LATERALLY LOADED WOOD COMPRESSION MEMBERS: FINITE ELEMENT AND RELIABILITY ANALYSIS

by

EXAUD NOE KOKA

B.Sc.Eng.(Hons).University Of DSM , TANZANIA,1984

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Department of Civil Engineering

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

October 1987

4 6

Department of Civil Engineering

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date: October 1987

ABSTRACT

This thesis consists of two parts. The first part describes the analysis and implementation of a finite element computer model for the general prediction of failure of wood members in bending or in combined bending and axial compression. Both instability and material strength failures are included. The program is verified using available analytical and test results. A good agreement with the results predicted by this program is observed.

The second part describes a procedure for the structural reliability evaluation of a compression member assuming random loads and material variables. The program developed here for the reliability study links the finite element program and the Rackwitz-Fiessler algorithm for the calculation of the reliability index β. The gradient of the failure function, which is a necessary input to the Rackwitz-Fiessler algorithm, is computed numerically using the finite element routine. The results of the reliability study for a typical column problem are compared against the available results obtained by following the code procedures [as outlined in CAN3-086.1-M84 (1984)] for different slenderness ratios.

A performance factor $\phi_p = 0.75$, for compression members of any length is recommended in order to obtain a more accurate and consistent level of reliability in the design

process. It is estimated that if this factor $\phi_p = 0.75$ is adopted in the current design practices, a level of reliability index of the order of 4.0 can be achieved.

TABLE OF CONTENTS

ABSTRACT ii

LIST OF FIGURES vi

LIST OF TABLES viii

NOTATION ix

Acknowledgment xi

1. INTRODUCTION 1
   1.1. Background 1
   1.2. Objectives 3
   1.3. Thesis Organisation 3

2. CURRENT CODE REQUIREMENTS AND PREVIOUS RESEARCH WORK .. 5
   2.1. Introduction 5
   2.2. Current Code Requirements 5
      2.2.1. Concentric Compression 5
      2.2.2. Combined bending and Compression 7
   2.3. Previous Research on Wood compression members . 8

3. FINITE ELEMENT ANALYSIS 10
   3.1. Introduction 10
   3.2. Assumptions 10
   3.3. Stress strain relationship 11
   3.4. Finite Element Approximation 17
      3.4.1. Introduction 17
      3.4.2. Kinematic Assumptions 17
      3.4.3. Problem formulation 20
      3.4.4. Interpolation Functions 22
      3.4.5. Strain Displacement Relations 26
      3.4.6. Virtual Work Equations 27
      3.4.7. Newton-Raphson Method 28
      3.4.8. Computation procedure 31
      3.4.9. Numerical Integration 33
   3.5. Convergence criterion for solution vector 34
   3.6. Obtaining the ultimate load Pmax 35
   3.7. Failure Criterion and Size effects 39
      3.7.1. Size effect in compression 40
      3.7.2. Size effect in tension 41
   3.8. Program Structure 44
   3.9. Discussion 44

4. PROGRAM VERIFICATION 46
   4.1. Introduction 46
   4.2. Comparison of results 46
      4.2.1. Presentation of Results 47

4.3. Discussion 53

5. APPLICATIONS 55
5.1. Introduction 55
5.2. Numerical example 56
5.3. Observations 58

6. RELIABILITY ANALYSIS 59
6.1. Introduction 59
6.2. The β method for Reliability analysis 62
6.2.1. Rackwitz-Fiessler Algorithm 64
6.3. Problem Formulation 65
6.3.1. Code Design Equation 67
6.4. The G function for the problem 67
6.5. The realiability Program 69
6.6. Reliability Results 71
6.6.1. Discussion of results 75

7. Conclusions and Recommendations 76
7.1. Conclusions 76
7.2. Recommendations 77

REFERENCES 79

APPENDIX A 81

LIST OF FIGURES

1. Axial load-slenderness relationship for concentric loading. 7

2. Stress strain assumptions for wood 13

3. Bilinear stress strain relationship for wood 14

4. Distribution of stresses and strains 15

5. Stress-strain relationship for various m. 17

6. Large deformation of a beam element 18

7. Large deformation of axis of beam element. 19

8. i^{th} finite element in the x-coordinate system. 21

9. Flow chart for obtaining the solution vector {X}. 33

10. Estimating the initial failure load Pi. 36

11. Iteration process for obtaining Pmax. 38

12. Stress profile across the section. 42

13. Comparison of program (with m=-1) and analytical results [3]. 48

14. Axial load-Slenderness plots for the data of Table 2,(no size effects included). 51

15(a). Axial load-Slenderness curve with size effect taken into account, e = 2mm. 51

15(b). Axial load-slenderness curves with varying k_c for a constant k_t. 52

16. Some loading cases and support conditions. 55

17(a). Ultimate strength interaction curves for simply supported columns subjected to uniformly distributed load. 57

17(b). Non-dimensionalized ultimate strength interaction curves of Figure 17(a). 58

18. Typical problem for reliability evaluation. 59

19. Probability density function for the variable G. 61

20. Definition of the reliability index β. 63

21. Reliability results as a function of ϕ_p; size effects not included. 74

22. Reliability results with size effect included in the reliability program. 75

LIST OF TABLES

1. Maximum deflections of a fixed ended uniformly loaded beam. (E_o = 10000 Mpa, 2x4-in section, L = 2m) 47

2. Axial load-slenderness data for a pin-ended 2x4-in beam (size effect neglected). 49

NOTATION

The following symbols are frequently used in this report

P = axial compressive load

NDS = National Design Specifications (USA).

σ = normal stress

A = cross sectional area

E_o = mean modulus of elasticity (in Mpa)

m = slope of the stress strain curve, falling branch

ϵ = strain

u = axial deformation

w = transverse deformation

B = width of cross section

H = depth of cross section

L = Length of member

ξ, η = normalized coordinates

$\{\delta\}$ = nodal displacement vector

M, M_1, M_2 = Shape functions for the w displacements

N, N_1 = Shape functions for the u displacements

K_T = global tangent stiffness matrix

$\{\delta_o\}$ = initial displacement vector

$\{\Delta\delta\}$ = incremental displacement vector

ΔP = load increment

$\{X_o\}$ = initial solution vector

$\{X\}$ = New solution vector

k_t = size effect factor in tension

k_c = size effect factor in compression

G = failure function

p_f = probability of failure

β = reliability index

P_f = factored compressive resistance parallel to grain

ϕ_p = resistance factor in pure comression

K_c = Slenderness factor

d = dead load variable

l = live load variable

Design Load Factors

α_D = dead load factor

α_L = live load factor

γ_1 = ratio of nominal dead load to nominal live load

Subscripts

c = compression

t = tension

P_{DN} = nominal dead load

P_{LN} = nominal live load

f = failure

ACKNOWLEDGMENT

I wish to express my deepest gratitude to my supervisor, Professor R. O. Foschi for his invaluable advice and guidance throughout the research work and in the preparation of this thesis.

Thanks are also due to Professor B. Madsen for his advice and support. The financial support from the Commonwealth Scholarship and Fellowship Committee and some additional funding from the department of Civil Engineering of the University of British Columbia is gratefully acknowledged.

Lastly, thanks are due to those who have rendered help in this project, directly or indirectly.

1. INTRODUCTION

1.1. BACKGROUND

Wood compression members subjected to lateral loads occur very frequently, such as in building frames, bridge or roof trusses and other important engineering structures. They are usually proportioned to satisfy some limiting stress criterion set by design specifications or codes. The stresses developed at any cross section in such members consist of:

1. the axial stress caused by the compressive forces ,
2. the primary bending stress due to the lateral loads, and
3. the secondary bending stress resulting from the amplification of the deflections produced by the compressive forces .

The secondary bending stress becomes particularly important for members with a high slenderness ratio and large compressive forces. The procedures for computing the secondary stresses in elastic columns are described in the literature on stability theory [1].

Although elastic analysis is used extensively in design computations, it does not give an accurate indication of the true load-carrying capacity, particularly for columns which are not very slender. Laterally loaded columns generally fail by excessive bending after the stresses in some portions of the member exceed a maximum value. To determine the ultimate strength of such columns, it is necessary to perform a stability analysis that considers the elasto-plastic behaviour of the material. Most available design codes and specifications use the traditional approach, which consists of assuming a linear elastic material with a maximum normal stress failure criterion .

Previous analytical and experimental studies on wood, as reported in the literature [5,6,7], have shown that :

1. wood has a non-linear stress strain relationship in compression, e.g.bilinear elasto-plastic relationship, and
2. this material characteristic contributes significantly to the behaviour of the column, particularly at small slenderness ratios.

Furthermore,there are still some problems which remain unsolved:

1. The codes do not give guidance for calculating moments resulting from beam-column deflections.
2. An account for possibilities of ductile yielding in the compression zone or tension failure in the tension zone is not given.

1.2. OBJECTIVES

This study is aimed at achieving three main objectives, namely :

1. To develop a finite element analysis for the general prediction of the failure of a compression member under transverse loads. The analysis will take into account the non-linearities due to slenderness effects (geometric),a non-linear stress-strain relationship for the material, and estimation of failure load controlled by either tension or compression.
2. The analysis will be implemented in a computer program. The computer program will allow flexibility in accomodating various support conditions and load configurations.
3. To evaluate the reliability of wood compression members assuming random loads and material variables.

1.3. THESIS ORGANISATION

Part 2 provides a summary of current design code recommendations and previous research on wood compression members. Part 3 describes a general formulation of the finite element analysis and the computer implementation. Part 4 provides a verification of the computer program developed in Part 3, using experimental results as reported by previous researchers [6,7]. Part 5 presents the application of the analysis to a laterally loaded compression member, where axial load versus transverse load interaction diagrams are developed for different slenderness ratios using a 2x4-in SPF cross section.

Part 6 discusses the concept of reliability evaluation. Here, a computer program for the evaluation of the reliability index β of a wood compression member is constructed, using the program developed in Part 3 and the Rackwitz-Fiessler algorithm. A summary of the results obtained from this study for a specific problem is given at the end of the chapter. And lastly, Part 7 provides a general conclusion of the report and some recommendations for further research and study.

2. CURRENT CODE REQUIREMENTS AND PREVIOUS RESEARCH WORK

2.1. INTRODUCTION

The failure characteristics of a compression member depend on its slenderness. The ultimate capacity of short compression members is directly related to the strength of the material in compression. With an increase in the length of the member, a change to a buckling type of failure is observed. Thus, a lateral instability failure is characteristic of slender compression members. For a member of intermediate length, there is a transition between these two types of failure regimes, in which case the load capacity depends on both the compression strength and the stiffness of the material.

2.2. CURRENT CODE REQUIREMENTS

2.2.1. Concentric Compression

Current design codes classify compression members into short, intermediate or slender members according to their slenderness ratio C_c. For rectangular cross-sections,

$$C_c = \frac{L}{d} \tag{1}$$

where

L = length of the member

d = dimension of the cross-section of the member in the direction of buckling.

Thus, Short members are considered to be those with slenderness ratios of 10 or less. They will normally fail by crushing parallel to the grain. Their design allowable load is based on the specified strength in compression parallel to grain, F_c, their cross sectional area A, and a performance factor ϕ_p,

$$\frac{P}{A} \leq F_c \phi_p \tag{2}$$

Slender members generally follow the Euler buckling relation and have slenderness ratios exceeding C_k, a number dependent on the mean elastic modulus E_o and the specified compression strength, F_c, of the column material. The number C_k is given by :

$$C_k = \sqrt{\frac{0.9\ E_c}{F_c}} \tag{3}$$

where E_c is called the modulus of elasticity for compression members and is equal to $0.74E_o$. For lumber, C_k can vary between 20 to 25, depending on the grade of the member under consideration.

Intermediate members have slenderness ratios between 10 and C_k. They are designed using a modified compression strength which empirically interpolates between slenderness ratios of 10 and C_k. The above classification is illustrated in Figure 1.

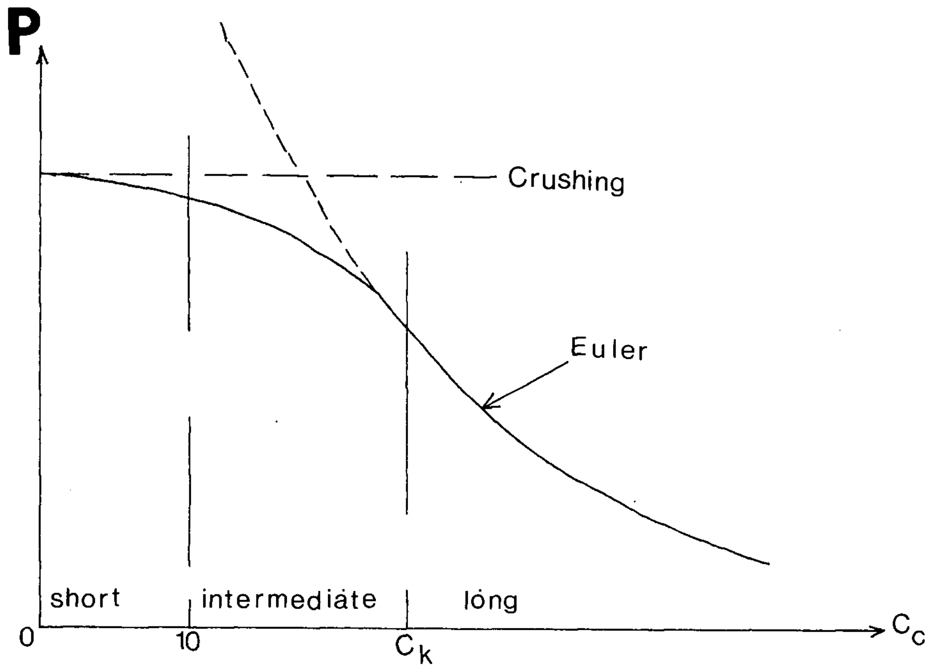


Figure 1. Axial load-slenderness relationship for concentric loading.

2.2.2. Combined bending and Compression

A compression member is often subjected to bending about either one or both axes, and the combined effect of the bending and axial loading must be considered. For this type of loading, most current codes specify a simple failure criterion based on a linear interaction between the axial load capacity of a concentrically loaded column and the moment capacity in bending alone. Therefore, this approach may be applicable as long as the wood member remains in the linear elastic range. Very little has been done so far to predict the behaviour beyond the linear range. This may be attributed, in part, to the uncertainities about the precise form of the curvilinear stress-strain relationship of wood in compression.

2.3. PREVIOUS RESEARCH ON WOOD COMPRESSION MEMBERS

Most previous studies on wood columns and beam-columns (Newlin and Trayer 1925; Wood 1950) have considered wood to be a linear elastic material which fails when a limiting compression stress is reached. Thus, Larsen and Theilgaard (1979) tested wood members with combined axial and transverse loads to verify their theory for beam-column behaviour. They used a second order linear differential equation to predict the deflections of elastic beams and beam-columns.

Bleau (1983) and Buchanan (1984), conducted an extensive joint experimental study on eccentrically loaded columns to calibrate and verify their strength models. Their models are able to predict the strength of full size lumber, using results of axial tension and compression tests on similar members. Buchanan used a mean modulus of elasticity, E_o, equal to 10000 Mpa to calibrate his model. Zann (1985), used Bleau's data (1983), with E_o equal to 10400 Mpa to calibrate his strength model. Zann's model (1985), is based on the NDS (1975) design recommendations, and takes account of biaxial account of biaxial bending. Although in both cases good agreement with the test results is reported, a question which remains unanswered is the fact that in each case a different E_o is used, and this E_o is not the one corresponding to the mean test results. Bleau (1983), reports an E_o of 9660 Mpa for the population tested.

The model developed in this report incorporates some of the ideas discussed by the previous researchers, and provides a more general solution to the beam-column problem. The method of formulation and the corresponding computer implementation will be discussed in Part 3 of this report.

3. FINITE ELEMENT ANALYSIS

3.1. INTRODUCTION

This chapter describes the formulation of a finite element analysis for predicting the failure of a wood member under direct axial compression and lateral loads. The theory and assumptions in this chapter will be described along with the basis of a computer program developed to implement the model.

3.2. ASSUMPTIONS

The following assumptions are made:

1. plane sections remain plane.
2. the stress-strain law for the material is known.
3. material properties are constant along the length of the member.
4. bending in only one plane is considered.
5. no torsional or out of plane deformations are considered.
6. duration of load effects are not considered.
7. shear deformations are small, hence neglected.

3.3. STRESS STRAIN RELATIONSHIP

Various studies [4,5,8] have focussed on the stress-strain behaviour of wood in compression parallel to grain, with the aim of deriving a mathematical relationship to represent this behaviour. Recently, Malhotra and Mazur (1970), proposed a stress strain relation of the form :

$$\epsilon = \frac{1}{E_o} [c\sigma-(1-c)f_c \ln(1- \frac{\sigma}{f_c})] \qquad (4)$$

where:

ϵ = strain.

σ = stress

f_c = maximum compression stress

E_o = mean modulus of elasticity

c = shape parameter.

For c = 0.99, the curve described by Equation (4) is shown as A in Figure 2.

A mathematical equation for the stress strain curve for clear dry wood in compression at various grain angles was also developed by O'Halloran(1973). The equation takes the following form

$$\sigma = E_o\epsilon - A\epsilon^n \qquad (5)$$

Where σ, ϵ, and E_o are defined above and A,n are equation constants determined by fitting the equation to a given set of experimental data. A plot of this equation is shown as curve B in Figure 2. This equation cannot be used beyond maximum strain because it may take on negative values very rapidly.

A comprehensive study on the stress-strain relationship of timber with defects, in compression parallel to grain, has been done by Glos (1978), as reported by Buchanan (1984). Based on experimental data, the curve shown as C in Figure 2 was proposed. This curve is characterised by a number of material parameters that depend on measurable wood properties, namely density, moisture content, knot ratio and the percentage of compression wood. Using this curve for modelling purposes necessarily involves the calibration of these parameters.

A simple bilinear proposal by Bazan (1980), as discussed by Buchanan (1984), appears to be the most recent one. In this proposal, it is assumed that the slope of the falling branch is a variable which can be arbitrarily taken as that value which produces maximum bending moment for any neutral axis depth. A plot of this curve is shown as D in Figure 2.

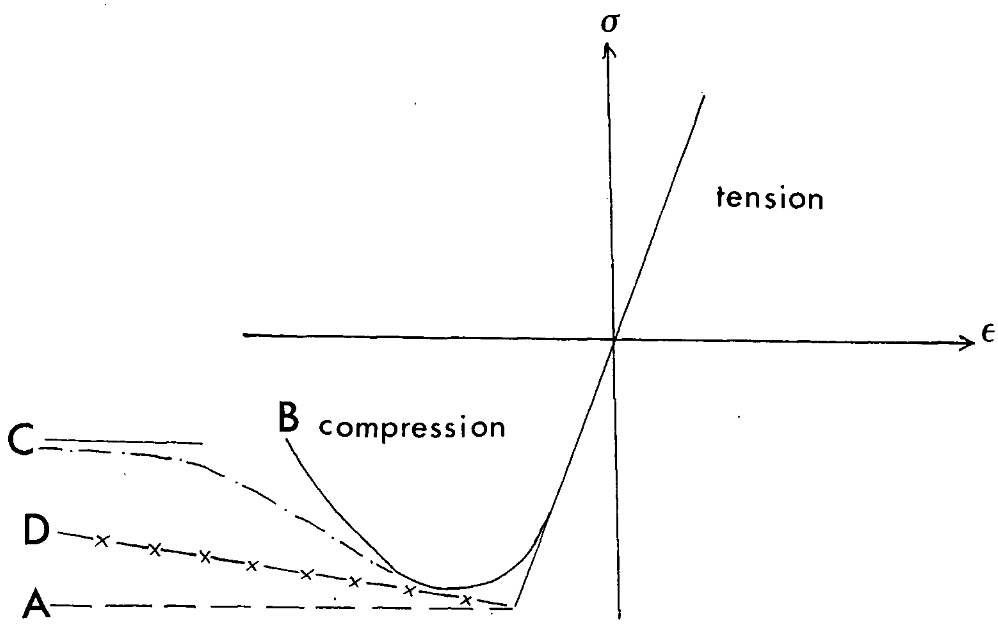


Figure 2. Stress strain assumptions for wood

The analysis in this study uses the simple bilinear curve D, with the exception that the slope of the falling branch of the stress-strain relation is considered to be a material property, in agreement with Buchanan (1984).

The curves in Figure 2 are characterised by a linear elastic and a non-linear part. Therefore the stresses can generally be expressed as

$$\sigma = E_o \epsilon + F(\epsilon) \tag{6}$$

The stress-strain relationship adopted in this study is as shown in Figure 3 and includes linear elastic behaviour in tension, with a bilinear relationship in compression and a falling branch after maximum stress.

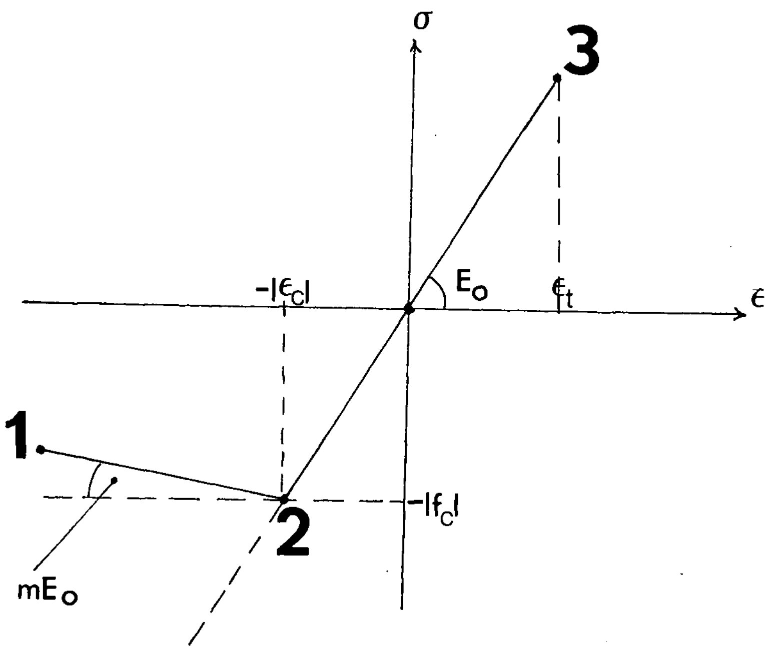


Figure 3. Bilinear stress strain relationship for wood

Using the above stress-strain relationship, the resulting distribution of stresses and strains in a rectangular beam is as shown in Figure 4.

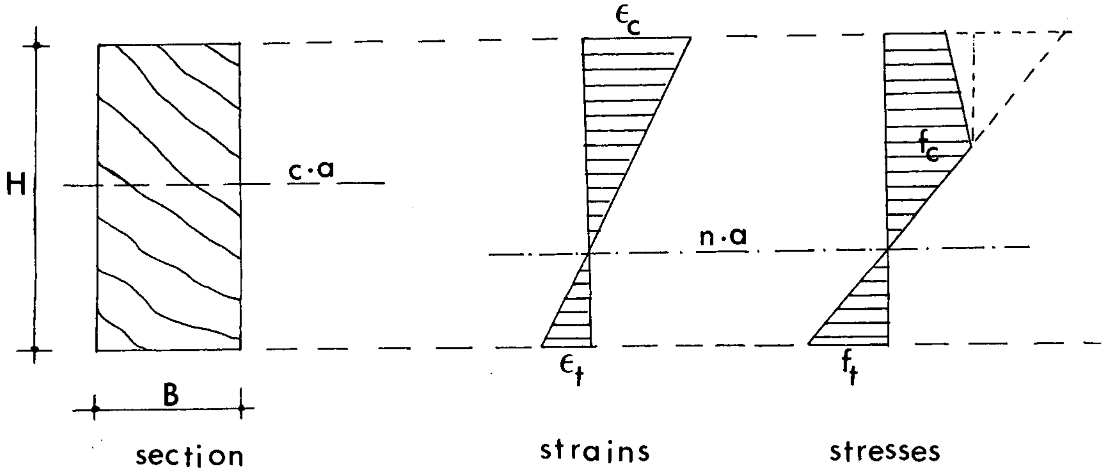


Figure 4. Distribution of stresses and strains

The curve in Figure 3 can be mathematically represented by the following expressions :

For segment 1-2 ;

$$\sigma = -|f_c| - |f_c|m - mE_o\epsilon \tag{7}$$

For segment 2-3 ;

$$\sigma = E_o\epsilon \tag{8}$$

Or, in combination,

$$\sigma = E_o\epsilon - [E_o\epsilon + |f_c|(1+m) + mE_o\epsilon](1 - \Delta(\epsilon + |\epsilon_c|)) \tag{9}$$

where $\Delta(\epsilon+|\epsilon_c|)$ is the step function defined as follows :

$$\text{if } \epsilon \geq -|\epsilon_c| \; ; \; \Delta = 1 \tag{10}$$

$$\text{if } \epsilon \leq -|\epsilon_c| \; ; \; \Delta = 0$$

Hence, for the case of elasto-perfectly plastic behaviour, m = 0; and Equation (9) reduces to

$$\sigma = E_o\epsilon - \{E_o\epsilon + |f_c|\} (1-\Delta(\epsilon + |\epsilon_c|)) \tag{11}$$

For the elastic case, m = -1, and we have Equation (9) both for tension and compression. This explanation is further illustrated in Figure 5 below.

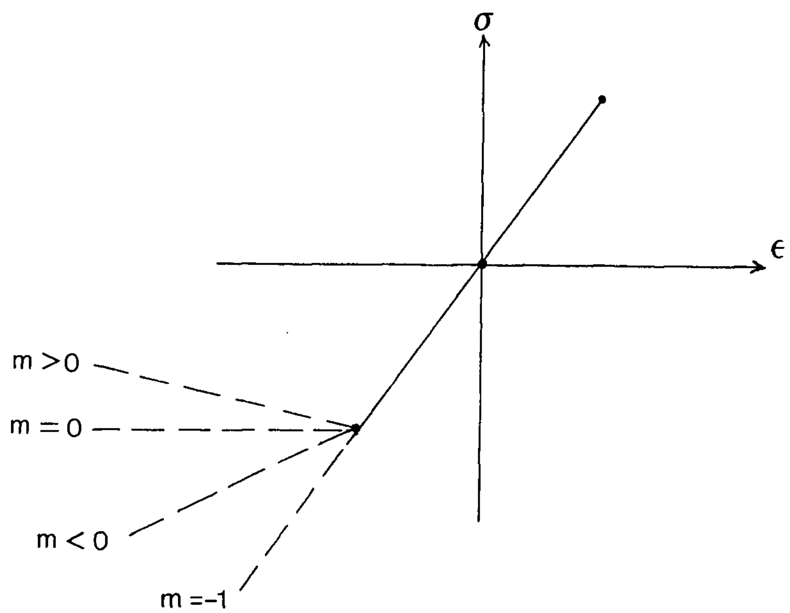


Figure 5. Stress-strain relationship for various m.

3.4. FINITE ELEMENT APPROXIMATION

3.4.1. Introduction

The finite element method is a very powerful and versatile technique presently available for the numerical solution of problems of the type considered here. The advantages of the method have been recognised and its applications extensively demonstrated particularly in steel and concrete structures, and for some wood structures such as wood floors, wood diaphragms and trusses. However, the application of the method to wood beam-column analysis has not been explored to an equivalent degree.

3.4.2. Kinematic Assumptions

As the displacements become large, a geometric non-linearity is introduced in the deformation of a beam-column. Consider a beam element undergoing large deformations but small strains. For the geometry shown in Figure 6, u and w are, respectively, axial and lateral displacements of the centreline of the beam. A and O are two points on the same plane such that O is on the beam centerline (axis) and A is at a distance z from O (positive z). Line OA represents conditions before deformation, while line O'A' represents conditions after deformation.

Assuming that plane sections remain plane, the rotation of the cross-section is $\theta = \dfrac{dw}{dx}$. Figure 7 shows two points, A and B, at the same distance z from the centreline. After deformation, these points are at A' and B'.

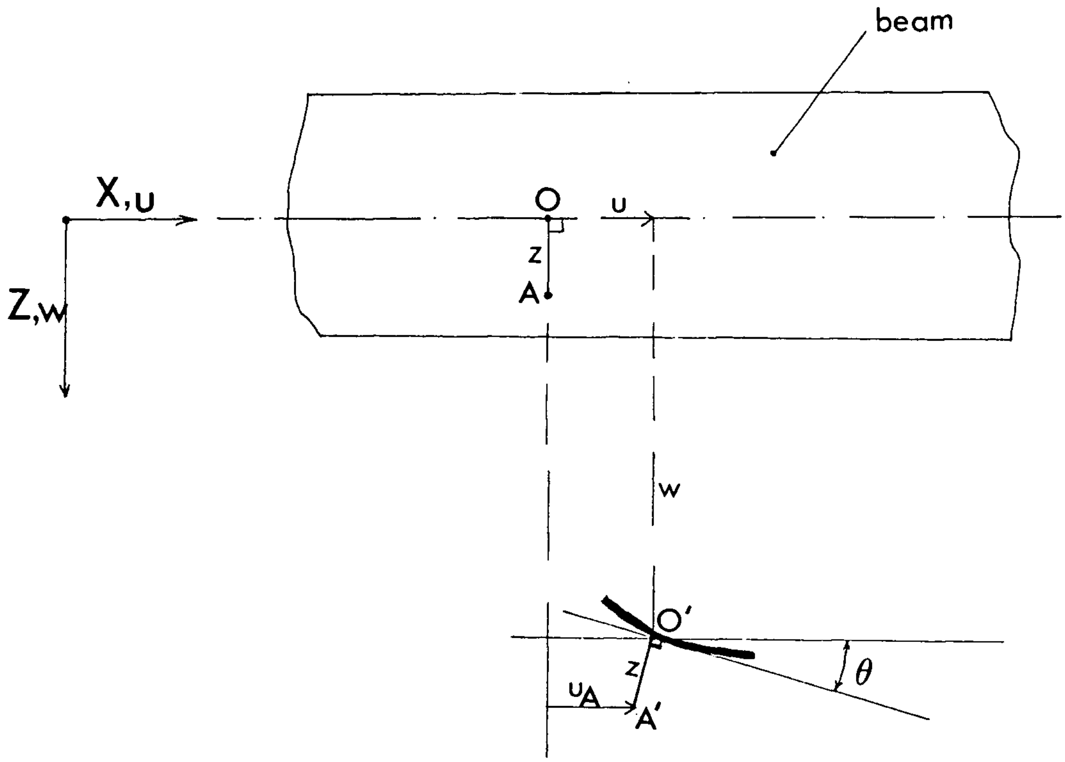


Figure 6. Large deformation of a beam element

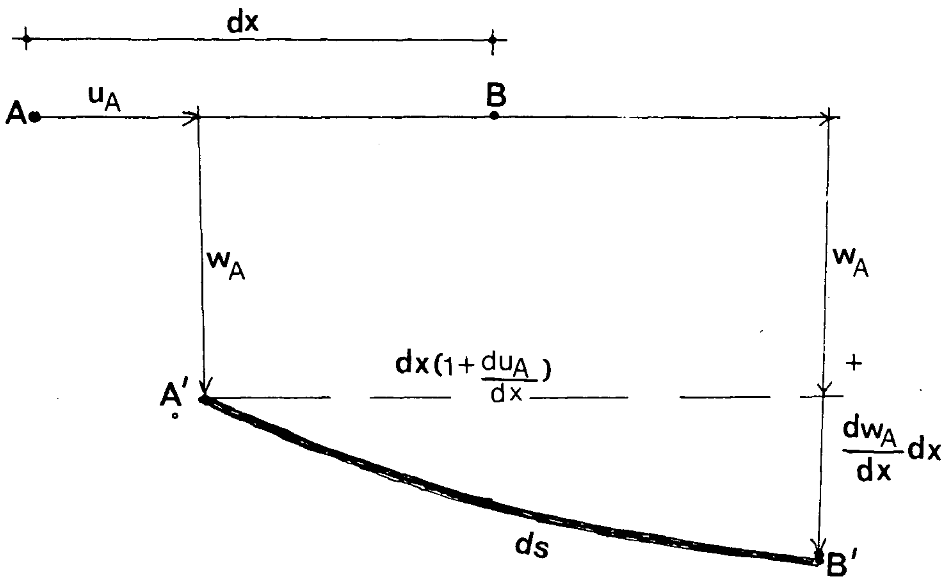


Figure 7. Large deformation of axis of beam element.

From the geometry of Figure 7 it follows that

$$ds^2 \simeq dx^2 \left[\left(1 + \frac{du_A}{dx}\right)^2 + \left(\frac{dw_A}{dx}\right)^2 \right] \tag{12}$$

If the expression above is expanded binomially, and if the higher order terms are neglected, the following simplified expression is obtained.

$$ds = dx\left(1 + \frac{du_A}{dx} + \frac{1}{2}\left(\frac{dw_A}{dx}\right)^2 + \ldots.\right) \tag{13}$$

Therefore, the corresponding strain ϵ_A at a distance z is

$$\epsilon_A = \frac{du_A}{dx} + \frac{1}{2} \left(\frac{dw_A}{dx} \right)^2 \tag{14}$$

But, from Figure 6, $u_A = u - z \dfrac{dw}{dx}$ (15)

thus,

$$\frac{du_A}{dx} = \frac{du}{dx} - z \frac{d^2w}{dx^2} \tag{16}$$

Also, neglecting higher order terms,

$$w_A = w \tag{17}$$

Thus combining Equations (14),(16) and (17), we get the strain at a height z as :

$$\epsilon = \frac{du}{dx} + \frac{1}{2} \left(\frac{dw}{dx} \right)^2 - z \frac{d^2w}{dx^2} \tag{18}$$

3.4.3. Problem formulation

A beam element with two end nodes is used in the formulation. Let us choose a local coordinate ξ ($-1 \le \xi \le 1$) in each element such as the one shown in Figure 8 below. Thus, along the x axis coordinate system each element has two end nodes, i and j separated by a length 2Δ.

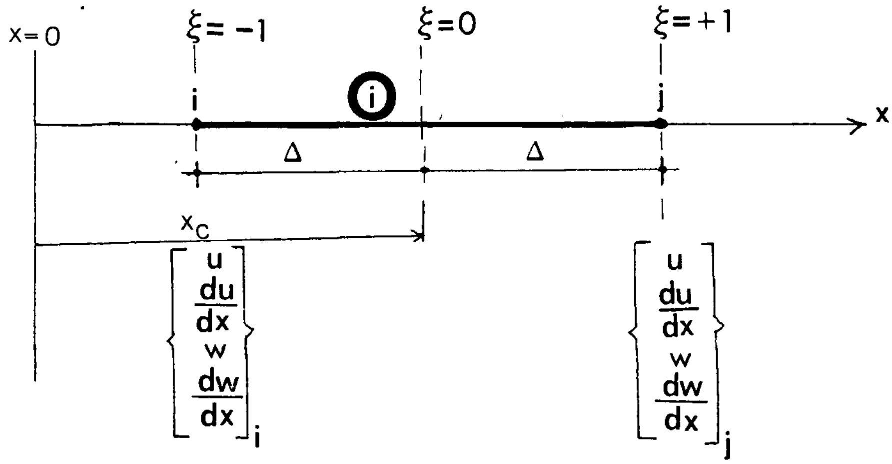


Figure 8. i^{th} finite element in the x-coordinate system.

Thus, the x-coordinate of any point within the element can be expressed as $x = x_c + \Delta\xi$. The elemental nodal degree of freedom vector is represented as in Equation (19). There are 4 degrees of freedom at each node, namely the two dispacements u and w and their respective first derivatives.

$$\{\delta\} = \begin{Bmatrix} u_i \\ (\frac{du}{dx})_i \\ w_i \\ (\frac{dw}{dx})_i \\ u_j \\ (\frac{du}{dx})_j \\ w_j \\ (\frac{dw}{dx})_j \end{Bmatrix} \tag{19}$$

3.4.4. Interpolation Functions

In this study, complete cubic interpolations are used to approximate the displacements u and w within an element. It is important to note that in order to satisfy compatibility conditions, the displacement u only requires a linear interpolation. However, a cubic interpolation was used to give an improved approximation of the axial stress with fewer elements.

A complete cubic interpolation requires 4 parameters to define the function. The displacements and the first derivatives at the two nodes provide sufficient parameters to fully describe a cubic polynomial function. The displacements u and w are thus given as follows:

$$u(\xi) = \left(\frac{1}{2} - \frac{3}{4}(\xi^2 + \frac{1}{4}\xi^3\right)u_i + \frac{1}{8}(1-\xi-\xi^2+\xi^3)\left(\frac{du}{dx}\right)_i + \left(\frac{1}{2} + \frac{3}{4}\xi + \frac{1}{4}\xi^3\right)u_j + \frac{1}{8}(-1-\xi+\xi^2+\xi^3)\left(\frac{du}{dx}\right)_j \tag{20}$$

$$w(\xi) = \left(\frac{1}{2} - \frac{3}{4}\xi + \frac{1}{4}\xi^3\right)w_i + \frac{1}{8}(1-\xi-\xi^2+\xi^3)\left(\frac{dw}{dx}\right)_i + \left(\frac{1}{2} + \frac{3}{4}\xi + \frac{1}{4}\xi^3\right)w_j + \frac{1}{8}(-1-\xi+\xi^2+\xi^3)\left(\frac{dw}{dx}\right)_j \tag{21}$$

$$\text{where} \qquad \xi = \frac{2x - 2x_c}{2\Delta}$$

In vector matrix notation, we can write

$$u = \{\mathbf{N}\}^T\{\delta\} \; ; \; \frac{du}{dx} = \{\mathbf{N}_1\}^T \{\delta\} \tag{22}$$

$$w = \{\mathbf{M}\}^T\{\delta\} \; ; \; \frac{dw}{dx} = \{\mathbf{M}_1\}^T \{\delta\} \; ; \; \frac{d^2w}{dx^2} = = \{\mathbf{M}_2\}^T\{\delta\}$$

where **N**, $\mathbf{N}_1$, **M**, $\mathbf{M}_1$ and $\mathbf{M}_2$, are vector functions of ξ given by the following expressions:

$$\{\mathbf{N}\} = \left\{\begin{bmatrix} \frac{1}{2} - \frac{3}{4}\xi + \frac{1}{4}\xi^3 \\ \frac{\Delta}{4}(1-\xi-\xi^2+\xi^3) \\ 0.0 \\ 0.0 \\ \frac{1}{2} + \frac{3}{4}\xi - \frac{1}{4}\xi^3 \\ \frac{\Delta}{4}(-1.0-\xi+\xi^2+\xi^3) \\ 0.0 \\ 0.0 \end{bmatrix}\right\} \tag{23}$$

$$\{N_1\} = \left\{ \begin{array}{c} \frac{1}{\Delta}\left(-\frac{3}{4}+\frac{3}{4}\xi^2\right) \\ \frac{1}{4}\left(-1.0-2\xi+3\xi^2\right) \\ 0.0 \\ 0.0 \\ \frac{1}{\Delta}\left(\frac{3}{4}-\frac{3}{4}\xi^2\right) \\ \frac{1}{4}\left(-1.0+2\xi+3\xi^2\right) \\ 0.0 \\ 0.0 \end{array} \right\} \qquad (24)$$

$$\{M\} = \left\{ \begin{array}{c} 0.0 \\ 0.0 \\ \frac{1}{2}-\frac{3}{4}\xi+\frac{1}{4}\xi^3 \\ \frac{\Delta}{4}\left(1.0-\xi-\xi^2+\xi^3\right) \\ 0.0 \\ 0.0 \\ \frac{1}{2}+\frac{3}{4}\xi-\frac{1}{4}\xi^3 \\ \frac{\Delta}{4}\left(-1.0-\xi+\xi^2+\xi^3\right) \end{array} \right\} \qquad (25)$$

$$\{M_1\} = \begin{Bmatrix} 0.0 \\ 0.0 \\ \frac{1}{\Delta}\left(-\frac{3}{4} + \frac{3}{4}\xi^2\right) \\ \frac{1}{4}\left(-1.0-2\xi+3\xi^2\right) \\ 0.0 \\ 0.0 \\ \frac{1}{\Delta}\left(\frac{3}{4} - \frac{3}{4}\xi^2\right) \\ \frac{1}{4}\left(-1.0+2\xi+3\xi^2\right) \end{Bmatrix} \tag{26}$$

$$\{M_2\} = \begin{Bmatrix} 0.0 \\ 0.0 \\ \frac{1}{\Delta^2}\left(\frac{3}{2}\xi\right) \\ \frac{1}{4\Delta}\left(-2.0+6\xi\right) \\ 0.0 \\ 0.0 \\ \frac{1}{\Delta^2}\left(-\frac{3}{2}\xi\right) \\ \frac{1}{4\Delta}\left(2.0+6\xi\right) \end{Bmatrix} \tag{27}$$

3.4.5. Strain Displacement Relations

For a laterally loaded column problem with large deformations, the w displacements will be much larger than the axial displacements u. Thus the strain displacement terms considered are similar to those derived in Equation (18) above, where:

$$\epsilon = \frac{du}{dx} - z\,\frac{d^2w}{dx^2} + \frac{1}{2}\left(\frac{dw}{dx}\right)^2 \tag{28}$$

Substituting the displacement functions into Equation (28), the results in symbolic form are

$$\underline{\epsilon} = [\underline{B}+\underline{B}(\delta)]\delta \tag{29}$$

where $\underline{B}$, represents the linear strain displacement terms, while $\underline{B}(\delta)$, which is a function of the δ parameters, includes the contribution of the non-linear strain displacement terms. Thus

$$\underline{\mathbf{B}} = \mathbf{N}_1{}^T - z\mathbf{M}_2{}^T, \text{ where } z = \frac{h}{2}\,\eta \tag{30}$$

$$\underline{\mathbf{B}}(\delta) = \frac{1}{2}\,\{\delta\}^T\,\underline{\mathbf{M}}_1\underline{\mathbf{M}}_1{}^T$$

3.4.6. Virtual Work Equations

The system of equations governing the problem is obtained via the principle of virtual work. Defining $\underline{\tilde{\delta}}$ as a virtual dislacement of the nodal variables, the resulting virtual strains, $\underline{\tilde{\epsilon}}$, are given by

$$\underline{\tilde{\epsilon}} = [\underline{B} + \underline{C}(\delta)]\underline{\tilde{\delta}} \tag{31}$$

where $\underline{C}(\delta) = \frac{\partial}{\partial \delta}\{\underline{B}(\delta)\}$ is a linear function of δ. Thus, if we neglect inertia forces, the virtual work equation reduces to

$$\int_V \underline{\tilde{\epsilon}}\sigma dv = \underline{\tilde{\delta}^T}P \tag{32}$$

where P is the consistent load vector, calculated using the shape functions as indicated in Equations (23), (24), (25), (26) and (27). V is the volume of the member. Substitution of the resulting equation for $\underline{\tilde{\epsilon}}$ into the Equation (32) leads to the system of governing equations for an element, that is,

$$\int_V [\underline{B} + \underline{C}(\delta)]^T \sigma dv = P \tag{33}$$

Assembling the element equations in the usual finite element manner leads to the global system of equations. In order to find the solution to the nonlinear system of Equations (33), it is convenient to introduce the vector function of δ, $\Phi(\delta)$, such that

$$\Phi(\delta) = \int_V [\underline{B} + \underline{C}(\delta)]^T \sigma dv - P \tag{34}$$

The solution δ now has to satisfy $\Phi(\delta) = 0$. The zeros of $\Phi(\delta)$ may be found numerically via the Newton-Raphson procedure as outlined below.

3.4.7. Newton-Raphson Method

This is a commonly used technique to solve non-linear equations. The method uses a first order approximation technique to solve non-linear equations through iteration. Thus, at $\delta+\Delta\delta$, the first order approximation for the function Φ will be

$$\Phi(\delta+\Delta\delta) = \Phi(\delta) + [\frac{d\Phi(\delta)}{\delta\{\delta\}}]\Delta\delta \tag{35}$$

where $\Phi(\delta)$ is a function of the displacement vector $\{\delta\}$. The above equation can further be simplified into

$$\Delta\delta = - K_T^{-1}\Phi(\delta) \tag{36}$$

$$= [\ \frac{d\{\Phi(\delta)\}}{d\{\delta\}}\ ]^{-1}\Phi(\delta) \tag{37}$$

where $[K_T]$ represents the tangent stiffness matrix. Differentiating the right hand side of Equation (34) by parts, we obtain a simplified expression for K_T. Equation (36) permits an iterative procedure to determine the vector $\{\delta\}$ starting from an initial approximation $\{\delta_o\}$. Thus, in general,

$$\delta_{i+1} = \delta_i - [K_T]^{-1}\Phi(\delta_i) \tag{38}$$

$$\Phi(\delta_i) = \int_V [B_o + B(\delta_i)]\sigma_i - P \tag{39}$$

where the matrix K_T is obtained as shown in Equation (40), which follows.

$$
\begin{aligned}
[K_T] = &+ E_o\int_V B^T C(\delta)dV \\
&+ E_o\int_V C^T(\delta) B dV \\
&+ E_o\int_V C^T(\delta)\ C(\delta)dV \\
&+ E_o\int_V B^T B dV \\
&- E_o(1.0 + m)\int_V (1.0 - \Delta(\epsilon + |\epsilon_C|))B^T\ B dV \\
&- E_o(1.0 - m)\int_V (1.0 - \Delta(\epsilon + |\epsilon_C|))B^T\ C(\delta)dV \\
&- E_o(1.0 - m)\int_V (1.0 - \Delta(\epsilon + |\epsilon_C|))C^T(\delta)B dV \\
&- E_o(1.0 - m)\int_V (1.0 - \Delta(\epsilon + |\epsilon_C|))C^T(\delta)\ C(\delta)dV \\
&+ \int_V M_1 M_1 \sigma dV
\end{aligned}
\tag{40}
$$

3.4.8. Computation procedure

The Cholesky decomposition of the matrix K_T is utilized to determine the vector $\{\Delta\delta\}$from Equation (38). The boundary conditions are first applied to the stiffness matrix K_T and to the vector $\{\Phi(\delta)\}$. The boundary condition codes for this program are as follows

$$1 = u$$

$$2 = \frac{du}{dx}$$

$$3 = w$$

$$4 = \frac{dw}{dx}$$

Thus, to enforce a boundary condition equal to zero, zeros are placed into the off diagonal locations for the row and column corresponding to the specified degree of freedom in $[K_T]$, while a zero is placed for the same degree of freedom in the returned load vector $\{\Phi(\delta)\}$. In addition to this, a value 1 is placed into the diagonal term of the specified degree of freedom in the $[K_T]$ matrix. Then the matrix $[K_T]$ is decomposed and finally a solution $\{\Delta\delta\}$ is obtained.

Each element of the vector $\{\Delta\delta\}$ is compared against an acceptable tolerance specified by the user to determine whether a need to do more iterations is necessary in order to refine the solution. A summary of the whole procedure is given in the flow chart in Figure 9.

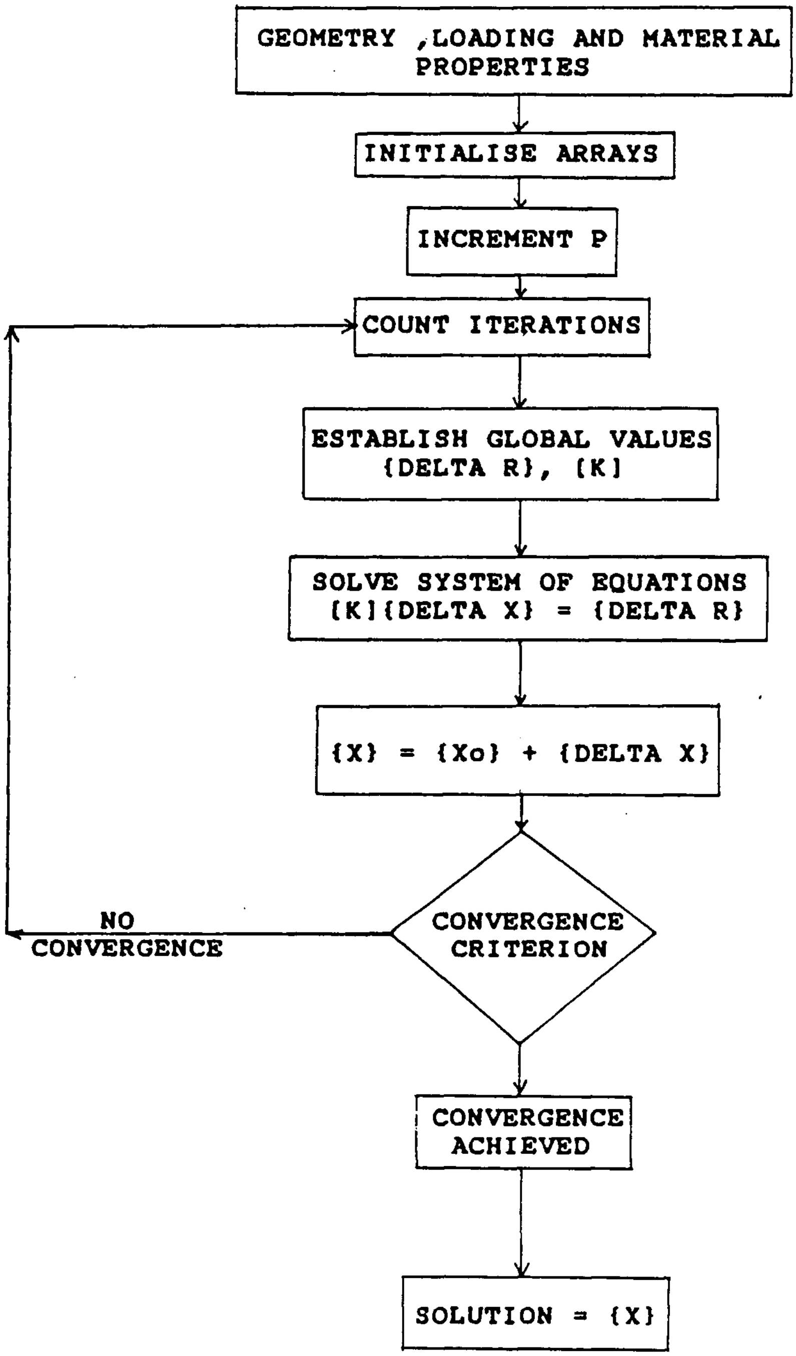


Figure 9. Flow chart for obtaining the solution vector {X}.

3.4.9. Numerical Integration

Since the volume integrals in the expression for K_T are complicated, it is difficult to obtain closed form solutions. Hence numerical integration is used. Gaussian quadrature scheme has been applied due to its suitability in the local coordinate system varying from -1 to +1.

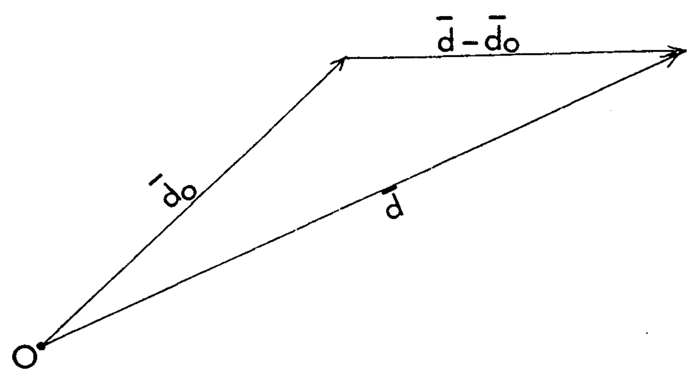
According to Zienkiewicz (1979), the maximum order of the polynomial appearing in the integral determines the number of Gaussian points necessary to accurately integrate the function. Thus, the term $\{\underline{B}\}$ in Equation (40) contains a fourth order polynomial in ξ and at the same time $\{\underline{B}\}$ is squared in the expression for $[K_T]$. Therefore, the highest order polynomial term in the integrals is of order 8. Knowing that a k point Gaussian scheme will integrate exactly a (2k-1) order polynomial, it follows that a 5-point Gaussian scheme is needed in the numerical integration. Thus, the integrals over the volume V become

$$I_V = \frac{2BH\Delta}{4}\int_{-1}^{1} d\xi \int_{-1}^{1} d\eta .. = \frac{2BH\Delta}{4}\sum_{i=1}^{N}\sum_{j=1}^{N} K_T(i,j) w_i w_j \qquad (41)$$

where N = 5 was chosen, and w_i and w_j are the corresponding Gaussian weights.

3.5. CONVERGENCE CRITERION FOR SOLUTION VECTOR

The convergence of the solution vector at every load step is checked by the Euclidean norm criterion. If we let $\bar{d}_o$ be the previous solution vector and $\bar{d}$ be the present solution, then, as shown below,



let $\Delta x = |\bar{d} - \bar{d}_o|$ represent the difference between the lengths of d and d_o. We can then write

$$|\bar{d}_o|^2 = X_o^2(i)$$

$$|\bar{d}|^2 = X^2(i)$$

where $X_o(i)$ and $X(i)$ are the components of d_o and d, respectively.

Then,

$$\Delta x = \sqrt{\sum (X(i) - X_o(i))^2} \tag{42}$$

The convergence criterion based on the Euclidean norm is defined as

$$\frac{\Delta x}{|\bar{d}_o|} \leq \text{specified tolerance} \tag{43}$$

3.6. OBTAINING THE ULTIMATE LOAD PMAX

The failure load Pmax is obtained by an iterative procedure. For fast convergence to the solution Pmax, the following approach for estimating an initial guess for the failure load is chosen. First of all, the crushing strength P_c as well as the Euler buckling load Pcr of the member are computed. Regardless of the support conditions and member length, the ultimate load will be less than the smallest value between P_c and Pcr and will lie within the shaded region of Figure 10.

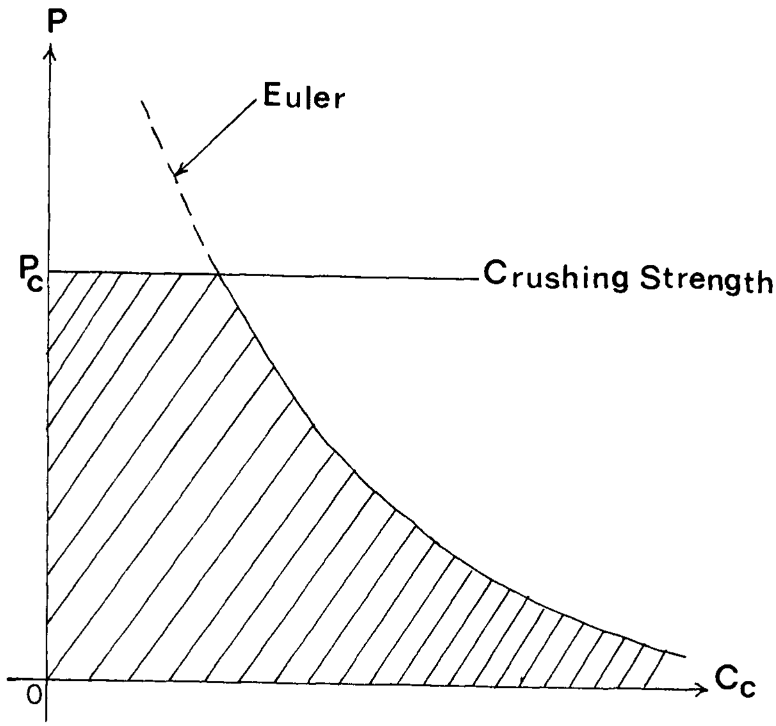


Figure 10. Estimating the initial failure load Pi.

The minimum of P_c and Pcr is then taken to be the initial failure load P_i. As a first step, we let the solution lie between two load values, namely P1=0 and P2=P_i. The average load P3=(P1+P2)/2 becomes the first trial load. The finite element solution is obtained for P=P3. If failure occurs, it means that the solution is between the values P=P1 and P=P3. Therefore we set the minimum and the maximum loads for next iteration as P1=P1 and P2=P3. A new P3 = (P1+P2)/2 is calculated and the finite element program re-run. If the member survives, it means that the solution is now between the values P=P3 and P=P2. Therefore, we set the minimum and the maximum loads for next iteration as P1=P3 and P2=P2. This process is repeated several times until an acceptable tolerance is reached between two successive estimates of Pmax. If this tolerance is defined as TOLP, the iterations are stopped when

$$TOP = \frac{P2-P3}{P3} \leq TOLP$$

The process is summarized in the flow chart of Figure 11, where TOLP is the allowable tolerance normally set by the user.

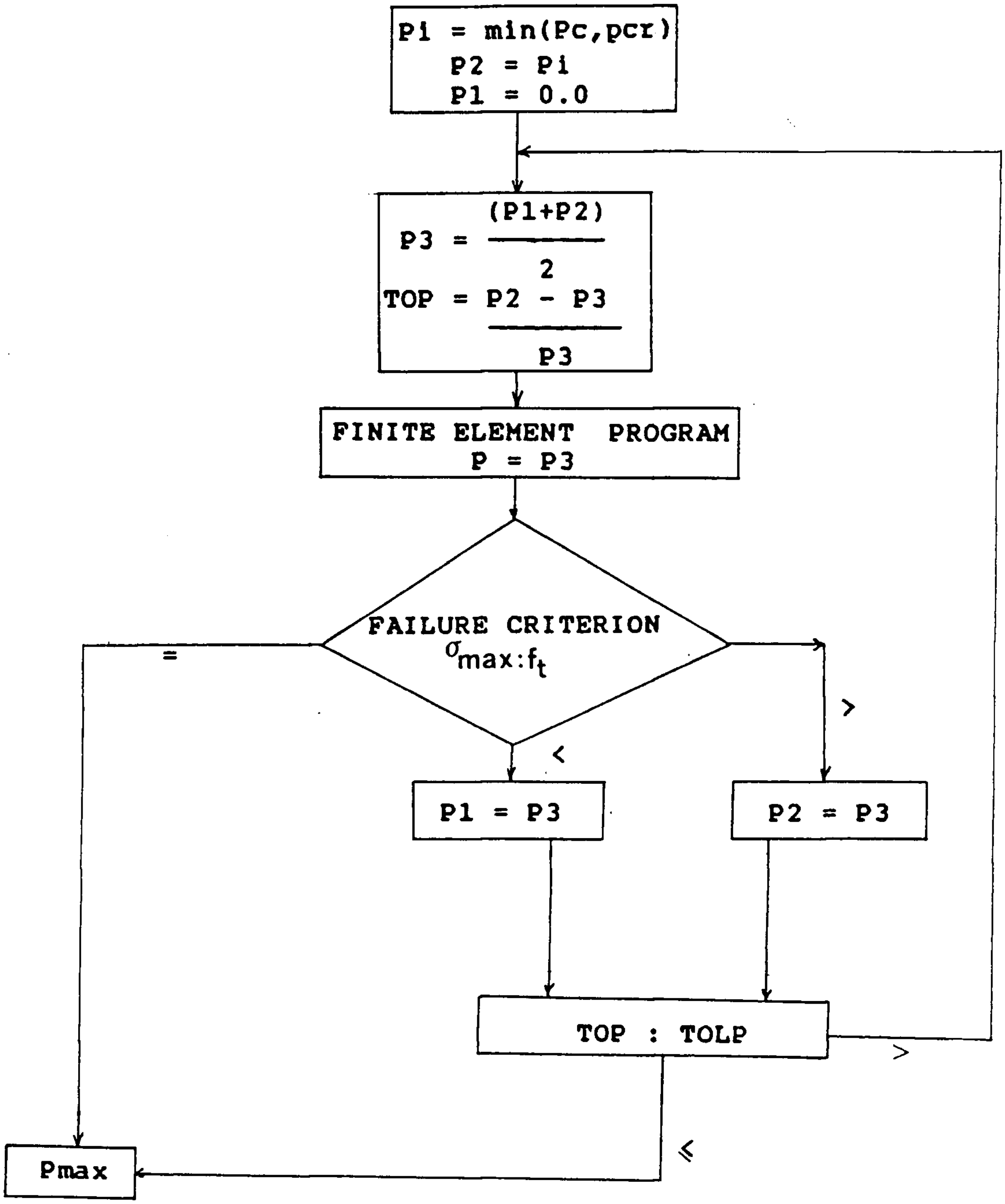


Figure 11. Iteration process for obtaining Pmax.

3.7. FAILURE CRITERION AND SIZE EFFECTS

Due to the brittle fracture phenomenon which is commonly observed in wood members, it may be important that the associated size effects be incorporated in the analysis. In a brittle material, a decrease in member strength is normally observed as a result of a corresponding increase in member size. If no size effects are considered, the failure criterion is

$$\sigma_{max} = F_t \tag{44}$$

where σ_{max} is the maximum tensile stress in the member. This criterion, although simple, does not result in different strengths between pure tension and pure bending. Such differences are accountable through the incorporation of size effects.

Weibull's theory of brittle fracture will be applied to incorporate the size effect phenomenon. Thus, for a member of volume V, failure is related to the integral

$$I = \int \sigma^k \, dv \tag{45}$$

with a corresponding failure criterion given by

$$I = (\sigma^*)^k \qquad (46)$$

where

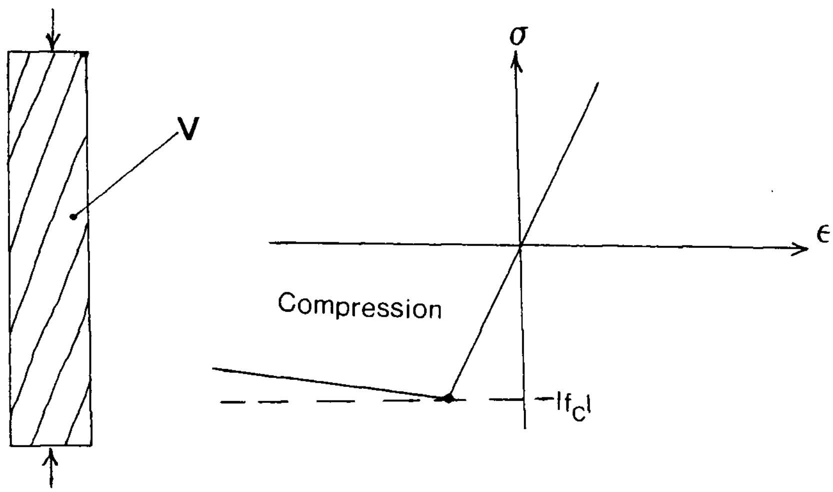
σ = stresses in the member,

σ^* = strength of a unit volume under uniform stress,

k = size effect factor,

V = volume of the stresses domain.

3.7.1. Size effect in compression



The parameter $|f_c|$ is the failure stress in pure compression (buckling restrained). This may be considered subject to size effects, according to

$$f_c^{k_c} V = (F_c^*)^{k_c} \tag{47}$$

$$f_c = \frac{F_c^*}{V^{1/k_c}} \tag{48}$$

where

F_c^* = failure stress in pure compression for a unit volume,

k_c = size effect parameter in compression,

V = total volume of the domain under compression that is, the entire member.

3.7.2. Size effect in tension

Let F_T be the strength in pure tension. Then, from Equations (44) and (45) we have at any probability level:

$$\int_{V_T} \sigma^{k_t} \, dv = F_T^{k_t} V \tag{49}$$

where k_t is the size effect factor in tension, V is the total volume and V_T is the domain of the tensile stresses. In the context of the analysis presented here, consider a finite element i and the local ξ-coordinate system, as shown in Figure 12. The stresses within the element are assumed to follow the stress strain relationship as indicated Figure 4.

Let us introduce a local coordinate η ($0 \le \eta \le 1$)such that $y = \eta h$. The tensile stresses will be linear in y, or $\sigma = \eta\sigma_T$, where σ_T is the maximum stress at the edge $y = h$.

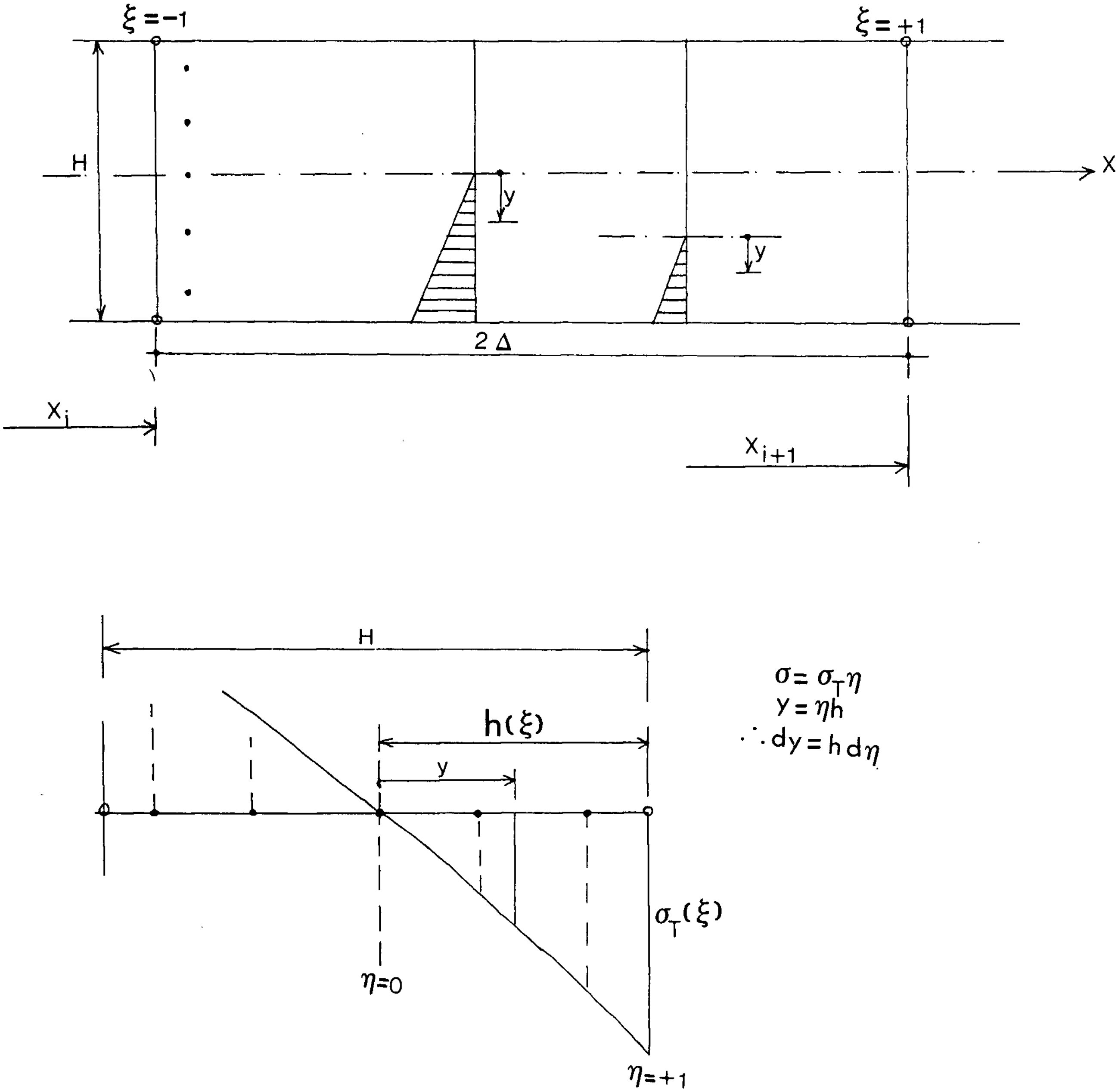


Figure 12. Stress profile across the section.

Equation (49) can then be expressed as

$$\sum_{i=1}^{N} BH\Delta \int_{-1}^{1} d\xi \int_{0}^{1} d\eta \ \sigma^{k_t} \frac{h(\xi \ )}{H} = F_T{}^{k_t}V \tag{50}$$

where N is the number of elements. Since $\sigma = \sigma_T(\xi)\eta$,

$$\int_{-1}^{1} \frac{h(\xi)}{H} \ \sigma_T{}^{k_t}(\xi) \quad d\xi \int_{0}^{1} \eta^{k_t} d\eta = \frac{1}{k_t+1} \int_{-1}^{1} \frac{h(\xi)}{H} \ \sigma_T{}^{k_t}(\xi) d\xi \tag{51}$$

then, Equation (50) becomes

$$\frac{1}{2N(k_t+1)} \sum_{i=1}^{N} \int_{-1}^{1} \frac{h(\xi)}{H} \ \sigma_T{}^{k_t}(\xi) d\xi = F_T{}^{k_t}$$

or, finally,

$$F_T{}^{k_t} = \frac{(\sigma_T max)^{k_t}}{2(k_t+1)N} \sum_{i=1}^{N} \int_{-1}^{1} \frac{h(\xi)}{H} \ \left(\frac{\sigma_T(\xi)}{\sigma_T max} \right)^{k_t} d\xi \tag{52}$$

The location of the neutral axis, $h(\xi)$, where the stresses σ change sign, can be obtained by interpolation of the stress field.

The implementation of the procedure in the finite element computer program follows the equations as derived above. A summary of the steps follow below.

1. $\sigma_T(\xi)$ is determined at all points ξ and for all elements;
2. obtain the largest of the $\sigma_T(\xi)$, $\sigma_T max$ to normalize the

stresses.

3. obtain $h(\xi)$ for any cross section. A section fully in compression will result in $h(\xi) = 0$.
4. Integrate over each element and add, according to Equation (52).
5. Compare the $\sigma_T$max with the maximum stress possible according to the failure criterion of Equation (52).

3.8. PROGRAM STRUCTURE

The computer program consists of a number of subroutines which read the structure's geometry and load data, carry out numerical integration, decompose matrices, solves system of equations and checks the convergence of the solution vector. The program enables the user to analyize beams, columns or beam-columns of various configurations. A time subroutine has been provided to give the amount of computer time used to solve each specific problem. This time is calculated in cpu seconds. A listing of the program has been provided in Appendix A.

3.9. DISCUSSION

The analysis developed here offers numerous possibilities. The material behaviour law can be modified to study different materials or the effect of several parameters in a single material. Also the dimensions of a member cross-section, the eccentricity of axial load, laterally acting loads and support conditions can be varied. In the following chapter, the model will be verified by considering some problems for which there are available experimental or theoretical results.

The computer program developed here does not take into account torsional or out of plane deformations. Also creep effects were not included in the analysis. It is also anticipated that there could be a significant variation of modulus of elasticity E_o along the length of the member. However, without loss of generality, and in the presence of reliable experimental data, the program can be easily modified to accommodate such variations in E_o. The approximation for the stress-strain relationship used is suitable for small and intermediate levels of strain, but obviously can not be extrapolated to very large strains.

4. PROGRAM VERIFICATION

4.1. INTRODUCTION

In this chapter, the finite element computer program developed in the previous chapter is verified with reference to

1. theoretical results from the theory of elastic beam-columns, and
2. the results of an extensive experimental program on a large number of timber members in structural sizes [as reported by Bleau (1983) and Buchanan (1984)].

The test material was SPF lumber, purchased in 16ft. (4.88m) lengths as 'Number 2 and Better' grade in Quebec, Canada. The program is verified using the mean test results, namely modulus of elasticity E_o = 9660 Mpa, compressive strength f_c = 32.3 Mpa and tensile strength f_t = 30.35 Mpa.

4.2. COMPARISON OF RESULTS

The first comparison considers analytical results [3] and the computer program's elastic predictions using m = -1, where m is the slope of the falling branch of the stress-strain curve in compression. The second comparison presents plots and tables of axial load versus slenderness ratio to compare the mean maximum load from tests with what the present analysis predicts for several end eccentricities

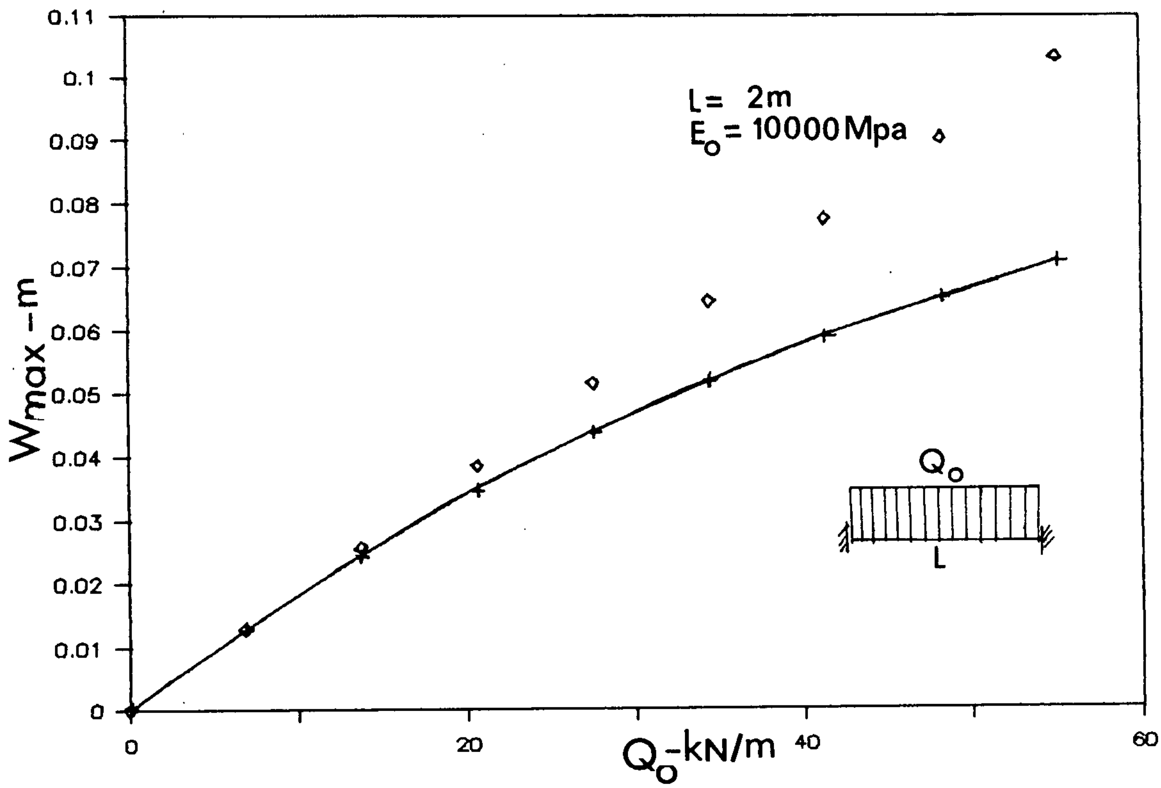
e. No size effects are considered. The third comparison is similar to the second one, except that in this case the size effect phenomenon is taken into consideration, and the effect of varying k_c for a chosen k_t is evaluated.

4.2.1. Presentation of Results

Table 1 shows a comparison of linear and non-linear theoretical results [3] and computer predictions for a uniformly loaded fixed ended beam. The data of Table 1 is plotted in Figure 13.

Qo	Wmax [m]		
[kn/m]	[Timoshenko]	[program]	[linear]
00.000	0.00000	0.00000	0.00000
06.885	0.01291	0.01266	0.01285
13.771	0.02447	0.02437	0.02570
20.656	0.03516	0.03469	0.03855
27.541	0.04406	0.04371	0.05140
34.426	0.05162	0.05159	0.06426
41.312	0.05874	0.05859	0.07711
48.197	0.06497	0.06485	0.08996
55.082	0.07076	0.07053	0.10281

Table 1. Maximum deflections of a fixed ended uniformly loaded beam. (E_o = 10000 Mpa, 2x4-in section, L = 2m)



◊ Linear, —Timoshenko, x program

2x4-in SPF

Figure 13. Comparison of program (with m=-1) and analytical results [3].

Table 2 shows a comparison of failure loads as obtained by the computer program to test results for different eccentricities e. A graphical plot of the data in this table is shown in Figure 14, with no size effects considered. Similar results with size effects included are shown in Figures 15(a) and 15(b).

	COMPUTER RESULTS		TEST RESULTS	
	e = 2mm	e = 39mm	e = 2mm	e = 39mm
C_c	Pmax [Kn]		Pmax[Kn]	
3.37	100.953	41.327		
5.10	100.111	40.066	104.35	48.21
6.74	98.008	38.593		
8.99	95.064	36.490		
11.24	90.437	34.177		
14.61	80.131	30.601	69.02	32.68
16.85	70.022	28.175		
19.10	60.125	25.676		
20.22	55.170	24.442	48.75	24.71
21.35	50.897	23.376		
24.72	40.024	20.356		
25.80	36.934	19.410	34.98	--
28.10	31.793	17.759		
32.60	24.220	14.829		
35.96	20.054	13.194	20.52	14.11
40.45	15.974	11.258		

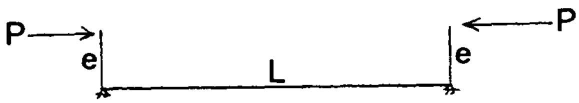


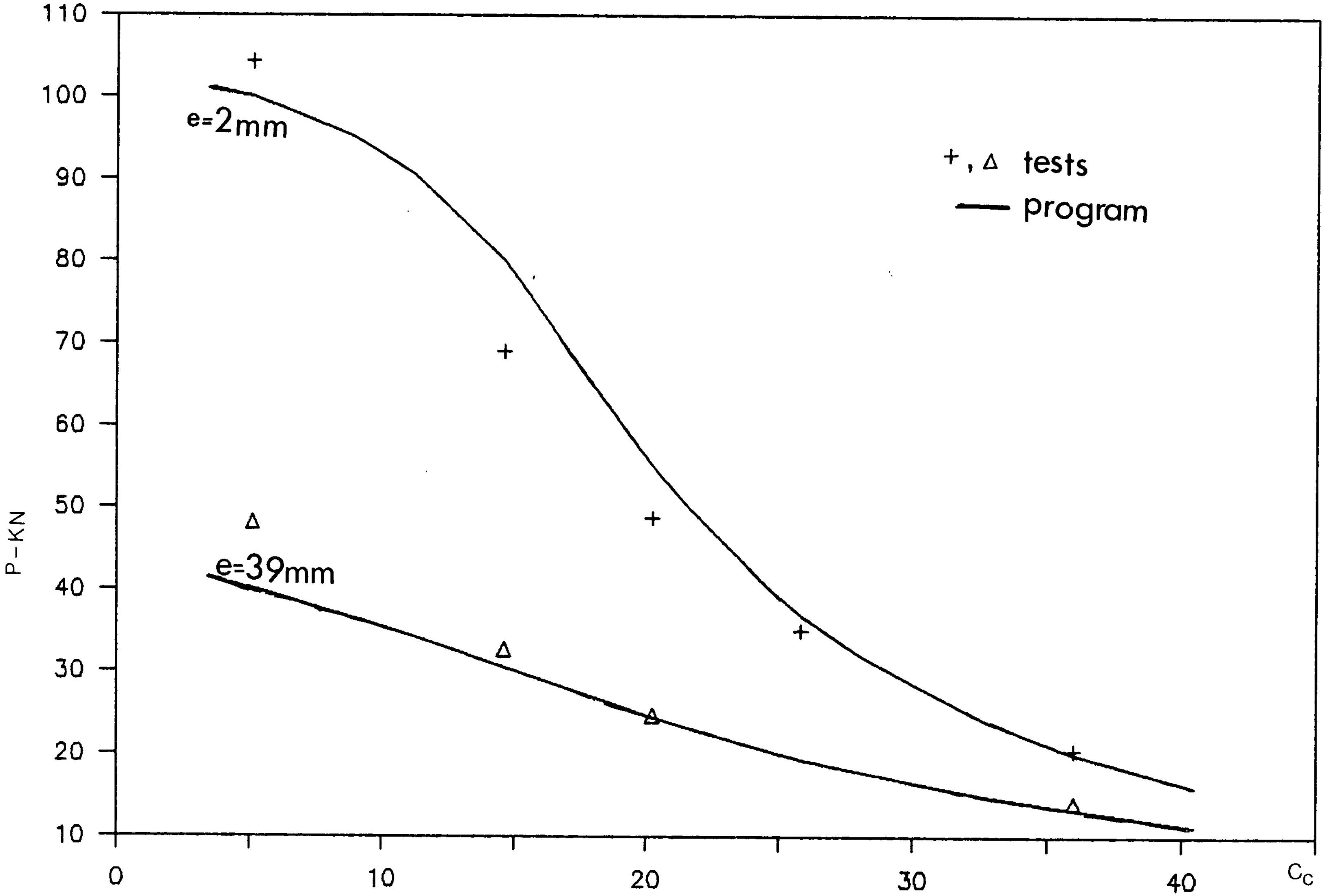
Table 2. Axial load-slenderness data for a pin-ended 2x4-in beam (size effect neglected).

<u>Data Input</u> :

2x4-in SPF section

mean E_o, f_c, f_t

Figure 14. Axial load-Slenderness plots for the data of Table 2,(no size effect).



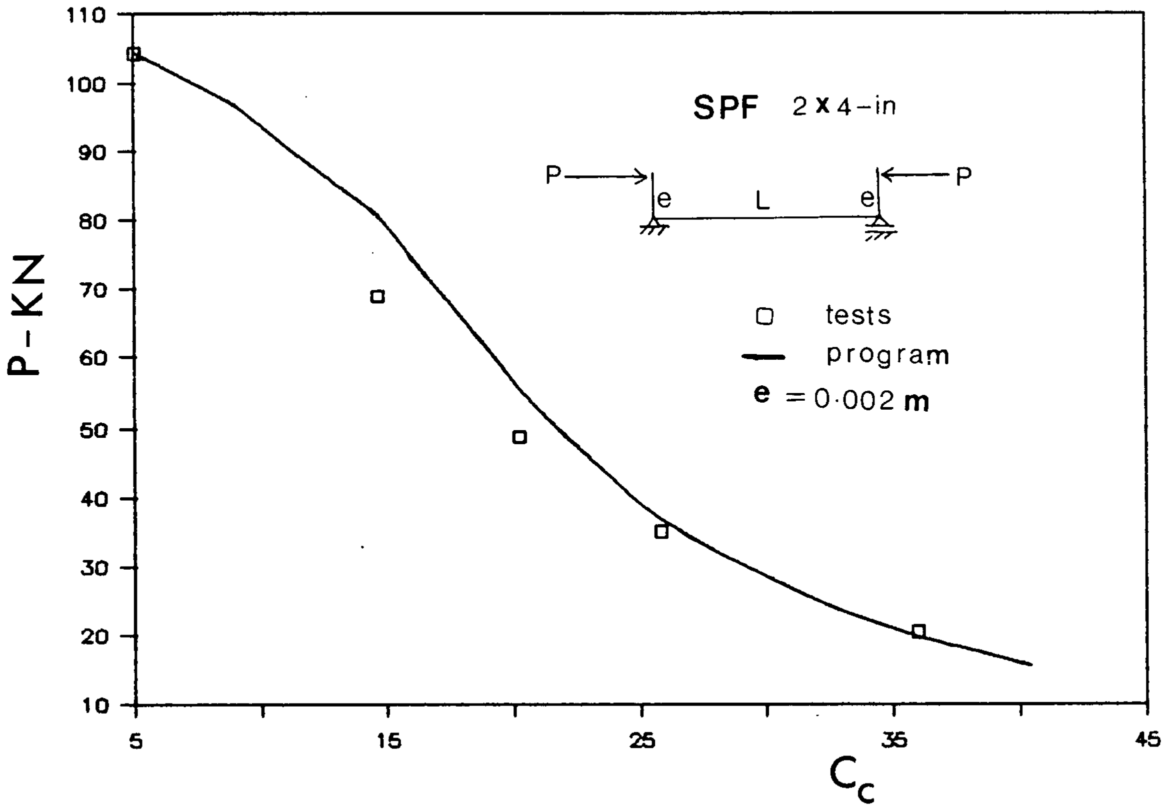


Figure 15(a). Axial load-Slenderness curve with size effect taken into account, e = 2mm.

Input Data:

mean E_o, f_c and f_t

k_c = 20.0 and k_t = 5.0

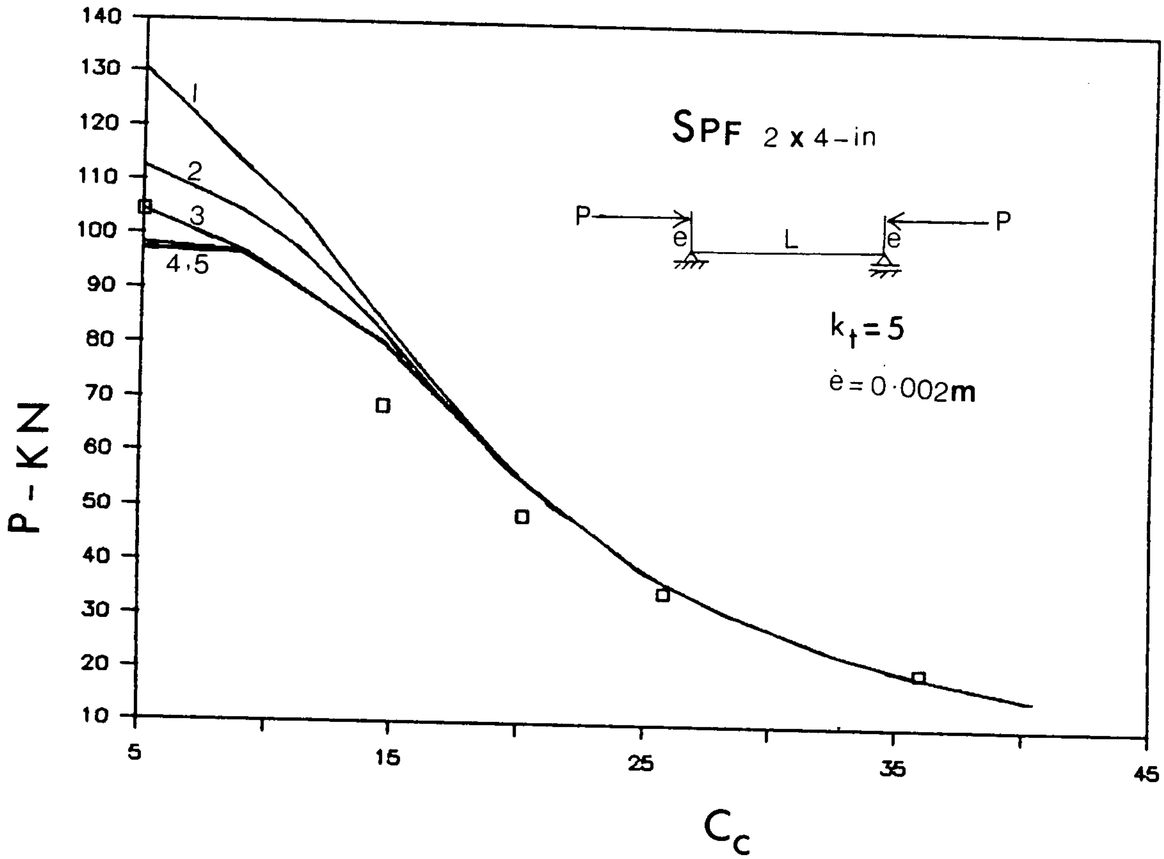


Figure 15(b). Axial load-slenderness curves with varying k_c for a constant k_t.

□- Tests, 1 - k_c = 5

2- k_c = 10, 3 - k_c = 20

4 - k_c = 100, 5 - k_c = 150

4.3. DISCUSSION

The results as presented in this chapter, show that there is a relatively good agreement between the test and the predictions by the computer program, the agreement being a very good one for members with a high slenderness ratio. For compression members in the intermediate range, the program predictions are slightly higher than the test results. For very short members the program predictions are slightly below. Several explanations can be put forward to explain these discrepancies. The most obvious one is that the stress-strain curve used for this study may not be a true representation of the actual behaviour. Nevertheless, since this feature may be changed in the analysis, the finite element technique developed here remains a powerful and general tool to study the behaviour and design considerations of timber columns and beam-columns. Application of this computer program to wood beam-columns will be disccussed in the following chapter.

As shown in Figure 15(a), when k_c = 20.0 and k_t = 5.0 are taken as input into the program, the results are slightly improved with respect to the ones where no size effect was considered. Also, it is noted that size effects in compression have little significance for very slender members, while these size effects play a major role in very short and intermediate members. The reason for this is that for short members, the volume subjected to tension is small or non-existent. For slender members, the failure is controlled by the modulus of elasticity and member instability. When k_c is very large, the results obtained are the same as the ones in Table 2, meaning that there is no size effect for large k_c. It appears from Figure 15(b) that $k_c = 20.0$ gives a best fit to the test results.

5. APPLICATIONS

5.1. INTRODUCTION

The application of the program to solve wood beam-columns will be discussed in this chapter. This program can handle multiple spans with different load and support configurations. Among them are the ones shown in figure 16.

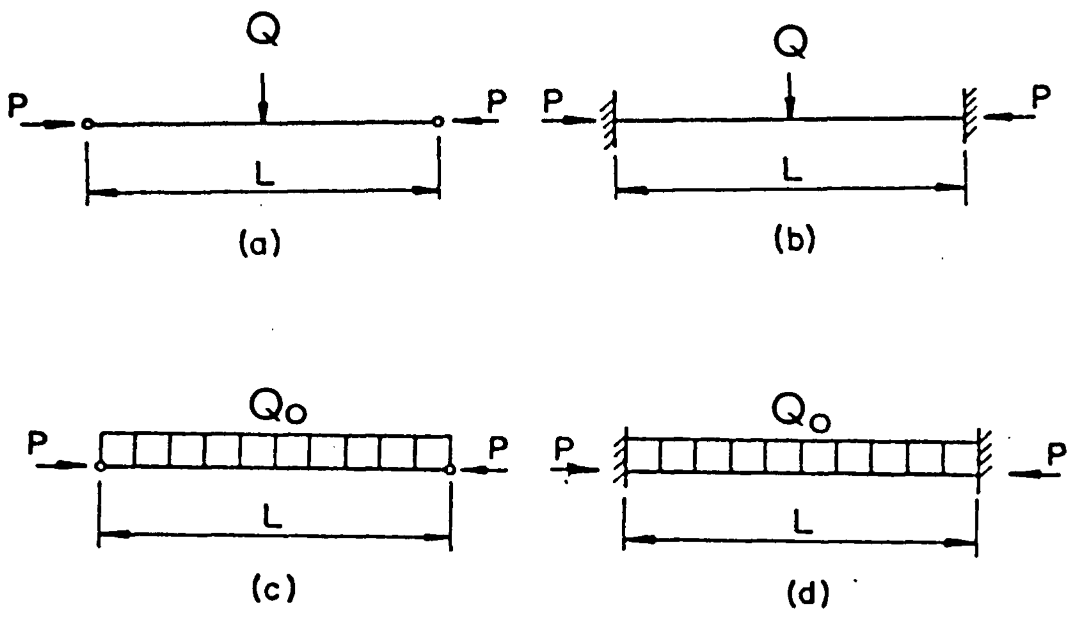


Figure16. Some loading cases and support conditions.

The members are assumed to be prismatic. The desired responses can usually be represented as load versus centre deflection curve or any other convenient way for a particular lateral load Qo. Once the complete curves are obtained, the maximum loads can be easily determined from the peak of the curves. The computer program developed in this study provides an easier approach to the above process in that one gets the maximum load directly by supplying the program with the appropriate information. In all cases of Figure 16, the lateral loads Q or Qo cause bending moments about the major axis of the cross-section. It is further assumed that weak axis buckling and lateral - torsional buckling are effectively prevented so that failure is always caused by excessive bending in the plane of the applied lateral load . In performing the numerical procedure, it is assumed that the lateral load Qo is applied first and maintained at a constant value as the axial compressive load P increases or decreases.

5.2. NUMERICAL EXAMPLE

As a numerical example, case (c) in Figure 16 has been considered in this study, using a 2x4-in SPF section and mean values for E_o, f_c and f_t. Also k_t = 5.0, m = 0.02 and k_c = 10.0 has been used in obtaining the P versus Qo results as shown in Figures 17(a) and 17(b).

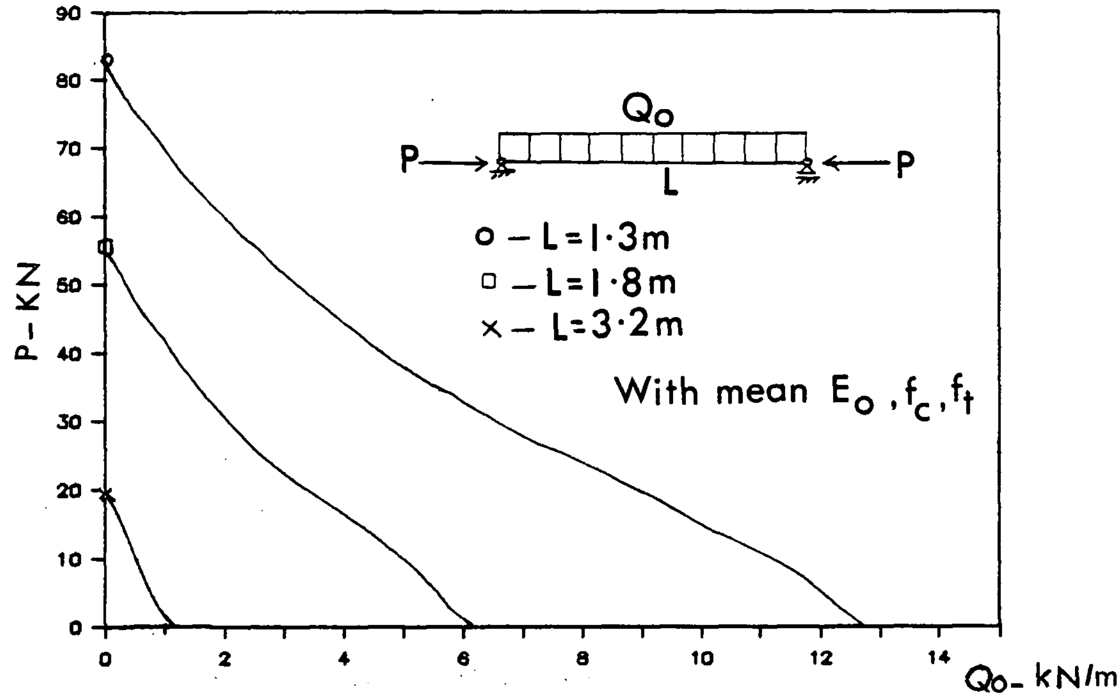


Figure 17(a). Ultimate strength interaction curves for simply supported columns subjected to uniformly distributed load.

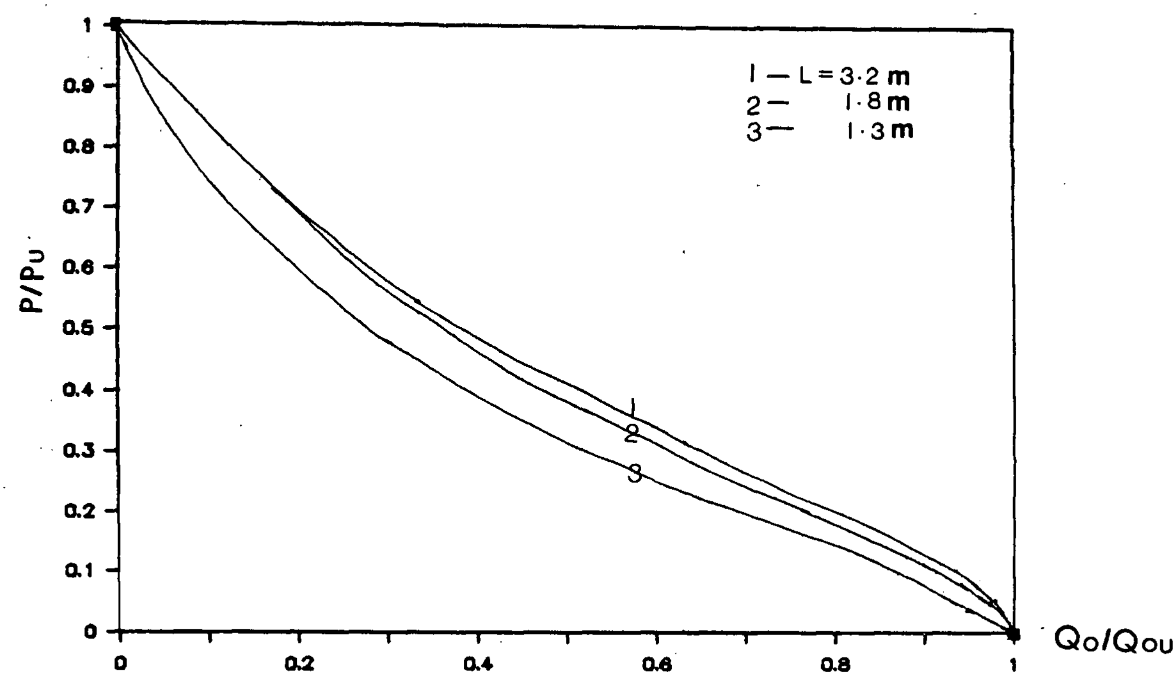


Figure 17(b). Non-dimensionalized ultimate strength interaction curves of Figure 17(a).

5.3. OBSERVATIONS

From Figures 17(a) and 17(b), it can be noticed that P-Qo relationships predicted by the computer model are not a linear one as it is normally assumed in the current design practice for different slenderness ratios. Additional research is needed here in order to come up with a simplified design procedure for wood beam-columns.

6. RELIABILITY ANALYSIS

6.1. INTRODUCTION

This chapter describes the procedure for the structural reliability analysis of a wood compression member. The problem to be studied is as shown in Figure 18; where P represents the applied axial compressive load (for only dead and live loads). L represents the length of the member while H and B represents the height and breadth of the cross section.

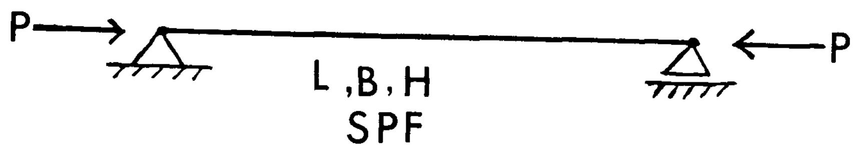


Figure 18. Typical problem for reliability evaluation.

The reliability of a member simply means the probability that it will perform as intended in a prescribed situation. It is influenced by the demands on the structure and the capacity of the structure to respond to those demands. In general, one can define a performance or failure function G to characterize the state of the structure in relation to some performance criterion. This function G can be expressed as

$$G = C - D$$

where

C = structural capacity

D = demands on the structure.

The function G as defined above is positive whenever the capacity exceeds the demand, therefore the structure meets the performance criterion. On the other hand, the function G will be negative whenever the demands exceed the capacity, resulting in the structure not meeting the required performance. When the function G is exactly equal to zero, the structure is on the threshold between meeting and failing to meet the performance criterion, and such a state is defined as "limit state".

The probability of failure p_f is the compliment of the reliability . Thus

$$p_f = 1.0 - \text{reliability}$$

According to the definition of G above, the probability of failure is then given as

$$p_f = \text{Probability } (G < 0)$$

Each design problem will contain a set of intervening variables, and depending on the nature of the problem, some of the variables may be random, obeying some distribution function. Thus, if some of the basic variables are random, it is obvious that G will be itself a random variable. The probability distribution for G could be derived from a knowledge of the individual probability distributions for the basic variables, and the result would be as shown in Figure 19.

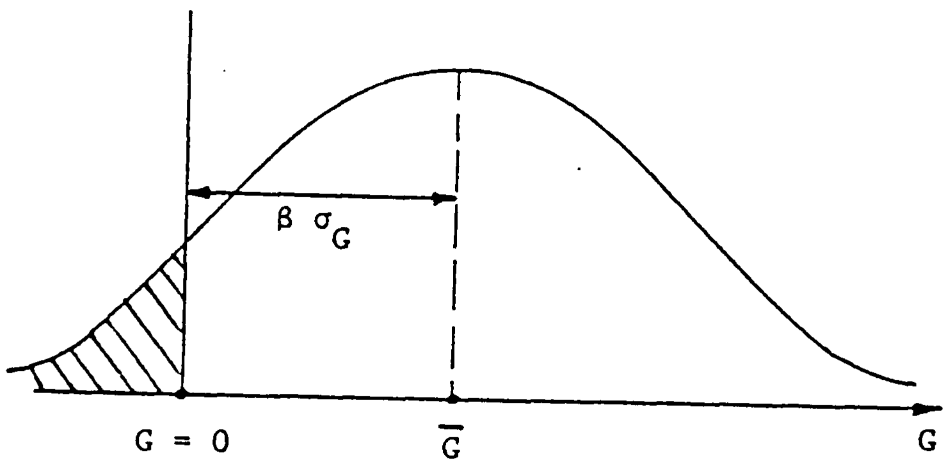


Figure 19. Probability density function for the variable G.

The probability of failure p_f will be the area under the curve to the left of the origin G = 0. If this probability of failure p_f exceeds some desired value, one or more of the design variables would be changed and p_f recalculated until it meets the required target. The probability distribution for G could be obtained by analytical means using multiple integrations and the joint probability distributions between the basic variables. This is a very tedious and difficult approach.

MonteCarlo simulation can be used to obtain the probability of failure in an approximate manner. In this approach the value of G is computed for a large number of combinations of the basic variables and p_f is estimated from the proportion of times the G was negative. The selection of values for the basic variables must obey their joint probability distributions, and when more than two variables are involved, the procedure becomes difficult, tedious and expensive. In the following section, an approximate and fast procedure for estimating p_f will be discussed.

6.2. THE β METHOD FOR RELIABILITY ANALYSIS

In order to estimate the probability of failure p_f with sufficient accuracy but without resorting to complicated integrations or computer simulations, Hasofer and Lind [1974] introduced the concept of reliability index β using geometric approach. Thus, for a design problem containing N uncorrelated random variables X_i, $i = 1,..,N$, with mean $\overline{X}_i$

and standard deviation σ_i, a set of "normalised " variables x_i is introduced. These variables have zero mean and standard deviation equal to 1.0, and are given as

$$x_i = \frac{X_i - \overline{X}_i}{\sigma_i} \tag{53}$$

The failure function can now be expressed in terms of the new, normalised variables x_i as shown schematicaly in the figure below, in which the horizontal plane represents the space of the variables x and the vertical axis the function G.

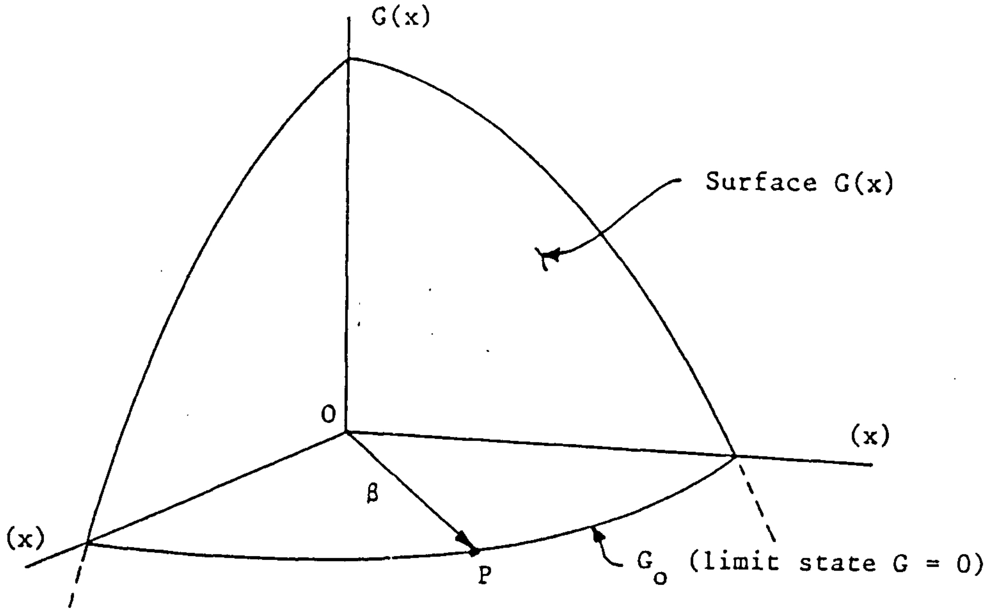


Figure 20. Definition of the reliability index β.

Hasofer and Lind showed that the reliability index β can be intepreted as the minimum distance between the origin O and the limit state G_o. This is a geometric problem which can be solved by successive iterations using, for example, Hasofer and Lind's proposed algorithm. Knowing β, the probability of failure is obtained from

$$p_f = \Phi(-\beta) \tag{54}$$

where Φ is the standardised normal probability function. For p_f to be exact, we require that all the basic variables be normally distributed and G be linear in the basic variables. Figure 20 shows the case when the mean point belongs to the "safe domain" G > 0. The combinations of x_i which correspond to G = 0 (the limit state) are represented by the curve G_o.

6.2.1. Rackwitz-Fiessler Algorithm

This is in actual fact the modification of the Hasofer and Lind Algorithm in order to improve the estimation of the probability of failure. The modification refers to the case when the basic variables are non-normal. Rackwitz and Fiessler [1978], suggested a transformation of the original random variables X_i into a set of normalised uncorrelated standard variables z_i using the following transformation

$$z_i = \Phi^{-1}[F(X_i)] \tag{55}$$

where Φ is the standard normal probability distribution function and $F(X_i)$ is the cumulative distribution function for the variable X_i. The standard algorithm from Hasofer and Lind is then used for the new variable z_i. This modification improves the prediction of p_f because it meets one of the two conditions mentioned earlier namely, that all variables be normally distributed. This algorithm is presently the accepted norm for the evaluation of the reliability index β.

6.3. PROBLEM FORMULATION

To illustrate the applicability of the theory discussed above to normal practice, let us consider the column problem of Figure 18. The cross sectional dimensions of the column are B for width and H for depth. The length of the column is represented as L. It is assumed simply supported under an axial compressive load P (for both dead and live loads). The demand on the structure is the applied load P. Thus,

$$D = P = P_D + P_L$$

where

D = demand

P_D = dead load

P_L = live load

If

$$d = \frac{P_D}{P_{DN}} \tag{56}$$

$$l = \frac{P_L}{P_{LN}} \tag{57}$$

where

P_{LN} = nominal (design) live load

P_{DN} = nominal (design) dead load

Then

$$D = P_{LN}[\gamma_1 d + l] \tag{58}$$

where d and l are considered to be random variables. The factor γ_1 is a constant defined as $\gamma_1 = \frac{P_{DN}}{P_{LN}}$ or the ratio of nominal dead load to nominal live load. The capacity C is the maximum load, Pmax, the member can carry; thus

$$C = Pmax = P\{E_o, fc, ft, B, H, L, m\} \tag{59}$$

and the failure function can be expressed as

$$G = C - D$$

$$G = Pmax - P \tag{60}$$

where

f_c = strength in compression

f_t = strength in tension.

m = slope of stress-strain curve in compression.

The problem can now be studied using the Rackwitz-Fiessler algorithm and the finite element computer program developed in part 3 of this thesis. However, it is convenient for the purpose of future code development to modify Equation (60) above to bring in the design equation format adopted for the code.

6.3.1. Code Design Equation

For members subjected to pure axial compression, the Canadian Code, CAN3-086.1-M84 (1984) specifies the following desing equation.

$$\alpha_D P_{DN} + \alpha_L P_{LN} \leq \phi_p \; A \; F_c \; K_c \tag{61}$$

where

ϕ_p = performance factor in compression.

A = cross sectional area of member.

K_c = slenderness factor

F_c = Fifth percentile compression strength

(α_D, α_L) = load factors (1.25 and 1.5 respectively).

6.4. THE G FUNCTION FOR THE PROBLEM

Considering the limiting case of Equation (61), we obtain the following equation:

$$P_{LN}[a_D\gamma_1 + a_L] = \phi_p \; A \; F_c \; K_c \tag{62}$$

where

$$P_{LN} = \frac{\phi_p AFcKc}{a_D\gamma_1 + a_L} \tag{63}$$

Combining Equations (60), (62) and (63),we can express the failure function as

$$G = Pmax - \frac{\phi_p AFcKc}{a_D\gamma_1 + a_L}[\gamma_1 d + l]$$

or

$$G = P\{E_o, f_c, f_t, B, H, L, m\} - \frac{\phi_p A \; Fc \; Kc}{a_D\gamma_1 + a_L}[\gamma_1 d + l] \tag{64}$$

For the purpose of this study, the following variables have been considered random

modulus of elasticity E_o

compressive strength f_c

tensile strength f_t

dead load variable d

live load variable l

and the following have been considered to be constants with mean average values

height of cross section H

breadth of cross section B

length of member L

slope m.

6.5. THE REALIABILITY PROGRAM

The computer program which implements the derivation above is attached in Appendix A. As part of the input, the program requests the number of random variables (in this case 5), the type of their distribution (according to a distribution code), and the relevant parameter information to characterize the distributions. The program can accept the following distributions

Code	Distribution
1	Normal
2	Lognormal
3	Weibull
4	Gumbel
5	Ranked Data

The fixed parameters γ_1, ϕ_p, a_D and a_L are provided by the user for each particular problem. The subroutine GXPR computes the function G and its gradient by calling the finite element subprogram. The GXPR routine returns the value of G and the gradient vector DELTA. For the column problem discussed in this thesis the elements of the gradient vector corresponding to the first 3 random variables were obtained numerically, while the remaining two were obtained by differentiating the failure function explicitly. Thus, the total elements of the gradient vector considering only five random variables are obtained as

$$\text{Delta}(1) = \frac{G(E_o^{+}) - G(E_o^{-})}{2\Delta E_o}$$

$$\text{Delta}(2) = \frac{G(f_c^{+}) - G(f_c^{-})}{2\Delta f_c}$$

$$\text{Delta}(3) = \frac{G(f_t^{+}) - G(f_t^{-})}{2\Delta f_t}$$

$$\text{Delta}(4) = \frac{\phi_p AFcKc}{a_D\gamma_1 + a_L}\gamma_1$$

$$\text{Delta}(5) = -\frac{\phi_p AFcKc}{a_D\gamma_1 + a_L}$$

The fixed parameters are passed onto the routines GXPR and COLUMN through a COMMON block.

6.6. RELIABILITY RESULTS

Keeping the ratio γ_1 = 1.0, m = 0.02 and using a 2x4-in SPF section, the factor ϕ_p was changed and the corresponding reliability index β was computed for columns of different slenderness ratios. Figure 21 shows the results for the reliability index β as a function of the performance factor ϕ_p for the case of no size effect considered in the program. Figure 22 shows reliability results with size effects included in the computer program. In obtaining the results for the two cases studied, the following information has been used for the random variables.

(1) E_o : 3-parameter Weibull.
Scale = 6738.0 Mpa
Location = 3514.0 Mpa
Shape = 3.97
Mean = 9660.0 Mpa

(2) f_c : 3-parameter Weibull
Scale = 33.845 Mpa
Location = 0.0
Shape = 7.8559 Mpa
Fifth percentile = 15.87 Mpa

(3) f_t : 3-parameter Weibull

Scale = 29.861 Mpa

Location = 4.03 Mpa

Shape = 2.911 Mpa

(4) d = dead load variable : Normal

Mean = 1.0

Standard deviation = 0.15

(5) l = live load variable : normal

Mean = 0.75

Standard deviation = 0.15

Figure 21. Reliability results as a function of ϕ_p; no size effect.

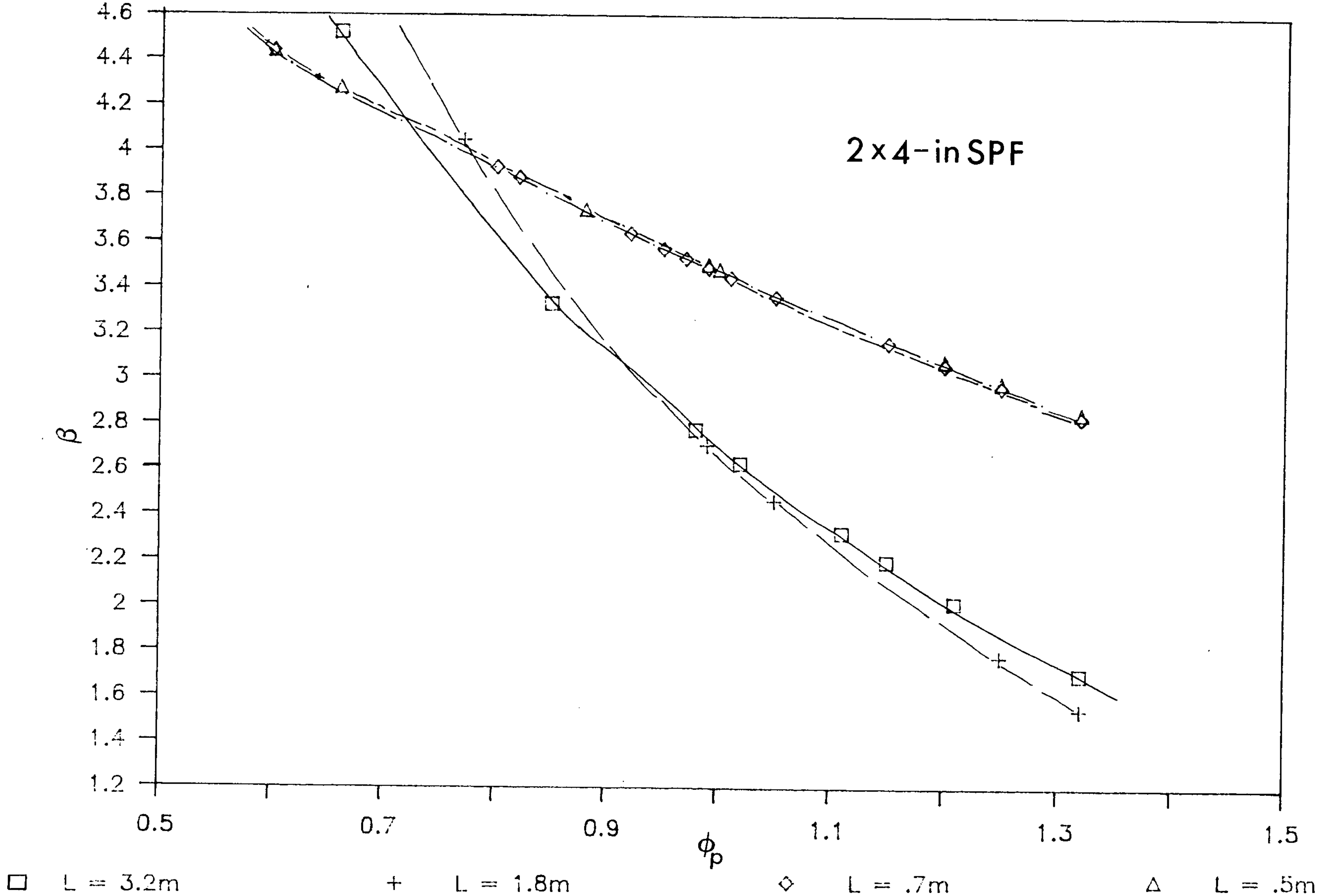
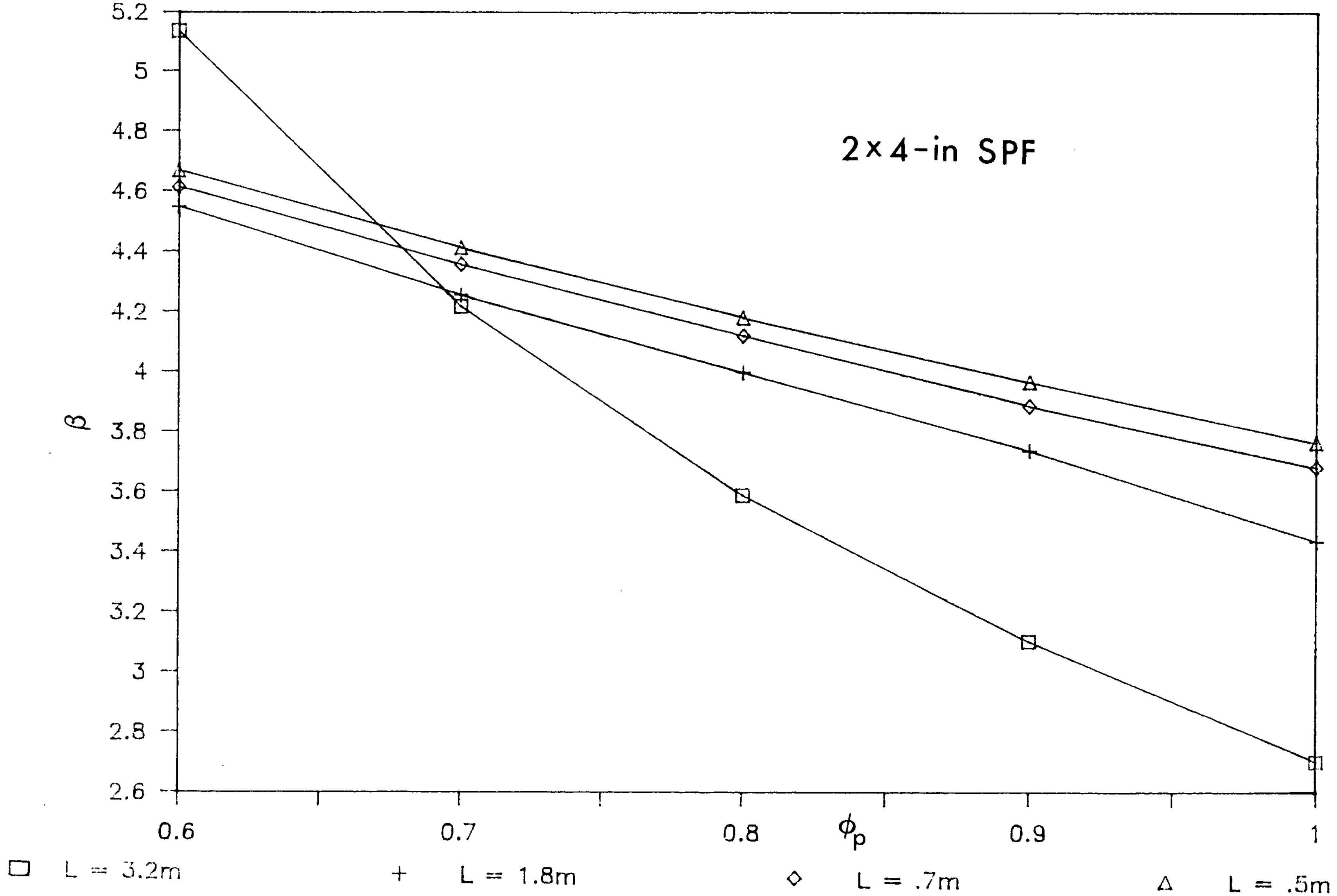


Figure 22. Reliability results with size effect included in the reliability program.



6.6.1. Discussion of results

From the results of Figure 21 it is noted that for a performance factor $\phi_p = 0.75$, a reliability index β of the order $\beta = 4.0$ is achieved for all slenderness ratios considered. Figure 22 shows a small increase of the reliability index for the same performance factor ϕ_p. Considering the two cases, a performance factor $\phi_p = 0.75$ can be taken as a reasonable value to be included in the current design practices for columns of any length.

The procedure outlined in this chapter for the reliability analysis of columns does not take into account the duration of load effect over the length of the servive life of the column. A reliability study for beams taking into account the duration of load effect is currently been done in the Department of Civil Engineering of the University of British Columbia. It will be of interest for further research, to integrate the model developed here to this study.

7. CONCLUSIONS AND RECOMMENDATIONS

7.1. CONCLUSIONS

From the results of part one of this study, it is seen that the finite element analysis, including large deformations and non-linear material properties, can model wood column behaviour satisfactorily. The model does require accurate and reliable input information on modulus of elasticity, compressive and tensile strengths. The results including size effects show that for the size effect parameters $k_c = 20.0$ and $k_t = 5.0$, the computer predictions for the maximum load Pmax agree fairly well with test results. However, k_c does have significance influence for short and intermediate columns, and should be known with some accuracy.

For the reliability results in part two of this study, it is observed that the current performance factors ϕ_p, as given in CAN3-086.1-M84 (1984), are more conservative than what this model predicts. A value of $\phi_p = 0.75$ appears to be a reasonable one for all slenderness ratios. It is estimated that if this new value of ϕ_p is adopted in the current design practice, it will give rise to a reliability index β of the order of 4.0. If a lower β is required, a different ϕ_p should be introduced for short and intermediate columns. This points to a deficiency in the "column formula" giving the slenderness adjustment factor K_c. Idealy, this factor should reflect the changes due to slenderness in such a way that the same ϕ_p - β relationship be obtained for all column lengths.

7.2. RECOMMENDATIONS

It is recommended that a performance factor ϕ_p = 0.75 be used in the current design practice for all slenderness ratios. However, prior to adopting this recommendation, there is need to do more research in this area. In particular, the research should cover duration of load effects, and the case of correlated variables; neither of which has been included in the analysis. The application of the Rackwitz-Fiessler algorithm requires all the variables involved to be uncorrelated. However, in some practical cases some or more of the intervening variables will be correlated. For example, in the context of the problems discussed in this report, the strength of beams, columns or beam-columns under combined axial and lateral loads will depend on the modulus of elasticity E_o, the compression strength f_c and the tensile strength f_t. For lumber, these variables are partially correlated and this must be dealt with, using for example the procedures available in the literature [12], before using the Rackwitz-Fiesler algorithm.

There is not enough data available at present on size effects in both tension and compression, hence further practical as well as theoretical study is necessary in order to come up with a realistic design recommendation applicable to lumber of all grades and species.

REFERENCES

1. Chen,W.F., and Astuta,T., 1976, Theory of Beam Columns, Vol. II, Space Behaviour and Design, McGraw-Hill Book Company, New York, 732p.

2. Zienkiewicz,O., 1971, The Finite Element method in Engineering Science. McGraw-Hill Book Company, London, England.

3. Timoshenko S., and Woinowsky-Krieger S., 1959, Theory of plates and shells. McGraw-Hill Book Company, New York., pp. 396-428.

4. Malhotra, S.K., and Mazur, S.J., 1970, Buckling Strength of solid timber Columns, *Engrg. J.* 13(A-4), I-VII.

5. Larsen,H.J., and Thielgaard, E. (1979)., Laterally loaded timber columns , *J. Struct. Div., ASCE*, 105(7) 1347-1363.

6. Bleau, R., (1983).,Etude de Resistance des Colonnes en bois de Qualite Commerciale, "Thesis presented to the University of Shebrooke at Quebec, in partial fulfilment of the requirements for the degree of Doctor of Philosophy".

7. Buchanan, A. H., Strength Model and design methods for

bending and axial load interaction in timber members., "Thesis presented to the University of British Columbia, at Vancouver, B.C., in partial fulfilment of the requirements for the degree of Doctor of Philosophy".

8. Zann, J.J., (1985)., Strength of lumber under combined bending and compression., "USDA Forest Service Research paper FPL 391., Washington, D.C".

9. Hasofer, A.M., and Lind, N.C., 1974., "Exact and Invariant Second Moment Coe Format", Journal of Engineering Mechanics Division, ASCE, Vol. 100, No. EM1, pp. 111-121.

10. Rackwitz, R., and Fiessler, B. 1978. "Structural Reliability Under Combined Load Sequences", Computers and Structures , Vol. 9, pp. 489-494.

11. Foschi, R.O., 1979., "A discussion on the application of the safety index concept to wood structures", Canadian Journal of Civil Engineering, Vol. 6, No. 1, pp. 51-58.

12. Der Kiureghian, A., 1986., "Structural Reliability Under Incomplete Probability Information", Journal of Engineering Mechanics , ASCE, Vol. 112, No. 1, pp. 85-104.

A P P E N D I X A

1. Program COLUMN.FOR (source code) 82

2. Sample Input/Output Files 96

3. Program RBETA.FOR (source code) 99

4. Sample Input/Output File 119

1. PROGRAM COLUMN.FOR

```
C    **************************************************************
C    *                COLUMN.FOR Version 2.0                      *
C    *                     2 August, 1987                         *
C    *                                                            *
C    *  A PROGRAM FOR THE CALCULATION OF THE ULTIMATE LOAD ON A  *
C    *                 COLUMN (OR BEAM-COLUMN)                    *
C    *                                                            *
C    *  MATERIAL BEHAVIOUR IS ELASTIC IN TENSION WITH BRITTLE     *
C    *  FRACTURE, AND ELASTIC IN COMPRESSION UP TO A LIMITING     *
C    *  COMPRESSION STRESS, WITH A FALLING LINEAR BRANCH BEYOND  *
C    *  THAT LIMIT. SIZE EFFECTS ARE CONSIDERED BOTH IN TENSION  *
C    *  AND COMPRESSION.                                          *
C    *  END LOAD IS A COMPRESSION LOAD.                           *
C    *  END LOAD CAN BE APPLIED ECCENTRICALLY. LATERAL LOADS      *
C    *  CAN BE DISTRIBUTED OR CONCENTRATED.                       *
C    *                                                            *
C    *  THE PROGRAM FINDS THE ULTIMATE END LOAD CORRESPONDING     *
C    *  TO A GIVEN END ECCENTRICITY AND GIVEN LATERAL LOADS.      *
C    *                                                            *
C    *  THE PROGRAM CAN ALSO FIND THE ULTIMATE LATERAL LOAD       *
C    *  WHEN THE END LOAD IS SPECIFIED TO BE ZERO ( NP = 0 ).     *
C    *                                                            *
C    *  PROBLEM DATA IS READ FROM UNIT #1.                        *
C    *  OUTPUT IS STORED IN UNIT #2.                              *
C    *                                                            *
C    *  PROGRAM WRITTEN BY E. KOKA AND R.O. FOSCHI, UBC.          *
C    *                                                            *
C    **************************************************************
C
      IMPLICIT REAL*8(A-H,O-Y)
      CHARACTER*20 NAMED1,NAMEA1,NANS
      DIMENSION IX(21,4),F(8),NBC(21),TKO(672),XE(8)
     1,R(84),XO(84),X(84),B(84),B1(8,8),B2(8,8),B3(8,8),B4(8,8)
     2,B5(8,8),B6(8,8),B7(8,8),B8(8,8),B9(8,8),Y(5),RE(8),XP(84)
     3,Q(20),IQ(20), ESTR(7), FI(7)
      COMMON/C1/GAP(5),GAW(5),EN1(8,5),EM1(8,5),EM2(8,5), NGAUSS
      COMMON/C2/DIFP, NINT
      COMMON/C3/DEFL,PDEFL
C   **********************************************************
C   *           DEFINE        VARIABLES                      *
C   **********************************************************
C   * NELEM    =   NO OF ELEMENTS                            *
C   * NJBC     =   NO OF JOINTS WITH B.C.                    *
C   * NBC(I)   =   NO OF B.C. AT NODE I                      *
C   * IX       =   B.C. CODE                                 *
C   *   1      =   U                                         *
C   *   2      =   UX                                        *
C   *   3      =   W                                         *
C   *   4      =   WX                                        *
C   * EN1      =   INTERPOLATION FUNCTIONS FOR u             *
C   * EM1,EM2  =   INTERPOLATION FUNCTIONS FOR w             *
C   * GAP      =   CORDINATE AT GAUSS POINT                  *
C   * GAW      =   CORRESPONDING WEIGHT                      *
C   * NGAUSS   =   NO OF GAUSS POINTS                        *
C   * NITER    =   MAX. NO OF ITERATIONS                     *
C   * TOP      =   TOLERANCE FOR LOAD STEP                   *
C   * EPSLON   =   TOLERANCE FOR SOLUTION VECTOR             *
C   * FC       =   MATERIAL STRENGTH IN COMPRESSION          *
```

```
C   * FT        =   MATERIAL STRENGTH IN TENSION                        *
C   * E0        =   MOE OF THE MATRIAL (RANDOM)                         *
C   * EN        =   SLOPE OF THE STRESS-STRAIN CURVE IN COMPR.          *
C   * SPAN      =   MEMBER LENGTH                                       *
C   * W         =   WIDTH OF SECTION                                    *
C   * H         =   DEPTH OF SECTION                                    *
C   * E         =   ECCENTRICITY OF AXIAL LOAD                          *
C   * NEQ       =   NO OF EQUATIONS TO BE SOLVED                        *
C   * NJOINT    =   NO OF NODES                                         *
C   * NDOF      =   NO OF VARIABLES PER NODE                            *
C   * NODEL     =   NO OF NODES PER ELEMENT                             *
C   * NDIMB     =   NO OF VARIABLES PER NODE                            *
C   * LBW,LHB   =   HALF BANDWIDTH INCLUD. THE DIAG.                    *
C   * NA        =   NO OF UNKNOWNS FOR TOTAL PROBLEM                    *
C   * RE        =   ELEMENTAL LOAD VECTOR                               *
C   * R         =   STRUCTURE LOAD VECTOR                               *
C   * B         =   GLOBAL LOAD VECTOR RETURNED                         *
C   * TKO       =   GLOBAL TANGENT MATRIX                               *
C   * XE        =   ELEMENT DISPLACEMENT VECTOR                         *
C   * X         =   GLOBAL SOLUTION VECTOR                              *
C   * B1,..B9   =   ARRAYS FOR TEMPORARY STORAGE                        *
C   *****************************************************************
      WRITE(*,6)
 6    FORMAT('  ENTER DATA FILE NAME '/)
      READ(*,8) NAMED1
      WRITE(*,7)
 7    FORMAT('  ENTER OUTPUT FILE  NAME  '/)
      READ(*,8) NAMEA1
 8    FORMAT(A)
       OPEN (1,FILE = NAMED1,STATUS='OLD')
       OPEN (2,FILE = NAMEA1,STATUS='NEW')
      READ(1,*) NELEM, NGAUSS
       NDOF = 4
       NJOINT = NELEM+1
       NG1 = NGAUSS + 1
       NG2 = NGAUSS + 2
      READ (1,*) NP,NQ,Q0
      IF (NQ.EQ.0) GO TO 44
      DO 43 I = 1, NQ
43    READ (1,*) IQ(I), Q(I)
44     E = 0.0D0
       IF (NP.NE.0) READ(1,*) E
       DO 65 I = 1, NJOINT
65     NBC(I)=0
      READ (1,*) NJBC
      DO 75 I = 1, NJBC
       READ (1,*) N, NBC(N)
       READ (1,*) (IX(N,J),J=1,NBC(N))
75     CONTINUE
       NEQ = NDOF*NJOINT
       NODEL = 2
C
C * READS MATERIAL STRENGTH IN COMPRESSION (FC) AND TENSION (FT),
C   BOTH CORRESPONDING TO THE SPECIFIED CROSS-SECTION AND THE
C   REFERENCE SPAN SREF. XKC AND XKT ARE THE WEIBULL SIZE EFFECT
C   SHAPE PARAMETERS IN COMPRESSION AND TENSION RESPECTIVELY.
C
      READ(1,*) FC,FT,SREF,XKC, XKT
```

```
      NDIMB = NODEL*NDOF
       LBW = NDIMB
       LHB = LBW
       NA = LBW*NEQ
C     READ MOE AND SLOPE m OF CURVE
      READ(1,*) E0,EN
C     READ PROBLEM SIZE L, B, H
      READ(1,*) SPAN,W,H
       AR = W*H
       XI = W*H**3/12.D0
       DEL = SPAN/(2.D0*NELEM)
       SLAMDA = SPAN/H
C      * ADJUST STRENGTHS TO THE ACTUAL VOLUME
       FC = FC *(SREF/SPAN)**(1.0/XKC)
       FT = FT *(SREF/SPAN)**(1.0/XKT)
C    OBTAIN SHAPE FUNCTIONS AND DERIVATIVES : N1,M1,M2
      CALL SHAPE(DEL)
      WRITE(*,79)
79    FORMAT('  TOLERANCE   FOR  PMAX  ? '/)
      READ(*,*) TOP
      WRITE(*,790)
790   FORMAT(' TOLERANCE FOR CONVERGENCE? '/)
      READ(*,*) EPSLON
      WRITE(*,791)
791   FORMAT(' MAX. NUMBER OF ITERATIONS? '/)
      READ(*,*) NITER
      WRITE(*,799)
799   FORMAT(' WANT TO SEE INTERMEDIATE RESULTS? (Y/N)'/)
      READ(*,8) NANS
      NINT = 0
      IF (NANS.EQ.'Y'.OR.NANS.EQ.'y') NINT = 1
      IF (NP.EQ.0) GO TO 761
      PC = AR*FC
      PCR = 3.14159D0**2*E0*XI/(SPAN**2)
      PI = PC
      IF(PCR .LE. PC) PI=PCR
       P2 = PI
       P1 = 0.0D0
      P3 = (P1 + P2)/2.0D0
      NFAIL = 0
      SMAX1 = 0.0
      GO TO 760
761   FQ1 = 0.0D0
      FQ2 = 1.0D0
      FQ3 = FQ2
      NFLAG = 0
760   DO 792 J = 1, NEQ
792   XP(J) = 0.0D0
C
C     START CALCULATIONS FOR TRIAL LOAD LEVELS
      CALL TIME(ZIM)
      ZIM0 = ZIM
3773  CONTINUE
       P = 0.0D0
       FQ = 1.0D0
       IF (NP.NE.0) P = P3
       IF (NP.EQ.0) FQ = FQ3
       IF (NINT.EQ.1.AND.NP.NE.0) WRITE(*,4000) P
```

```
        IF (NINT.EQ.1.AND.NP.EQ.0) WRITE(*,4001) FQ
4000   FORMAT(//' SOLUTION FOR P =',E15.6,' :'/)
4001   FORMAT(//' SOLUTION FOR LATERAL LOAD FACTOR=',E15.6,':'/)
C     INITIALISE ARRAYS
       DO 80 J = 1, NEQ
       XO(J) = XP(J)
80     R(J) = 0.D0
C     EXTERNAL LOAD VECTOR R
       IF (Q0.EQ.0.0D0) GO TO 87
       DO 81 J = 1, 8
       RE(J)=0.D0
81     CONTINUE
       RE(3) = FQ*Q0*DEL
       RE(4) = FQ*Q0*DEL**2/3.D0
       RE(7) = RE(3)
       RE(8) = -RE(4)
      DO 83 NE = 1, NELEM
      DO 82 JJ = 1, 8
       K = (NE-1)*NDOF + JJ
       R(K) = R(K) + RE(JJ)
82    CONTINUE
83    CONTINUE
87    IF (NQ.EQ.0) GO TO 185
       DO 180 J = 1, NQ
       JS = (IQ(J)-1)*NDOF + 3
180    R(JS) = R(JS) + Q(J)*FQ
185    EM = P*E
       JJ = (NJOINT-1)*NDOF + 1
       R(JJ) = R(JJ)-P
       R(1) = R(1) + P
       R(4)=R(4)-EM
       R(NEQ)=R(NEQ)+EM
      ITER=0
C
C      BEGIN ITERATIONS AT THE TRIAL LOAD LEVEL
777    CONTINUE
       DO 84  I = 1, NA
84      TKO(I) = 0.0D0
        DO 85 K = 1, NEQ
85      B(K) = -R(K)
       DO 645 IE = 1, NELEM
C   INITIALIZE ARRAYS
       DO 88 I = 1, 8
        F(I) = 0.0D0
       DO 86 J = 1, I
        B1(I,J) = 0.0D0
        B2(I,J) = 0.0D0
        B3(I,J) = 0.0D0
        B4(I,J) = 0.0D0
        B5(I,J) = 0.0D0
        B6(I,J) = 0.0D0
        B7(I,J) = 0.0D0
        B8(I,J) = 0.0D0
        B9(I,J) = 0.0D0
86     CONTINUE
88     CONTINUE
C       PICK ELEMENT SOLUTION FROM GLOBAL VECTOR
       DO 90 JJ = 1, 8
```

```
         K = (IE - 1)*NDOF + JJ
         XE(JJ) = XO(K)
90      CONTINUE
        DO 101 K = 1, NGAUSS
         Y(K) = 0.D0
        DO 91 I=1, 8
         Y(K) = Y(K)+XE(I)*EM1(I,K)
91      CONTINUE
C       OBTAINING COMPONENTS OF EKT
        DO 93 I = 1, 8
        DO 93 J = 1, I
         B1(I,J) = B1(I,J)+E0*DEL*EN1(I,K)*Y(K)*AR*
     1  EM1(J,K)*GAW(K)
         B2(I,J) = B2(I,J)+E0*DEL*EM1(I,K)*Y(K)*AR*
     1  EN1(J,K)*GAW(K)
         B3(I,J) = B3(I,J)+E0*DEL*EM1(I,K)*Y(K)*AR*
     1  Y(K)*EM1(J,K)*GAW(K)
         B4(I,J) = B4(I,J)+(E0*AR*DEL*EN1(I,K)*EN1(J,K)+
     1  E0*XI*DEL*EM2(I,K)*EM2(J,K))*GAW(K)
93      CONTINUE
        DO 100 L = 1, NGAUSS
C       STRESSES AND STRAINS AT GAUSS POINT
        STR = 0.5D0*Y(K)**2
         DO 96 MO = 1, 8
        STR = STR+(EN1(MO,K)-GAP(L)*H*0.5D0*EM2(MO,K))*XE(MO)
96      CONTINUE
        STRE = STR+FC/E0
        FAC = 1.0D0
        IF(STRE.GE.0.D0) FAC=0.0D0
         STRESS = E0*STR-((E0+EN*E0)*STR+FC*(1.D0+EN))*FAC
         DO 99 I = 1, 8
         DO 98 J = 1, I
        B5(I,J) = B5(I,J)+DEL*0.5D0*AR*(EN1(I,K)-GAP(L)*
     1  H*0.5D0*EM2(I,K))*(E0+E0*EN)*FAC*(EN1(J,K)-H*0.5D0*
     2  GAP(L)*EM2(J,K))*GAW(K)*GAW(L)
         B6(I,J) = B6(I,J)+DEL*0.5D0*AR*(EN1(I,K)-GAP(L)*
     1   H*0.5D0*EM2(I,K))*(E0+EN*E0)*FAC*Y(K)*EM1(J,K)*
     2  GAW(K)*GAW(L)
         B7(I,J) = B7(I,J)+DEL*0.5D0*EM1(I,K)*Y(K)*AR*
     1  (E0+E0*EN)*FAC*(EN1(J,K)-H*0.5D0*GAP(L)*EM2(J,K))*
     2   GAW(K)*GAW(L)
        B8(I,J) = B8(I,J)+DEL*0.5D0*EM1(I,K)*Y(K)*AR*
     1  (E0+E0*EN)*FAC*Y(K)*EM1(J,K)*GAW(K)*GAW(L)
        B9(I,J) = B9(I,J)+AR*STRESS*EM1(I,K)*EM1(J,K)*
     1  GAW(K)*GAW(L)*DEL*0.5D0
98      CONTINUE
        F(I) = F(I)+AR*DEL*0.5D0*STRESS*((EN1(I,K)-H*0.5D0*
     1  GAP(L)*EM2(I,K))+Y(K)*EM1(I,K))*GAW(K)*GAW(L)
99      CONTINUE
100     CONTINUE
101     CONTINUE
C      OBTAIN ELEMENT TANGENT MATRIX
C      EKT IS THE (I,J) COMPONENT OF THE ELEMENT TANGENT MATRIX
        DO 105 I = 1, 8
        II = (IE-1)*NDOF + I
        B(II) = B(II) + F(I)
        DO 102 J = 1, I
        JJ = (IE-1)*NDOF + J
```

```
      EKT = B1(I,J)+B2(I,J)+B3(I,J)+B4(I,J)-
     1 B5(I,J)-B6(I,J)-B7(I,J)-B8(I,J)+B9(I,J)
      IJ = (JJ-1)*(LBW-1) + II
      TKO(IJ) = TKO(IJ)+EKT
102   CONTINUE
105   CONTINUE
645   CONTINUE
C    INTRODUCE BOUNDARY CONDITIONS
      DO 111 IJO = 1, NJOINT
      IF (NBC(IJO).EQ.0) GO TO 111
      DO 110 J = 1, NBC(IJO)
      II = (IJO -1)*NDOF + IX(IJO,J)
      LBW1 = LBW - 1
      DO 108 K = 1, LBW1
       JJ = II - LBW + K
       IF (JJ.LE.0) GO TO 1080
       IJ = (JJ-1)*(LBW-1) + II
      TKO(IJ) = 0.0D0
1080   JJ = II + K
       IF (JJ.GT.NEQ) GO TO 108
       IJ = (II-1)*(LBW-1) + JJ
      TKO(IJ) = 0.0D0
108   CONTINUE
       IJ = (II - 1)*(LBW-1) + II
      TKO(IJ) = 1.0D0
       B(II) = 0.0D0
110   CONTINUE
111   CONTINUE
C
C     SOLUTION OF THE SYSTEM
C
      CALL DECOMP(NEQ,LBW,TKO,IERROR)
      IF(IERROR .EQ. 1) GO TO 3774
      CALL SOLVN(NEQ,LBW,TKO,B)
      DO 112 I = 1, NEQ
      X(I) = XO(I)-B(I)
112   CONTINUE
      CALL CONVRG(XO,X,IER,NEQ,EPSLON,ITER)
      ITER = ITER + 1
      IF (ITER.EQ.NITER) GO TO 431
      IF (IER.EQ.2) GO TO 430
      IF(IER.EQ.0) GO TO 118
      DO 115 I = 1, NEQ
115   XO(I) = X(I)
      GO TO 777
430   IERROR = 1
      GO TO 3774
431   WRITE(2,900) NITER, P
900   FORMAT(' NO CONVERGENCE IN',I3,'ITERATIONS AT P=',E13.6/)
      GO TO 901
C
C    AFTER CONVERGENCE, OBTAIN STRESSES AND STRAINS AT
C    THE CURRENT LOAD LEVEL
C
118   CONTINUE
      EMAXP = 0.0D0
      EMAXN = 0.0D0
      SUME = 0.0D0
```

```
      DO 550 IE = 1, NELEM
      DO 500 J = 1, 8
      K = (IE-1)*NDOF +J
      XE(J) = X(K)
500   CONTINUE
      DO 540 K = 1, NGAUSS
       FACTOR = 0.0
      DO 501 I = 1, 8
501   FACTOR = FACTOR + XE(I)*EM1(I,K)
      EPLUS = 0.5D0 * FACTOR**2
      EMINUS = EPLUS
      DO 505 I = 1, 8
       EPLUS = EPLUS + (EN1(I,K)-H*0.5D0*EM2(I,K))*XE(I)
       EMINUS = EMINUS + (EN1(I,K)+H*0.5D0*EM2(I,K))*XE(I)
505   CONTINUE
       IF(EPLUS.GT.0.0D0 .AND. EMINUS.GT.0.0) GO TO 506
       IF(EPLUS.GT.0.0D0 .AND. EMINUS.LE.0.0) GO TO 507
       IF(EPLUS.LE.0.0D0 .AND. EMINUS.LE.0.0) GO TO 508
       IF(EPLUS.LE.0.0D0 .AND. EMINUS.GT.0.0) GO TO 509
506   EPOS = EPLUS
      IF(EMINUS.GT.EPOS) EPOS=EMINUS
      ENEG = 0.0D0
      GO TO 530
507    EPOS = EPLUS
       ENEG = EMINUS
      GO TO 510
508   EPOS = 0.0D0
      ENEG = EPLUS
      IF (DABS(EMINUS).GT.DABS(ENEG)) ENEG = EMINUS
      GO TO 530
509    EPOS = EMINUS
      ENEG = EPLUS
C
C     * FINDS THE POSITION OF THE NEUTRAL AXIS
C
510   ESTR(1) = EMINUS
      FI(1) = -1.0D0
      ESTR(NG2) = EPLUS
      FI(NG2) = 1.0D0
      DO 512 L = 1, NGAUSS
      SUM = 0.5*FACTOR**2
      DO 511 I = 1,8
511   SUM = SUM + (EN1(I,K) - GAP(L)*H/2.0*EM2(I,K))*XE(I)
      ESTR(L+1) = SUM
      FI(L+1) = GAP(L)
512   CONTINUE
      DO 515 I = 1, NG1
      PROD = ESTR(I)*ESTR(I+1)
      IF (PROD.LE.0.0D0) GO TO 516
515   CONTINUE
516   XN = FI(I) - ESTR(I)*(FI(I+1)-FI(I))/(ESTR(I+1)-ESTR(I))
      IF (ESTR(I).EQ.0.0D0) GO TO 518
      IF (ESTR(I).LT.0.0D0) HN = (1.0D0 - XN)*H/2.0D0
      IF (ESTR(I).GT.0.0D0) HN = (1.0D0 + XN)*H/2.0D0
      GO TO 520
518   IF (ESTR(I+1).LT.0.0D0) HN = (1.0D0 + XN)*H/2.0D0
      IF (ESTR(I+1).GT.0.0D0) HN = (1.0D0 - XN)*H/2.0D0
520   SUME = SUME + (HN/H)*(E0*EPOS)**XKT*GAW(K)
```

```
530    IF(EPOS.LT.EMAXP) GO TO 538
         EMAXP = EPOS
538    IF(DABS(ENEG).LT.DABS(EMAXN)) GO TO 540
         EMAXN = ENEG
540     CONTINUE
550     CONTINUE
         SMAXP = E0*EMAXP
         SMAXN = E0*EMAXN
         IF (DABS(SMAXN).LE.FC) GO TO 560
         SMAXN = SMAXN -((E0 + EN*E0)*EMAXN + FC*(1.0 + EN))
560      IF (SUME.EQ.0.0D0.OR.SMAXP.EQ.0.0D0) GO TO 563
         SUME = SUME/(2.0*NELEM*(XKT+1.0)*SMAXP**XKT)
         FTT = FT * SUME**(-1.0D0/XKT)
         GO TO 564
563      FTT = FT
564      IF (SMAXP.GE.FTT) GO TO 3774
         DEFL = 0.0D0
         DO 565 IE = 1, NELEM
         J = (IE-1)*NDOF + 3
         IF (DABS(X(J)).GT.DABS(DEFL)) DEFL = X(J)
565      CONTINUE
         J = NEQ - 1
         IF (DABS(X(J)).GT.DABS(DEFL)) DEFL = X(J)
         IF (NP.EQ.0) PDEFL = FQ3
         IF (NP.NE.0) PDEFL = P3
3774    CONTINUE
         IF (NINT.EQ.0) GO TO 8810
         IF (IERROR.EQ.1) WRITE(*,8888)
         IF (IERROR.EQ.0.AND.SMAXP.LT.FTT) WRITE(*,8889) SMAXP
         IF (IERROR.EQ.0.AND.SMAXP.GE.FTT) WRITE(*,8890) SMAXP
8888     FORMAT(' IERROR=1,FAILS (DIVERGENCE OR SINGULAR MATRIX)'/)
8889     FORMAT(' IERROR=0 SMAXP = ',E15.6,'  ----- SURVIVES'/)
8890     FORMAT(' IERROR=0 SMAXP = ',E15.6,'  ----- FAILS'/)
8810     CONTINUE
         IF (NP.EQ.0) GO TO 4500
         IF (IERROR.EQ.1) GO TO 7330
         IF (SMAXP.GT.FTT) GO TO 7331
         IF (SMAXP.EQ.FTT) GO TO 7337
         P1 = P3
         IF (SUME.EQ.0.0D0.OR.SMAXP.EQ.0.0D0) GO TO 5650
         SMAX1 = SMAXP*SUME**(1.0D0/XKT)
         GO TO 5655
5650     SMAX1 = SMAXP
5655     DO 833 J = 1, NEQ
833      XP(J) = X(J)
         GO TO 8334
7330     P2 = P3
         GO TO 8334
7331     P2 = P3
         NFAIL = 1
         SMAX2 = SMAXP*SUME**(1.0D0/XKT)
8334     IF (P1.EQ.0.0D0) GO TO 8338
        TOLP = (P2-P1)/P1
        IF (TOLP.LE.TOP) GO TO 7338
        GO TO 8336
8338    IF (P2.LE.0.1D0) GO TO 7338
8336    IF (NFAIL.EQ.1) GO TO 8340
        P3 = (P1 + P2)/2.0
```

```
      GO TO 3773
8340  P3 = P1 + (P2-P1)*(FT-SMAX1)/(SMAX2-SMAX1)
       GO TO 3773
7337  P = P3
      PP = P3
      PAV = P3
      GO TO 7339
7338  IF (P1.EQ.0.0D0) P2 = 0.0D0
      P = P2
      PP = P1
      PAV = (P+PP)/2.0
7339  CALL TIME(ZIM)
      ZIM = ZIM - ZIMO
     WRITE(2,570) PP,P,PAV,SMAXP,SMAXN,DEFL,PDEFL,SLAMDA
570  FORMAT(2X,' FAILURE BETWEEN LOADS ',E15.6,2X,' AND',
    12X,E15.6/' AVERAGE=',E15.6/' EDGE STRESS (+) =',E15.6/
    2' EDGE STRESS (-) =',E15.6/' MAX. DEFLECTION =',E15.6,
    3 '  AT LOAD =',E15.6/' SLENDERNESS = ',F6.2/)
      WRITE(*,683) ZIM
683   FORMAT(' TIME =',F7.1,' SECS.'/)
      WRITE(*,570) PP,P,PAV,SMAXP,DEFL,PDEFL,SLAMDA
      GO TO 901
4500  IF (IERROR.EQ.1) GO TO 4330
       IF (SMAXP.GT.FTT) GO TO 4330
       IF (SMAXP.EQ.FTT) GO TO 4337
       IF (NFLAG.EQ.1) GO TO 4331
       FQ1 = FQ2
       FQ2 = 2.0D0*FQ2
       GO TO 4580
4331   FQ1 = FQ3
4580   DO 4833 J = 1,NEQ
4833   XP(J) = X(J)
       GO TO 4334
4330   NFLAG = 1
       FQ2 = FQ3
4334   IF (FQ1.EQ.0.0D0) GO TO 5338
       TOLP = (FQ2-FQ1)/FQ1
       IF (TOLP.LE.TOP) GO TO 4338
5338   IF (NFLAG.EQ.0) FQ3 = FQ2
       IF (NFLAG.EQ.1) FQ3 = (FQ1+FQ2)/2.0D0
       GO TO 3773
4337   P = FQ3
       PP = FQ3
       PAV = FQ3
       GO TO 4339
4338   P = FQ2
       PP = FQ1
       PAV = (P+PP)/2.0
4339   CALL TIME(ZIM)
       ZIM = ZIM-ZIMO
     WRITE(2,670) PP,P,PAV,SMAXP,SMAXN,DEFL,PDEFL
670  FORMAT(2X,' FAILURE BETWEEN LOAD FACTORS ',E15.6,2X,' AND',
    12X,E15.6/' AVERAGE =',E15.6/' EDGE STRESS (+) =',E15.6/
    2' EDGE STRESS (-) =',E15.6/' MAX. DEFLECTION = ',E15.6,
    3' AT LOAD FACTOR =',E15.6/)
      WRITE(*,683) ZIM
      WRITE(*,670) PP,P,PAV,SMAXP,SMAXN,DEFL,PDEFL
901   CONTINUE
```

```
3131   CONTINUE
       CLOSE (1,STATUS='KEEP')
       CLOSE (2,STATUS='KEEP')
       STOP
       END
C
C     END OF MAIN PROGRAM
C
       SUBROUTINE SHAPE(DEL)
C*    THIS SUBROUTINE CALCULATES DERIVATIVES OF SHAPE FUNCTIONS
       IMPLICIT REAL*8(A-H,O-Y)
       COMMON/C1/GAP(5),GAW(5),EN1(8,5),EM1(8,5),EM2(8,5),NGAUSS
       IF (NGAUSS.EQ.5) GO TO 5
       IF (NGAUSS.EQ.4) GO TO 4
C      *** 3 POINT GAUSSIAN INTEGRATION
       GAP(1) = -0.774596669241483D0
       GAP(2) = 0.0D0
       GAP(3) = -GAP(1)
       GAW(1) = 0.555555555555556D0
       GAW(2) = 0.888888888888889D0
       GAW(3) = GAW(1)
       GO TO 10
C      *** 4 POINT GAUSSIAN INTEGRATION
4      GAP(1) = -0.861136311594053D0
       GAP(2) = -0.339981043584856D0
       GAP(3) = -GAP(2)
       GAP(4) = -GAP(1)
       GAW(1) = 0.347854845137454D0
       GAW(2) = 0.652145154862546D0
       GAW(3) = GAW(2)
       GAW(4) = GAW(1)
       GO TO 10
C      *** 5 POINT GAUSSIAN INTEGRATION
5      GAP(1) = -0.906179845938664D0
       GAP(2) = -0.538469310105683D0
       GAP(3) = 0.0D0
       GAP(4) = -GAP(2)
       GAP(5) = -GAP(1)
         GAW(1) = 0.236926885056189D0
         GAW(2) = 0.478628670499366D0
         GAW(3) = 0.568888888888889D0
         GAW(4) = GAW(2)
         GAW(5) = GAW(1)
C      INITIALISES EN1,EM1,EM2
10     DO 150 IL = 1, 8
       DO 350 IK = 1, NGAUSS
         EN1(IL,IK) = 0.0D0
         EM1(IL,IK) = 0.0D0
         EM2(IL,IK) = 0.0D0
350    CONTINUE
150    CONTINUE
       DO 250 I = 1, NGAUSS
         EN1(1,I) = (-0.75D0+0.75D0*GAP(I)**2)/DEL
         EN1(2,I) = (-1.D0-2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
         EN1(5,I) = (0.75D0-0.75D0*GAP(I)**2)/DEL
         EN1(6,I) = (-1.D0+2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
          EM1(3,I) = (-0.75D0+0.75D0*GAP(I)**2)/DEL
          EM1(4,I) = (-1.D0-2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
```

```
            EM1(7,I) = (0.75D0-0.75D0*GAP(I)**2)/DEL
            EM1(8,I) = (-1.D0+2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
            EM2(3,I) = 1.5D0*GAP(I)/(DEL**2)
            EM2(4,I) = (-2.D0+6.D0*GAP(I))/(4.D0*DEL)
            EM2(7,I) = -1.5D0*GAP(I)/(DEL**2)
            EM2(8,I) = (2.D0+6.D0*GAP(I))/(4.D0*DEL)
250     CONTINUE
        RETURN
        END
C
        SUBROUTINE DECOMP(NN,LHB,AA,IERROR)
C * THIS SUBROUTINE DECOMPOSES A MATRIX USING CHOLESKY
C   METHOD FOR BANDED,SYMMETRIC,POS. DEFN. MATRICES
        IMPLICIT REAL*8(A-H,O-Y)
        DIMENSION AA(672)
C       TKO IS STORED COLUMNWISE.
         IERROR = 0
         KB = LHB-1
C       DECOMPOSITION
          IF(AA(1).LE.0.D0) IERROR=1
          IF(IERROR.EQ.1) RETURN
          AA(1) = DSQRT(AA(1))
          IF(NN.EQ.1) RETURN
        DO 551 I = 2, LHB
551       AA(I) = AA(I)/AA(1)
        DO 590 J = 2, NN
          J1 = J-1
          IJD = LHB*J-KB
          SUM = AA(IJD)
          KO = 1
          IF(J.GT.LHB) KO=J-KB
        DO 555 K = KO, J1
          JK = KB*K+J-KB
555       SUM = SUM-AA(JK)*AA(JK)
          IF(SUM.LE.0.D0) IERROR=1
          IF(IERROR.EQ.1) RETURN
          AA(IJD) = DSQRT(SUM)
        DO 568 I = 1, KB
          II = J+I
          KO = 1
        IF (II.GT.LHB) KO=II-KB
        SUM = AA(IJD+I)
         IF(I.EQ.KB) GO TO 565
        DO 540 K = KO, J1
         JK = KB*K+J-KB
         IK = KB*K+II-KB
540     SUM = SUM-AA(IK)*AA(JK)
565     AA(IJD+I) = SUM/AA(IJD)
568      CONTINUE
590      CONTINUE
        RETURN
        END
C
       SUBROUTINE SOLVN(NN,LHB,AA,S)
C* THIS SUBROUTINE SOLVES THE SYSTEM OF EQUATIONS USING
C  THE DECOMPOSED MATRIX FROM THE PREVIOUS SUBROUTINE
        IMPLICIT REAL*8(A-H,O-Y)
        DIMENSION AA(672),S(84)
```

```
C      FORWARD SUBSTITUTION
        KB = LHB-1
        S(1) = S(1)/AA(1)
       IF(NN.EQ.1) GO TO 685
       DO 680 I = 2, NN
        I1 = I-1
        KO = 1
       IF(I.GT.LHB) KO=I-KB
        SUM = S(I)
        II = LHB*I-KB
       DO 675 K = KO, I1
        IK = KB*K+I-KB
675     SUM = SUM-AA(IK)*S(K)
        S(I) = SUM/AA(II)
680    CONTINUE
C      BACKWARD SUBSTITUTION
685     N1 = NN-1
        LB = LHB*NN-KB
        S(NN) = S(NN)/AA(LB)
       IF(NN.EQ.1) RETURN
       DO 699 I = 1, N1
       I1 = NN-I+1
       NI = NN-I
       KO = NN
      IF (I.GT.KB) KO=NI+KB
       SUM = S(NI)
      II = LHB*NI-KB
       DO 690 K = I1, KO
       IK = KB*NI+K-KB
690   SUM = SUM-AA(IK)*S(K)
      S(NI) = SUM/AA(II)
699   CONTINUE
      RETURN
      END
C
      SUBROUTINE CONVRG(XO,X,IER,NEQ,EPSLON,ITER)
C*  THIS SUBROUTINE CHECKS THE CONVERGENCE
C   OF SOLUTION VECTOR
       IMPLICIT REAL*8(A-H,O-Y)
       COMMON/C2/DIFP,NINT
       DIMENSION XO(84),X(84)
        IER = 0
        PARX0 = 0.0D0
        PARDIF = 0.0D0
        PARX = 0.0D0
       DO 602 I = 1, NEQ
        PARX0 = PARX0 + XO(I)**2
        PARX = PARX + X(I)**2
602     PARDIF = PARDIF + (X(I)-XO(I))**2
       IF (NINT.EQ.1) WRITE(*,1002)  PARX0, PARX, PARDIF
1002   FORMAT(' NORMX0=',E13.6,'NORMX=',E13.6,'NORMDIF=',E13.6/)
        IF (ITER.EQ.0) GO TO 606
        IF (PARDIF.GE.DIFP) GO TO 605
606     DIFP = PARDIF
        IF (PARX0.EQ.0.0D0) GO TO 603
        DIF = DSQRT(PARDIF/PARX0)
        IF (DIF.LE.EPSLON) GO TO 604
603     IER = 1
```

```
       RETURN
604   RETURN
605    IER = 2
       RETURN
      END
     SUBROUTINE TIME(TIM)
     CALL GETTIM(IH,IM,IS,IHS)
     TIM = IH*3600 + IM*60 + IS + IHS/100.0
     RETURN
     END
```

2. SAMPLE INPUT/OUTPUT FILES.

SAMPLE INPUT DATA FILE FOR AXIAL COMPRESSION

```
10 5 1 10
1 0 0.0
-0.001
2
1 2
1 3
11 1
3
32300.0 30350.0 2.0 10.0 5.0
9660000.0 -1.0
3.2 0.038 0.089
```

SAMPLE OUTPUT FILE

```
FAILURE BETWEEN LOADS  0.199730e+02  AND   0.201354e+02
AVERAGE =   0.200542e+02
EDGE STRESS (+) =   0.701410e+04
EDGE STRESS (-) =  -0.188233e+05
MAX. DEFLECTION =   0.312672e-01 AT LOAD = 0.199730e+02
SLENDERNESS =  35.96
```

SAMPLE INPUT DATA FILE FOR PURE BENDING

```
8 5 10 1
0 1 0.0
5 1.0
2
1 2
1 3
9 1
3
32300.0 30350.0 2.0 10.0 5.0
9660000.0 -1.0
3.2 0.038 0.089
```

SAMPLE OUTPUT FILE

```
FAILURE BETWEEN LOAD FACTORS 0.406250e+01 AND 0.409375e+01
AVERAGE =    0.407813e+01
EDGE STRESS (+) =    0.646474e+05
EDGE STRESS (-) =   -0.643671e+05
MAX. DEFLECTION = 0.128601e+00 AT LOAD FACTOR = 0.406250e+01
```

3. PROGRAM RBETA.FOR (SOURCE CODE)

```
C     ******************************************************
C     *              RBETA.FOR   Version 2.0               *
C     *   (SHORTENED VERSION WITH SIZE EFFECTS CONSIDERED)  *
C     *                15 August, 1987                      *
C     *                                                     *
C     *  A PROGRAM FOR THE EVALUATION OF THE REIABILITY INDEX *
C     *          BETA OF A COLUMN (OR BEAM-COLUMN)          *
C     *                                                     *
C     * MATERIAL BEHAVIOUR IS ELASTIC IN TENSION WITH BRITTLE *
C     * FRACTURE, AND ELASTIC IN COMPRESSION UP TO A LIMITING *
C     * COMPRESSION STRESS, WITH A FALLING LINEAR BRANCH    *
C     * BEYOND THAT LIMIT.                                  *
C     *                                                     *
C     *   END LOAD IS  APPLIED  CENTRALLY. LATERAL LOADS    *
C     *   CAN BE DISTRIBUTED ALONG THE LENGTH OF THE MEMBER  *
C     *                                                     *
C     *   THE PROGRAM FINDS THE RELIABILITY INDEX BETA FOR A *
C     *   BEAM-COLUMN TAKING INTO ACCOUNT 5 RANDOM VARIABLES *
C     *   WHICH CONSTITUTE THE LOAD AND MATERIAL RESISTANCES *
C     *                                                     *
C     *  PROBLEM DATA IS READ FROM UNIT  #1                 *
C     *  OUTPUT IS STORED IN UNIT #2.                       *
C     *                                                     *
C     ******************************************************
C
C       * MAXIMUM OF 10 VARIABLES *
C       * MAXIMUM OF 20 ELEMENTS  *
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 INVNPR,NORMPR
      DIMENSION X(10),Y(10),U(10),DELTA(10),SIG(10)
     1 ,AVER(10),STD(10),F1X(10),F2X(10)
     2 ,SCALE(10),SHAPE(10),A(10),B(10),X0(10),XW(10)
      COMMON/CX1/GAP(5),GAW(5),EN1(8,5),EM1(8,5),EM2(8,5),NGAUSS
      COMMON/C2/F11,F21
      COMMON/C4/W,H,SPAN,PLN,GAMA1,SREF,XKC,XKT
      COMMON/CX4/NELEM,NBC(21),IX(21,4)
      REAL*8 LOC(10),MU(10),NN(10), NNN(10)
        INTEGER*2 ICODE(10)
       INTEGER*2 MXC(10), MEX(10)
      OPEN(1,FILE='DET',STATUS='OLD')
      OPEN(2,FILE='OT',STATUS='NEW')
      PI2 = DSQRT(8.0*DATAN(1.0D0))
      CONST=1.0D0/PI2
C
C    *******************************************************
C    *        DEFINE       VARIABLES                       *
C    *******************************************************
C    *  FCN    =  COMPRESSIVE STRENGTH, FIFTH PERCENTILE    *
C    *  W      =  WIDTH OF SECTION                          *
C    *  H      =  DEPTH OF SECTION                          *
C    *  GAMA1  =  RATIO OF NOMINAL DL TO LL                 *
C    *  ALFD   =  DEAD LOAD FACTOR                          *
C    *  ALFL   =  LIVE LOAD FACTOR                          *
C    *  EMIN   =  MODULUS OF ELASTICITY, MEAN VALUE         *
C    *  NELEM  =  NO OF ELEMENTS                            *
C    *  NJBC   =  NO OF JOINTS WITH B.C.                    *
C    *  NGAUSS =  NO OF INTEGRATION POINTS                  *
```

```
C   *   NBC(I) =  NO OF B.C. AT NODE I                          *
C   *    IX          =  B.C. CODE                                  *
C   *     1          =  U                                          *
C   *     2          =  UX                                         *
C   *     3          =  W                                          *
C   *     4          =  WX                                         *
C   *     N          =  NO OF RANDOM VARIABLES FOR TOTAL PROB.     *
C   *   EN1,EM1,EM2 = INTERPOLATION FUNCTIONS                      *
C   *       GAP      = CORDINATE AT GAUSS POINT                    *
C   *       GAW      = CORRESPONDING WEIGHT                        *
C   *       NELEM    = NO OF ELEMENTS                              *
C   *       NGAUSS   = NO OF GAUSS POINTS                          *
C   *       NITER    = MAX. NO OF ITERATIONS                       *
C   *       TOP      = TOLERANCE FOR LOAD                          *
C   *       EPSLON   = TOLERANCE FOR SOLUTION VECTOR               *
C   *       FC       = MATERIAL STRENGTH IN COMPRESSION            *
C   *       FT       = MATERIAL STRENGTH IN TENSION                *
C   *       E0       = MOE OF THE MATRIAL                          *
C   *       EN       = SLOPE OF THE STRESS-STRAIN CURVE            *
C   *       SPAN     = MEMBER LENGTH                               *
C   *       W        = WIDTH OF SECTION                            *
C   *       H        = DEPTH OF SECTION                            *
C   *       E        = ECCENTRICITY OF AXIAL LOAD                  *
C   *       NEQ      = NO OF EQUATIONS TO BE SOLVED                *
C   *       NJOINT   = NO OF NODES                                 *
C   *       NDOF     = NO OF VARIABLES PER NODE                    *
C   *       NODEL    = NO OF NODES PER ELEMENT                     *
C   *       SREF     = REFERENCE SPAN                              *
C   *       XKC      = SIZE EFFECT SHAPE PARAMETER (COMP.)         *
C   *       XKT      = SIZE EFFECT SHAPE PARAMETER (TENS.)         *
C   *       NDIMB    = NO OF VARIABLES PER NODE                    *
C   *       LBW,LHB = HALF BANDWIDTH INCLUD. THE DIAG.             *
C   *       NA       = NO OF UNKNOWNS FOR TOTAL PROBLEM            *
C   ***********************************************************
C
      READ(1,*) FCN,W,H
      READ(1,*) GAMA1,ALFD,ALFL
      READ(1,*) EMIN,SREF,XKC,XKT
      READ(1,*) NELEM,NJBC,NGAUSS
C
C*  READ BOUNDARY CONDITION CODES FOR THE PROBLEM
      DO 4455 I = 1, (NELEM+1)
      NBC(I) = 0
      DO 4433 J = 1, 4
      IX(I,J) = 0
 4433 CONTINUE
 4455 CONTINUE
      DO 9922 K = 1, NJBC
      READ(1,*) NJ,NBC(NJ)
      READ(1,*) (IX(NJ,JV),JV=1,NBC(NJ))
 9922 CONTINUE
C
      READ(1,*) N, (ICODE(I), I = 1,N)
      READ(1,*) (MXC(I), I = 1,N)
      DO 7779 I = 1,N
      MEX(I) = 0
      IF (MXC(I).EQ.0) GO TO 7779
```

```
       GO TO 7780
7779   CONTINUE
       GO TO 7782
7780   WRITE(*,7784)
7784   FORMAT(' ENTER EXPONENTS FOR DISTRIBUTION OF EXTREMES'/)
       READ(*,*) (MEX(I), I=1,N)
7782   CONTINUE
      READ(1,*) TOLB
      READ(1,*) NITER
C
C ENTER THE CODES FOR EACH VARIABLE AND THEIR PARAMETERS
C
      DO 9 IC = 1,N
      ICD = ICODE(IC)
      GO TO(11,12,13,14),ICD
C
C NORMAL ( CODE=1)
C
11    READ(1,*) AVER(IC),STD(IC)
      GO TO 9
C
C LOGNORMAL (CODE=2)
C
12    READ(1,*) AVER(IC),STD(IC)
      GO TO 9
C
C WEIBULL  ( CODE=3 )
C
13    READ(1,*) LOC(IC),SCALE(IC),SHAPE(IC)
      GO TO 9
C
C GUMBEL EXTREME TYPE I  ( CODE=4 )
C
14    READ(1,*) B(IC),A(IC)
C
9     CONTINUE
C
C ENTER INITIAL VECTOR X AND CHECK FOR CONSISTENCY IN THE CASE
C OF THE WEIBULL DISTRIBUTION
C
      DO 805 I = 1, N
151   READ(1,*) X(I)
      IF(ICODE(I).NE.3) GO TO 805
      IF (X(I).GT.LOC(I)) GO TO 805
      WRITE(*,1270)
1270  FORMAT (' CHANGE INITIAL VALUE TO EXCEED THE'
     1         ,/,' LOCATION PARAMETER FOR THE WEIBULL'/)
      GO TO 151
805   CONTINUE
C
      DO 702 I = 1, N
702   X0(I) = X(I)
155   NCOUNT=0
       NBET = 0
       IERR1 = 0
       IERR = 0
       READ(1,*) SPAN,R
       DELT = SPAN/(2.0D0*NELEM)
```

```
C
C   CALC VECTORS EN1, EM1, EM2
       CALL SHAP(DELT)
C
C        CORRECT FOR SLENDERNESS EFFECTS
       PC = W*H*FCN
       PCR = (3.14159D0**2)*EMIN*(W*H**3)/(12.D0*SPAN**2)
        CC = SPAN/H
        CA = DSQRT(0.9D0*0.74D0*EMIN/FCN)
       CK1 = 1.D0 - (1.D0/3.0D0)*((CC/CA)**4)
       CK2 = 3.14159D0**2*0.74D0*EMIN/(12.0*FCN*CC**2)
       IF (CC .GT. 10.0D0) GO TO 2080
       CK = 1.0D0
       GO TO 4080
2080   IF (CC .GT. CA) GO TO 3080
       CK = CK1
       GO TO 4080
3080   CK = CK2
4080   CONTINUE
C
C       OBTAIN NOMINAL DESIGN LOAD
       PLN = R*W*H*FCN*CK/(ALFD*GAMA1+ALFL)
C
      WRITE(2,1080)(ICODE(I), I=1,N)
1080  FORMAT       ('      CODES : ',10I5)
.C
C START ITERATIONS: GIVEN THE VECTOR X(I), COMPUTE
C THE FAILURE FUNCTION GXP AND THE GRADIENT DELTA
C USING THE SUBROUTINE GXPR, WHICH MUST BE PROVIDED
C EXTERNALLY BY THE USER FOR EACH PARTICULAR CASE.
C
2     CONTINUE
      DO 7722 J = 1, N
7722  XW(J) = X(J)
      CALL GXPR(XW,N,DELTA,GXP)
C
C CALC F1X(X), AND F2X(X)
C
      CALL FFX(N,X,AVER,STD,F1X,F2X,ICODE,LOC,SCALE,SHAPE,A,B,
     1  IERR, MXC, MEX)
      IF(IERR.EQ.1) GO TO 65
C
C CALC Y-VALUES
C
       DO 8 I = 1, N
       Y(I) = INVNPR(F1X(I))
8     CONTINUE
C
C CALC SIGMA AND MU VECTORS
C
      DO 10 I = 1, N
       IF (F2X(I).LE.0.0D0) GO TO 68
       DSIG = DLOG(CONST) - Y(I)*Y(I)*0.5D0 - DLOG(F2X(I))
       IF (DSIG.LT.-709.0D0) GO TO 865
      SIG(I) = DEXP(DSIG)
      GO TO 87
865   SIG(I) = 0.0D0
87    MU(I)=-SIG(I)*Y(I)+X(I)
```

```
10    CONTINUE
C
C CALC NN
      SUM=0.0D0
      DO 55 I=1,N
55    SUM = SUM + SIG(I)*SIG(I)*DELTA(I)*DELTA(I)
      SUM = DSQRT(SUM)
      DO 20 I = 1,N
      NN(I) = -SIG(I)*DELTA(I)/SUM
20    NNN(I) = DABS(NN(I))
C
C CALC BETA
      SDMU=0.0D0
      SDX = 0.0D0
      DO 25 I=1,N
      SDMU = SDMU + DELTA(I)*MU(I)
25    SDX = SDX + DELTA(I)*X(I)
      BETA = (GXP + SDMU - SDX)/SUM
      DO 30 I = 1,N
30    U(I) = BETA * NN(I)
      NCOUNT = NCOUNT+1
      IF (NCOUNT.GT.NITER) GO TO 66
      IF(NCOUNT.EQ.1) GO TO 32
       DIFFB = DABS(BETA - BETAP)
      BETAP = BETA
      NBET = 1
      DO 80 I = 1, N
      TX = SIG(I)*U(I) + MU(I)
80    X(I) = TX
      CONFAC = (TOLB-DIFFB)
      IF (CONFAC.GT.0.0) GO TO 50
      GO TO 2
32    BETAP = BETA
      DO 35 I = 1,N
35    X(I) = SIG(I)*U(I) + MU(I)
      GO TO 2
50    WRITE(2,51) BETA
      WRITE(2,710) NCOUNT
710   FORMAT(5X,'ITERATIONS =',I5)
      WRITE(2,703) TOLB
703   FORMAT(5X,'TOLB =',F8.4)
705   WRITE(2,1280)(X0(I),I=1,N)
1280  FORMAT('     VECTOR XO   ',10E13.5)
      WRITE(2,1300)(X(I),I=1,N)
1300  FORMAT('     VECTOR X    ',10E13.5)
      WRITE(2,1320)(NNN(I),I=1,N)
1320  FORMAT('     SENSITIVITY COEFFS. ',10F8.4)
      WRITE(2,2088) SPAN,R
2088  FORMAT('     L =',E13.6,' fp =',E13.6/)
51    FORMAT(5X,'BETA = ',F10.3)
      GO TO 900
65    IF (NBET.EQ.1) GO TO 880
      WRITE(2,1340) IERR
      GO TO 900
880   WRITE(2,1340) IERR
      GO TO 900
68    IERR1 = 1
      IF (NBET.EQ.1) GO TO 882
```

```
      WRITE(2,1341) IERR1
      GO TO 900
882   WRITE(2,1341) IERR1
      WRITE(2,1342) BETA
      GO TO 900
66    WRITE(2,1350)NITER
1350  FORMAT       (' NO CONVERGENCE IN ',I5,' ITERATIONS')
1340   FORMAT(' IERR =',I2,' ERROR: NEGATIVE LOGNORMAL OR',/,
     1         '              WEIBULL VARIABLE LESS THAN ITS',/,
     2         '              LOCATION PARAMETER.',/,
     3         '              TRY NEW INITIAL POINT')
1341  FORMAT(' IERR1 =',I2,' NEGATIVE OR ZERO DENSITY F2X(I)',/,
     1         '              TRY NEW INITIAL POINT'/)
1342   FORMAT(' LAST BETA WAS =', F10.3)
900   CONTINUE
      CLOSE (UNIT=1,STATUS='KEEP')
      CLOSE (UNIT=2,STATUS='KEEP')
      STOP
      END
C
      SUBROUTINE FFX(N,X,AVER,STD,F1X,F2X,ICODE,LOC,SC,SK,A,B,
     1 IERR, MXC, MEX)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 NORMPR
      DIMENSION SC(N),SK(N),A(N),B(N),X(N),AVER(N)
     1,         STD(N),F1(N),F2X(N)
       COMMON/C2/F11,F21
        INTEGER*2 ICODE(10)
       INTEGER*2 MXC(10), MEX(10)
      REAL*8 LOC(N),MU
      PI2=(8.0*DATAN(1.0D0))
      DO 20 I = 1,N
      IC = ICODE(I)
      GO TO(11,12,13,14),IC
C
C NORMAL
C
11    RATIO = (X(I) - AVER(I))/STD(I)
      F1X(I) = NORMPR(RATIO)
      F2X(I)=DEXP(-0.5D0*RATIO*RATIO)/(STD(I)*DSQRT(PI2))
       IF (MXC(I).EQ.0) GO TO 20
       CALL EXTR(F1X(I), F2X(I),MXC(I), MEX(I))
      GO TO 20
C
C LOGNORMAL
C
12    DLN = DLOG(1.0 + (STD(I)/AVER(I))**2)
      MU = DLOG(AVER(I)) - 0.5*DLN
      SDP = DSQRT(DLN)
      IF (X(I).LE.0.0) GO TO 99
      PARAM = (DLOG(X(I)) - MU)/SDP
      F1X(I) = NORMPR(PARAM)
      POW = DEXP(-0.5D0*PARAM*PARAM)
      F2X(I) = POW/(SDP*X(I)*DSQRT(PI2))
        IF (MXC(I).EQ.0) GO TO 20
       CALL EXTR(F1X(I), F2X(I), MXC(I), MEX(I))
      GO TO 20
C
```

```
C WEIBULL
C
13     IF (X(I).LE.LOC(I)) GO TO 99
       POW = -((X(I) - LOC(I))/SC(I))**SK(I)
       POW = DEXP(POW)
       F1X(I) = 1.0D0 - POW
       F2X(I) = (SK(I)/SC(I))*((X(I)-LOC(I))/SC(I))**(SK(I)
      1- 1.0)*POW
        IF (MXC(I).EQ.0) GO TO 20
        CALL EXTR(F1X(I), F2X(I), MXC(I), MEX(I))
       GO TO 20
C
C GUMBEL EXTREME TYPE I
C
14     POW = -A(I)*(X(I) - B(I))
       POW = DEXP(POW)
       F1X(I) = DEXP(-POW)
       F2X(I) = A(I)*POW * F1X(I)
        IF (MXC(I).EQ.0) GO TO 20
        CALL EXTR(F1X(I), F2X(I), MXC(I), MEX(I))
20     CONTINUE
       RETURN
99     IERR=1
       RETURN
       END
C
        SUBROUTINE EXTR(F1,F2,NC,M)
        IMPLICIT REAL*8(A-H,O-Z)
        INTEGER*2 NC, M
        IF (NC.EQ.2) GO TO 10
        F2 = M*F2*F1**(M-1)
        F1 = F1**M
        RETURN
10      F2 = M*F2*(1.0D0 - F1)**(M-1)
        F1 = 1.0D0 - (1.0D0 - F1)**M
        RETURN
        END
C
         FUNCTION NORMPR(X)
C        * NORMAL PROBABILITY INTEGRAL (X)*
         IMPLICIT REAL*8(A-H,O-Z)
         REAL*8 NORMPR
         DIMENSION E(16),H(16)
         PI = 2.0D0 * DSQRT(DATAN(1.0D0))
         IF (DABS(X).GT.5.0D0) GO TO 20
       E(1) = 0.989400934991650E0
       E(2) = 0.944575023073233E0
       E(3) = 0.865631202387832E0
       E(4) = 0.755404408355003E0
       E(5) = 0.617876244402644E0
       E(6) = 0.458016777657227E0
       E(7) = 0.281603550779259E0
       E(8) = 0.095012509837637E0
       H(1) = 0.027152459411754E0
       H(2) = 0.062253523938648E0
       H(3) = 0.095158511682493E0
       H(4) = 0.124628971255534E0
       H(5) = 0.149595988816577E0
```

```
      H(6) = 0.169156519395003E0
      H(7) = 0.182603415044924E0
      H(8) = 0.189450610455068E0
      DO 1 I = 1,8
      E(17-I) = -E(I)
1      H(17-I) = H(I)
        Y = X/DSQRT(2.0D0)
        S = 0.0
        DO 10 I = 1, 16
        Z = Y * E(I)
        Z = DEXP(-Z*Z)
        S = S + Z*H(I)
10      CONTINUE
        ERF = Y * S/PI
        NORMPR = (1.0D0 + ERF)/2.0D0
        RETURN
20      IF (DABS(X).GT.37.5D0) GO TO 25
        S = 1.0D0 - 1.0D0/(X**2) + 3.0D0/(X**4) - 15.0D0/(X**6)
     1  + 105.0D0/(X**8) - 945.0D0/(X**10) + 10395.0D0/(X**12)
        S = S*DEXP(-X*X/2.0D0)/DABS(X)
        S = S*DSQRT(2.0D0)/PI
        IF (X.GT.0.0D0) NORMPR = 1.0D0 - S/2.0D0
        IF (X.LT.0.0D0) NORMPR = S/2.0D0
        RETURN
25      IF (X.GT.0.0D0) NORMPR = 1.0D0
        IF (X.LT.0.0D0) NORMPR = 0.0D0
        RETURN
        END
C
C
        FUNCTION INVNPR(Y)
C       * INVERSE NORMAL PROBABILITY *
C
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 INVNPR
        REAL*8 NORMPR
        PI = DSQRT(8.0D0*DATAN(1.0D0))
        TOL = 1.0E-8
        IF (Y.EQ.0.50) GO TO 80
        X0 = -PI*(0.50D0 - Y)
        X1 = X0
5       S = NORMPR(X1) - Y
        S = S * DEXP(X1*X1/2.0D0) * PI
        X2 = X1 - S
        DIF = DABS(X2-X1)
        IF (DABS(DIF).LE.TOL) GO TO 20
        X1 = X2
        GO TO 5
80      INVNPR = 0.0
        RETURN
20      INVNPR=X2
        RETURN
        END
C
      SUBROUTINE COLUMN(XW,N,PAV)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(8),TKO(672),XE(8),XW(N)
     1,R(84),XO(84),X(84),B(84),B1(8,8),B2(8,8),B3(8,8),B4(8,8)
```

```
     2,B5(8,8),B6(8,8),B7(8,8),B8(8,8),B9(8,8),Y(5),RE(8),XP(84)
     3,Q(20),IQ(20), ESTR(7), FI(7)
      COMMON/CX1/GAP(5),GAW(5),EN1(8,5),EM1(8,5),EM2(8,5),NGAUSS
      COMMON/CX2/DIFP,NINT
      COMMON/C3/DEFL,PDEFL
      COMMON/C4/W,H,SPAN,PLN,GAMA1,SREF,XKC,XKT
      COMMON/CX4/NELEM,NBC(21),IX(21,4)
C   *******************************************************
C   *  EN1,EM1,EM2 = INTERPOLATION FUNCTIONS              *
C   *      GAP     = CORDINATE AT GAUSS POINT             *
C   *      GAW     = CORRESPONDING WEIGHT                 *
C   *      NELEM   = NO OF ELEMENTS                       *
C   *      NGAUSS  = NO OF GAUSS POINTS                   *
C   *      NITER   = MAX. NO OF ITERATIONS                *
C   *      TOP     = TOLERANCE FOR LOAD                   *
C   *      EPSLON  = TOLERANCE FOR SOLUTION VECTOR        *
C   *      FC      = MATERIAL STRENGTH IN COMPRESSION     *
C   *      FT      = MATERIAL STRENGTH IN TENSION         *
C   *      E0      = MOE OF THE MATRIAL                   *
C   *      EN      = SLOPE OF THE STRESS-STRAIN CURVE     *
C   *      SPAN    = MEMBER LENGTH                        *
C   *      W       = WIDTH OF SECTION                     *
C   *      H       = DEPTH OF SECTION                     *
C   *      E       = ECCENTRICITY OF AXIAL LOAD           *
C   *      NEQ     = NO OF EQUATIONS TO BE SOLVED         *
C   *      NJOINT  = NO OF NODES                          *
C   *      NDOF    = NO OF VARIABLES PER NODE             *
C   *      NODEL   = NO OF NODES PER ELEMENT              *
C   *      SREF    = REFERENCE SPAN                       *
C   *      XKC     = SIZE EFFECT SHAPE PARAMETER (COMP.)  *
C   *      XKT     = SIZE EFFECT SHAPE PARAMETER (TENS.)  *
C   *      NDIMB   = NO OF VARIABLES PER NODE             *
C   *      LBW,LHB = HALF BANDWIDTH INCLUD. THE DIAG.     *
C   *      NA      = NO OF UNKNOWNS FOR TOTAL PROBLEM     *
C   *******************************************************
        CONST = GAMA1
        E0 = XW(1)*1000.D0
        FC = XW(2)*1000.D0
        FT = XW(3)*1000.D0
        TOP = 0.01D0
        EPSLON = 0.001D0
        NITER = 10
        EN = 0.02D0
        NDOF = 4
        NJOINT=NELEM+1
        NG1 = NGAUSS + 1
        NG2 = NGAUSS + 2
         NP = 1
         NQ = 0
         Q0 = 0.0D0
       IF (NQ.EQ.0) GO TO 44
       DO 43 I = 1, NQ
43     READ(1,*) IQ(I), Q(I)
44     ECEN = -0.002D0
       IF (NP.NE.0) E=ECEN
        NEQ = NDOF*NJOINT
        NODEL = 2
        NDIMB = NODEL*NDOF
```

```
      LBW = NDIMB
      LHB = LBW
      NA = LBW*NEQ
      AR = W*H
      XI = W*H**3/12.D0
      DEL = SPAN/(2.D0*NELEM)
C
C      * ADJUST STRENGTHS TO THE ACTUAL VOLUME
      FC = FC *(SREF/SPAN)**(1.0/XKC)
      FT = FT *(SREF/SPAN)**(1.0/XKT)
      NINT = 0
      IF (NP.EQ.0) GO TO 761
      PC = AR*FC
      PCR = 3.14159D0**2*E0*XI/(SPAN**2)
      PI = PC
      IF(PCR .LE. PC) PI=PCR
      P2 = PI
      P1 = 0.0D0
      P3 = (P1 + P2)/2.0D0
      NFAIL = 0
      SMAX1 = 0.0
      GO TO 760
761   FQ1 = 0.0D0
      FQ2 = 1.0D0
      FQ3 = FQ2
      NFLAG = 0
760   DO 792 J = 1, NEQ
792   XP(J) = 0.0D0
C
C     START CALCULATIONS FOR TRIAL LOAD LEVELS
3773  CONTINUE
      P = 0.0D0
      FQ = 1.0D0
      IF (NP.NE.0) P = P3
      IF (NP.EQ.0) FQ = FQ3
      IF (NINT.EQ.1.AND.NP.NE.0) WRITE(*,4000) P
      IF (NINT.EQ.1.AND.NP.EQ.0) WRITE(*,4001) FQ
4000  FORMAT(//' SOLUTION FOR P =',E15.6,' :'/)
4001  FORMAT(//' SOLUTION FOR LATERAL LOAD FACTOR=',E15.6,':'/)
C     INITIALISE ARRAYS
      DO 80 J = 1, NEQ
      XO(J) = XP(J)
  80  R(J) = 0.D0
C
C     EXTERNAL LOAD VECTOR R
C     RE = ELEMENT LOAD VECTOR
      IF (Q0.EQ.0.0D0) GO TO 87
      DO 81 J = 1, 8
      RE(J)=0.D0
 81   CONTINUE
      RE(3) = FQ*Q0*DEL
      RE(4) = FQ*Q0*DEL**2/3.D0
      RE(7) = RE(3)
      RE(8) = -RE(4)
      DO 83 NE=1, NELEM
      DO 82 JJ = 1, 8
      K = (NE-1)*NDOF + JJ
      R(K) = R(K) + RE(JJ)
```

```
 82   CONTINUE
 83   CONTINUE
 87   IF (NQ.EQ.0) GO TO 185
       DO 180 J = 1, NQ
       JS = (IQ(J)-1)*NDOF + 3
180    R(JS) = R(JS) + Q(J)*FQ
185    EM = P*E
       JJ = (NJOINT-1)*NDOF + 1
       R(JJ) = R(JJ)-P
       R(1) = R(1) + P
       R(4) = R(4)-EM
       R(NEQ) = R(NEQ)+EM
      ITER = 0
C
C      BEGIN ITERATIONS AT THE TRIAL LOAD LEVEL
 777   CONTINUE
       DO 84  I = 1, NA
 84     TKO(I) = 0.0D0
        DO 85 K = 1, NEQ
 85     B(K) = -R(K)
       DO 645 IE = 1, NELEM
C   INITIALIZE ARRAYS
       DO 88 I = 1, 8
        F(I) = 0.0D0
       DO 86 J = 1, I
        B1(I,J) = 0.0D0
        B2(I,J) = 0.0D0
        B3(I,J) = 0.0D0
        B4(I,J) = 0.0D0
        B5(I,J) = 0.0D0
        B6(I,J) = 0.0D0
        B7(I,J) = 0.0D0
        B8(I,J) = 0.0D0
        B9(I,J) = 0.0D0
 86    CONTINUE
 88    CONTINUE
C       PICK ELEMENT SOLUTION FROM GLOBAL VECTOR
       DO 90 JJ = 1, 8
        K = (IE - 1)*NDOF + JJ
        XE(JJ) = XO(K)
 90    CONTINUE
       DO 101 K = 1, NGAUSS
        Y(K) = 0.D0
       DO 91 I=1, 8
        Y(K) = Y(K) + XE(I)*EM1(I,K)
 91    CONTINUE
C      OBTAINING COMPONENTS OF EKT
       DO 93 I = 1, 8
       DO 93 J = 1, I
        B1(I,J) = B1(I,J)+E0*DEL*EN1(I,K)*Y(K)*AR*
     1 EM1(J,K)*GAW(K)
        B2(I,J) = B2(I,J)+E0*DEL*EM1(I,K)*Y(K)*AR*
     1 EN1(J,K)*GAW(K)
        B3(I,J) = B3(I,J)+E0*DEL*EM1(I,K)*Y(K)*AR*
     1 Y(K)*EM1(J,K)*GAW(K)
        B4(I,J) = B4(I,J)+(E0*AR*DEL*EN1(I,K)*EN1(J,K)+
     1 E0*XI*DEL*EM2(I,K)*EM2(J,K))*GAW(K)
 93    CONTINUE
```

```
      DO 100 L = 1, NGAUSS
C     STRESSES AND STRAINS AT GAUSS POINT
      STR = 0.5D0*Y(K)**2
       DO 96 MO = 1, 8
      STR = STR+(EN1(MO,K)-GAP(L)*H*0.5D0*EM2(MO,K))*XE(MO)
 96   CONTINUE
      STRE = STR+FC/E0
      FAC = 1.0D0
      IF(STRE.GE.0.D0) FAC=0.0D0
       STRESS = E0*STR-((E0+EN*E0)*STR+FC*(1.D0+EN))*FAC
       DO 99 I = 1, 8
       DO 98 J = 1, I
      B5(I,J) = B5(I,J)+DEL*0.5D0*AR*(EN1(I,K)-GAP(L)*
     1 H*0.5D0*EM2(I,K))*(E0+E0*EN)*FAC*(EN1(J,K)-H*0.5D0*
     2 GAP(L)*EM2(J,K))*GAW(K)*GAW(L)
       B6(I,J) = B6(I,J)+DEL*0.5D0*AR*(EN1(I,K)-GAP(L)*
     1  H*0.5D0*EM2(I,K))*(E0+EN*E0)*FAC*Y(K)*EM1(J,K)*
     2 GAW(K)*GAW(L)
       B7(I,J) = B7(I,J)+DEL*0.5D0*EM1(I,K)*Y(K)*AR*
     1 (E0+E0*EN)*FAC*(EN1(J,K)-H*0.5D0*GAP(L)*EM2(J,K))*
     2  GAW(K)*GAW(L)
      B8(I,J) = B8(I,J)+DEL*0.5D0*EM1(I,K)*Y(K)*AR*
     1 (E0+E0*EN)*FAC*Y(K)*EM1(J,K)*GAW(K)*GAW(L)
      B9(I,J) = B9(I,J)+AR*STRESS*EM1(I,K)*EM1(J,K)*
     1 GAW(K)*GAW(L)*DEL*0.5D0
 98   CONTINUE
      F(I) = F(I)+AR*DEL*0.5D0*STRESS*((EN1(I,K)-H*0.5D0*
     1 GAP(L)*EM2(I,K))+Y(K)*EM1(I,K))*GAW(K)*GAW(L)
 99   CONTINUE
 100  CONTINUE
 101  CONTINUE
C*  OBTAIN ELEMENT TANGENT MATRIX
C   EKT IS THE (I,J) COMPONENT OF THE ELEMENT TANGENT MATRIX
      DO 105 I = 1, 8
      II = (IE-1)*NDOF + I
      B(II) = B(II) + F(I)
      DO 102 J = 1, I
      JJ = (IE-1)*NDOF + J
      EKT = B1(I,J)+B2(I,J)+B3(I,J)+B4(I,J)-
     1 B5(I,J)-B6(I,J)-B7(I,J)-B8(I,J)+B9(I,J)
      IJ = (JJ-1)*(LBW-1) + II
      TKO(IJ) = TKO(IJ)+EKT
 102  CONTINUE
 105  CONTINUE
 645  CONTINUE
C    INTRODUCE BOUNDARY CONDITIONS
      DO 111 IJO = 1, NJOINT
      IF (NBC(IJO).EQ.0) GO TO 111
      DO 110 J = 1, NBC(IJO)
      II = (IJO -1)*NDOF + IX(IJO,J)
      LBW1 = LBW - 1
      DO 108 K = 1, LBW1
       JJ = II - LBW + K
       IF (JJ.LE.0) GO TO 1080
       IJ = (JJ-1)*(LBW-1) + II
      TKO(IJ) = 0.0D0
1080   JJ = II + K
       IF (JJ.GT.NEQ) GO TO 108
```

```
         IJ = (II-1)*(LBW-1) + JJ
        TKO(IJ) = 0.0D0
 108    CONTINUE
         IJ = (II - 1)*(LBW-1) + II
        TKO(IJ) = 1.0D0
         B(II) = 0.0D0
 110    CONTINUE
 111    CONTINUE
C
C       SOLUTION OF THE SYSTEM
        CALL DECOMP(NEQ,LBW,TKO,IERROR)
        IF(IERROR .EQ. 1) GO TO 3774
        CALL SOLVN(NEQ,LBW,TKO,B)
        DO 112 I = 1, NEQ
        X(I) = XO(I)-B(I)
 112    CONTINUE
        CALL CONVRG(XO,X,IER,NEQ,EPSLON,ITER)
        ITER = ITER + 1
        IF (ITER.EQ.NITER) GO TO 431
        IF (IER.EQ.2) GO TO 430
        IF(IER.EQ.0) GO TO 118
        DO 115 I = 1, NEQ
 115    XO(I) = X(I)
        GO TO 777
430     IERROR = 1
        GO TO 3774
431    WRITE(2,900) NITER, P
900    FORMAT(' NO CONVERGENCE IN',I3,' ITERATIONS AT P=',E13.6/)
        GO TO 901
C*    AFTER CONVERGENCE, OBTAIN STRESSES AND STRAINS
C     AT THE CURRENT LOAD LEVEL
C
 118    CONTINUE
        EMAXP = 0.0D0
        EMAXN = 0.0D0
        SUME = 0.0D0
        DO 550 IE = 1, NELEM
        DO 500 J = 1, 8
        K = (IE-1)*NDOF +J
        XE(J) = X(K)
500     CONTINUE
        DO 540 K = 1, NGAUSS
         FACTOR = 0.0
        DO 501 I = 1, 8
501     FACTOR = FACTOR + XE(I)*EM1(I,K)
        EPLUS = 0.5D0 * FACTOR**2
        EMINUS = EPLUS
        DO 505 I = 1, 8
         EPLUS = EPLUS + (EN1(I,K)-H*0.5D0*EM2(I,K))*XE(I)
         EMINUS = EMINUS + (EN1(I,K)+H*0.5D0*EM2(I,K))*XE(I)
505     CONTINUE
         IF(EPLUS.GT.0.0D0 .AND. EMINUS.GT.0.0) GO TO 506
         IF(EPLUS.GT.0.0D0 .AND. EMINUS.LE.0.0) GO TO 507
         IF(EPLUS.LE.0.0D0 .AND. EMINUS.LE.0.0) GO TO 508
         IF(EPLUS.LE.0.0D0 .AND. EMINUS.GT.0.0) GO TO 509
506     EPOS = EPLUS
        IF(EMINUS.GT.EPOS) EPOS=EMINUS
        ENEG = 0.0D0
```

```
      GO TO 530
507    EPOS = EPLUS
       ENEG = EMINUS
      GO TO 510
508   EPOS = 0.0D0
      ENEG = EPLUS
      IF (DABS(EMINUS).GT.DABS(ENEG)) ENEG = EMINUS
      GO TO 530
509    EPOS = EMINUS
      ENEG = EPLUS
C
C     * FINDS THE POSITION OF THE NEUTRAL AXIS
510   ESTR(1) = EMINUS
      FI(1) = -1.0D0
      ESTR(NG2) = EPLUS
      FI(NG2) = 1.0D0
      DO 512 L = 1, NGAUSS
      SUM = 0.5*FACTOR**2
      DO 511 I = 1,8
511   SUM = SUM + (EN1(I,K) - GAP(L)*H/2.0*EM2(I,K))*XE(I)
      ESTR(L+1) = SUM
      FI(L+1) = GAP(L)
512   CONTINUE
      DO 515 I = 1, NG1
      PROD = ESTR(I)*ESTR(I+1)
      IF (PROD.LE.0.0D0) GO TO 516
515   CONTINUE
516   XN = FI(I) - ESTR(I)*(FI(I+1)-FI(I))/(ESTR(I+1)-ESTR(I))
      IF (ESTR(I).EQ.0.0D0) GO TO 518
      IF (ESTR(I).LT.0.0D0) HN = (1.0D0 - XN)*H/2.0D0
      IF (ESTR(I).GT.0.0D0) HN = (1.0D0 + XN)*H/2.0D0
      GO TO 520
518   IF (ESTR(I+1).LT.0.0D0) HN = (1.0D0 + XN)*H/2.0D0
      IF (ESTR(I+1).GT.0.0D0) HN = (1.0D0 - XN)*H/2.0D0
520   SUME = SUME + (HN/H)*(E0*EPOS)**XKT*GAW(K)
530  IF(EPOS.LT.EMAXP) GO TO 538
       EMAXP = EPOS
538  IF(DABS(ENEG).LT.DABS(EMAXN)) GO TO 540
       EMAXN = ENEG
540   CONTINUE
550   CONTINUE
       SMAXP = E0*EMAXP
       SMAXN = E0*EMAXN
       IF (DABS(SMAXN).LE.FC) GO TO 560
       SMAXN = SMAXN -((E0 + EN*E0)*EMAXN + FC*(1.0 + EN))
560    IF (SUME.EQ.0.0D0.OR.SMAXP.EQ.0.0D0) GO TO 563
       SUME = SUME/(2.0*NELEM*(XKT+1.0)*SMAXP**XKT)
       FTT = FT * SUME**(-1.0D0/XKT)
       GO TO 564
563    FTT = FT
564    IF (SMAXP.GE.FTT) GO TO 3774
       DEFL = 0.0D0
       DO 565 IE = 1, NELEM
       J = (IE-1)*NDOF + 3
       IF (DABS(X(J)).GT.DABS(DEFL)) DEFL = X(J)
565    CONTINUE
       J = NEQ - 1
       IF (DABS(X(J)).GT.DABS(DEFL)) DEFL = X(J)
```

```
      IF (NP.EQ.0) PDEFL = FQ3
      IF (NP.NE.0) PDEFL = P3
 3774 CONTINUE
      IF (NINT.EQ.0) GO TO 8810
      IF (IERROR.EQ.1) WRITE(*,8888)
      IF (IERROR.EQ.0.AND.SMAXP.LT.FTT) WRITE(*,8889) SMAXP
      IF (IERROR.EQ.0.AND.SMAXP.GE.FTT) WRITE(*,8890) SMAXP
8888  FORMAT(' IERROR = 1,FAILS (DIVERGENCE OR SINGULAR MATRIX)'/)
8889  FORMAT(' IERROR = 0 SMAXP = ',E15.6,'  ----- SURVIVES'/)
8890  FORMAT(' IERROR = 0 SMAXP = ',E15.6,'  ----- FAILS'/)
8810  CONTINUE
      IF (NP.EQ.0) GO TO 4500
      IF (IERROR.EQ.1) GO TO 7330
      IF (SMAXP.GT.FTT) GO TO 7331
      IF (SMAXP.EQ.FTT) GO TO 7337
      P1 = P3
      IF (SUME.EQ.0.0D0.OR.SMAXP.EQ.0.0D0) GO TO 5650
      SMAX1 = SMAXP*SUME**(1.0D0/XKT)
      GO TO 5655
5650  SMAX1 = SMAXP
5655  DO 833 J = 1, NEQ
833   XP(J) = X(J)
      GO TO 8334
7330  P2 = P3
      GO TO 8334
7331  P2 = P3
      NFAIL = 1
      SMAX2 = SMAXP*SUME**(1.0D0/XKT)
8334  IF (P1.EQ.0.0D0) GO TO 8338
      TOLP = (P2-P1)/P1
      IF (TOLP.LE.TOP) GO TO 7338
      GO TO 8336
8338  IF (P2.LE.0.1D0) GO TO 7338
8336  IF (NFAIL.EQ.1) GO TO 8340
      P3 = (P1 + P2)/2.0
      GO TO 3773
8340  P3 = P1 + (P2-P1)*(FT-SMAX1)/(SMAX2-SMAX1)
      GO TO 3773
7337  P = P3
      PP = P3
      PAV = P3
      GO TO 7339
7338  IF (P1.EQ.0.0D0) P2 = 0.0D0
      P = P2
      PP = P1
      PAV = (P+PP)/2.0
7339  CONTINUE
      GO TO 901
4500  IF (IERROR.EQ.1) GO TO 4330
      IF (SMAXP.GT.FTT) GO TO 4330
      IF (SMAXP.EQ.FTT) GO TO 4337
      IF (NFLAG.EQ.1) GO TO 4331
      FQ1 = FQ2
      FQ2 = 2.0D0*FQ2
      GO TO 4580
4331  FQ1 = FQ3
4580  DO 4833 J = 1,NEQ
4833  XP(J) = X(J)
```

```
        GO TO 4334
4330    NFLAG = 1
        FQ2 = FQ3
4334    IF (FQ1.EQ.0.0D0) GO TO 5338
        TOLP = (FQ2-FQ1)/FQ1
        IF (TOLP.LE.TOP) GO TO 4338
5338    IF (NFLAG.EQ.0) FQ3 = FQ2
        IF (NFLAG.EQ.1) FQ3 = (FQ1+FQ2)/2.0D0
        GO TO 3773
4337    P = FQ3
        PP = FQ3
        PAV = FQ3
        GO TO 4339
4338    P = FQ2
        PP = FQ1
        PAV = (P+PP)/2.0
4339    CONTINUE
901     RETURN
        END
C
        SUBROUTINE SHAP(DELT)
C*  THIS SUBROUTINE CALCULATES DERIVATIVES OF SHAPE FUNCTIONS
        IMPLICIT REAL*8(A-H,O-Z)
       COMMON/CX1/GAP(5),GAW(5),EN1(8,5),EM1(8,5),EM2(8,5),NGAUSS
        IF (NGAUSS.EQ.5) GO TO 5
        IF (NGAUSS.EQ.4) GO TO 4
C       *** 3 POINT GAUSSIAN INTEGRATION
        GAP(1) = -0.774596669241483D0
        GAP(2) = 0.0D0
        GAP(3) = -GAP(1)
        GAW(1) = 0.555555555555556D0
        GAW(2) = 0.888888888888889D0
        GAW(3) = GAW(1)
        GO TO 10
C       *** 4 POINT GAUSSIAN INTEGRATION
4       GAP(1) = -0.861136311594053D0
        GAP(2) = -0.339981043584856D0
        GAP(3) = -GAP(2)
        GAP(4) = -GAP(1)
        GAW(1) = 0.347854845137454D0
        GAW(2) = 0.652145154862546D0
        GAW(3) = GAW(2)
        GAW(4) = GAW(1)
        GO TO 10
C       *** 5 POINT GAUSSIAN INTEGRATION
5       GAP(1) = -0.906179845938664D0
        GAP(2) = -0.538469310105683D0
        GAP(3) = 0.0D0
        GAP(4) = -GAP(2)
        GAP(5) = -GAP(1)
          GAW(1) = 0.236926885056189D0
          GAW(2) = 0.478628670499366D0
          GAW(3) = 0.568888888888889D0
          GAW(4) = GAW(2)
          GAW(5) = GAW(1)
C       INITIALISES EN1,EM1,EM2
10      DO 150 IL = 1, 8
        DO 350 IK = 1, NGAUSS
```

```
          EN1(IL,IK) = 0.0D0
          EM1(IL,IK) = 0.0D0
          EM2(IL,IK) = 0.0D0
 350    CONTINUE
 150    CONTINUE
        DO 250 I = 1, NGAUSS
          EN1(1,I) = (-0.75D0+0.75D0*GAP(I)**2)/DELT
          EN1(2,I) = (-1.D0-2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
          EN1(5,I) = (0.75D0-0.75D0*GAP(I)**2)/DELT
          EN1(6,I) = (-1.D0+2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
           EM1(3,I) = (-0.75D0+0.75D0*GAP(I)**2)/DELT
           EM1(4,I) = (-1.D0-2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
           EM1(7,I) = (0.75D0-0.75D0*GAP(I)**2)/DELT
           EM1(8,I) = (-1.D0+2.D0*GAP(I)+3.D0*GAP(I)**2)*0.25D0
           EM2(3,I) = 1.5D0*GAP(I)/(DELT**2)
           EM2(4,I) = (-2.D0+6.D0*GAP(I))/(4.D0*DELT)
           EM2(7,I) = -1.5D0*GAP(I)/(DELT**2)
           EM2(8,I) = (2.D0+6.D0*GAP(I))/(4.D0*DELT)
 250    CONTINUE
        RETURN
        END
C
        SUBROUTINE DECOMP(NN,LHB,AA,IERROR)
C*    THIS SUBROUTINE DECOMPOSES A MATRIX USING CHOLESKY
C     METHOD FOR BANDED,SYMMETRIC,POS. DEFN. MATRIX
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION AA(672)
C       TKO IS STORED COLUMN - WISE.
         IERROR = 0
         KB = LHB-1
C       DECOMPOSITION
          IF(AA(1).LE.0.D0) IERROR=1
          IF(IERROR.EQ.1) RETURN
          AA(1) = DSQRT(AA(1))
          IF(NN.EQ.1) RETURN
        DO 551 I = 2, LHB
 551      AA(I) = AA(I)/AA(1)
        DO 590 J = 2, NN
          J1 = J-1
          IJD = LHB*J-KB
          SUM = AA(IJD)
          KO = 1
          IF(J.GT.LHB) KO=J-KB
        DO 555 K = KO, J1
          JK = KB*K+J-KB
 555      SUM = SUM-AA(JK)*AA(JK)
          IF(SUM.LE.0.D0) IERROR=1
          IF(IERROR.EQ.1) RETURN
          AA(IJD) = DSQRT(SUM)
        DO 568 I = 1, KB
          II = J + I
          KO = 1
        IF (II.GT.LHB) KO=II-KB
        SUM = AA(IJD+I)
         IF(I.EQ.KB) GO TO 565
        DO 540 K = KO, J1
         JK = KB*K+J-KB
         IK = KB*K+II-KB
```

```
 540    SUM = SUM-AA(IK)*AA(JK)
 565    AA(IJD+I) = SUM/AA(IJD)
 568     CONTINUE
 590     CONTINUE
        RETURN
        END
C
       SUBROUTINE SOLVN(NN,LHB,AA,S)
C* THIS SUROUTINE SOLVES CALLS A MATRIX SOLVER TO THE SYSTEM
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION AA(672),S(84)
C       FORWARD SUBSTITUTION
         KB = LHB-1
         S(1) = S(1)/AA(1)
        IF(NN.EQ.1) GO TO 685
        DO 680 I = 2, NN
         I1 = I-1
         KO = 1
        IF(I.GT.LHB) KO=I-KB
         SUM = S(I)
         II = LHB*I-KB
        DO 675 K = KO, I1
         IK = KB*K+I-KB
 675     SUM = SUM-AA(IK)*S(K)
         S(I) = SUM/AA(II)
 680    CONTINUE
C       BACKWARD SUBSTITUTION
 685     N1 = NN-1
         LB = LHB*NN-KB
         S(NN) = S(NN)/AA(LB)
        IF(NN.EQ.1) RETURN
        DO 699 I = 1, N1
        I1 = NN-I+1
        NI = NN-I
        KO = NN
       IF (I.GT.KB) KO=NI+KB
        SUM = S(NI)
       II = LHB*NI-KB
        DO 690 K = I1, KO
        IK = KB*NI+K-KB
 690   SUM = SUM-AA(IK)*S(K)
       S(NI) = SUM/AA(II)
 699   CONTINUE
       RETURN
       END
C
       SUBROUTINE CONVRG(XO,X,IER,NEQ,EPSLON,ITER)
C* THIS SUBROUTINE CHECKS THE CONVERGENCE OF SOLUTION VECTOR
        IMPLICIT REAL*8(A-H,O-Z)
        COMMON/CX2/DIFP,NINT
        DIMENSION XO(84),X(84)
         IER = 0
         PARX0 = 0.0D0
         PARDIF = 0.0D0
         PARX = 0.0D0
        DO 602 I = 1, NEQ
         PARX0 = PARX0 + XO(I)**2
         PARX = PARX + X(I)**2
```

```
602      PARDIF = PARDIF + (X(I)-XO(I))**2
        IF (NINT.EQ.1) WRITE(*,1002)  PARX0, PARX, PARDIF
1002    FORMAT(' NORMX0=',E13.6,'NORMX=',E13.6,'NORMDIF=',E13.6/)
        IF (ITER.EQ.0) GO TO 606
        IF (PARDIF.GE.DIFP) GO TO 605
606     DIFP = PARDIF
         IF (PARX0.EQ.0.0D0) GO TO 603
         DIF = DSQRT(PARDIF/PARX0)
         IF (DIF.LE.EPSLON) GO TO 604
603      IER = 1
         RETURN
604     RETURN
605      IER = 2
         RETURN
        END
C
        SUBROUTINE GXPR(XW,N,DELTA,GXP)
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION XW(N),DELTA(N)
        COMMON/CX1/GAP(5),GAW(5),EN1(8,5),EM1(8,5),EM2(8,5),NGAUSS
        COMMON/C4/W,H,SPAN,PLN,GAMA1,SREF,XKC,XKT
        COMMON/CX4/NELEM,NBC(21),IX(21,4)
         CALL COLUMN(XW,N,PU)
        GXP = PU - PLN*(GAMA1*XW(N-1) + XW(N))
        I = 0
 6644   I = I + 1
        XW(I) = XW(I)*1.01D0
        CALL COLUMN(XW,N,PU1)
        XW(I) = XW(I)*0.99D0/1.01D0
        CALL COLUMN(XW,N,PU2)
        XW(I) = XW(I)/0.99D0
        DELTA(I) = (PU1 - PU2)/(0.02D0*XW(I))
        IF (I.GE.(N-2)) GO TO 1202
        GO TO 6644
 1202   DELTA(N-1) = -PLN*GAMA1
        DELTA(N) = -PLN
        RETURN
        END
```

4. SAMPLE INPUT/OUTPUT FILE

SAMPLE INPUT DATA FILE FOR RELIABILITY ANALYSIS

```
15870.0 0.038 0.089
1.0 1.25 1.5
9660000.0 2.0 10.0 5.0
4 2 3
1 2
1 3
5 1
3
5 3 3 3 1 1
0 0 0 0 0
0.01
10
3514.0 6738.0 3.97
0.0 33.845 7.8559
4.03 29.861 2.9111
1.0 0.15
0.75 0.15
4538.4
7.036
8.358
1.025
0.881
3.2 0.6
```

SAMPLE OUTPUT FILE

```
CODES :    3    3    3    1    1
BETA = 5.136
ITERATIONS = 4
TOLB =   0.0100
VECTOR  XO            : 4693.6   7.053   8.538  1.025   0.881
VECTOR   X            : 3878.8  32.302  30.358  1.3053  1.0553
SENSITIVITY COEFFS. : 0.8282  0.0000  0.0000  0.3963  0.3963
```

L = 3.2 ϕ_p = 0.6

EXPLANATIONS

Vector Xo : Initial (trial) value for the variables.

Vector X : Coordinates of the most likely failure point (design point)

Sensitivity coefficients : Sensitivity of β to each of the variables. In this case β is most sensitive to X(1), X(3) and X(5). It is not sensitive to X(2) and X(3).

Solution corresponding to : L = 3.2m, ϕ_p = 0.6