nm-1 RANGE nj-njs+1 nj-njs+2 nj-1 nj
nm nj nj-njs+1 ;CONSTANTS E e ALL G g=e/(2*(1+nu)) g ALL
MEMBER PROPERTIES ;nr=0 nes=2*(na+nc) m=-nes*6 j=-njs ;:200
nr=nr+1 m=m+6*nes m+1 THRU m+6*na ELEMENT T b
m+6*na+1 THRU m+6*(na+nc) ELEMENT T d
m+6*(na+nc)+1 THRU m+6*(2*na+nc) ELEMENT T b
m+6*(2*na+nc)+1 THRU m+6*(2*na+2*nc) ELEMENT T d
IF nr<nx GOTO 200 ;! Next end stiffeners of 4 times average thick.
nm-2*nst+1 THRU nm-nst RECTANGLE D 4*(d+b)/2*warp+1E-9
nm-nst+1 THRU nm RECTANGLE D 4*(d+b)/2*warp+1E-9 !Warp is on/off.
LOADING CASE 1 ;JOINT LOADS ;! Vert. loads with eccentricity 'ec'.
w=wy*xinc/2 1 THRU u STEP nst FORCE Y w-w*ecy/((a-d)/2)
1 THRU u STEP u-1 FORCE Y -(w-w*ecy/((a-d)/2))/2
njz THRU s STEP nst FORCE Y w+w*ecy/((a-d)/2)
njz THRU s STEP s-njz FORCE Y -(w+w*ecy/((a-d)/2))/2
w=wz*xinc/2 1 THRU u STEP nst FORCE Z w+w*ecz/((c-b)/2)
1 THRU u STEP u-1 FORCE Z -(w+w*ecz/((c-b)/2))/2
r THRU v STEP nst FORCE Z w-w*ecz/((c-b)/2)
r THRU v STEP v-r FORCE Z -(w-w*ecz/((c-b)/2))/2 ;SOLVE
nli=ARR(12,4,1) lli=ARR(12,4,2) ;IF lli<nli THEN nli=lli ;m=0
maxs=0 mem=0 ;:300 ;m=m+1 ;rn=(nli-1)*nm+m ;fax=ARR(13,rn,13)
fby=ARR(13,rn,17) fbz=ARR(13,rn,18) fab=ABS(fax)+ABS(fby)+ABS(fbz)
IF fab>maxs THEN maxs=fab mem=m ;IF m<nm GOTO 300 ;n=ARR(1,mem,2)
diss=ARR(8,n,3) j=0 maxd=0 jnt=0 ;:310 ;j=j+1 n=ARR(8,j,2)
row=6*(n-1)+1 dx=ARR(6,row,nli) row=row+1 dy=ARR(6,row,nli)
row=row+1 dz=ARR(6,row,nli) dab=SQR(dx^2+dy^2+dz^2)
IF dab>maxd THEN maxd=dab jnt=j ;IF j<nj GOTO 310 ;n=ARR(8,jnt,2)
disd=ARR(8,n,3) ;! '*>' pipes summary to 'public.stk'. ;*/5
* Maximum combined axial & bend. stress +maxs         member +mem
* Distance from start to maximum stress +diss
* Maximum combined displacements        +maxd         joint  +jnt
* Distance from start to maximum displ. +disd ;IF maxs>yield
* Combined stress= +maxs exceeds yield stress= +yield ;ENDIF
IF run>0
*>Run No. +run     ┌ Max. comb. axial & bend. stress +maxs kN/m2
*>Self check at    │ Dist. from start to max. stress +diss m
*>working load.    │ Maximum combined displacement   +maxd m
*>                 └ Dist. from start to max. displ. +disd m
ENDIF ;IF run>0 AND maxs>yield
*>*** Combined stress= +maxs exceeds yield stress= +yield
ENDIF ;< ;FINISH
```

▪ 1.11 GENERAL NOTES

PARAMETER

No.	Name	Start	End	Type	Dependency conditions	
1	a	0.1	0.9	9		
2	b	0.01	0.09	9	>a/20	<a/10
3	c	0.14	1.26	9	>a	<3*a
4	d	0.01	0.09	9	>c/20	<b
5	lx	0.6	6	10	>3*c	<30*c
6	e	10E6	205E6	0		
7	nu	0.33	0.33	3		
8	na	4	4	0	Type=200 ≡ divisible by two.	
9	nc	4	8	200	=INT(na*c/a+0.5)	
10	nli	1	1	2	Roark's treatment	
11	warp	1	1	2	is for no warping.	
12	fy	1	10	0		
13	fz	1	10	0		
14	mx	1	10	0		

NOTES ON THIS VERIFIED MODEL

Verification is by comparison with displacements computed using Roark's Formulas for Stress and Strain, Fourth Edition 1965.

The figure below shows the topology assuming na=4 & nc=6 (default settings for these parameters). A summary of the joint numbering (which corresponds to the parameters na & nc set by the engineer) follows the results.

```
─┬─      ─┬─       1─────21─────41─────61─────81────101─  ─1021
 │ Top    │        │      │      │      │      │      │       │
 │ flange │        2─────22─────42─────62─────82────102─  ─1022
 │        │        │      │      │      │      │      │       │
a-d     na elem.   3─────23─────43─────63─────83────103─  ─1023
 │        │        │      │      │      │      │      │       │
 │        │        4─────24─────44─────64─────84────104─  ─1024
 │        │        │      │      │      │      │      │       │
─┴─      ─┴─       5─────25─────45─────65─────85────105─  ─1025

─┬─      ─┬─       5─────25─────45─────65─────85────105─  ─1025
 │  Top   │        │      │      │      │      │      │       │
 │  of    │        6─────26─────46─────66─────86────106─  ─1026
 │  right │        │      │      │      │      │      │       │
 │  web   │        7─────27─────47─────67─────87────107─  ─1027
 │        │        │      │      │      │      │      │       │
c-b     nc elem.   8─────28─────48─────68─────88────108─  ─1028 ──X
 │        │        │      │      │      │      │      │       │
 │ Bottom │        9─────29─────49─────69─────89────109─  ─1029
 │  of    │        │      │      │      │      │      │       │
 │  right │       10─────30─────50─────70─────90────110─  ─1030
 │  web   │        │      │      │      │      │      │       │
─┴─      ─┴─      11─────31─────51─────71─────91────111─  ─1031

─┬─      ─┬─      11─────31─────51─────71─────91────111─  ─1031
 │ Bottom │        │      │      │      │      │      │       │
 │ flange │       12─────32─────52─────72─────92────112─  ─1032
 │        │        │      │      │      │      │      │       │
a-d     na elem.  13─────33─────53─────73─────93────113─  ─1033
 │        │        │      │      │      │      │      │       │
 │        │       14─────34─────54─────74─────94────114─  ─1034
 │        │        │      │      │      │      │      │       │
─┴─      ─┴─      15─────35─────55─────75─────95────115─  ─1035
```

```
──┬──       ──┬──     15──────35──────55──────75──────95─────115─  ─1035
  │ Bottom    │        │       │       │       │       │       │     │
  │  of       │       16──────36──────56──────76──────96─────116─  ─1036
  │  left     │        │       │       │       │       │       │     │
  │  web      │       17──────37──────57──────77──────97─────117─  ─1037
  │           │        │       │       │       │       │       │     │
c-b      nc elem.     18──────38──────58──────78──────98─────118─  ─1038 ──X
  │           │        │       │       │       │       │       │     │
  │  Top      │       19──────39──────59──────79──────99─────119─  ─1039
  │  of       │        │       │       │       │       │       │     │
  │  left     │       20──────40──────60──────80─────100─────120─  ─1040
  │  web      │        │       │       │       │       │       │     │
──┴──       ──┴──      1──────21──────41──────61──────81─────101─  ─1021

                       ├────────────────nx elements────────────────┤
                      Z├────────────────lx=length──────────────────┤
```

It is permissible to have an assignment on any line of NL-STRESS data but an assignment does not contribute an item of data; thus if it is intended that the value assigned be an item of data, then the name of the variable used should be given after its assignment. A semi-colon is used to separate statements when more than one line. Type E or click Edit to change the page headings and parameters. When in Edit mode, type Ctrl+H for Help; press any key to clear Help.

Adding general loading, intermediate supports and stiffeners is easy once a plot has been produced showing the joint numbers for the number of elements chosen. For simplicity, it is suggested that stiffeners be modelled by stiff bars. The free end has a stiffener already modelled, the can be rendered effective or ineffective by setting warp to 1 or 0 respectively.

If an additional support is required at say joints 91 to 95 in the Y direction, then the following should be added to the JOINT RELEASES table:
91 THRU 95 FORCE Y -1

Such changes as that described above, take minutes rather than hours when starting anew.

Before embarking on verification it is prudent to try different meshes viz:

Meshing na,nc	1,2	2,3	3,5	4,6	5,8	6,9
%age difference	14%	5%	1.7%	1%	1%	1%

There is no advantage in increasing the mesh above 4,6. For:

```
esa=(a-d)/na    esc=(c-b)/nc        !┐ Element sizes along a & c
eav=(esa+esc)/2 nx=INT(lx/eav+.5)   !┘ & number along the length.
```

extracted from the data, then with na=4 & nc=6 there are 20 joints around the RHS, using the default data then the average element size is found thus: esa=(.1-.008)/4=0.023 esc=(.15-.008)/6=.02366 thus eav=(0.023+0.02366)/2 =0.023333

For say 32000 members, i.e. 5300 elements, the number of elements in the length =5300/20 =265 i.e. a length of 0.02333*265 ≈6.2m. This gives a span:width ratio of the RHS of 6.2/0.1 =62 which should be slender enough for most checking, so limit the number of elements to say 5300 and adjust 'nx' accordingly, as in the data.

CONCLUSIONS

Roark's formulas ignore Poisson's ratio. Generally, when Poisson's ratio is ignored, it is assumed to be equal to 1/3 as appropriate for steel, accordingly this value was used in the data.

End displacements and rotation for the rectangular structural hollow section analysed using NL-STRESS & computed using Roark's formulas agree to an average percentage difference of 0.2440% with a maximum percentage difference of 2.3333% in run 633, when tested for 996 runs using the data given in the PARAMETER table.