# THE ANALYSIS OF RIGID-VISCOPLASTIC PLANE STRUCTURES SUBJECTED TO LARGE IMPULSIVE LOADING

by

P.D. Griffin

a thesis submitted for the degree of Doctor of Philosophy
in Engineering

August 1982

Department of Civil Engineering
University of Cape Town.

## ABSTRACT

This thesis is concerned with the analysis of plane ductile beams and frames which are subjected to large impulsive loading. The elastic response is ignored, and the material is considered as rigid-viscoplastic in order to take rate effects into account. Computational advantage is obtained by modelling this behaviour by a homogeneous viscous constitutive relation, as the rigid phase is absent. As opposed to the standard displacement method finite element formulation where interpolation functions describing the velocity field across elements are given, a formulation is used in which nodal velocities, moments and element axial forces are carried as parameters. Three methods of analysis are presented; firstly, the mode approximation technique is described, where the actual behaviour of the structure is approximated in closed form by the product of a mode shape and a function of time. A new algorithm for the determination of the mode shape is presented. The mode technique is then extended to include geometric effects by means of the instantaneous mode solution technique. Secondly, a method is given whereby at each instant the accelerations (by the Tamuzh principle) and the rates of change of moment (by virtual velocities formulation) are found, and velocities and moments are integrated forward independently to obtain a solution. Finally, a direct method of analysis is described, where nodal forces conjugate to a given velocity field are calculated (by the principle of virtual velocities), and hence from the equations of motion, accelerations are determined. An implicit forward integration scheme is employed to advance the solution in time. Illustrative examples are presented which show that these techniques give very good and computationally efficient predictions of the displaced shape of the structures under consideration, even when displacements are in the order of the dimensions of the structure.

# DECLARATION

I, Paul Dominic Griffin, declare that this thesis is essentially my own work and has not been submitted for a degree at another university.

P.D. Griffin
August 1982.

# CONTENTS

# NOMENCLATURE

## SPECIAL SYMBOLS

| | |
|---|---|
| [ ] | a matrix |
| $\underset{\sim}{c}$ | a vector c |
| $\dot{c}$ | the differential of c with respect to time |
| $\lvert c \rvert$ | the absolute value of c |
| T(superscript) | the transpose of a matrix |
| - 1(superscript) | the inverse of a matrix |
| d | differentiation with respect to |
| $\partial$ | partial differentiation with respect to |

## LOWER CASE CHARACTERS

| | |
|---|---|
| h | rectangular section depth |
| $\ell$ | length of an element |
| n | power in constitutive relation |
| p | generalised loads |
| $\dot{q}$ | generalised strain rate |
| s | spacial variable (two-dimensional) |
| t | time variable |
| u | displacement |
| $\dot{u}$ | velocity |
| $\ddot{u}$ | acceleration |
| x | spacial variable (one-dimensional) |

UPPER CASE CHARACTERS

| | |
|---|---|
| $A$ | rectangular section area |
| $D$ | dissipation rate |
| $M$ | moments |
| $\dot{M}$ | moment rate |
| $M_o$ | yield moment |
| $N$ | axial force |
| $N_o$ | axial yield force |
| $Q$ | generalised internal forces |
| $T$ | time function in mode analysis |
| $X$ | global cartesian X - axis |
| $Y$ | global cartesian Y - axis |

MATRICES AND VECTORS

| | |
|---|---|
| $[B]$ | deformation matrix |
| $[G]$ | lumped mass matrix |
| $\underset{\sim}{I}$ | impulse vector |
| $\underset{\sim}{M}$ | moment vector |
| $\underset{\sim}{\dot{M}}$ | moment rate vector |
| $[m]$ | influence matrix of nodal moments |
| $\underset{\sim}{N}$ | axial force vector |
| $[n]$ | influence matrix of element axial forces |
| $\underset{\sim}{P}$ | load vector |
| $\underset{\sim}{u}$ | displacement vector |
| $\underset{\sim}{\dot{u}}$ | velocity vector |
| $\underset{\sim}{\ddot{u}}$ | acceleration vector |
| $\underset{\sim}{X}$ | nodal force vector |
| $\underset{\sim}{\dot{\varepsilon}}$ | strain rate vector |
| $\underset{\sim}{\dot{\kappa}}$ | curvature rate vector |
| $\underset{\sim}{\phi}$ | mode shape vector |

## GREEK CHARACTERS

| | |
|---|---|
| $\gamma$ | specific mass |
| $\Delta$ | increment in |
| $\dot{\varepsilon}$ | axial strain rate |
| $\dot{\varepsilon}_o$ | strain rate material constant |
| $\theta$ | rotation |
| $\dot{\theta}$ | rotation rate |
| $\ddot{\theta}$ | rotational acceleration |
| $\dot{\kappa}$ | curvature rate |
| $\dot{\kappa}_o$ | curvature rate material constant |
| $\mu$ | stress matching factor |
| $\nu$ | power matching factor |
| $\sigma_o$ | yield stress |
| $\phi$ | mode shape |

## SUBSCRIPTS

| | |
|---|---|
| e | element |
| i,j,k | the i-th, j-th, k-th iteration |
| max | maximum |
| t | time |

## SUPERSCRIPTS

| | |
|---|---|
| i,j,k | the i-th, j-th, k-th, iteration |
| m | modal quantities |
| o | initial value |
| s | statically admissible system |
| t | time |

# CHAPTER 1

# INTRODUCTION

The behaviour of ductile metal beam and frame structures subjected to very severe impulsive loading has been the subject of a large number of theoretical and experimental studies since the Second World War. The problem is a complex one, both due to nonlinear material behaviour and the large plastic deformations which occur.

Early analytical solutions (for example, Bleich and Salvadori [17]), made use of an elastic-plastic constitutive relation and standard elastic mode techniques. Permanent plastic deformations were included by introducing plastic hinges. Such techniques were unable to incorporate large plastic deformations and were thus limited to small impulses. Nevertheless, valuable special solutions were obtained (for example, Duwez, Clark and Bohenblust [18]).

The incorporation of both elastic and plastic effects in the constitutive relation proved very difficult even when post-elastic behaviour was idealised as perfectly plastic. It was recognised, however, that since a structure subjected to large impulsive loading undergoes plastic deformations far in excess of possible elastic deformations, elastic effects could be ignored (see, for example, Lee and Symonds [20], Parkes [5], and Symonds [21], [24]). Geometric effects were recognised as being significant but for simplicity were assumed small. These assumptions were incorporated in what became known as the *simple rigid-plastic* theory. Although this simple rigid-plastic theory provided an analytical method for determining the major deformations in an impulsively loaded structure, and as a first order theory sometimes provided excellent results when compared to experiment, it proved useful only in limited applications.

Experimental work by, among others, Manjoine [3], Aspden and Campbell [4] and Parkes [23] highlighted the importance of including rate sensitivity in the plastic model, particularly for steel and titanium alloy structures. Parkes [5] proposed a crude rate sensitive model in which the static yield stress in the rigid-plastic theory was modified simply by a constant factor appropriate to the average strain rate in the structure. This approach led to an improved solution, but nevertheless overestimated deflections for the analysis of a cantilever beam struck transversely at its tip, as factoring the static yield moment did not lead to correct predictions of the pattern of plastic deformation in the structure.

Analytical results were greatly improved by including the strain rate behaviour directly into the constitutive relation (for example, Ting [7], Ting and Symonds [6], Bodner and Symonds [10] and Bodner [11]). This rigid-viscoplastic model was based on empirical stress-strain rate relations suggested by Manjoine [3] for steel, and Parkes [23] for aluminum alloys. In uniaxial form, the relation is

$$\frac{\dot{\varepsilon}}{\dot{\varepsilon}_o} = \left(\frac{\sigma}{\sigma_o} - 1\right)^n \quad \text{for} \quad \sigma \geq \sigma_o \quad ;$$

$$\dot{\varepsilon} = 0 \quad \text{for} \quad \sigma \leq \sigma \leq \sigma_o \quad . \tag{1.1}$$

In this equation $\dot{\varepsilon}$, $\sigma$ are strain rate and stress respectively and $\dot{\varepsilon}_o$, $\sigma_o$ are material constants with the dimensions of strain rate and stress respectively. The constants $\varepsilon_o$, $\sigma_o$ and n were obtained from experiment; n is large, usually greater than 4. Improved correlation with experiment was obtained using this constitutive relation, but despite the simplifying assumptions of no elastic phase, no strain hardening and small deflections, the analyses remained rather complicated and not easily generalised.

A much simpler approach for estimating the permanent deformations of structures subjected to high intensity dynamic loading is the use of mode approximations, suggested by Martin and Symonds [1] for rigid-plastic structures. For such structures in which displacements are small, mode solutions are admitted in which the velocity field $\dot{\underset{\sim}{u}}^m(s,t)$ is given by the product of a function $\underset{\sim}{\phi}$ of the spatial variable s and a function T of time t;

$$\dot{\underset{\sim}{u}}^m(s,t) = \underset{\sim}{\phi}(s)T(t) \quad . \tag{1.2}$$

It can be shown that the actual solution $\dot{\underset{\sim}{u}}(s,t)$ for an impulsively loaded problem with initial velocities $\underset{\sim}{u}(s,o) = \dot{\underset{\sim}{u}}^o(s)$ converges onto a mode solution: the mode approximation was based on the concept of replacing the actual solution $\dot{\underset{\sim}{u}}(s,t)$ by a mode solution $\dot{\underset{\sim}{u}}^m(s,t)$, with the initial amplitude T(o) suitably chosen. A fundamental requirement of the application of this approach is that a method for determining $\underset{\sim}{\phi}(s)$ should be available. Once $\underset{\sim}{\phi}(s)$ is known, the initial amplitude T(o) was chosen so as to minimise a measure $\Delta_o$ of the initial difference between the actual solution and the mode solution.

$$\Delta^o = \int \frac{\gamma(s)}{2} \left[\dot{\underset{\sim}{u}}^o(s)-T(o)\underset{\sim}{\phi}(s)\right]\left[\dot{\underset{\sim}{u}}^o(s)-T(o)\underset{\sim}{\phi}(s)\right]ds \quad , \tag{1.3}$$

where $\gamma(s)$ is the specific mass of the structure, and integration is carried out over the entire structure. If a number of possible mode shapes are available, the 'best' mode shape would be taken as that which gives the smallest value of $\Delta_o$ when equn. (1.3) was minimised with respect to T(o). This approach was successfully used to analyse a variety of simple structures where the choice of mode shape was fairly clear. The method was not readily applied to more complex problems where the choice of $\phi(s)$ was not apparent. Although a variational principle by which mode shapes could be determined was known quite early in the development of the topic (Martin [19]), a means of implementation of the principle to the numerical calculation of mode shapes was not immediately available.

Symonds [12] and later Bodner [11] extended the mode approximation technique to include rate sensitivity, and with very simple calculations were able to successfully predict the response of a cantilever subjected to transverse impact at its tip. The inclusion of rate sensitivity was treated more formally by Lee and Martin [8] using the rigid-viscoplastic constitutive relation of equn. (1.1). Since, for the rigid-viscoplastic model, mode solutions do not exist as they do for rigid-plastic materials because the constitutive equations are not homogeneous, an alternative approximation scheme was proposed in which corresponding to each level of kinetic energy throughout the motion a "piecewise stationary mode" shape was determined by applying the variational principle for the mode shape. For structures whose natural response does not change significantly throughout the time span of deformation, they showed that this approach gave a good approximation to the actual solution. Excellent agreement was obtained with previous analytical and experimental results for the tip loaded cantilever (Ting [7], Bodner and Symonds [10]), but the technique was not set out in a way which could be easily generalised to more complex problems.

In order to obtain a rate sensitive constitutive model which permitted the separation of variables required for an exact mode solution, Symonds [9] proposed that the non-homogeneous relation of equn. (1.1) be replaced by an equivalent homogeneous viscous relation between stress and strain rate of the form

$$\frac{\dot{\varepsilon}}{\dot{\varepsilon}_o} = \left(\frac{\sigma}{\mu\sigma_o}\right)^{\nu n} , \qquad (1.4)$$

where $\mu$, $\nu$ are factors which were chosen so that equn. (1.4) was appropriately matched to equn. (1.1), and hence to test results. Further, he presented an iterative scheme to determine the mode shape which permitted greater flexibility in application of the method. The results obtained using this technique agreed reasonably well with the stationary mode solution for the tip loaded cantilever of Lee and Martin [8], which indicated that, when suitably matched, the homogeneous viscous relation of equn. (1.4) could replace the rigid-viscoplastic relation of equn. (1.1) without significant loss of accuracy.

The mode solution technique when applied using the homogeneous viscous law holds rigorously throughout the timespan of deformation in the case where the displacements are small. Since for structures which are subjected to large impulses geometric effects are in general of great significance, the technique outlined above is only of limited value. The basic concept may, however, be extended to include large deflections by using the *instantaneous mode technique* (Symonds and Chon [13]). The response time of the structure is divided into a number of small intervals $\Delta t$, and it is assumed that the geometry is fixed during each interval. At the beginning of a typical interval, a mode shape is computed on the basis of the current geometry, and is used to compute the response during that interval. At the end of the interval the displacement increments are computed and used to update the geometry of the structure, and the process is repeated for the next increment. The mode amplitude at the beginning of each interval is determined by the same procedure that is used to compute the initial amplitude in the small displacement case. The method is not exact, but for a suitably chosen time step can give excellent results in some structures (see for example, Symonds and Chon [14], Symonds and Raphanel [16]).

While mode solution techniques have given valuable insight into the behaviour of dynamically loaded structures by modelling the 'natural' response of a structure, caution is required in their application. An implicit assumption in the mode approximation technique is that final deformations are predominantly of the modal shape, and that any localised, non-modal response which occurs contributes negligibly to the overall behaviour of the structure. This assumption is not in general true, since for certain classes of problems large non-modal deformations may take place before a modal pattern of behaviour occurs, if indeed it occurs at all. Since in the mode solution procedure the initial velocity imparted to the structure is replaced by an equivalent velocity field in the mode shape, the initial effect of the impulse, when stresses are at their maximum, is not described in these cases.

To obtain a comprehensive and general solution, recourse must be made to direct methods of analysis, where equations of motion and equilibrium are solved at each instant of time. Due to their complexity, such approaches are computationally far more costly than approximate methods and often require lengthy numerical procedures. Computer programs of varying degrees of sophistication which perform such analyses have been available for some time. Earlier programs used the finite difference technique with an elastic-plastic or rigid-plastic material model (Witmer, Balmer, Leech and Pian [24], Balmer [25], Hashmi, At Hassani and Johnson [26]). More recently a variety of finite element programs have been developed for application in the automotive and aviation industries for crash simulation (for example, KRASH [27], ACTION [28], DYCAST [29] and WRECKER [30], cited by Pifko and Winter [31]). Elastic-plastic or rigid plastic constitutive relations were assumed but rate sensitivity was not included.

In this thesis we shall be concerned with various aspects of the analysis of impulsively loaded structures composed of a homogeneous viscous material. Attention will be directed primarily towards the numerical solution of such problems, using both the mode approximation and direct time integration techniques.

Specifically we shall be concerned with the dynamic analysis of homogeneous viscous beam and frame structures which lie in one plane, which are cantilevered or supported only at their ends, and which have a specific mass $\gamma(s)$ per unit length.

At time $t = 0$, a large distributed impulse $I(s)$ is imparted to part of the undeformed structure, which is assumed to result in an initial velocity field $\dot{\underset{\sim}{u}}(s,o) = \dot{\underset{\sim}{u}}^{o}(s) = I(s)/\gamma(s)$ in the plane of the structure. No further external loading is considered for $t > 0$.

In the beam and frame structures under consideration, the homogeneous viscous constitutive equations must relate bending moments M and axial forces N to conjugate curvature rates $\dot{K}$ and middle surface strain rates $\dot{\varepsilon}$ (Fig. 1.1 b.c). The constitutive equations are greatly simplified by taking the section to be that of a sandwich beam; the two flanges have an area $\frac{1}{2}A$ and are held at distance h apart by material which carries shear force but undergoes negligible shear deformations. (Fig. 1.1a)

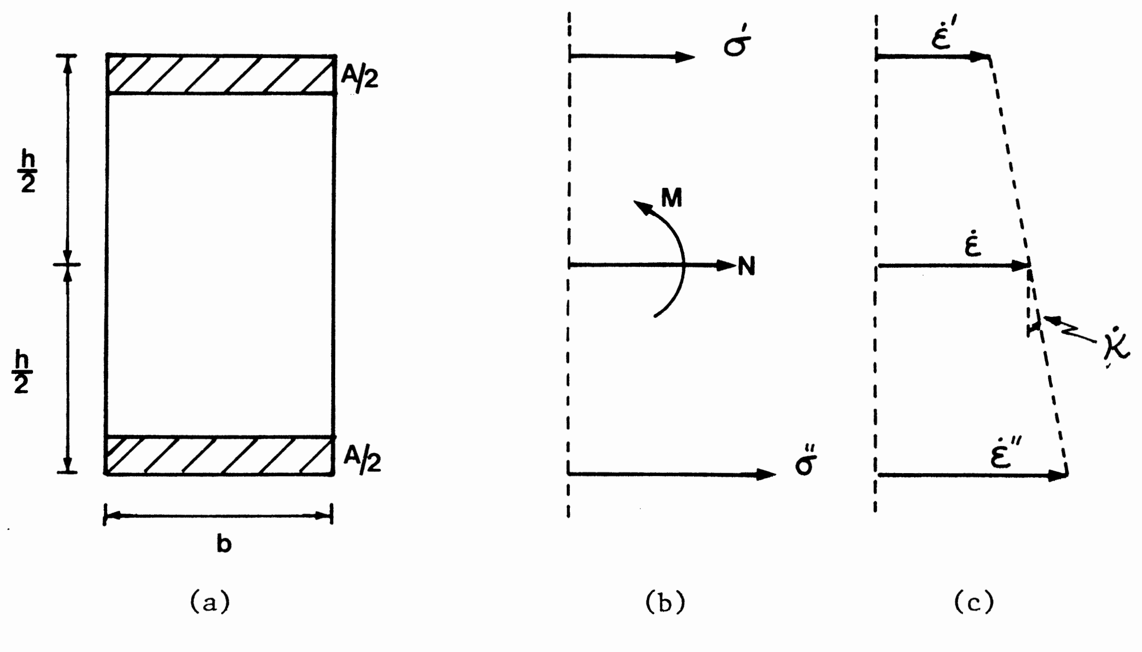



Figure 1.1 Sandwich beam idealization.

Applying equn. (1.1) to the sandwich beam the following constitutive equation is obtained

$$N_o \dot{\varepsilon} = \frac{\partial\Psi}{\partial N/N_o} \quad ; \quad M_o \dot{\kappa} = \frac{\partial\Psi}{\partial M/M_o} \tag{1.5a}$$

where

$$\Psi = \frac{N_o \dot{\varepsilon}_o}{2(n+1)} \left[ \left| \frac{N}{N_o} + \frac{M}{M_o} \right| - 1^{\,n+1} + \left| \frac{N}{N_o} - \frac{M}{M_o} \right| - 1^{\,n+1} \right] \quad (1.5b)$$

and where

$$N_o = A\sigma_o \quad , \quad M_o = \frac{1}{2} Ah\sigma_o \quad (1.5c)$$

Following Symonds and Chon [13], the rigid-viscoplastic relation of equns. (1.5) will be replaced by homogeneous viscous relations equns. (1.6); and in this case we find

$$N'_o \dot{\varepsilon} = \frac{\partial \Psi}{\partial N/N'_o} \quad ; \quad M'_o \dot{\kappa} = \frac{\partial \Psi}{\partial M/M'_o} \quad (1.6a)$$

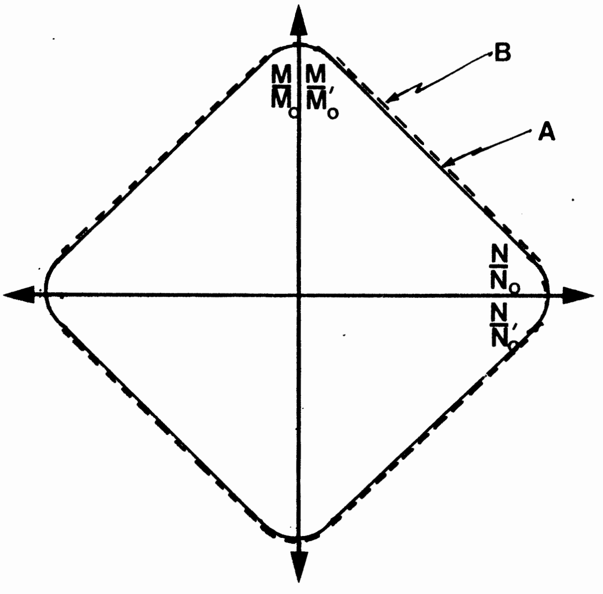
where

$$\Psi = \frac{N'_o \dot{\varepsilon}_o}{2(n'+1)} \left[ \left| \frac{N}{N'_o} + \frac{M}{M'_o} \right|^{n'+1} + \left| \frac{N}{N'_o} - \frac{M}{M'_o} \right|^{n'+1} \right] \quad (1.6b)$$

and where

$$\sigma'_o = \mu\sigma_o \ , \quad n' = \nu n \ , \quad N'_o = A\sigma'_o \ \text{ and } \ M'_o = \frac{1}{2} Ah\sigma'_o \quad (1.6c)$$

The constitutive laws of equns. (1.5) and (1.6) are shown in Fig. 1.2 .




Curve A : rigid-viscoplastic behaviour (equations (1.5))

Curve B : matched viscous behaviour (equations (1.6))

both for constant $\Psi$

Figure 1.2 Constitutive laws for rate-dependent sandwich beam sections

For a suitable choice of $\mu$, $\nu$, the equns. (1.6) can provide a close approximation to the rigid-viscoplastic relations of equns. (1.5). The choice of $\mu$ and $\nu$ will be discussed in detail in a later section. The adoption of the homogeneous viscous constitutive relation permits the separation of variables required for the application of the mode approximation technique, and greatly simplifies the calculations required in direct solution procedures.

In formulating numerical solutions, the beam or frame must be discretised: node positions are defined along the centre line of the structure, and the velocities and rotation rates of the nodes become kinematic variables. Mass will be lumped at the nodes in the conventional way (Newmark [32]), so that the elements connecting the nodes are massless but are assumed to be able to transmit axial force, shear force and bending moment from one node to another. When a homogeneous viscous constitutive relation is used, however, conventional finite element methods are not easily applied. With massless elements the bending moments should vary linearly between nodes. If, however, the usual cubic interpolation functions for transverse velocity is used between nodes together with the homogeneous viscous relation (especially when $n'$ is large) the variation of moments will be highly non-linear. Alternatively if linear variation of moments is assumed, the interpolation function for transverse velocity cannot be explicitly computed, making the formulation of relations between element end forces and moments, and velocities and rotation rates very difficult. An important consideration in the solution procedures to be presented in this thesis is that the interpolation functions for the velocity field across an element will not be explicitly defined.

In Chapter 2, the mode solution technique will be discussed in detail, and a new iterative scheme for the determination of the mode shape $\underset{\sim}{\phi}$, with proof of convergence, will be given. The extension of this new technique to include geometric nonlinearities, using the instantaneous mode technique, is then described. Limitations of the solution procedure are also discussed.

In Chapter 3, two direct time integration techniques for dynamic analysis are presented; firstly, a method based on the principle of Tamuzh [33] where small displacement assumptions will be adopted, and secondly a more direct method of analysis in which the force method of analysis and the principle of virtual velocities are used to determine equilibrating forces and hence accelerations in the structure. Further, an implicit time integration scheme is presented which leads to an efficient solution of the dynamic problem for values of n' which are very large.

Various methods whereby the homogeneous viscous constitutive relations may be matched to the rigid-viscoplastic material model are discussed in Chapter 4, and their limitations are noted. In Chapter 5, the implementation of the analytical techniques presented here to the computer is discussed, and flow charts of the various solution methods are presented. The results of the analysis of a variety of beam and frame structures using the mode approximation technique and direct methods of analysis are given in Chapter 6 as an illustration of the concepts put forward in this thesis.

In Appendix A, a user manual for the implementation of the programs which perform the instantaneous mode solution method and the direct method of analysis is presented, followed in Appendix B and C by listings of the programs. Finally, a list of papers which were co-authored with Prof. J.B. Martin and which have been accepted for publication is given in Appendix D.

# CHAPTER 2

# THE MODE APPROXIMATION TECHNIQUE

## 2.1 The Basis of the Mode Approximation Technique

In order to apply the mode approximation technique the constitutive equations must permit a separation of variables in the solution. The homogeneous viscous constitutive model satisfies this requirement; bending moment M and axial force N are related to conjugate curvative rate $\dot{\kappa}$ and middle surface strain rate $\dot{\varepsilon}$, so that

$$N_o \dot{\varepsilon} = \frac{\partial \Psi}{\partial N/_{No}} \quad ; \quad M_o \dot{\kappa} = \frac{\partial \Psi}{\partial M_{/Mo}} \quad , \tag{2.1a}$$

where

$$\Psi = \frac{N_o \dot{\varepsilon}_o}{2(n+1)} \left[ \left| \frac{N}{N_o} + \frac{M}{M_o} \right|^{n+1} + \left| \frac{N}{N_o} - \frac{M}{M_o} \right|^{n+1} \right] \tag{2.1b}$$

and where

$$N_o = A\sigma_o, \quad M_o = \frac{1}{2} Ah\sigma_o \quad . \tag{2.1c}$$

Equations (2.1) are identical to equns. (1.6) with the prime on $\sigma_o$ and n omitted for clarity.

We shall consider beam and frame structures which are cantilevered or supported only at their ends, which lie on one plane and which have a specific mass $\gamma(s)$ per unit length. Small displacement assumptions will be adopted so that equations of motion are formulated in the original configuration throughout the time span of deformation. This assumption is essential for the mode technique to hold rigorously.

Impulsive laoding problems are characterised by initial applied velocities $\dot{\underset{\sim}{u}}^{o}(s)$, with no external loads for $t > 0$. The solution of the problem will be denoted by the velocity field $\dot{\underset{\sim}{u}}(s,t)$, with $\dot{\underset{\sim}{u}}(s,o) = \dot{\underset{\sim}{u}}^{o}(s)$.

Mode approximations are based on the concept that the actual velocity field $\dot{\underset{\sim}{u}}(s,t)$ can be replaced by a solution where the velocity field is of the form

$$\dot{\underset{\sim}{u}}^{m}(s,t) = \underset{\sim}{\phi}(s)T(t) \quad , \tag{2.2}$$

where $\underset{\sim}{\phi}(s)$ is the mode shape and $T(t)$ is a function of time.

The mode approximation technique is based on the observation that the actual solution converges onto the mode shape, so that all that is lost in the approximation, if $\underset{\sim}{\phi}$ and T are properly chosen, is the transient behaviour of the structure before the velocity field adopts the mode shape.

If $\underset{\sim}{\phi}(s)$ is known, the function $T(t)$ can be explicitly found. Its initial value, $T(o)$ is given by a generalised momentum balance; this is equivalent to minimising the function $\Delta^{o}(\dot{\underset{\sim}{u}}(s), \dot{\underset{\sim}{u}}^{m}(s,o))$ with respect to $T(o)$, where

$$\Delta^{o} = \frac{1}{2}\int \gamma(s)(\dot{\underset{\sim}{u}}^{o}(s)-\dot{\underset{\sim}{u}}^{m}(s,o))\ (\dot{\underset{\sim}{u}}^{o}(s)-\dot{u}^{m}(s,o))ds \tag{2.3}$$

Substituting $\underset{\sim}{u}^{m}(s,o) = \underset{\sim}{\phi}(s)T(o)$ from equn. (2.2) into equn. (2.3) and minimising with respect to $T(o)$, we obtain

$$T(o) = \frac{\int\dot{\gamma}(s)\ \underset{\sim}{\phi}(s)\ \dot{\underset{\sim}{u}}^{o}(s)ds}{\int\gamma(s)\ \underset{\sim}{\phi}(s)\ \dot{\underset{\sim}{\phi}}\ (s)ds} \quad . \tag{2.4}$$

Equation (2.4) ensures that momentum is conserved.

The function T(t) can then be determined from a work rate balance for the structure. If $M^m(s,t)$ and $N^m(s,t)$ denote the modal bending moments and axial forces conjugate to modal curvature rates $\dot{\kappa}^m(s,t)$ and middle surface axial strain rates $\dot{\varepsilon}^m(s,t)$, respectively, then

$$-\int \gamma(s)\ddot{\underset{\sim}{u}}^m(s,t)\dot{\underset{\sim}{u}}^m(s,t)ds = \int M^m(s,t)\dot{\kappa}(s,t)ds + \int N^m(s,t)\dot{\varepsilon}^m(s,t)ds \quad (2.5)$$

where $\ddot{\underset{\sim}{u}}^m(s,t)$ are the accelerations associated with the mode. Equation (2.5) may be rewritten as

$$-\int \gamma(s)\ddot{\underset{\sim}{u}}^m(s,t)\dot{\underset{\sim}{u}}^m(s,t)ds = \int D(\dot{\kappa}(s,t),\dot{\varepsilon}(s,t))ds \quad , \quad (2.6a)$$

where $D(\dot{\kappa}(s,t), \dot{\varepsilon}(s,t)$ is a homogeneous dissipation function of degree $\left(\frac{n+1}{n}\right)$ in the components of the strain rate quantities $\dot{\kappa}$, $\dot{\varepsilon}$. Since strain rates and velocities $\dot{\underset{\sim}{u}}(s,t)$ are related through a linear set of strain rate-displacement rate equations, the right hand side of equn. (2.6a) may be expressed as a homogeneous function of degree $\left(\frac{n+1}{n}\right)$ in the components of velocities $\dot{\underset{\sim}{u}}(s,t)$, so that

$$-\int \gamma(s)\ddot{\underset{\sim}{u}}^m(s,t)\ \dot{\underset{\sim}{u}}^m(s,t)ds = \int D(\dot{\underset{\sim}{u}}(s,t))ds \quad . \quad (2.6b)$$

Substituting from equn. (2.2) into equn. (2.6b), we obtain

$$-T\frac{dT}{dt}\int \gamma(s)\underset{\sim}{\phi}(s)\underset{\sim}{\phi}(s)ds = T^{\frac{n+1}{n}}\int D(\underset{\sim}{\phi}(s))ds \quad , \quad (2.7a)$$

hence

$$\frac{dT}{dt} = - \frac{T^{1/n} \int D(\underset{\sim}{\phi}(s))ds}{\int \gamma(s)\underset{\sim}{\phi}(s)\underset{\sim}{\phi}(s)ds} \quad . \tag{2.7b}$$

The solution of this differential equation can be written as

$$T(t) = T(o) \left\{ 1 - \frac{n-1}{n} kt \right\}^{n/n-1} \tag{2.8a}$$

where

$$k = \frac{1}{T(o)} \frac{n-1}{n} \left\{ \frac{\int D(\underset{\sim}{\phi}(s))ds}{\int \gamma(s)\underset{\sim}{\phi}(s)\underset{\sim}{\phi}(s)ds} \right\} \quad . \tag{2.8b}$$

It can be seen from these expressions that the mode shape $\underset{\sim}{\phi}(s)$ must be known, together with the dissipation rate associated with the mode $D(\underset{\sim}{\phi}(s))$ and the energy of the mode $\int \gamma(s)\underset{\sim}{\phi}(s)\underset{\sim}{\phi}(s)ds$, in order to determine $T(t)$. All this information is provided by a new algorithm for the determination of the mode shape $\underset{\sim}{\phi}(s)$, which will be presented in the next section.

## 2.2 An Algorithm for the Determination of the Mode Shape

The major source of difficulty in many previous attempts to apply the mode approximation technique is the choice of a suitable mode shape $\underset{\sim}{\phi}(s)$. While for simple structures reasonable results can be achieved simply by basing this choice on intuition, a more consistent and general approach was required. Symonds [9] suggested an iterative procedure for the determination of $\underset{\sim}{\phi}(s)$ based on the method used by Lee and Martin [8] in their "piecewise stationary mode" technique and applied it successfully to the analysis of a tiploaded cantilever beam.

Here a new procedure to determine $\underset{\sim}{\phi}(s)$ will be presented. This procedure may be applied to any beam or frame structure being considered in the scope of this thesis. It is based on the method to determine mode shapes in rigid-plastic structures given by Martin [15] and has been adapted here to the homogeneous viscous material model. The steps in this new procedure are outlined below:

Step 0 : Select as the trial mode shape $\underset{\sim}{\phi}^i(s)$ the given initial velocity $\dot{\underset{\sim}{u}}^o$.

Step 1 : Apply 'loads' $\gamma(s)\underset{\sim}{\phi}^i(s)$ to the structure, and determine corresponding moments M and axial forces N in the structure.

Step 2 : Compute velocities $\dot{\underset{\sim}{u}}^i(s)$ resulting from this loading, using the principle of virtual velocities.

Step 3 : Normalise the velocities $\dot{\underset{\sim}{u}}^i(s)$ by dividing by the product $\int\gamma(s)\underset{\sim}{\phi}^i(s)\underset{\sim}{\phi}^i(s)ds$, to give a new trial mode shape,

$$\overset{i+1}{\underset{\sim}{\phi}(s)} = \frac{\dot{\underset{\sim}{u}}^i(s)}{\int\gamma(s)\underset{\sim}{\phi}^i(s)\underset{\sim}{\phi}^i(s)ds} \tag{2.9}$$

Return to Step 1, replacing $\underset{\sim}{\phi}^i(s)$ by $\underset{\sim}{\phi}^{i+1}(s)$.

The iterative procedure outlined above is continued until acceptable convergence of the mode shape has been attained. Numerical trials have shown that convergence is rapid, requiring only three or four cycles of the procedure before acceptable convergence is obtained, even for relatively complex structures. In Section 2.3, a proof of convergence of the algorithm for homogeneous viscous constitutive relations will be given and in Section 2.4 the numerical implementation of the algorithm will be discussed. Finally, the extention of the mode concept to geometrically non-linear analyses will be described in Section 2.5 .

## 2.3 Proof of Convergence of the Mode Algorithm

In order to establish convergence, it is convenient to write the constitutive equations in terms of derivatives of potention functions depending on kinematic quantities. In general terms, we may write the relation between internal forces $Q_j$ ($j = 1, 2, \ldots n$) and associated generalised strain rates $\dot{q}_j$ in the form

$$Q_j = \frac{\partial}{\partial \dot{q}_j} \left\{ \frac{n}{n+1} D(\dot{q}_j) \right\} , \tag{2.10}$$

where $D(\dot{q}_j)$ is the dissipation function,

$$D(\dot{q}_j) = Q_j \, \dot{q}_j \tag{2.11}$$

expressed in terms of $\dot{q}_j$. The dissipation function $D(\dot{q}_j)$ is homogeneous of degree $(n+1)/n$ in the components of $\dot{q}_j$. Because of the homogeneity of the relation, the term in parenthesis in equn. (2.10) can be seen to derive from

$$\int Q_j \, d\dot{q}_j = \frac{n}{n+1} Q_j \dot{q}_j = \frac{n}{n+1} D(\dot{q}_j) \quad . \tag{2.12}$$

For homogeneous viscous materials, mode shapes are given by stationary values of the functional

$$J(\dot{\tilde{u}}) = \int \frac{n}{n+1} D(\dot{\tilde{u}})ds - \lambda \int \gamma \, \dot{\tilde{u}} \, \dot{\tilde{u}} \, ds \quad , \tag{2.13}$$

where the generalised strain rates $\dot{q}_j$ and velocities $\dot{\tilde{u}}$ are related through a linear set of strain rate-displacement rate equations and $\lambda$ is a Lagrange multiplier. The minimum principle has been discussed in detail by Lee [35].

In the proposed algorithm, the problem is linearised. We choose a trial mode $\tilde{\phi}^i(s)$, and seek to minimise

$$J^*(\dot{\tilde{u}}) = \int \frac{n}{n+1} D(\dot{\tilde{u}})ds - \int \gamma \, \tilde{\phi}^i \, \dot{\tilde{u}} \, ds \quad . \tag{2.14}$$

We assume that the least value of the functional is given by $J^*(\dot{\tilde{u}}^i)$. The minimisation of $J(\dot{\tilde{u}})$ is exactly equivalent to the classical static problem of determining the velocities $\dot{\tilde{u}}^i$ due to static loading $\gamma\tilde{\phi}^i$ on the structure, and can be carried out by a variety of methods.

We adopt a normalisation rule for the mode shape, requiring that

$$\int \gamma \, \tilde{\phi} \, \tilde{\phi} \, ds = A \tag{2.15}$$

where A is an arbitrarily chosen constant. The next trial mode shape $\tilde{\phi}^{i+1}$ is then obtained from the velocities $\dot{\tilde{u}}^i$ through the relation,

$$\tilde{\phi}^{i+1} = \alpha \, \dot{\tilde{u}}^i \quad , \tag{2.16}$$

where

$$\alpha^2 \int \gamma \, \dot{\tilde{u}}^i \, \dot{\tilde{u}}^i \, ds = \int \gamma \, \tilde{\phi}^i \, \tilde{\phi}^i \, ds = A \tag{2.17}$$

Note that $\alpha$ is positive definite. We also define a parameter $\beta$ such that

$$\beta \int \gamma \, \underset{\sim}{\phi}^i \, \underset{\sim}{\phi}^i \, ds = \int \gamma \, \underset{\sim}{\phi}^i \, \underset{\sim}{\dot{u}} \, ds \quad . \tag{2.18}$$

It can readily be established that $\beta > 0$ : since $\underset{\sim}{\dot{u}}^i$ are the velocities resulting from the loading $\gamma\underset{\sim}{\phi}^i$, it follows that

$$\int \gamma \, \underset{\sim}{\phi}^i \, \underset{\sim}{\dot{u}}^i \, ds > 0 \quad , \tag{2.19}$$

and since

$$\int \gamma \, \underset{\sim}{\phi}^i \, \underset{\sim}{\phi}^i \, ds > 0 \quad , \tag{2.20}$$

it further follows from equn. (2.18) that $\beta$ is positive definite.

As a result of the normalisation rule (equn. (2.15)), the mode shape is given by the stationary values of

$$\int \frac{n}{n+1} \, D \, (\underset{\sim}{\phi}) \, ds \quad . \tag{2.21}$$

If we can show that

$$\int \frac{n}{n+1} \, D(\underset{\sim}{\phi}^{i+1}) \, ds \leq \int \frac{n}{n+1} \, D(\underset{\sim}{\phi}^i) \, ds \quad , \tag{2.22}$$

the proposed algorithm will lead, after repeated applications, to a local minimum value of J, and hence to a mode shape.

To establish this result, consider a velocity field given by $\beta\underset{\sim}{\phi}^i$. In view of the requirement that $\underset{\sim}{\dot{u}}^i$ minimises J* (equn. (2.14)), we have

$$\int \frac{n}{n+1} D\,(\beta\,\underset{\sim}{\phi}^{i})\,ds - \beta\int \gamma\,\underset{\sim}{\phi}^{i}\,\underset{\sim}{\phi}^{i}\,ds$$

$$\geq \int \frac{n}{n+1} D\,(\underset{\sim}{\dot{u}}^{i})\,ds - \int \gamma\,\underset{\sim}{\phi}^{i}\,\underset{\sim}{\dot{u}}^{i}\,ds \quad . \tag{2.23}$$

Using equn. (2.18), this reduces to

$$\int \frac{n}{n+1} D\,(\beta\,\underset{\sim}{\phi}^{i})\,ds \geq \int \frac{n}{n+1}\,(\underset{\sim}{\dot{u}}^{i})\,ds \quad . \tag{2.24}$$

Observing that D is homogeneous and of degree (n+1)/n, and substituting from equn (2.16), inequality (2.24) can be written as

$$(\alpha\beta)^{\frac{n+1}{n}} \int \frac{n}{n+1} D\,(\underset{\sim}{\phi}^{i})\,ds \geq \int \frac{n}{n+1} D\,(\underset{\sim}{\phi}^{i+1})\,ds \quad . \tag{2.25}$$

Inequality (2.22) will follow from inequality (2.25) if we can establish that

$$(\alpha\beta) \leq 1 \quad . \tag{2.26}$$

It may be noted from equns. (2.16) and (2.18) that

$$\int \gamma\,\underset{\sim}{\phi}^{i}\,\underset{\sim}{\dot{u}}^{i}\,ds = \beta\int \gamma\,\underset{\sim}{\phi}^{i}\,\underset{\sim}{\phi}^{i}\,ds = \frac{1}{\alpha}\int \gamma\,\underset{\sim}{\phi}^{i}\,\underset{\sim}{\phi}^{i+1}\,ds \ ,$$

and hence

$$(\alpha\beta)\int \gamma\,\underset{\sim}{\phi}^{i}\,\underset{\sim}{\phi}^{i}\,ds = \int \gamma\,\underset{\sim}{\phi}^{i}\,\underset{\sim}{\phi}^{i+1}\,ds \quad . \tag{2.27}$$

Further, it follows that

$$0 \leq \int \gamma\, (\underset{\sim}{\phi}^{i}-\underset{\sim}{\phi}^{i+1})(\underset{\sim}{\phi}^{i}-\underset{\sim}{\phi}^{i+1})\, ds$$

$$= \int \gamma\, \underset{\sim}{\phi}^{i}\, \underset{\sim}{\phi}^{i}\, ds - 2 \int \gamma\, \underset{\sim}{\phi}^{i}\, \underset{\sim}{\phi}^{i+1}\, ds + \int \gamma\, \underset{\sim}{\phi}^{i+1}\, \underset{\sim}{\phi}^{i+1}\, ds$$

$$= 2 \{\int \gamma\, \underset{\sim}{\phi}^{i}\, \underset{\sim}{\phi}^{i} - \int \gamma\, \underset{\sim}{\phi}^{i}\, \underset{\sim}{\phi}^{i+1}\, ds \} \quad , \qquad (2.28)$$

after using the normalisation rule equn. (2.15). Hence, with the use of equn. (2.28),

$$\int \gamma\, \underset{\sim}{\phi}^{i}\, \underset{\sim}{\phi}^{i}\, ds \geq \int \gamma\, \underset{\sim}{\phi}^{i}\, \underset{\sim}{\phi}^{i+1}\, ds = (\alpha\beta) \int \gamma\, \underset{\sim}{\phi}^{i}\, \underset{\sim}{\phi}^{i}\, ds. \qquad (2.29)$$

It follows from this that equn. (2.26) holds, and that $\underset{\sim}{\phi}^{i+1}$ is a better approximation to the mode shape than $\underset{\sim}{\phi}^{i}$, in that it is associated with a lower value of the function J.

The algorithm will converge onto mode shapes which are local minima in the functional J, and not, in general, onto saddle points or local maxima. More than one minimum may exist for the problem, and thus the mode onto which the algorithm converges may depend on the initial trial mode shape. It has been found in numerical work to date that if the initial velocity field is used as the first trial mode, the algorithm will provide the mode onto which the solution converges. This cannot be shown to be rigorously true, but is likely to be correct in almost all practical problems.

## 2.4 Numerical Implementation of the Mode Approximation Technique

In Section 2.2 an iterative procedure to determine the mode shape $\underset{\sim}{\phi}(s)$ was described. Here its numerical application to beam and frame structures will be discussed.

In formulating numerical solutions, the structure must be discretised: node positions are defined along the centre line of the structure, and straight massless elements connect adjacent nodes. It will be assumed that the displacements, and hence velocities, at the constrained nodes or supports are identically zero. Rotations and rotation rates will only be included in the description of displacement and displacement rates where they are constrained, and therefore zero. Furthermore, three independent constrained node displacement components are designated as those required to prevent rigid body motion of the structure, and these components are not included in the description of displacements and displacement rates. By this process we define a statically determinate "released" structure. The remaining displacements, velocities and accelerations are defined by the vectors $\underset{\sim}{u}(t)$, $\dot{\underset{\sim}{u}}(t)$ and $\ddot{\underset{\sim}{u}}(t)$ respectively.

Mass is lumped at nodes and a diagonal mass matrix [G] is defined in such a way that the kinetic energy of the structure is given by

$$K = \frac{1}{2}\, \dot{\underset{\sim}{u}}^{T}\, [G]\, \dot{\underset{\sim}{u}} \quad , \tag{2.30}$$

at any instant, where superscript T denotes the transpose. The mass terms corresponding to constrained velocity components of $\dot{\underset{\sim}{u}}$ can be arbitrarily defined; this includes the rotatory inertia associated with constrained (support) rotation rates. No other rotatory inertia terms appear in [G].

Generalised stresses consist of bending moment M at each node and axial force N in each element. Moments are distributed linearly across each element: if a, b are adjacent nodes separated by distance

$\ell_e$, and $M_a$, $M_b$ are the nodal moments, the bending moment at distance s from node a is given by

$$M(s) = M_a(1 - \frac{s}{\ell_e}) + M_b(\frac{s}{\ell_e}) \quad . \tag{2.31}$$

The axial force N is assumed to remain constant along an element. Conjugate to M and N are curvature rate $\dot{K}$ and axial strain rate $\dot{\varepsilon}$ which are related to M, N through the constitutive relations given by equn. (2.1). The nodal moments and element axial forces are ordered and form the vectors $\underset{\sim}{M}$, $\underset{\sim}{N}$.

At time t = 0 an impulse $\underset{\sim}{I}$ is applied to each node of the structure. The impulsive load imparts an initial velocity $\underset{\sim}{u}^o$ to each node, given by

$$\underset{\sim}{I} = [G]\dot{\underset{\sim}{u}}^o = [G]\dot{\underset{\sim}{u}}(o) \quad . \tag{2.32}$$

In Step 0 of the algorithm given in Section 2.2, $\dot{\underset{\sim}{u}}^o$ is selected as the initial trial mode shape, so that

$$\underset{\sim}{\phi}^o = \dot{\underset{\sim}{u}}^o \quad . \tag{2.33}$$

Subsequent trial mode shapes will be denoted by $\underset{\sim}{\phi}^i$, where superscript i denotes the i-th iteration.

In Step 1 of the algorithm, bending moments $\underset{\sim}{M}$ and axial forces $\underset{\sim}{N}$ which result from loading $[G]\underset{\sim}{\phi}^i$ on the structure must be calculated. For statically determinate structures, $\underset{\sim}{M}$ and $\underset{\sim}{N}$ are computed in the normal way from the equations of equilibrium. For hyperstatic structures, the force method of analysis is used: let the forces conjugate to the degrees of freedom in the structure which are constrained to be zero and which have been included in the definition of the vector $\dot{\underset{\sim}{u}}(t)$ be denoted by $\underset{\sim}{X}$. The nodal moments $\underset{\sim}{M}$ may be expressed as

$$\underset{\sim}{M} = \underset{\sim}{M}^s + [m]\,\underset{\sim}{X} \quad , \tag{2.34}$$

where $\underset{\sim}{M}^s$ are the bending moments in the statically determinate released structure resulting from loading $[G]\underset{\sim}{\phi}^i$. Each row of the influence matrix [m] is the set of nodal moments due to a unit value of some component of $\underset{\sim}{X}$. Similarly, the vector of axial forces $\underset{\sim}{N}$ is

$$\underset{\sim}{N} = \underset{\sim}{N}^s + [n]\,\underset{\sim}{X} \quad , \tag{2.35}$$

where $\underset{\sim}{N}^s$ are the axial forces in the statically determinate released structure resulting from loading $[G]\underset{\sim}{\phi}^i$, and each row of the influence matrix [n] is the set of axial forces due to a unit value of some component of $\underset{\sim}{X}$. Further we can define the components $m_j$, $n_j$ as being the bending moment along each element, and the axial force in each element, respectively, which result from a unit value of the j-th component $X_j$ of the vector $\underset{\sim}{X}$.

Using the constitutive equns. (2.1) and equns. (2.34) and (2.35) we can write the curvature rate $\dot{\kappa}$ and the strain rate $\dot{\varepsilon}$ at each point on the structure in terms of $\underset{\sim}{X}$. We denote the components of u which correspond to constrained but "released" nodes as $\dot{\underset{\sim}{u}}_j$. With the curvature rates $\dot{\kappa}$, axial strain rates $\dot{\varepsilon}$ and $\dot{u}_j$ as the kinematic system, and a unit value of the j-th component of $\underset{\sim}{X}$ together with its associated $m_j$, $n_j$, as the static system, the principle of virtual velocities gives the j-th component $\dot{u}_j$ of $\dot{\underset{\sim}{u}}$ as

$$\dot{u}_j = \sum_{\text{elements}} \{ \int_{\ell_e} m_j \dot{\kappa} ds + n_j \dot{\varepsilon} \ell_e \} \tag{2.36a}$$

$$= F(\underset{\sim}{X}) \tag{2.36b}$$

where $\ell_e$ is the length of an element. Noting that $\dot{u}_j$ is identically zero as it corresponds to a constrained node, equn. (2.36) may be repeated for each $\dot{u}_j$ conjugate to components of $\underset{\sim}{X}$, giving

$$\underset{\sim}{F}(\underset{\sim}{X}) = 0 \quad . \tag{2.37}$$

The solution of equn. (2.37) for $\underset{\sim}{X}$ is a nonlinear problem, and a full Newton-Raphson iterative procedure is used to determine the solution. A matrix of partial derivatives of $\underset{\sim}{F}$ with respect to $\underset{\sim}{X}$ is defined,

$$[A(\underset{\sim}{X})] = \frac{\partial \underset{\sim}{F}}{\partial \underset{\sim}{X}} \quad . \tag{2.38}$$

In the iterative scheme below the k-th trial value of $\underset{\sim}{X}$ is denoted by $\underset{\sim}{X}^k$. An improved trial value, $\underset{\sim}{X}^{k+1}$, is given by

$$\underset{\sim}{X}^{k+1} = \underset{\sim}{X}^k - [A^k]^{-1} \{ \underset{\sim}{F}(\underset{\sim}{X}^k) \} \quad . \tag{2.39}$$

The process is repeated until an estimate of $\underset{\sim}{X}$ of acceptable accuracy is obtained. At this point, the moment and axial force vectors $\underset{\sim}{M}$, $\underset{\sim}{N}$ may be evaluated from equns. (2.34) and (2.35). Step 1 of the algorithm is thus completed.

In Step 2, the velocities $\underset{\sim}{u}^i$ in the structure which result from loading $[G]\underset{\sim}{\phi}^i$ are required. We redefine the influence matrix [m] as the set of nodal moments in the structure due to a unit value of the components of $[G]\underset{\sim}{\phi}^i$. The bending moments $m_j$ then become the moments along each element resulting from a unit value of the j-th component of $[G]\underset{\sim}{\phi}^i$. Similary we redefine the influence matrix [n] as the set of axial forces in the structure due to a unit value of the components of $[G]\underset{\sim}{\phi}^i$. The axial forces $n_j$ then become the forces in each element resulting from a unit of the j-th component of $[G]\underset{\sim}{\phi}^i$. Applying the principle of virtual velocities, the j-th velocity component is

$$\dot{u}_j = \sum_{\text{elements}} \left( \int_{\ell_e} m_j \dot{K} ds + n_j \dot{\varepsilon} \ell_e \right) \qquad (2.40)$$

where $\ell_e$ is the length of an element, and where $\dot{\kappa}$, $\dot{\varepsilon}$ are obtained from equn. (2.1), using the bending moments $\underset{\sim}{M}$ and axial forces $\underset{\sim}{N}$ calculated in the previous step. Equation (2.40) is applied at each unconstrained node in turn to obtain the velocity vector $\dot{\underset{\sim}{u}}^i$.

In Step 3, $\dot{\underset{\sim}{u}}^i$ is normalised by dividing by the product $\underset{\sim}{\phi}^{i^T}[G]\underset{\sim}{\phi}^i$ to obtain an improved estimate of the mode shape,

$$\underset{\sim}{\phi}^{i+1} = \frac{\dot{\underset{\sim}{u}}^i}{\underset{\sim}{\phi}^{i^T}[G]\underset{\sim}{\phi}^i} \qquad (2.41)$$

A new load $[G]\underset{\sim}{\phi}^{i+1}$ is then formed, and the resultant bending moments $\underset{\sim}{M}$, axial forces $\underset{\sim}{N}$ and velocities $\dot{\underset{\sim}{u}}^{i+1}$ are calculated. Equation (2.41) is reapplied to revise $\underset{\sim}{\phi}$, and the iterative procedure is repeated until satisfactory convergence of $\underset{\sim}{\phi}$ has been obtained.

Once the algorithm has converged onto a mode shape, T(t) in equn. (2.8a) and (2.8b) can be calculated. Noting that the algorithm for the determination of the mode entails repeated solutions of the static problem in which loads $[G]\underset{\sim}{\phi}$ are applied to the structure, a work rate balance for this static problem is

$$\underset{\sim}{\phi}^T[G]\underset{\sim}{u} = \int D(\dot{\underset{\sim}{u}})ds \quad , \tag{2.42}$$

where $D(\dot{\underset{\sim}{u}})$ is a homogeneous dissipation function of degree $\frac{n+1}{n}$.
If $\alpha^*$ is a factor such that

$$\underset{\sim}{\phi} = \alpha^*\dot{\underset{\sim}{u}} \quad , \tag{2.43}$$

we have from equn. (2.42)

$$\frac{1}{\alpha^*}\underset{\sim}{\phi}^T[G]\underset{\sim}{\phi} = \frac{1}{(\alpha^*)^{n+1/n}}\int D(\underset{\sim}{\phi})ds \quad , \tag{2.44a}$$

hence

$$(\alpha^*)^{1/n} = \frac{\int D(\underset{\sim}{\phi})ds}{\underset{\sim}{\phi}^T[G]\underset{\sim}{\phi}} \tag{2.44b}$$

It follows then that the quotient in equn. (2.44b) required in equn. (2.8b) can be obtained from the normalization coefficient in the last step of the iterative procedure to determine the mode shape. This completes the mode solution algorithm and permits construction of mode solutions of the form of equn. (2.2). In the next section, the instantaneous mode technique will be described, whereby the mode solution technique is extended to include geometric nonlinearities.

## 2.5 The Instantaneous Mode Technique

Solutions obtained using the mode approximation technique described in the previous section hold rigorously if displacements are assumed to be small. In the geometrically nonlinear case mode solutions of the form of equn. (2.2) cannot be found, and the mode approximation technique is not directly applicable. The instantaneous mode concept (Symonds and Chon [13]) assures, however, that the response is such that the solution tends towards a stage which at each instant is close to that which satisfies the minimum principle for the mode (equn. (2.13)). In consequence, an approximate solution can be found by assuming that at an instant t the velocity field can be written as

$$\dot{\underset{\sim}{u}}(s,t) = \underset{\sim}{\phi}^t T^t \quad , \tag{2.45}$$

where $\underset{\sim}{\phi}^t$ is the mode shape computed for the instantaneous geometry of the structure, and the rate of change $\dot{T}^t$ is given by (see equn. (2.7b))

$$\dot{T}^t = -(T^t)^{1/n} \frac{\int D(\underset{\sim}{\phi}^t)ds}{\int \gamma \underset{\sim}{\phi}^t \underset{\sim}{\phi}^t ds} \quad . \tag{2.46}$$

Initial conditions are exactly the same as in the geometrically linear case, and the initial configuration of the structure is used to compute the initial mode shape, the value of T(o) and the initial value of $\dot{T}^t$. Thereafter we integrate forward in time, updating the geometry and using the new configuration to compute a new mode shape.

The determination of the mode shape is a geometrically linear problem at each instant; the method described in the previous section is used to find the mode shape for any updated configuration. The geometric nonlinearity is thus accounted for purely in the updating of the displaced shape.

In order to integrate the solution forward in time, a predictorcorrector method with an average rate of change is used. Hence with $t+1$ denoting the instant $t+\Delta t$, we put

$$T^{t+1} = T^{t} + \frac{\Delta t}{2}\left(\dot{T}^{t}+\dot{T}^{t+1}\right) \quad , \tag{2.47a}$$

$$\underset{\sim}{u}^{t+1} = \underset{\sim}{u}^{t} + \frac{\Delta t}{2}\left(T^{t}\underset{\sim}{\phi}^{t}+T^{t+1}\underset{\sim}{\phi}^{t+1}\right) \quad . \tag{2.47b}$$

In applying these equations we assume that $T^t$, $\underset{\sim}{\phi}^t$ and $\dot{\underset{\sim}{u}}^t$ are known. This is not sufficient information to compute $T^{t+1}$, $\underset{\sim}{u}^{t+1}$ from equns. (2.47), however, and an iterative scheme must be used. If subscript i indicates the i-th iteration, we put

$$T^{t+1}_{i+1} = T^{t} + \frac{\Delta t}{2}\left(\dot{T}^{t}+\dot{T}^{t+1}_{i}\right) \quad , \tag{2.48a}$$

$$\underset{\sim}{u}^{t+1}_{i+1} = \underset{\sim}{u}^{t} + \frac{\Delta t}{2}\left(T\underset{\sim}{\phi}^{t}+T^{t+1}_{i}\underset{\sim}{\phi}^{t+1}_{i}\right) \quad . \tag{2.48b}$$

The initial values of $\dot{T}^{t+1}_{i}$, $\underset{\sim}{\phi}^{t+1}_{i}$ are taken as $\dot{T}^{t}$, $\underset{\sim}{\phi}^{t}$; thereafter the updated configuration (obtained using $\underset{\sim}{u}^{t+1}$) is used to recompute $\underset{\sim}{\phi}^{t+1}$, and then $T^{t+1}$ from equn. (2.46), and the process is repeated. The iteration continued until satisfactory convergence in the values of $T^{t+1}$, $\underset{\sim}{u}^{t+1}$ is obtained. Numerical trials to date have indicated that convergence is rapid, requiring only two or three iterations to obtain satisfactory convergence.

The mode solution technique described here provides a simple and efficient numerical scheme for the solution of the dynamic problem in beam and frame structures, but is limited in application to problems where the response of the structure is predominantly of the modal type

In the following section, direct methods of analysis will be described which are not restricted to this class of problem, and are therefore more generally applicable.

# CHAPTER 3

# DIRECT METHODS OF ANALYSIS

As described in the previous chapter, the mode solution technique is an approximate method of analysis in which the actual velocity imparted to a structure is replaced by a velocity field of the mode shape. A fundamental assumption implicit in the technique is that the predominant pattern of behaviour of the structure throughout the time-span of deformation is of the mode shape chosen. As outlined in Section 2.5, the technique may be extended to include large geometric effects by changing the mode shape at suitably chosen time intervals, but the basic assumption that the structure behaves in a modal fashion for a discrete length of time remains. For certain classes of problems this approach provides solutions which are in excellent agreement with experimental results. These are impulsively loaded structures whose true response converges very rapidly onto a modal pattern of behaviour; for example, cantilever beams struck transversely at their tip, or symmetrically loaded rectangular frames. If, however, the true behaviour of the structure is such that convergence onto a mode shape is slow, or does not occur, then approximating the actual velocity of the structure by a velocity field in a mode shape may lead to unsatisfactory prediction of final deformations. Non-symmetrically loaded rectangular frames are such problems. Further, even when the mode approximation techniques does provide a reasonable final deformation pattern, the actual initial response of the structure, when stresses are at their maximum, is not given, as the structure is assumed to behave from time t = 0 in a modal fashion. The actual initial transient response, before a modal pattern of behaviour is adopted, is therefore ignored.

In order to quantify these non-modal effects, direct methods of analysis must be used. Here, two approaches will be discussed; firstly, a method based on the principle of Tamuzh [33], and secondly a more conventional direct method of analysis. In the first approach, accelerations (by the Tamuzh principle) and rates of change of moment (by a virtual velocities formulation) are found, and velocities and moments are integrated forward independently by an explicit forward integration scheme. In the second method, nodal forces in the structure corresponding to a given velocity field are determined, and from the equations of motion, accelerations are calculated. An implicit forward integration scheme is then used to determine velocities and nodal forces at subsequent time.

In the next section, the direct method of analysis based on the principle of Tamuzh [33] will be discussed. The primary aim of this formulation was to obtain solutions to dynamically loaded structures whose material characteristics were highly non-linear and in which conventional interpolation functions were not explicitly defined. The formulation presented here will be restricted to straight beams, and small displacement assumptions will be adopted, although the basic ideas may be extended to provide a more general solution procedure.

### 3.1.1 General Formulation for Geometrically Linear Problems Based on the Tamuzh Principle

In this formulation, only straight beams which are either cantilevered or supported at their ends will be considered. Node positions are defined by their co-ordinate on the x-axis of a cartesian co-ordinate system, and loads and displacement rates are assumed to lie in the x - y plane. Displacement rates or rotation rates at supports are specified to be zero. Transverse loads p(x,t) are applied along the beam and it is assumed that no loading occurs in the longitudinal direction. Initial transverse velocities $\dot{u}^o(x,t) = \dot{u}(x,o)$ are given. Shear and axial strain rates are assumed to be zero, and hence the only generalised strain rate which will be considered is the curvature rate $\dot{\kappa}$. The transverse displacement rate $\dot{u}(x,t)$, the rotation rate $\dot{\theta}(x,t)$ and the curvature rate $\dot{\kappa}$ must thus satisfy the relations

$$\dot{\theta} = \frac{\partial \dot{u}}{\partial x} \quad , \quad \dot{\kappa} = \frac{\partial^2 \dot{u}}{\partial x^2} \quad . \tag{3.1}$$

Generalised stresses consist of the shear force S and the bending moment M. The dynamic equation is

$$\frac{\partial^2 M}{\partial x^2} + \gamma \ddot{u} - p = 0 \tag{3.2}$$

where $\ddot{u} = \frac{\partial \dot{u}}{\partial t}$ is the acceleration, and $\gamma$ is the specific mass of the structure.

The constitutive equation will be assumed to take the form

$$\frac{\dot{\kappa}}{\dot{\kappa}_o} = \left(\frac{M}{M'_o}\right)^{n'} \quad , \tag{3.3a}$$

where $\dot{\kappa}_o$, $M'_o$ are material constants with dimensions of curvature rate and moment respectively, and $M'_o$, $n'$ are chosen such that equn. (3.3) matches the rigid-viscoplastic relation of equn. (3.3b).

$$\frac{\dot{\kappa}}{\dot{\kappa}_o} = \left(\frac{M}{M_o} - 1\right)^n \qquad \text{for} \quad n \geq M_o \quad ,$$

$$\dot{\kappa} = 0 \qquad \text{for} \quad 0 \leq M \leq M_o \quad . \tag{3.3b}$$

Matching techniques will be discussed in Chapter 4.

An an instant t it is assumed that the velocities are known, and hence through equns. (3.1) and (3.3a) the bending moments can be found. Accelerations may then be calculated using the equation of motion (3.2). Alternatively, we may use the principle of Tamuzh [33] to obtain the actual acceleration $\ddot{u}$ as those which provide an unconstrained minimum of the functional

$$J(\ddot{u}) = \int \frac{1}{2}\gamma\ddot{u}^2 dx - \int \rho\ddot{u}dx + \int M\ddot{\kappa}dx \quad . \tag{3.4}$$

In this expression, M is known and $\ddot{\kappa} = \partial^2\ddot{u}/\partial x^2$ are the curvature accelerations conjugate to $\ddot{u}$. Once the accelerations have been obtained, the velocities and hence, by the force method of analysis, the moments M at subsequent time may be found.

### 3.1.2 Numerical Formulation of the Problem

Since we are concerned here only with straight, transversely loaded beams which are assumed to undergo small displacements, the motion of each node is described by a transverse velocity $\dot{u}$ and a rotation $\dot{\theta}$ . The velocity field across a typical element ab is shown in Fig. 3.1 .

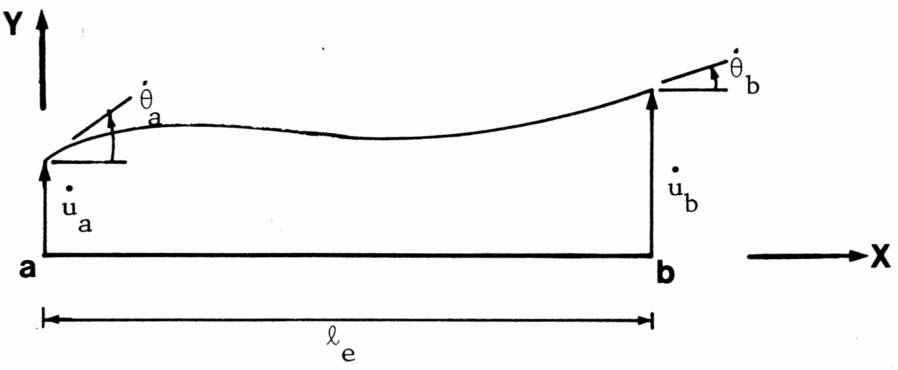
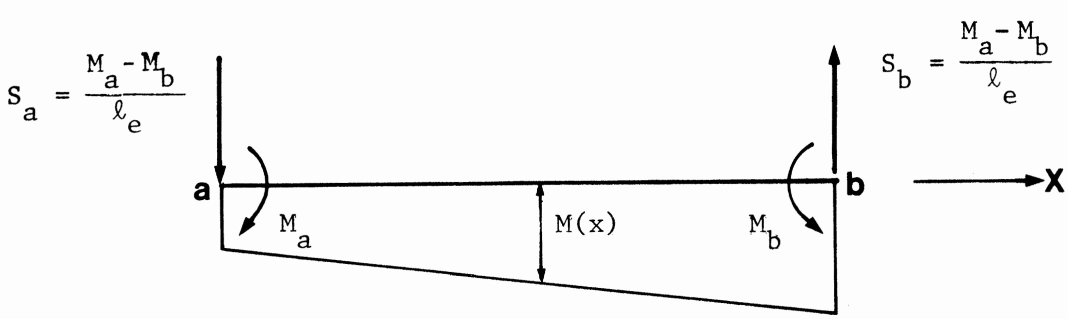



Figure 3.1 Element velocities

The velocity interpolation function will not be specified. Instead we assume that the end moments and shear forces on a typical element, shown in Fig. 3.2 , are in static equilibrium, and we thus imply a linear variation in bending moment across an element




$$M(x) = M_a(1 - x/\ell_e) + M\,(x/\ell_e)$$

Figure 3.2 End moments and shear forces.

Mass is lumped at node positions, and rotatory inertia of the lumped masses will be ignored. Accordingly we may define a diagonal mass matrix [G] comprising the lumped mass at each node, a velocity vector $\dot{\underset{\sim}{u}}$ and acceleration vector $\ddot{\underset{\sim}{u}}$ comprising only the unconstrained transverse velocity and acceleration components, respectively at each node,

$$\dot{\underset{\sim}{u}} = (\dot{u}_1\ \dot{u}_2\ \ldots.)^T \tag{3.5a}$$

and

$$\ddot{\underset{\sim}{u}} = (\ddot{u}_1\ \ddot{u}_2\ \ldots.)^T \tag{3.5b}$$

Consistent with the neglect of rotatory inertia, we assume that the external moment at any unconstrained node is zero. An external transverse force $\underset{\sim}{P}$ , conjugate to $\underset{\sim}{u}$ , is then defined.

The matrices and vectors defined above permit us to write the first two terms of equn. (3.4) in discrete form. To formulate the third term, we write the principle of virtual velocities (or accelerations) using the static and kinematic systems for the element shown in Figs. (3.3a) and (3.3b).

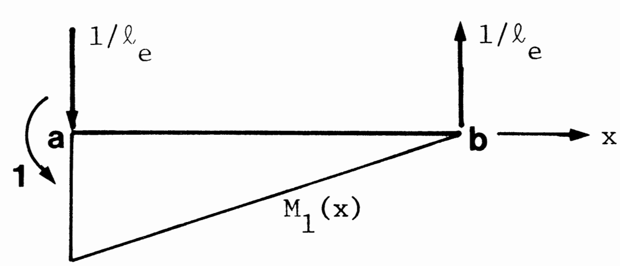



Figure 3.3a

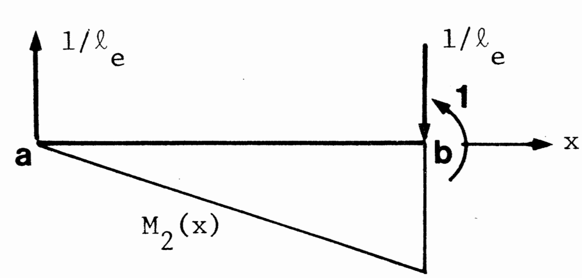



Figure 3.3b

Hence

$$\int_{ab} M\ddot{\kappa}dx = -\frac{(M_a-M_b)}{\ell_e}\ddot{u}_a - M_a\ddot{\theta}_a + \frac{(M_a-M_b)}{\ell_e}\ddot{u}_b + M_b\ddot{\theta}_b \,. \tag{3.6}$$

Only one moment value is identified at each node, and it follows that with the sign convertion shown in Fig. 3.2 moment equilibrium is satisfied. If further we add the contributions of each element given in equn. (3.6), it can be seen that the moment-rotational acceleration products will cancel out at interior nodes and vanish at supports because either the moment or rotational acceleration is zero. We may thus write

$$\int_{s} M\ddot{\kappa}dx = \ddot{\underset{\sim}{u}}^T[B]\underset{\sim}{M} \quad , \tag{3.7}$$

where $\underset{\sim}{M}$ is a vector of moments and [B] is a modified deformation matrix. This matrix can be assembled from element matrices of the form

$$[B]_e = \begin{bmatrix} 1/\ell_e & -1/\ell_e \\ -1/\ell_e & 1/\ell_e \end{bmatrix} \tag{3.8}$$

Tamuzh's functional now becomes

$$J = \frac{1}{2}\ddot{\underset{\sim}{u}}^T[G]\ddot{\underset{\sim}{u}} - \ddot{\underset{\sim}{u}}^T\underset{\sim}{P} + \ddot{\underset{\sim}{u}}^T[B]\underset{\sim}{M} \quad , \tag{3.9}$$

and the least value of this unconstrained quadratic expression in $\ddot{\underset{\sim}{u}}$ is given when

$$[G]\ddot{\underset{\sim}{u}} = \underset{\sim}{P} - [B]\underset{\sim}{M} \quad . \tag{3.10a}$$

The solution

$$\ddot{\underset{\sim}{u}} = [G]^{-1}(\underset{\sim}{P}-[B]\underset{\sim}{M}) \quad , \tag{3.10b}$$

provides the accelerations at time t provided that the external loads $\underset{\sim}{P}$ and the moments $\underset{\sim}{M}$ are known. This permits us to integrate forward in time to find the velocities at $t+\Delta t$, where $\Delta t$ is the time step.

In order that the procedure may move forward in time, however, we must be able to determine the moments $\underset{\sim}{M}$ at $(t+\Delta t)$ given the velocities $\dot{\underset{\sim}{u}}$ at $(t+\Delta t)$. Becuase of the non-linearity of the constitutive equn. (3.3a), this is not a trivial problem, and entails solving a system of non-linear simultaneous equations. An alternative scheme is used here in which the rates of change of moment $\dot{M}$ at time t are determined, and then used to determine M at $(t+\Delta t)$ by parallel forward integration.

To carry this out we use the principle of virtual velocities with the static systems for a typical element shown in Fig. 3.3a and Fig. 3.3b, and the kinematic system of Fig. 3.1 . Hence

$$-\frac{1}{\ell_e}\dot{u}_a - \dot{\theta}_a + \frac{1}{\ell_e}\dot{u}_b = \int_{ab} M_1\dot{\kappa}dx \quad , \tag{3.11a}$$

$$\frac{1}{\ell_e}\dot{u}_a - \frac{1}{\ell_e}\dot{u}_b + \dot{\theta}_b = \int_{ab} M_2\dot{\kappa}dx \quad , \tag{3.11b}$$

where

$$
\begin{aligned}
M_1 &= 1 - x/\ell_e \\
M_2 &= x/\ell_e \\
\dot{\kappa} &= \dot{\kappa}_o \left( \frac{M}{M'_o} \right)^{n'} \\
&= \dot{\kappa}_o \left\{ \frac{M_a(1-x/\ell_e) + M_b(x/\ell_e)}{M'_o} \right\}^{n'}
\end{aligned}
\tag{3.11c}
$$

The equations are now differentiated with respect to time, and give a relation between accelerations and rates of change of moment of the form

$$
\begin{bmatrix} C_1 & C_2 \\ C_2 & C_3 \end{bmatrix}
\begin{matrix} M_a \\ M_b \end{matrix}
=
\begin{bmatrix} -1/\ell_e & -1 & 1/\ell_e & 0 \\ 1/\ell_e & 0 & -1/\ell_e & 1 \end{bmatrix}
\begin{bmatrix} \ddot{u}_a \\ \ddot{\theta}_a \\ \ddot{u}_b \\ \ddot{\theta}_b \end{bmatrix}
\tag{3.12}
$$

In this equation

$$
C_1 = \frac{n'\dot{\kappa}_o}{M'^{n'}_o} \int_{ab} (1-x/\ell_e)^2 \{M_a(1-x/\ell_e) + M_b(x/\ell_e)\}^{n'-1} ds \quad , \tag{3.13a}
$$

$$
C_2 = \frac{n'\dot{\kappa}_o}{M'^{n'}_o} \int_{ab} (1-x/\ell_e)(x/\ell_e) \{ M_a(1-x/\ell_e) + M_b(x/\ell_e) \}^{n'-1} ds \quad , \tag{3.13b}
$$

and

$$
C_3 = \frac{n'{}_o}{M'^{n'}_o} \int_{ab} (x/\ell_e)^2 \{ M_a(1-x/\ell_e) + M_b(x/\ell_e) \}^{n'-1} ds \quad . \tag{3.13c}
$$

Equations (3.12) may be re-ordered, and written for each element as

$$\begin{bmatrix} C_1 & C_2 & 1 & 0 \\ C_2 & C_3 & 0 & -1 \end{bmatrix} \begin{bmatrix} \dot{M}_a \\ \dot{M}_b \\ \ddot{\theta}_a \\ \ddot{\theta}_b \end{bmatrix} = \begin{bmatrix} -1/\ell_e & 1/\ell_e \\ 1/\ell_e & -1/\ell_e \end{bmatrix} \begin{bmatrix} \ddot{u}_a \\ \ddot{u}_b \end{bmatrix} \qquad (3.14)$$

Equations (3.14) are then assembled into a global system, and may be expressed as

$$[D] \begin{bmatrix} \dot{\underset{\sim}{M}} \\ \hline \ddot{\underset{\sim}{\theta}} \end{bmatrix} = -[B]\ddot{\underset{\sim}{u}} \quad . \qquad (3.15)$$

Once $\ddot{\underset{\sim}{u}}$ is known, equn. (3.15) enables $\dot{\underset{\sim}{M}}$ and the rotational accelerations at the unconstrained nodes $\ddot{\underset{\sim}{\theta}}$ to be determined. In practise $\ddot{\theta}$ is not required, and is condensed out in the normal way.

The solution of equns. (3.10b) and (3.15) thus provides the accelerations and rates of change of moment at time t. Using an *explicit* parallel forward integration procedure, the velocities and moments at time (t+$\Delta$t) can be found. A modified Euler method is used. For a typical time step t in the solution process, we know the velocities and moments, and hence are able to determine $\ddot{\underset{\sim}{u}}_t$ and $\dot{\underset{\sim}{M}}_t$, the rates of change of velocity and moment. If t + 1 denotes the time step (t+$\Delta$t), a first estimate of $\dot{\underset{\sim}{u}}_{t+1}$ and $\underset{\sim}{M}_{t+1}$ is

$$\dot{\underset{\sim}{u}}_{t+1} = \dot{\underset{\sim}{u}}_t + \Delta t \, \ddot{\underset{\sim}{u}}_t \quad , \qquad (3.16a)$$

and

$$\underset{\sim}{M}_{t+1} = \underset{\sim}{M}_t + \Delta t \, \dot{\underset{\sim}{M}}_t \quad . \qquad (3.16b)$$

From these estimates, we use equns. (3.10b) and (3.15) to compute $\ddot{\underset{\sim}{u}}_{t+1}$

and $\dot{\underset{\sim}{M}}_{t+1}$. An improved estimate of $\dot{\underset{\sim}{u}}_{t+1}$ and $\underset{\sim}{M}_{t+1}$ is then found by averaging the rate quantities, so that

$$\dot{\underset{\sim}{u}}_{t+1} = \dot{\underset{\sim}{u}}_t + \frac{\Delta t}{2}(\ddot{\underset{\sim}{u}}_t + \ddot{\underset{\sim}{u}}_{t+1}) \quad , \qquad (3/17a)$$

and

$$\underset{\sim}{M}_{t+1} = \underset{\sim}{M}_t + \frac{\Delta t}{2}(\dot{\underset{\sim}{M}}_t + \dot{\underset{\sim}{M}}_{t+1}) \quad .$$

This process of refining the estimate of $\dot{\underset{\sim}{u}}_{t+1}$ and $\underset{\sim}{M}_{t+1}$ and recomputing $\ddot{\underset{\sim}{u}}_{t+1}$, $\dot{\underset{\sim}{M}}_{t+1}$ continues until convergence to a prescribed degree of accuracy is reached.

In order to commence the forward integration procedure, the initial moments are required. They must be dynamically admissible and must be compatible with the initial given velocities. This is a static problem in which nodes are treated as constrained, with unknown reactions. The solution is obtained using the force method of analysis and will be discussed in detail in Section 3.2.1, where both bending and axial effects will be treated.

The solution procedure given in this section was used to solve the problem of a cantilever beam subjected to an impulsive load at its tip, and the results obtained will be discussed in Chapter 6. The general experience with the approach was that it was a numerically inefficient one, mainly due to the numerical instability which arose due to the forward integration technique used. Extremely small time steps were required to ensure that divergence did not take place, which resulted in a computationally costly solution, even for the simple problem considered. A far more efficient solution scheme was obtained by combining the direct method of analysis outlined above with the mode solution technique.

At a typical instant of time t, we assume that $\dot{\underset{\sim}{u}}_t$ is known, and we can compute the mode velocity $\dot{\underset{\sim}{u}}_t^m$ from equn. (2.2). Using equn. (3.10b) and from equn. (2.2), we can find $\ddot{\underset{\sim}{u}}_t$ and $\ddot{\underset{\sim}{u}}_t^m$, respectively. The difference $(\ddot{\underset{\sim}{u}}_t - \ddot{\underset{\sim}{u}}_{t+1}^m)$ is then formed. We use this rate of change, by the same Euler method described above, to determine $(\dot{\underset{\sim}{u}}_{t+1} - \dot{\underset{\sim}{u}}_{t+1}^m)$ and hence find $\dot{\underset{\sim}{u}}_{t+1}$. The solution procedure continues until the difference between the velocities obtained by the direct method, and those obtained by the mode approximation technique is small, at which stage the direct method is dropped from the analysis procedure and the conventional mode analysis is adopted. This simple modification leads to a very considerable increase in the efficiency of the analysis as forward integration is performed on a decaying transient, and not on the actual accelerations.

In the next section a direct method of analysis which may be applied to beam and frame structures and which undergo large displacements will be given. An *implicit* forward integration scheme is presented which leads to a much more efficient solution procedure.

## 3.2 Direct Solutions Using An Implicit Forward Integration Scheme

Here we shall consider beam *and* polygonal frame planar structures which are supported at their ends and which undergo large displacements. As before, the problem is discretised by identifying nodes along the centre line of the structure, and it will be assumed that displacements, and hence velocities at the constrained nodes or supports are identically zero. Rotations, or rotation rates will only be included in the description of the displacements and displacement rates if they are constrained. Anticipating the force method formulation which is to follow, we designate three independent constrained node displacement components as those required to prevent rigid body motion of the structure, but which are not included in the description of the displacements and displacement rates. By this process we define a statically determinate "released" structure. The remaining displacement, velocity and acceleration components are grouped into the vectors $\underset{\sim}{u}(t)$, $\dot{\underset{\sim}{u}}(t)$ and $\ddot{\underset{\sim}{u}}(t)$ respectively, where t denotes time.

Mass is lumped at the nodes, and a diagonal mass matrix [G] is defined in such a way that the kinetic energy of the structure is given by

$$K = \frac{1}{2}\dot{\underset{\sim}{u}}^T[G]\dot{\underset{\sim}{u}} \quad , \tag{3.18}$$

at any instant. The mass terms corresponding to constrained velocity components of $\dot{\underset{\sim}{u}}$ can be arbitrarily defined; this includes the rotatory inertia associated with constrained (support) rotation rates. No other rotatory inertia terms appear in [G].

At time $t = 0$, an impulse is applied to the structure, represented by vector $\underset{\sim}{I}$. The impulsive load imparts an initial velocity to each node, given by

$$\underset{\sim}{I} = [G]\dot{\underset{\sim}{u}}^o = [G]\dot{\underset{\sim}{u}}(o) \quad . \tag{3.19}$$

We wish to determine the resulting motion of the structure, with initial displacements $\underset{\sim}{u}(o) = 0$ and initial velocities $\dot{\underset{\sim}{u}}(o)$ given by equn. (3.19).

As in the mode solution technique, a homogeneous viscous constitutive relation for a sandwich beam will be adopted, which is given by equns. (2.1).

### 3.2.1 Initial Moments and Axial Forces in the Structure

As a sub-problem of the general problem of integrating the equations of motion, the moments and axial forces at time $t = 0$ must be determined. This can be treated as a static problem. We have a *statically determinate* structure (the supports being the three node displacement components which prevent rigid body motion), with the node velocity components $\dot{\underset{\sim}{u}}^{o} = \dot{\underset{\sim}{u}}(o)$ completely prescribed. Note that $\dot{\underset{\sim}{u}}(o)$ contains both velocity components defined by the impulsive load and velocity components constrained to be zero. In addition, the geometry of the structure is defined by the initial displacements $\underset{\sim}{u}(o) = 0$ .

Using the principle of virtual work, we can readily compute the node velocities $\dot{\underset{\sim}{u}}$ in terms of the nodal forces $\underset{\sim}{X}$ . First, we formulate the nodal moments, represented by the vector $\underset{\sim}{M}$, in terms of the loads $\underset{\sim}{X}$ ;

$$M = [m]X \quad . \tag{3.20}$$

Each row of the influence matrix [m] is the set of nodal moments due to a unit value of some component of $\underset{\sim}{X}$ . Moments are distributed linearly across each element; if a, b are adjacent nodes separated by distance $\ell_e$, and $M_a$, $M_b$ are the node moments, the bending moment distance s from node a is given by

$$M(s) = M_a\left(1 - \frac{s}{\ell_e}\right) + M_b\left(\frac{s}{\ell_e}\right) \quad . \tag{3.21}$$

Using these relations, we can define the bending moment $m_j$ along each element resulting from a unit value of the j-th component $X_j$ of the load vector $\underset{\sim}{X}$ .

The axial forces are constant along each element, and are represented by an element axial force vector $\underset{\sim}{N}$ , given by

$$\underset{\sim}{N} = [n]\underset{\sim}{X} \quad . \tag{3.22}$$

Each row of the influence matrix [n] is the set of element axial forces due to a unit value of some component of $\underset{\sim}{X}$. From this, we can define the axial force $n_j$ in each element resulting from a unit value of the j-th component $X_j$ of the load vector $\underset{\sim}{X}$ .

Using the constitutive equn. (2.1) we can write the curvature rate $\dot{\kappa}$ and the strain rate $\dot{\varepsilon}$ at each point on the structure in terms of $\underset{\sim}{X}$ . With the curvature rates $\dot{\kappa}$, axial strain rates $\dot{\varepsilon}$ and velocities $\dot{\underset{\sim}{u}}$ as the kinematic system, and a unit value of the j-th component of $\underset{\sim}{X}$ , together with its associated $m_j$, $n_j$, as the static system, the principle of virtual velocities gives the j-th component $\dot{u}_j$ of $\dot{\underset{\sim}{u}}$ as

$$u_j = \sum_{\text{elements}} \int_{\ell_e} m_j \dot{\kappa} ds + n_j \dot{\varepsilon} \ell_e \tag{3.23}$$

where $\ell_e$ is the length of an element. This process is repeated for each component of $\dot{\underset{\sim}{u}}$, giving finally

$$\dot{\underset{\sim}{u}} = \underset{\sim}{F}(\underset{\sim}{X}) \tag{3.24}$$

It is a straight forward computational problem to determine $\dot{\underset{\sim}{u}}$ given $\underset{\sim}{X}$; we require, however, $\underset{\sim}{X}$ given $\dot{\underset{\sim}{u}}$ . This is a nonlinear problem, and a full Newton-Raphson iterative procedure is used to determine the solution. Equation (3.24) is written as

$$\dot{\underset{\sim}{u}} - \underset{\sim}{F}(\underset{\sim}{X}) = 0 \quad , \tag{3.25}$$

and a matrix of partial derivatives of $\underset{\sim}{F}$ with respect to $\underset{\sim}{X}$ is defined,

$$[A(\underset{\sim}{X})] = \frac{\partial \underset{\sim}{F}}{\partial \underset{\sim}{X}} \quad . \tag{3.26}$$

In the iterative scheme the k-th trial value of $\underset{\sim}{X}$ is denoted by $\underset{\sim}{X}^k$. An improved trial value, $\underset{\sim}{X}^{k+1}$ , is then given by

$$\underset{\sim}{X}^{k+1} = \underset{\sim}{X}^k - [A^k]^{-1} \{ \underset{\sim}{F}(\underset{\sim}{X}^k) - \dot{\underset{\sim}{u}} \} \quad . \tag{3.27}$$

The process is repeated until an estimate of $\underset{\sim}{X}$ of acceptable accuracy is obtained. At this point, the moment and axial force vectors $\underset{\sim}{M}$ , $\underset{\sim}{N}$ may be evaluated.

This procedure is applied to the determination of the initial moments and axial forces, given the initial velocities and the initial geometry. Note, however, that it might be applied at any instant, provided that the velocities $\dot{\underset{\sim}{u}}(t)$ and the configuration, described by $\underset{\sim}{u}(t)$, is given. We shall make use of this in the next section, but for instants after $t = 0$ the iteration scheme will be broadened to include forward integration.

### 3.2.2 An Implicit Time Integration Scheme

The forward integration of the equations of motion of impulsively loaded homogeneous viscous structures is not trivial, owing to the high degree of nonlinearity of the constitutive equations. Explicit forward integration schemes, although simple to formulate and implement, were found in general to be inadequate as they resulted in an unstable solution unless very small time steps were taken. In this section we present an implicit integration scheme in which equilibrium iterations are performed at each time step in order to improve the accuracy of the solution.

Let subscript t, t+1 denote the instants t, t+$\Delta$t respectively, and let superscript i denote the i-th iteration in the algorithm which will be outlined below. At time t velocities $\dot{\underset{\sim}{u}}_t$ and displacements $\underset{\sim}{u}_t$ are known, as are the nodal forces $\underset{\sim}{X}_t$. The nodal forces at time $t = 0$ are calculated by the procedure set out in the previous section; thereafter $\underset{\sim}{X}_t$ is calculated in the forward integration algorithm.

From the equation of motion, with the assumption that no external forces are applied to the structure at $t > 0$,

$$[G]\ddot{\underset{\sim}{u}}_t + \underset{\sim}{X}_t = 0 \quad , \tag{3.28a}$$

or

$$\ddot{\underset{\sim}{u}}_t = - [G]^{-1} \underset{\sim}{X}_t \quad . \tag{3.28b}$$

Rewriting equn. (3.24) at time t+1, we have

$$\dot{\underset{\sim}{u}}_{t+1} = \underset{\sim}{F}(\underset{\sim}{X}_{t+1}) = \underset{\sim}{F}_{t+1} \quad . \tag{3.29}$$

It is implicitly assumed that the function $\underset{\sim}{F}$, evaluated according to equns. (3.23) and (3.24), refers to the geometry of the structure at time t+1 . Thus $\underset{\sim}{F}_{t+1}$ can be found only when $\underset{\sim}{u}_{t+1}$ (or an estimate of $\underset{\sim}{u}_{t+1}$) is available.

Nonlinear geometrical effects are thus taken into account; because we are working with a *viscous* material, and are computing velocities in an instantaneously defined configuration, no further complications arise from the inclusion of large displacements.

Increments in $\ddot{\underset{\sim}{u}}$, $\dot{\underset{\sim}{u}}$, $\underset{\sim}{u}$ and $\underset{\sim}{X}$ are defined by the equations

$$
\begin{aligned}
\ddot{\underset{\sim}{u}}_{t+1} &= \ddot{\underset{\sim}{u}}_t + \Delta\ddot{\underset{\sim}{u}} \quad ,\\
\dot{\underset{\sim}{u}}_{t+1} &= \dot{\underset{\sim}{u}}_t + \Delta\dot{\underset{\sim}{u}} \quad ,\\
\underset{\sim}{u}_{t+1} &= \underset{\sim}{u}_t + \Delta\underset{\sim}{u} \quad ,\\
\text{and} \quad \underset{\sim}{X}_{t+1} &= \underset{\sim}{X}_t + \Delta\underset{\sim}{X} \quad .
\end{aligned}
\tag{3.30}
$$

Substituting $\ddot{\underset{\sim}{u}}_{t+1}$ , $\underset{\sim}{X}_{t+1}$ from equns. (3.30) into the equation of motion (3.28a) at time t+1 , we have

$$[G](\ddot{\underset{\sim}{u}}_t+\Delta\ddot{\underset{\sim}{u}}) + (\underset{\sim}{X}_t+\Delta\underset{\sim}{X}) = 0 \quad , \tag{3.31a}$$

hence

$$[G]\Delta\ddot{\underset{\sim}{u}} + \Delta\underset{\sim}{X} = -\left([G]\ddot{\underset{\sim}{u}}_t+\underset{\sim}{X}_t\right) \quad . \tag{3.31b}$$

Substituting also into equn. (3.29), we may put

$$\dot{\underset{\sim}{u}}_t + \Delta\dot{\underset{\sim}{u}} = \underset{\sim}{F}_t + [A_t]\Delta\underset{\sim}{X} \quad , \tag{3.32a}$$

where

$$[A_t] = [A(\underset{\sim}{X}_t)] = \left.\frac{\partial\underset{\sim}{F}}{\partial\underset{\sim}{X}}\right|_t \quad , \tag{3.32b}$$

and is given by the last evaluation of equn. (3.26) in the iterative procedure to determine $\underset{\sim}{X}_t$ , described in the previous section. Note that as the constitutive relation used is homogeneous, the partial derivatives of $\underset{\sim}{F}$ with respect to $\underset{\sim}{X}$ may be formulated explicitly. Integration is then carried out over the length of an element, and the contributions of each element is summed over the structure.

From equns. (3.32a) and (3.32b) ,

$$\Delta\dot{\underset{\sim}{u}} = [A_t]\Delta\underset{\sim}{x} + (\underset{\sim}{F}_t - \dot{\underset{\sim}{u}}_t) \quad . \tag{3.32c}$$

Using the trapezoidal rule, we put

$$\dot{\underset{\sim}{u}}_{t+1} = \dot{\underset{\sim}{u}}_t + \frac{\Delta t}{2}(\ddot{\underset{\sim}{u}}_t + \ddot{\underset{\sim}{u}}_{t+1}) \quad , \tag{3.33}$$

and hence, from the first of equns. (3.30) ,

$$\Delta\ddot{\underset{\sim}{u}} = \ddot{\underset{\sim}{u}}_{t+1} - \ddot{\underset{\sim}{u}}_t = \frac{2}{\Delta t}\Delta\dot{\underset{\sim}{u}} - 2\ddot{\underset{\sim}{u}}_t \quad . \tag{3.34}$$

Substituting equn. (3.34) into equn. (3.31b), we have

$$\frac{2}{\Delta t}[G]\Delta\dot{\underset{\sim}{u}} + \Delta\underset{\sim}{X} = -(-[G]\ddot{\underset{\sim}{u}}_t + \underset{\sim}{X}_t) \quad . \tag{3.35}$$

Finally, substituting for $\Delta\dot{\underset{\sim}{u}}$ from equn. (3.32c), and rearranging, we have

$$\{[A_t] + \frac{\Delta t}{2}[G^{-1}]\}\Delta\underset{\sim}{X} = -(\underset{\sim}{F}_t - \dot{\underset{\sim}{u}}_t) + \frac{\Delta t}{2}(\ddot{\underset{\sim}{u}}_t - [G]^{-1}\underset{\sim}{X}_t) \tag{3.36}$$

Equation (3.36) is solved for $\Delta\underset{\sim}{X}$ , and $\Delta\dot{\underset{\sim}{u}}$ follows from equn. (3.32c). Equations (3.30) then give $\dot{\underset{\sim}{u}}_{t+1}$ , $\underset{\sim}{X}_{t+1}$ , and $\underset{\sim}{u}_{t+1}$ is found by a further application of the trapezoidal rule

$$\underset{\sim}{u}_{t+1} = \underset{\sim}{u}_t + \frac{\Delta t}{2}(\dot{\underset{\sim}{u}}_t + \dot{\underset{\sim}{u}}_{t+1}) \quad . \tag{3.37}$$

This procedure will be numerically stable, but will introduce errors which will propagate as the solution advances in time. In particular, equn. (3.32a) does not include the effects of change in geometry, and hence the equation of motion at time t+1 will not be exactly satisfied. In order to improve estimates of $\Delta\dot{\underset{\sim}{u}}$, $\Delta\underset{\sim}{X}$ and to incorporate the error in the equation at the previous time step, an iterative scheme is introduced. Letting superscript (i+1) denote the (i+1)-th iteration, we write equns. (3.28a) and (3.25) as

$$[G]\ddot{\underset{\sim}{u}}^{i+1}_{t+1} + \underset{\sim}{X}^{i+1}_{t+1} = 0 \quad , \tag{3.38}$$

and

$$\dot{\underset{\sim}{u}}^{i+1}_{t+1} = \underset{\sim}{F}(\underset{\sim}{X}^{i+1}_{t+1}) = \underset{\sim}{F}^{i+1}_{t+1} \quad . \tag{3.39}$$

Redefining the increments of equns. (3.30) as residuals, we have

$$\ddot{\underset{\sim}{u}}^{i+1}_{t+1} = \ddot{\underset{\sim}{u}}^{i}_{t+1} + \Delta\ddot{\underset{\sim}{u}}^{i}_{t+1} \quad , \tag{3.40a}$$

$$\dot{\underset{\sim}{u}}^{i+1}_{t+1} = \dot{\underset{\sim}{u}}^{i}_{t+1} + \Delta\dot{\underset{\sim}{u}}^{i}_{t+1} \quad , \tag{3.40b}$$

$$\underset{\sim}{u}^{i+1}_{t+1} = \underset{\sim}{u}^{i}_{t+1} + \Delta\underset{\sim}{u}^{i}_{t+1} \quad , \tag{3.40c}$$

$$\underset{\sim}{X}^{i+1}_{t+1} = \underset{\sim}{X}^{i}_{t+1} + \Delta\underset{\sim}{X}^{i}_{t+1} \quad . \tag{3.40d}$$

From the trapezoidal rule at the i-th and (i+1)-th iteration, we may write

$$\dot{\underset{\sim}{u}}^{i}_{t+1} = \dot{\underset{\sim}{u}}_{t} + \frac{\Delta t}{2}(\ddot{\underset{\sim}{u}}_{t} + \ddot{\underset{\sim}{u}}^{i}_{t+1}) \tag{3.41a}$$

and

$$\dot{\underset{\sim}{u}}^{i+1}_{t+1} = \dot{\underset{\sim}{u}}_{t} + \frac{\Delta t}{2}(\ddot{\underset{\sim}{u}}_{t} + \ddot{\underset{\sim}{u}}^{i+1}_{t+1}) \tag{3.41b}$$

and hence

$$\Delta\dot{\underset{\sim}{u}}^{i}_{t+1} = \frac{\Delta t}{2}\,\Delta\ddot{\underset{\sim}{u}}^{i}_{t+1} \quad . \tag{3.41c}$$

From the first of equns. (3.40) and equn. (3.41c), we find

$$\ddot{\underset{\sim}{u}}^{i+1}_{t+1} = \ddot{\underset{\sim}{u}}^{i}_{t+1} + \frac{2}{\Delta t}\,\Delta\dot{\underset{\sim}{u}}^{i}_{t} \quad . \tag{3.42}$$

Substituting equns. (3.42) and (3.40d) into equn. (3.38), and rearranging we have

$$\frac{2}{\Delta t}\,[G]\Delta\dot{\underset{\sim}{u}}^{i}_{t+1} + \Delta\underset{\sim}{X}^{i}_{t+1} = -\,([G]\ddot{\underset{\sim}{u}}^{i}_{t+1}+\underset{\sim}{X}^{i}_{t+1}) \quad . \tag{3.43}$$

From equn. (3.32a), we write

$$\dot{\underset{\sim}{u}}^{i}_{t+1} + \Delta\dot{\underset{\sim}{u}}^{i}_{t+1} = \underset{\sim}{F}^{i}_{t+1} + [A^{i}_{t+1}]\Delta\underset{\sim}{X}^{i}_{t+1} \quad , \tag{3.44a}$$

where

$$[A^{i}_{t+1}] = \left.\frac{\partial \underset{\sim}{F}}{\partial \underset{\sim}{X}}\right|_{\underset{\sim}{X}^{i}_{t+1}} \quad . \tag{3.44b}$$

The matrix $[A^{i}_{t+1}]$ is re-evaluated at the beginning of each equilibrium iteration by taking the partial derivatives of the current value of $\underset{\sim}{F}$ given for the (i+1)-th iteration by equn. (3.39), with respect to the current value of the body forces, $\underset{\sim}{X}^{i}_{t+1}$ and for the configuration denoted by $\underset{\sim}{u}^{i}_{t+1}$ .

In order to find the final expression for $\Delta\underset{\sim}{X}^{i}_{t+1}$, we substitute equn. (3.44a), into equn. (3.43), and rearrange so that

$$\{[A^{i}_{t+1}] + \frac{\Delta t}{2}\,[G]^{-1}\}\Delta\underset{\sim}{X}^{i}_{t+1} = -\,(\underset{\sim}{F}^{i}_{t+1}-\dot{\underset{\sim}{u}}^{i}_{t+1}) - \frac{\Delta t}{2}\,(\ddot{\underset{\sim}{u}}^{i}_{t+1}-[G]^{-1}\underset{\sim}{X}^{i}_{t+1}) \tag{3.45}$$

Equation (3.45) provides $\Delta \underset{\sim}{X}^{i}_{t+1}$ , and $\Delta \dot{\underset{\sim}{u}}^{i}_{t+1}$ is then obtained from equn. (3.43). By the same process which led to equn. (3.41c), we have

$$\Delta \dot{\underset{\sim}{u}}^{i}_{t+1} = \frac{\Delta t}{2} \Delta \ddot{\underset{\sim}{u}}^{i}_{t+1} \quad . \tag{3.46}$$

We may thus find revised estimates $\dot{\underset{\sim}{u}}^{i+1}_{t+1}$ , $\underset{\sim}{u}^{i+1}_{t+1}$ , $\underset{\sim}{X}^{i+1}_{t+1}$ from equns. (3.40). The iterative procedure is repeated until the residual quantities $\Delta \dot{\underset{\sim}{u}}$ , $\Delta \underset{\sim}{u}$ , $\Delta \underset{\sim}{X}$ are acceptably small.

Once the solution quantities at time t+1 have been computed to the required tolerance, the solution proceeds to the next time step. The algorithm has been found to be an efficient procedure for the homogeneous structures under consideration. Much larger time steps than can be used in an explicit scheme are possible, and, even including the iteration within the time step, this leads to a much less costly computational scheme.

In Chapter 4, we shall discuss the matching strategy to choose $\sigma'_o$ and $n'$ .

# CHAPTER 4

## THE MATCHING PROCEDURE

The use of a homogeneous viscous relation for rigid-plastic dynamic analysis is based on the supposition that the rigid-visco-plastic relation

$$\frac{\dot{\varepsilon}}{\dot{\varepsilon}_o} = \left(\frac{\sigma}{\sigma_o} - 1\right)^n \qquad \text{for} \quad \sigma \geq \sigma_o \quad ,$$

$$\frac{\dot{\varepsilon}}{\dot{\varepsilon}_o} = 0 \qquad \text{for} \quad \sigma < \sigma_o \quad , \tag{4.1}$$

can be adequately approximated in any particular problem by a relation for the form

$$\frac{\dot{\varepsilon}}{\dot{\varepsilon}_o} = \left(\frac{\sigma}{\sigma'_o}\right)^{n'} \quad , \tag{4.2a}$$

with

$$\sigma'_o = \mu\sigma_o \quad , \tag{4.2b}$$

$$n' = \nu n \quad . \tag{4.2c}$$

A strategy for choosing $\mu$ or $\nu$ is thus an essential part of the application of the homogeneous viscous material in dynamic problems.

Symonds [9] suggested that the factors $\mu$ and $\nu$ should be chosen such that equns. (4.1) and (4.2a) have a common intercept and slope at a value of strain rate which is the largest occuring in the structure at $t = 0$. If this largest value is denoted by $\dot{\varepsilon}_{max}$, this strategy gives

$$\nu = \frac{1 + \left(\frac{\dot{\varepsilon}_{max}}{\dot{\varepsilon}_o}\right)^{1/n}}{\left(\frac{\dot{\varepsilon}_{max}}{\dot{\varepsilon}_o}\right)^{1/n}} \quad , \qquad (4.3a)$$

$$\mu = \frac{1 + \left(\frac{\dot{\varepsilon}_{max}}{\dot{\varepsilon}_o}\right)^{1/n}}{\left(\dot{\varepsilon}_{max}\right)^{1/\nu n}} \qquad (4.3b)$$

This matching is shown diagrammatically in Fig. 4.1 ; the rigid-viscoplastic relation is given by curve 1, and curve 2 depicts the homogeneous viscous relation matched by the procedure outlined above.

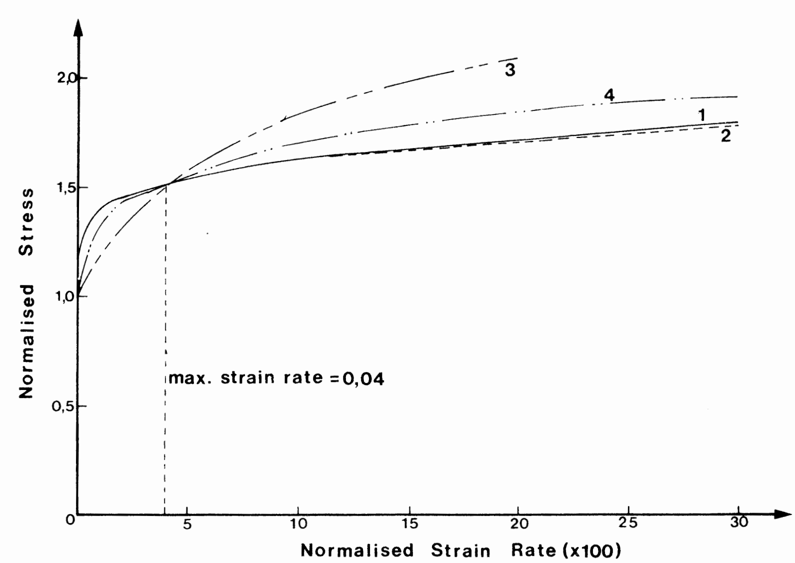



Figure 4.1 The matching procedure.

Curve 1 : rigid-viscoplastic curve; $n = 5$

Curve 2 : homogeneous viscous curve matched on slope and intercept at $\dot{\varepsilon}/\dot{\varepsilon}_o = 0{,}05$; $\mu = 1{,}904$ ; $n' = 14{,}52$

Curve 3 : homogeneous viscous curve matched on intercept alone at $\dot{\varepsilon}/\dot{\varepsilon}_o = 0{,}04$; $\mu = 2{,}904$; $n = n' = 5$

Curve 4 : homogeneous viscous curve matched on intercept alone at $\dot{\varepsilon}/\dot{\varepsilon}_o = 0{,}04$ with increased $n'$; $\mu = 2{,}204$; $n' = 8{,}75$.

In general, this strategy appears to be effective in simple problems; difficulties occur under two circumstances, however, when a generalization is attempted. The first is when it is difficult to estimate, or interpret, the maximum initial strain rate in the structure. The second is that the value of n' is typically in the range 10 - 15. This results in considerable numerical difficulties, particularly in direct solution techniques.

In certain of the analyses performed here using both the mode and the direct methods of analysis, the full matching procedure of equns. (4.3) was found to be unnecessary. Satisfactory results were obtained by setting $n' = n$, and choosing $\mu$ so that the homogeneous viscous curve intersects the rigid-viscoplastic curve at $\dot{\varepsilon}_{max}$. This has the obvious numerical advantage of keeping the value of n' low, thereby eliminating potential numerical problems. The scheme is illustrated by curve 3 in Fig. 4.1 .

This scheme does not, however, always lead to a satisfactory solution; as the magnitude of $\dot{\varepsilon}_{max}$ increases, the difference between curve 1 and curve 3 becomes large if the matching scheme given above is used. The true material behaviour is therefore not correctly modelled. In order to better approximate curve 1, a compromise may be made whereby the value of n' is chosen as large as possible, with $n' > n$, and with the choice being dictated by the ability of the solution procedure to carry through the analysis without computational difficulties. As before $\mu$ is chosen so that the rigid-viscoplastic relation and the homogeneous relation intersect at $\dot{\varepsilon}_{max}$. This is shown diagrammatically by curve 4 in Fig. 4.1 .

Whilst the compromise matching procedure has disadvantages in that a trial estimate of the largest n' which can be tolerated must be made, it seems a reasonable approach in the context of beams and frames. The results of analyses performed using this approach have shown that the higher the value of n' the better the correlation with test data. The errors introduced by low values of n' are not consistent, and thus upper or lower bounds cannot be established.

The best choice of the strain rate magnitude on which the matching is based is also open to question. Symonds [39] has also suggested that matching can be based on an average strain rate. Another possible approach in numerical analysis is to rematch at the beginning of each time step. While the compromise procedure given in this section has provided the best results in this study, further work is required to give firm guidelines on the matching strategy in any particular case.

Finally, consideration must be given to rematching equns. (4.3) if the predominant mode of deformation changes during the analysis. In the majority of analyses performed, flexural deformations are of prime importance and therefore the maximum curvature rate $\dot{\kappa}_{max}$ is used in equn. (4.3). However in the case of a fixed end beam, for example, deformation is initially flexural, but changes to a strong membrane action when the transverse displacement becomes comparable to the depth. In this case, a decision was made to rematch at the time interval when it was found that $N/N_o \geq 0.1$. New matching factors were then calculated based on the current maximum axial strain rate, using equns. (4.3). Thereafter, no further changes were made to the matching factors.

# CHAPTER 5

## COMPUTER IMPLEMENTATION OF THE MODE AND THE CONVENTIONAL DIRECT SOLUTION TECHNIQUE

Two computer programs, GNLIMST (Geometrically Non-linear Instantaneous Mode Solution Technique) and DAGNVS (Direct Analysis of Geometrically Non-linear Viscous Structures) have been developed to implement the solution procedures given in Chapter 2 and Section 3.2, respectively, and have been used successfully to analyse a variety of beam and frame structures.

The data input for each program is identical, comprising material constants ($\dot{\varepsilon}_o$, $\sigma_o$, n) , the co-ordinates of the discretised structure, node masses, the initial velocity field and control parameters such as time step size and output requirements. The data input will be discussed in detail in the user manual given in Appendix A followed by listings of GNLIMST and DAGNVS in Appendix B and Appendix C respectively. In the following two sections a description of how the two programs implement the above numerical techniques will be given.

### 5.1 Numerical Implementation of the Instantaneous Mode Technique using GNLIMST

GNLIMST is a FORTRAN program which is structured in modular form, that is, it consists of a driver routine which calls a number of subroutines, each of which performs a specific independent task.

Once the data has been read (subroutine INPUT) and displayed in order that it may be verified (subroutine DATA), the initial mode shape of the structure must be calculated. A macro flow chart of this procedure is shown in Fig. 5.1 . Before the mode solution algorithm may commence, the influence matrices [m] and [n] described in Section 2.4 are assembled. Each row of these matrices is the set of nodal moments and element axial forces respectively in the structure resulting from a unit load applied in turn at a node in the global X and Y directions. If the structure is hyperstatic, degrees of freedom must be defined as input data which are to be released so that the structure becomes statically determinate. The influence matrices [m] and [n] are dependent on the current geometry of the structure, and must therefore be revised at each time step if geometric effects are to be included. The numerical formulation of [m] and [n] is a straight forward static problem which is easily automated. This procedure is performed in subroutine STAT.

The mode algorithm may now commence. A mode shape is selected, the first trial being set equal to the given initial velocity and subsequent trials being calculated by the algorithm which follows. A load vector is then formed (subroutine LOAD), given by the product of the lumped mass matrix and the current mode shape. The bending moments and axial forces resulting from this loading are now calculated. If the structure is statically determinate, these are given by the product of the respective influence matrices and the load vector. For hyperstatic structures the previous products define a statically admissible bending moment and axial force diagram which is required in the force method of analysis used to determine the redundants. The above products are calculated in subroutine LOAD.

For hyperstatic problems, iteration is required to determine the redundants. Compatibility equations are formed, corresponding to each released degree of freedom, using equn. (2.36). The curvature rate and axial strain rate required in equn. (2.36) are calculated from the constitutive relation equn. (2.1). The bending moment diagram and axial force diagram required in equn. (2.1) may be formed in terms of the redundants (equns. (2.34) and (2.35)). Assuming that the axial force is constant along an element, and that the bending moment varies linearly between nodes, and noting that equn. (2.1) is homogeneous in the unknown redundants, equns. (2.36) may be evaluated, the integration over an element being performed explicitly, and the contribution from each element being summed over the structure. These compatibility equations are formed in subroutine COMEQU.

Since the above compatibility equations correspond to released degrees of freedom the velocity at these degrees of freedom should, when the correct choice of redundant forces is chosen, be zero. In general, we have no a priori knowledge of the numerical value of the redundant forces, and trial values must be assumed which when substituted into equn. (2.36) give non zero velocities. In order to determine the correct value of the redundants, the Newton-Raphson iterative solution technique is used. A matrix of partial derivatives of the compatibility equations with respect to the redundants is formed. Again, due to the homogeneity of the constitutive relations, these may be formulated explicitly. Integration is carried out over an element, and the contribution of each element is summed over the structure (subroutine PDIFF).

Using equn. (2.39) an improved estimate of the redundants is found. This requires the solution of a set of linear equations. Partial pivoting is used for improved numerical accuracy (subroutine PIVOT [41]). The value of the redundants is revised using equn. (2.39) in subroutine DELTA, and convergence is checked. Convergence is assumed when the velocities evaluated in COMEQU are acceptably small or the change in the rudundants is within a predefined tolerance (0.5%). If convergence has occurred, then the bending moment and axial force diagrams are evaluated using equns. (2.34) and (2.35) respectively. If not, the revised values of the redundants are used in equns. (2.34) and (2.35) to form a new trial bending moment and axial force diagram, and the program returns to subroutine COMEQU.

Once the redundants have been found, the velocity field corresponding to the bending moment and axial force diagram determined above is calculated using the principle of virtual velocities (equn. (2.40)). Since the bending moment and axial force diagrams are known, the curvature rate and axial strain rate may be evaluated explicitly over an element. The bending moment and axial force diagrams resulting from a unit load applied in the global X and Y directions are also known from subroutine STAT. Equation (2.40) may then be evaluated over an element, and the contribution from each element summed over the structure to give the velocity field in the global X and Y directions at each node (subroutine VELOC).

This velocity field is normalised using equn. (2.41) to give a new trial mode shape. This shape is compared to the previous trial to determine whether convergence onto the true mode shape has occured to within an acceptable tolerance. If convergence has not occurred, further iteration in the mode algorithm is required, and the program returns to subroutine LOAD with the current mode shape being used to calculate loads. If convergence *has* occurred, the time function required in equn. (2.2) and its derivative with respect to time may be calculated using equns. (2.8). In turn, equn. (2.8) is evaluated using equns. (2.4) and (2.44b), the latter being calculated from the normalisation coefficient used in the last step of the iteration procedure to determine the mode shape. The check on the mode shape convergence and the calculation of the time function and its derivative is performed in subroutine MODECH.

Matching the homogeneous viscous relation to the rigid-visco-plastic model is now performed. In the program GNLIMST, matching is performed on intercept alone. The maximum curvature rate is calculated from the maximum (known) bending moment in the structure, using the uniaxial form of equn (2.1), given in generalised terms by equn. (4.1). The stress factory $\mu$ is then calculated from equn. (4.3b) with $\nu = 1$. All subsequent calculations are performed using the matched yield stress given by equn. (4.2b). The matching procedure is carried out in subroutine MATCH. Note that a small displacement solution to the problem may now be found.

If geometric effects are to be included, the instantaneous mode solution technique, shown by the macro flowchart in Fig. 5.2, is used. From the time function calculated above, an estimate of the total time of deformation $t_f$ may be obtained by setting the right hand side of equn.(2.8a) to zero and solving for $t = t_f$. The total time is divided into a suitable number of intervals to give a time step $\Delta t$.

The mode shape, the time function and the derivative of time function evaluated at the current time step are stored (subroutine STORE).

Iteration is then performed to evaluate the mode shape, the velocity and displacement fields at the subsequent time step ($t+\Delta t$). The time function, velocities and displacements are evaluated using equns. (2.48a), (2.45) and (2.48b) at $t+\Delta t$, and the geometry of the structure is revised accordingly (subroutine UPDATE). Since the structural configuration has changed, a new mode shape must be determined using the updated geometry and the current velocity field calculated above. The procedure outlined above and shown in Fig. 5.1 is thus repeated to obtain a trial mode shape, time function and derivative of the time function at $t+\Delta t$. Equations (2.48a) and (2.48b) are re-evaluated to obtain an improved estimate of the time function and deflections at this time step, and the process continues until convergence has been obtained in the time function and deflection quantities. Note that since the estimate of deflections at $t+\Delta t$ is revised after each iteration, a new mode shape must be formed.

Once convergence has been obtained, the mode shape, velocities and displacements at $t+\Delta t$ are known, and the solution may proceed to the next time step with the current velocity and geometry being taken as initial conditions.

The solution proceeds until the structure comes to rest. On request, computer plots of the deformed shape of the structure at successive time intervals may be created (subroutine PICTUR).

In the next section, the program DAGNVS will be discussed.

## 5.2 Numerical Implementation of the Conventional Direct Solution Technique using DAGNVS

Like GNLIMST, DAGNVS is a FORTRAN program consisting of a number of subroutines contolled by a driver program. A macro flow chart of the main steps in the program is shown in Fig. 5.3 .

Once the data has been read (subroutine INPUT) and displayed for verification (subroutine DATA), a mode shape and time function are calculated in the same way as described in the previous section in order to obtain a rough estimate of the total time of deformation. A suitable time step may then be defined. The direct analysis procedure may now commence.

Moments and axial forces corresponding to the applied initial velocities are calculated as follows. Influence matrices [m] and [n] described in Section 3.2.1 are assembled. Each row of these matrices is the set of nodal moments and element axial forces, respectively, in the structure resulting from a unit load applied in turn at a node, in the global X and Y directions. If the structure is hyperstatic, degrees of freedom must be defined as input data which are to be released so that the structure becomes statically determinate. This static calculation for the determination of [m] and [n] is performed in subroutine STAT.

Trial values of the nodal forces which when applied as static loads to the structure result in the given velocity field are chosen, and using equns. (3.20) and (3.22), trial bending moments and axial forces may be calculated in terms of the nodal forces. Using the constitutive relations equn. (2.1), and noting that bending moment is assumed to vary linearly between nodes (equn. (3.21)) and that axial force is constant along an element, the curvature rate and axial strain rate for an element may be formulated in terms of the nodal forces. Since the constitutive relations equn. (2.1) are homogeneous in the nodal forces, the integral in equn. (3.23), a virtual velocities formulation, may be evaluated explicitly for each element. The contribution of each element is summed over the structure. This process is repeated for every component of the velocity field. Since for trial values of the nodal forces, the right hand side of equn. (3.23) will not in general equal the given velocity field, iteration is required to obtain a solution. The full Newton-Raphson procedure is used. The right hand side of equn. (3.23) consists of a set of nonlinear equations in the nodal forces. A matrix of partial derivatives of the compatibility equn. (3.23) with respect to the nodal forces is formed. Again, due to the homogeneity of the constitutive relations, these may be formulated explicitly. Integration is carried out over an element, and the contribution from each element is summed over the structure. The formulation of the compatibility equns. (3.23) and their partial derivatives (equns. (3.26)) is performed in subroutine COMDIF.

Using equn. (3.27) an improved estimate of the nodal forces is found. This requires the solution of a set of linear equations (subroutine PIVOT). The value of the nodal forces is revised and convergence is checked. Convergence is assumed when the velocities given by the compatibility equations evaluated in subroutine COMDIF equal the given velocities to acceptable accuracy, or the change in the nodal forces is within a predefined tolerance (0.5%). If convergence has occurred, then the bending moment and axial force diagrams are evaluated using equns. (3.20) and (3.22) respectively. If not, the revised values of the nodal forces are used in equns. (3.20) and equns. (3.22) to form a new trial bending moment and axial force diagram, and hence a revised estimate of the right hand side of the compatibility equns. (3.23) and the partial derivatives equn. (3.26).

Matching the homogeneous viscous relation to the rigid-visco-plastic model is now performed. Matching may be performed either on intercept alone, at the slope and intercept or at intercept but with $n' > n$, all at the maximum initial curvature rate, as described in Chapter 4. If either of the latter two matching schemes is used, iteration is required to obtain final matching factors, as the magnitude of the maximum curvature rate changes as the value of $n'$ is revised. The calculations for the nodal forces must thus be repeated if $n'$ is changed, which results in revised bending moments and axial forces and hence revised matching factors. This iteration procedure is repeated until acceptable convergence in the matching factor(s) is obtained. The matching procedure is performed in subroutine VMATCH.

The implicit time integration scheme may now commence. Accelerations are calculated from the equations of motion (equn. (3.28b)) in subroutine ACCAXC. As a first estimate of the nodal forces, velocities and displacements at subsequent time $t+\Delta t$, equns. (3.36), (3.32c) and (3.37) are evaluated, respectively. Note that the matrix of partial derivatives required in equn. (3.36) is given by the last evaluation of equn. (3.26) in the iterative procedure to determine the nodal forces. Equation (3.36) is formulated in subroutine IMPLIC, and consists of a system of linear equations which are solved for the change in nodal forces using subroutine PIVOT.

The change in velocity using equn. (3.32c), and the displacements from equn. (3.37) are calculated in subroutine REVISE, and the co-ordinates of the nodes, and hence the structural geometry is updated.

Iteration is required to refine the estimate of quantities at $t+\Delta t$ if equilibrium is to be maintained. The influence matrices (subroutine STAT) are revised due to the change in geometry, and new estimates of the compatibility equns. (3.23) and partial derivatives (equn. (3.26)) are calculated with the revised nodal force values (subroutine COMDIF). Equations (3.45) are formulated in subroutine IMPLIC. This system of linear equations is solved for the out of balance nodal forces by subroutine PIVOT, and hence, using equn. (3.43) residual velocities may be calculated. With this improved estimate of velocity, displacements at $(t+\Delta t)$ may be revised (subroutine REVISE). This iteration procedure continues until the residual nodal forces and velocities are acceptably small. The solution quantities (nodal forces and velocities) are thus found at $t+\Delta t$, and the solution proceeds to the next time step, until the structure comes to rest.

As in program GNLIMST, computer plots of the deformed shape of the structure at successive time intervals may be requested (subroutine PICTUR).

In Appendix A a user manual for both GNLIMST and DAGNVS is given, followed by a program listings of the two codes in Appendix B and Appendix C respectively.

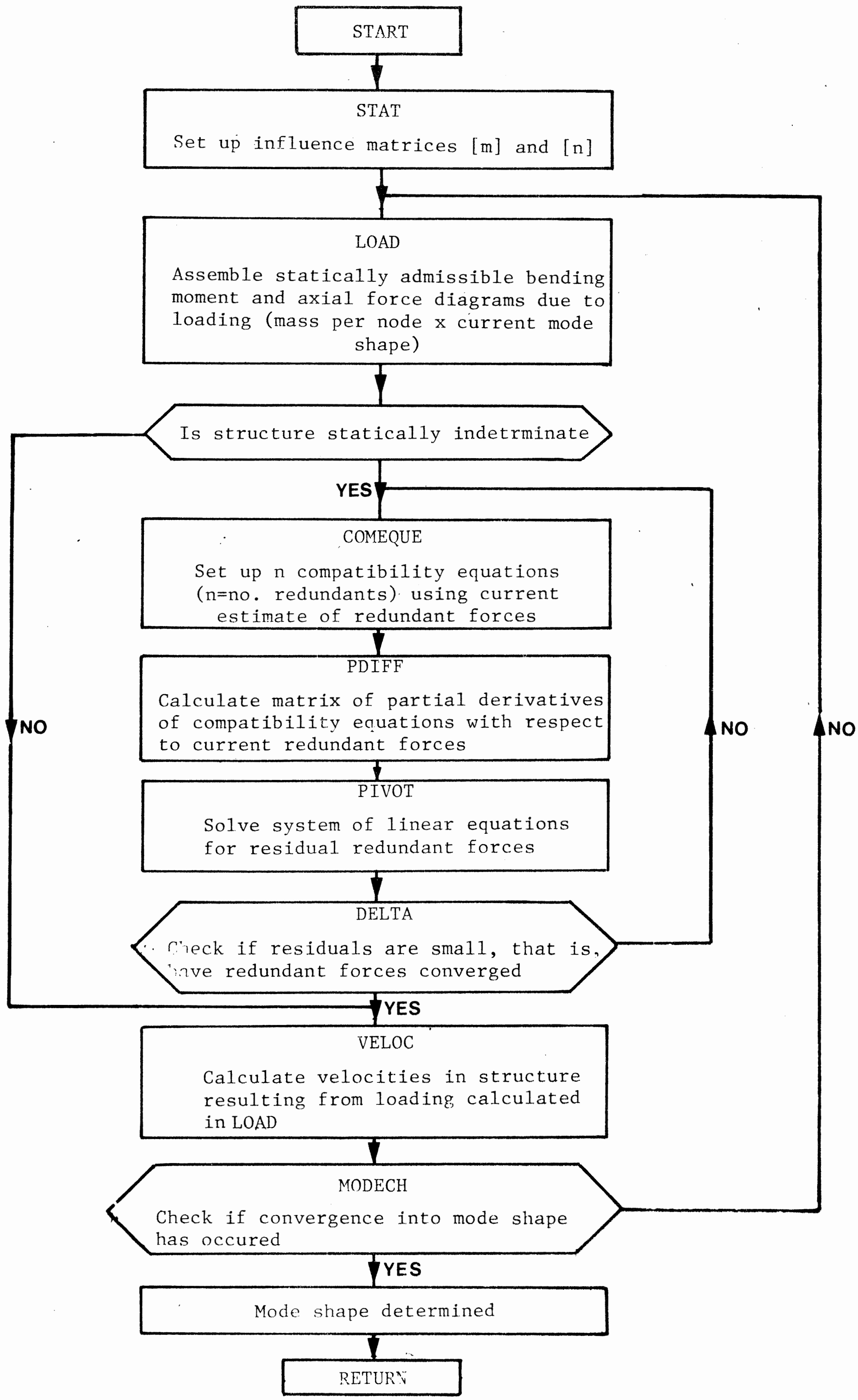

START
STAT
Set up influence matrices [m] and [n]
LOAD
Assemble statically admissible bending moment and axial force diagrams due to loading (mass per node x current mode shape)
Is structure statically indetrminate
YES
COMEQUE
Set up n compatibility equations (n=no. redundants) using current estimate of redundant forces
PDIFF
Calculate matrix of partial derivatives of compatibility equations with respect to current redundant forces
PIVOT
Solve system of linear equations for residual redundant forces
DELTA
Check if residuals are small, that is, have redundant forces converged
NO
NO
NO
YES
VELOC
Calculate velocities in structure resulting from loading calculated in LOAD
MODECH
Check if convergence into mode shape has occured
YES
Mode shape determined
RETURN

Figure 5.1

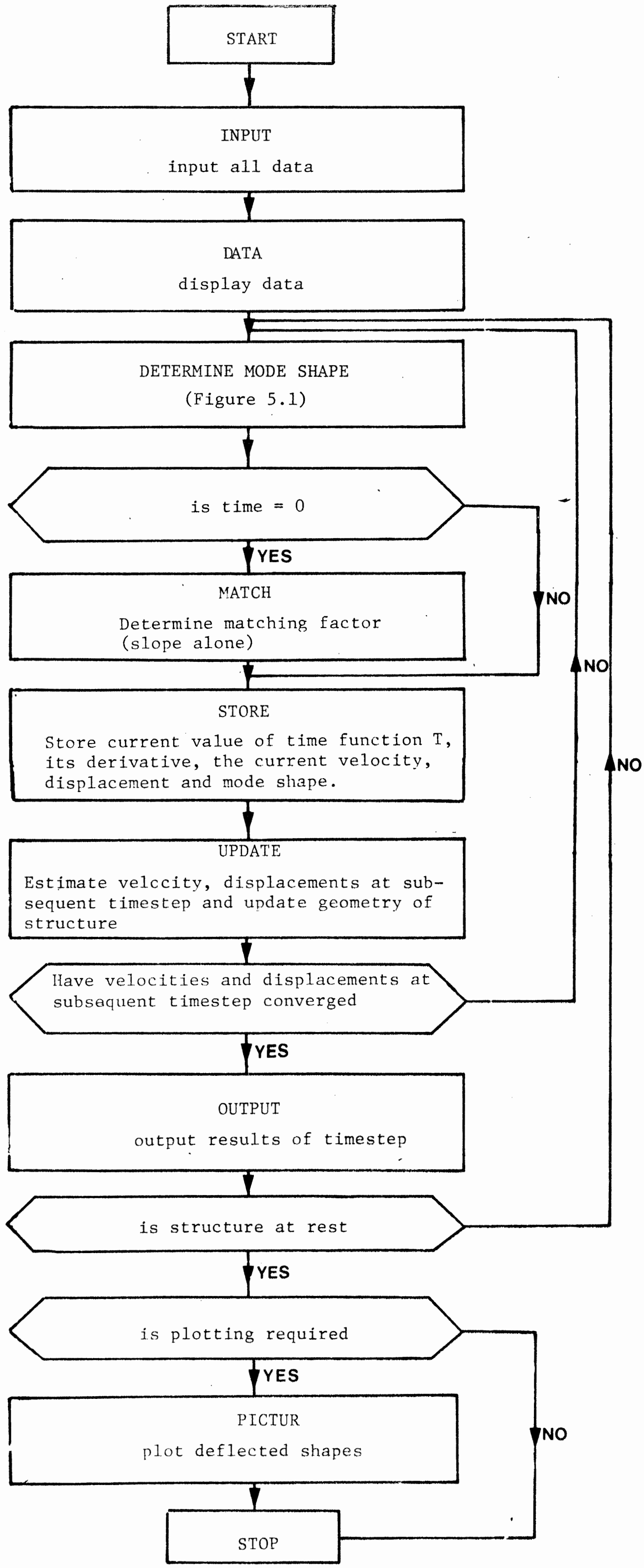

START
INPUT
input all data
DATA
display data
DETERMINE MODE SHAPE
(Figure 5.1)
is time = 0
YES
NO
MATCH
Determine matching factor
(slope alone)
NO
STORE
Store current value of time function T, its derivative, the current velocity, displacement and mode shape.
NO
UPDATE
Estimate velccity, displacements at subsequent timestep and update geometry of structure
Have velocities and displacements at subseaquent timestep converged
YES
OUTPUT
output results of timestep
is structure at rest
YES
is plotting required
YES
PICTUR
plot deflected shapes
NO
STOP

Figure 5.2

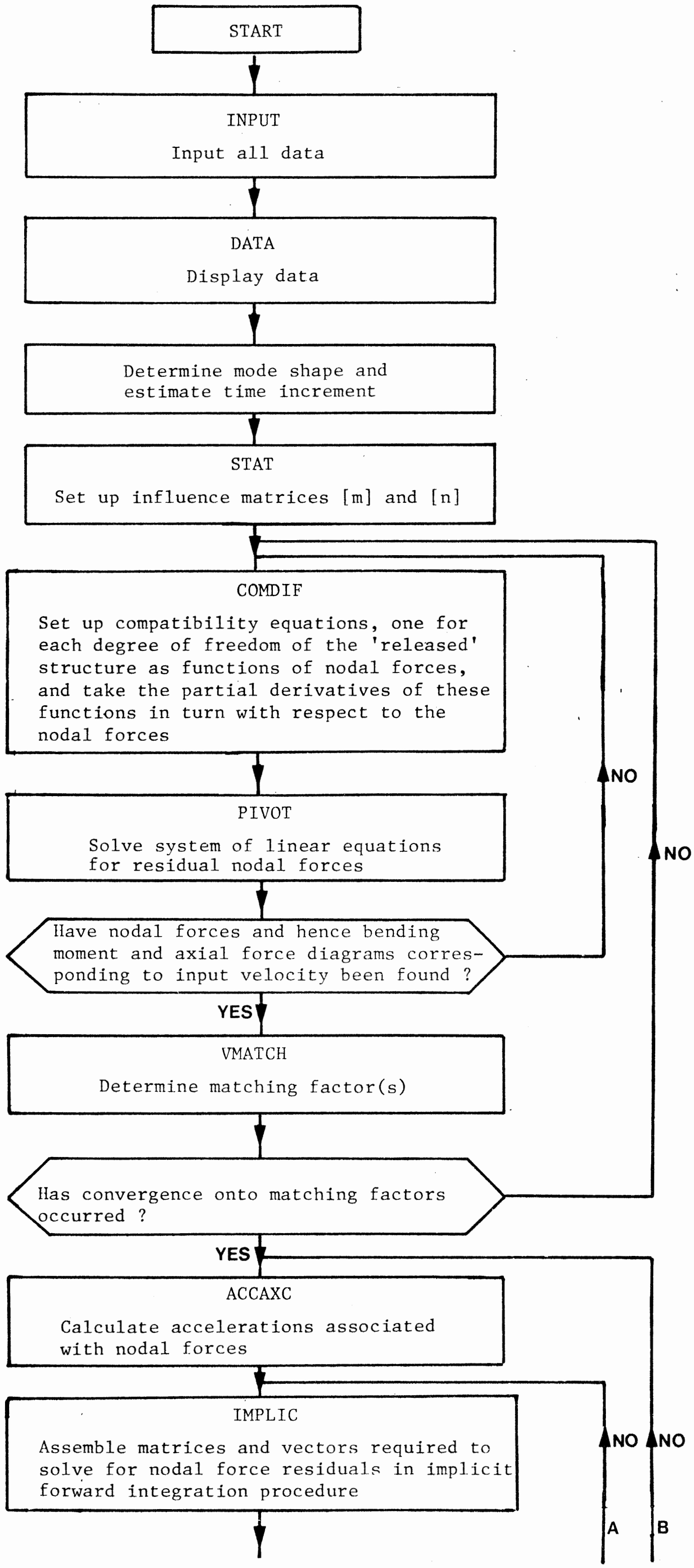



Figure 5.3 (continued overleaf)

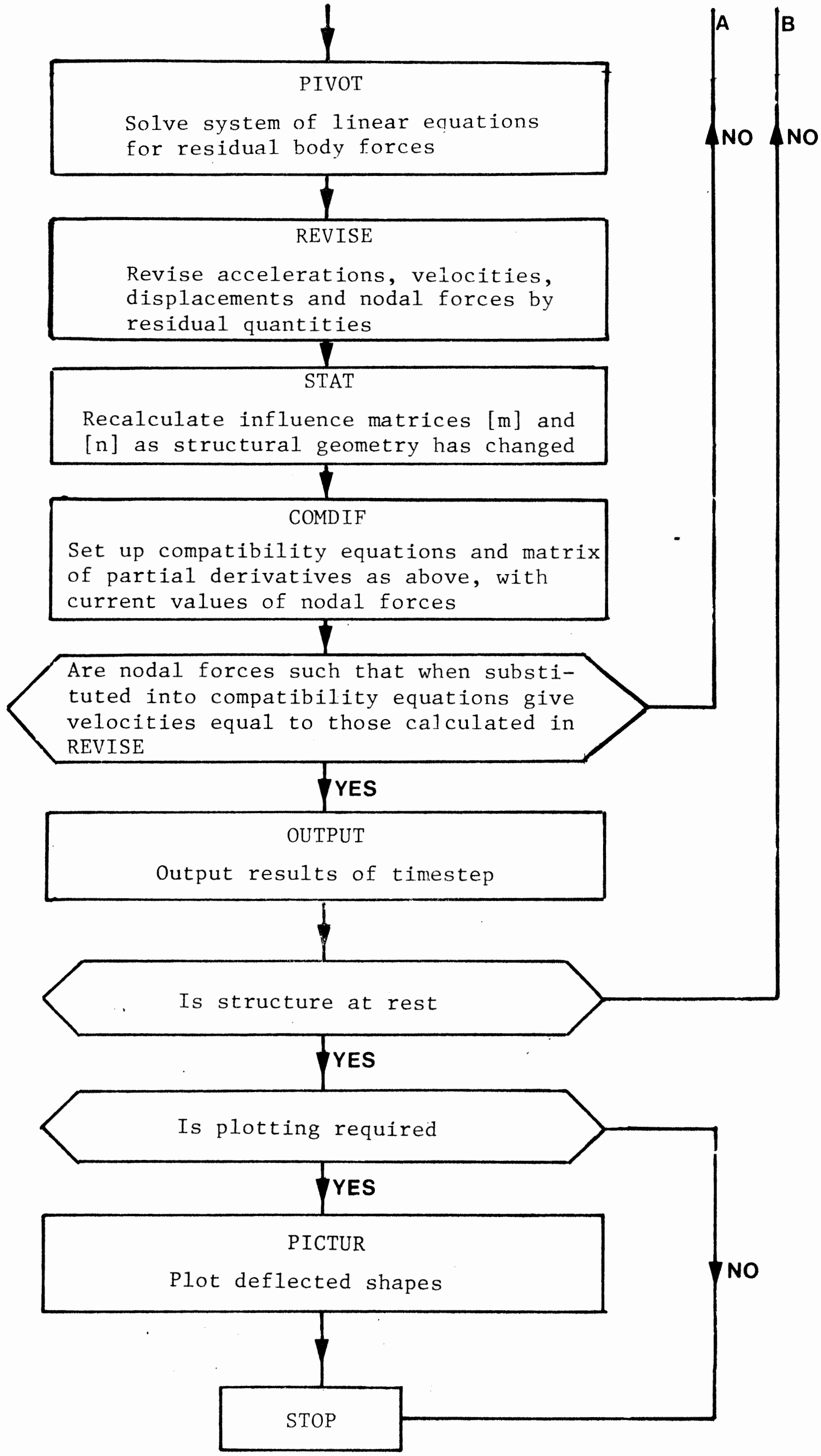



Figure 5.3 (continued)

# CHAPTER 6

## ILLUSTRATIVE EXAMPLES

The sequence of development of the two programs outlined in the previous chapter was in response to the need for the analysis of structures which underwent increasingly sophisticated modes of deformation. The first development was a program which incorporated the mode solution technique outlined in Chapter 2, and the direct method of analysis based on the Tamuzh Principle, given in Section 3.1.1 . The program was limited in application to beam structures which underwent small displacements, its primary aim being to test the mode algorithm and the homogeneous viscous laws presented in this thesis. In order to analyse beam *and* frame structures and incorporate geometric effects, GNLIMST was developed, and was successfully used to analyse a variety of beam and frame structures. For certain classes of structural problems, however, poor results were obtained using GNLIMST, which emphasised the need for a more general method of analysis, which is given in Section 3.2, and implemented by the program DAGNVS.

To illustrate the application of the two programs, the results of analyses of five types of structures are presented. They are:

(a) a cantilever struck transversely at its tip ,

(b) steel and aluminum rectangular, fixed end, portal frames subjected to a uniform transverse impulse applied to the full length of the beam,

(c) a fixed steel beam, with the ends clamped against longitudinal displacement, subjected to a uniform transverse impulse along the length of the beam,

(d) steel and aluminum rectangular, fixed end, portal frames subjected to a uniform sideways impulse applied along the length of one column, and

(e) aluminum rectangular, fixed end, portal frames subjected to a uniform transverse impulse applied to half the length of the beam.

The problem of the cantilever beam struck transversely at its tip has received considerable attention both experimentally and analytically (see, for example, Bodner and Symonds [10], Ting [7], Lee and Martin [8]). A particular beam, E4, from the tests by Bodner and Symonds [10] was analysed using the mode approximation technique with small displacement assumptions, the instantaneous mode solution technique, and the conventional direct method of analysis outlined in Section 3.2, hereafter referred to as the direct method of analysis. The experiments by Bodner and Symonds [10] gave a tip rotation of 53°, and they estimated the total time of deformation $t_f$ to be 0.052s. Deflections were not presented in their results. Since rotations are not calculated in all the techniques outlined in this thesis, the only parameter which may be used to compare directly the results obtained here to test data and most previous analytical solutions is $t_f$. Nevertheless, the cantilever is an important standard problem, and indicates the capabilities of the various analytical methods which have been presented in this thesis. From a small displacement rigid-plastic analysis which included strain rate sensitivity Bodner and Symonds [10] estimated $t_f$ to be 0.064s. Ting [7], using a rigid-viscoplastic material model with small displacement assumptions calculated $t_f$ to be 0.065s. Lee and Martin [8], using a rigid-viscoplastic material model and the 'piecewise stationary mode' technique, and Symonds [9] using the mode approximation technique and a matched viscous constitutive relation, estimated $t_f$ to be 0.064s and 0.066s respectively. Both analyses neglected geometric effects. In the latter analysis Symonds estimated the transverse tip displacement to be 0.348m. Here, the results of three methods of analysis are presented. Firstly, a small displacement analysis using the mode solution technique combined with a direct method of analysis based on the Tamuzh principle, as outlined in Section 3.1.2, was performed. Matching was not performed on the bases outlined in Chapter 4, but on the total estimated time of deformation $t_f$ given by Symonds [9], so that a comparison could be made with his deflection result. For $n = n' = 5$ the maximum transverse tip displacement was 0.347m, and with $n' = 14{,}065$ which Symonds suggested, the displacement was 0.348m. The instantaneous mode solution technique, a large displacement analysis, gave the tip deflection as 0.320m and the total time of deformation as 0.065s. Using the direct method of analysis, the transverse displacement at the tip was 0.330m, and $t_f$ was 0.065s. Different methods of matching the homogeneous viscous constitutive material model used here to the rigid-viscoplastic relation were employed for the latter two analyses. In the instantaneous mode analysis, matching was performed on intercept alone at the maximum initial mode curvature rate, so that $n = n' = 5$, which gave a stress factor $\mu = 3{,}25$. In the direct method of analysis, matching was performed both on shape and intercept at the maximum initial curvature rate obtained by the direct analysis, which resulted in the power factor $\nu = 2{,}21$ ($n' = 11.1$), and a stress matching factor $\mu = 1.99$. These values compare reasonably with the matching factors obtained by Symonds [9] of $\nu = 2{,}813$ and $\mu = 1{,}917$

for his matched mode approximation analysis. Plots of the displaced shape at successive time intervals for both the small and large displacement mode analyses are given in Fig. 6.1, together with the physical description of the cantilever. In Fig. 6.2, a similar plot is given of the results obtained using the direct method of analysis. The deformed shape at successive time intervals compares excellently with the sequence of photographs of the deforming cantilever given by Bodner and Symonds [10].

Fig. 6.3 compares the results obtained using the instantaneous mode solution technique to the tests on steel frames by Bodner and Symonds [37] and to the theoretical analyses by Symonds and Raphanel [16]. The latter considered three models; a rigid-perfectly plastic approximation, a rigid-perfectly plastic model with strain rate correction, and a rigid-perfectly plastic model which included both strain rate and elastic effects. The results obtained here agree well with the experimental data and with Symonds and Raphanel's strain rate sensitive rigid-plastic model. For all the analyses presented here, matching was performed on intercept alone, with the stress matching factor ranging from $\mu = 2{,}6$ for the smallest impulse to $\mu = 2{,}4$ for the largest impulse. A plot of the displaced shape of one of the above frames at successive time intervals is given in Fig. 6.4.

In Fig. 6.5, the results of analyses of aluminum frames by the instantaneous mode solution technique are given, and compared to the experimental and theoretical results of Hashmi and Al-Hassani [34], and the analyses of Symonds and Raphanel [16], outlined above. Hashmi and Al-Hassani employed a rigid-perfectly plastic model and included geometric effects. Symonds and Raphanel treated the aluminum frames as rate insensitive. Here, $\dot{\varepsilon}_o$ was set at 6500/sec. The results obtained agree excellently with experimental observations, and with the rigid-perfectly plastic analysis of Symonds and Raphanel [16]. As for the previous steel frames, matching was performed on intercept alone, with the stress matching factor ranging from $\mu$ = 7,6 for the smallest impulse to $\mu$ = 6,25 for the largest impulse considered.

The analysis of a fixed end beam subjected to uniform transverse impulse illustrates a problem where deformation proceeds from purely flexural to predominantly axial for sufficiently large initial impulse. As shown in Fig. 6.6, the results obtained using the instantaneous mode technique agree very well with the experimental work by Symonds and Jones [38] and with the analyses by Symonds [39] who used the mode approximation technique with large displacements and elastic effects included. In these analyses matching was initially performed at intercept alone at the maximum initial curvature rate, giving $\mu$ = 1,77 for the smallest impulse and $\mu$ = 1,62 for the largest impulse. Rematching was performed when axial effects became significant, which was adjudged to occur when the normalized axial force $N/N_o'$ exceeded 0,1. The new matching factors, calculated on intercept alone at the maximum axial strain rate at that instant, were found to range between 3,7 and 4,0 for the lowest and highest impulse respectively.

Fig. 6.7 shows a comparison between the results obtained using the direct method of analysis and the test results by Wegener [40] for rectangular steel portal frames subjected to uniform sideways impulse along the length of one column. Matching was performed at the maximum initial curvature rate on intercept alone, with $n = n' = 5$. The stress factor $\mu$ was found torange between 2,20 for the highest impulse to 2,37 for the lowest impulse considered. As shown in Fig. 6.7, the analyses agree excellently with experimental values. In Fig. 6.8 the deformed shape of a typical frame is shown, together with the original and deformed nodal positions, and a physical description of the frame.

A similar series of analyses was performed on rectangular aluminum frames, and the results were compared with the test data obtained by Hashmi and Al-Hassani [34]. The frame is shown in Fig. 6.9, with its geometric and material properties. Curve 1 in Fig. 6.10 shows results of the tests performed by Hashmi and Al-Hassani. The results of analyses using the direct method with a homogeneous viscous relation, matched on intercept alone at the maximum initial curvature rate, with $n = n' = 4$, is given by Curve 3, Fig. 6.10. The discrepancy between the two curves can be ascribed to the crudity of the matching procedure used. Considerable numerical difficulties were encountered, however, when analyses using a constitutive relation matched on intercept *and* slope were attempted. Such a matching procedure required that $n'$ exceed 12. For $n'$ greater than 8, the initial moments and axial forces required by the direct analysis procedure could not be obtained using the numerical procedures outlined in Section 3.2.1. Results were obtained by using a value of $n'$ as high as would permit a solution; this was found for these examples to lie in the range between 6,0 and 8,0. These results are shown by Curve 2 in Fig. 6.10. In Fig. 6.11, computer plots of the displaced shape at successive time intervals for a typical side loaded aluminum frame are given.

Similar numerical difficulties were encountered in the direct analyses of aluminum rectangular frames of the type shown in Fig. 6.9, subjected to a uniform impulse over half the beam length. In Fig. 6.12, the test data of Hashmi and Al-Hassani [34] is shown by Curve 1. Curve 2 shows the results obtained by the present analysis when matching was performed on intercept alone at the maximum curvature rate, with $n = n' = 4$. Much better correlation with experimental results was obtained when a higher $n'$ was used. In this series of analyses, solutions were achieved for $n'$ between 8 and 10, and are shown by Curve 3 in Fig. 6.12. Plots of the displaced shape at successive time intervals for the analysis of a typical frame in this series of analyses is shown in Fig. 6.13.

In the last three sets of examples, that is the steel and aluminum frames subjected to sideways impulse, and the aluminum frames subjected to impulse over half the beam it was noted that the direct method of analysis was used. It was found in these examples that the true behaviour of the structure either converged very slowly onto the mode shape predicted by the mode solution technique, or did not converge at all. A deformation pattern predicted by the mode analysis technique would thus be significantly different from the true structural behaviour. As can be seen from the results shown in Figs. 6.7, 6.10 and 6.12, when suitable matching coefficients are chosen the direct analysis technique presented here predicts deformations which agree very well with the true deformations of the structure.

All analyses were performed on the University of Cape Town UNIVAC 1100/81 computer. As indicated by the C.P.U. times given in Fig. 6.1, Fig. 6.2 and Fig. 6.4, the instantaneous mode solution technique as implemented here is a computationally efficient analytical scheme, as well as a method which provides reliable solutions to certain classes of rigid-viscoplastic dynamic problems. Analyses using the direct method of analysis are more costly, as would be expected. For the analysis of the cantilever beam with full matching scheme, shown in Fig. 6.2. the C.P.U. time was 2 min 40 sec. For the analysis of the aluminum frames subjected to impulse over half the length of the beam which were discussed above, the C.P.U. time ranged between 10 min 54 sec for the lowest impulse (n' = 10) and 32 min 13 sec for the highest impulse (n'= 8). The C.P.U. time increases substantially with either an increase in the number of time steps, or with increasing n'.

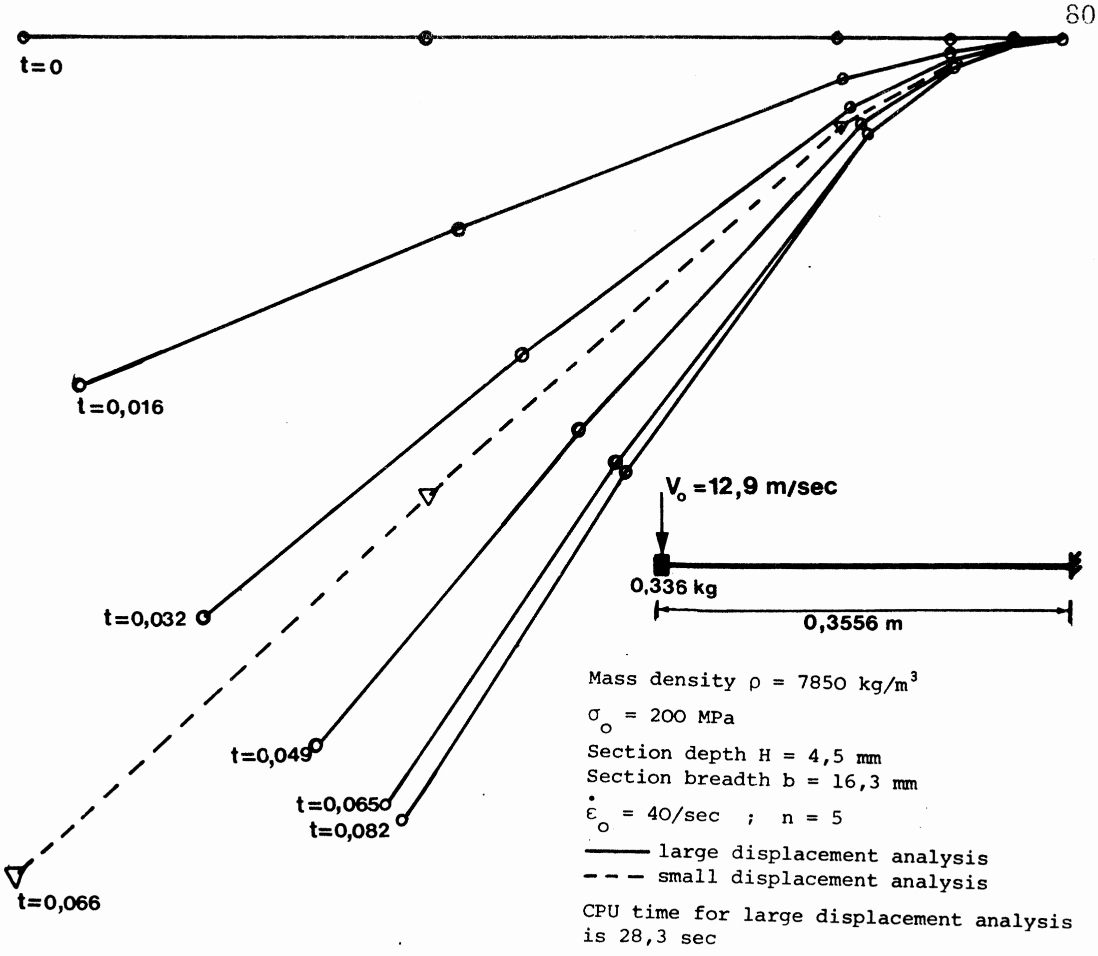



Figure 6.1 Displaced shape of cantilever beam as successive time intervals for large displacement analysis, and final displaced shape for small displacement analysis, using the mode solution technique

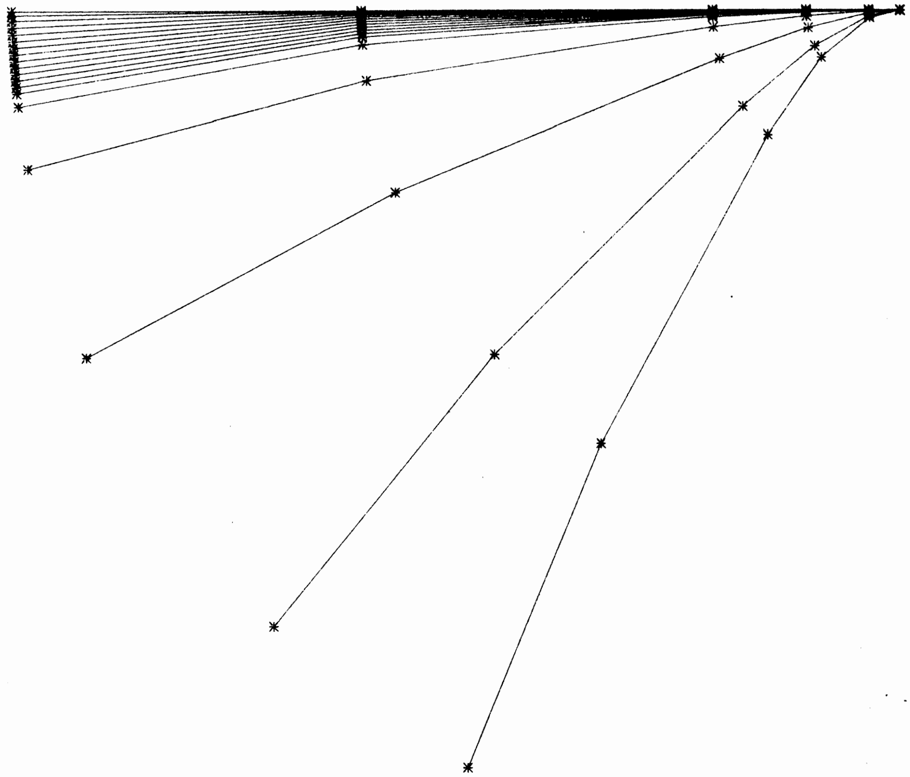
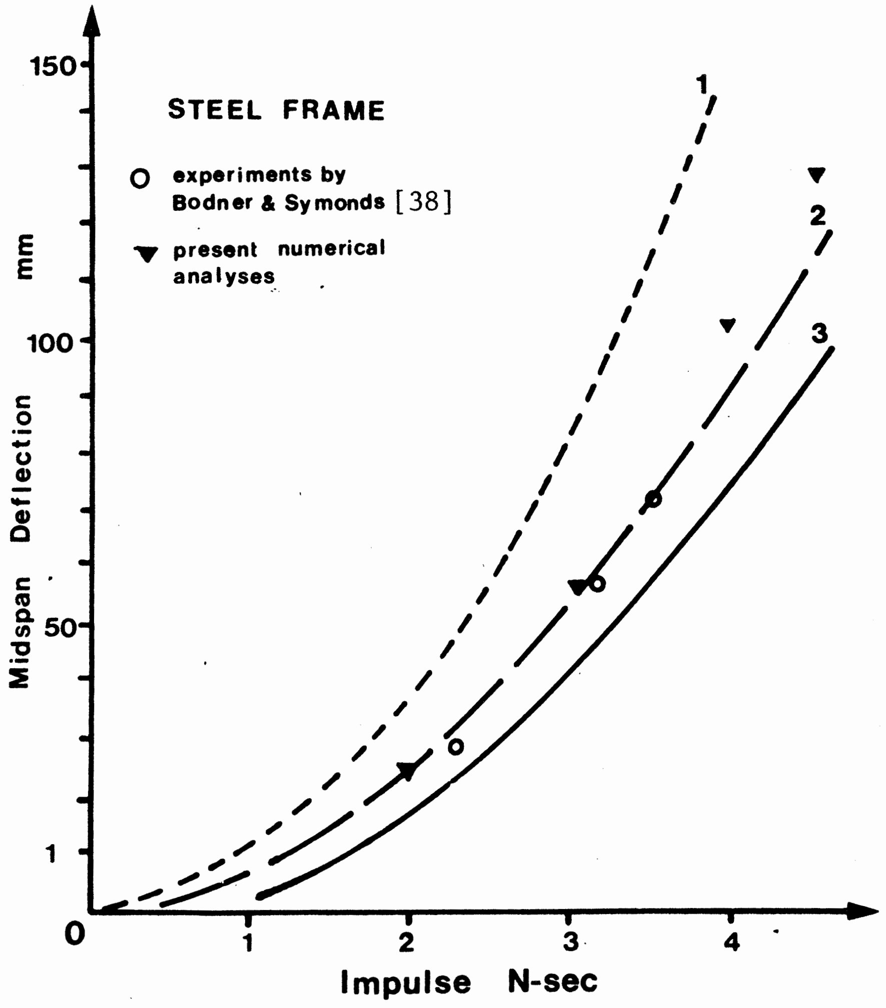
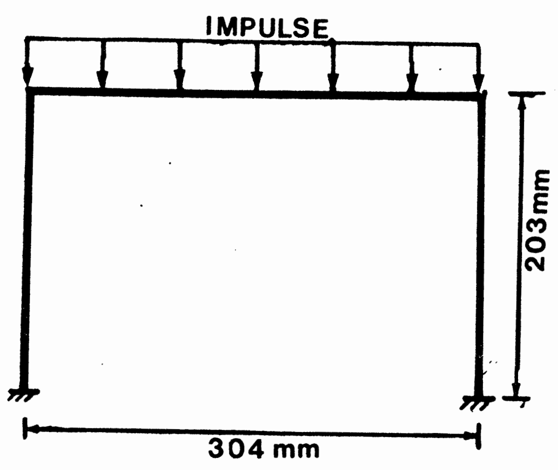


Figure 6.2 Displaced shape of cantilever beam at successive time intervals using the direct method of analysis. C.P.U. time is 2 min 40 sec.




**1** Rigid - perfectly plastic analysis

**2** Rigid - perfectly plastic analysis with strain rate correction

**3** Rigid - perfectly plastic analysis with strain rate and elastic corrections

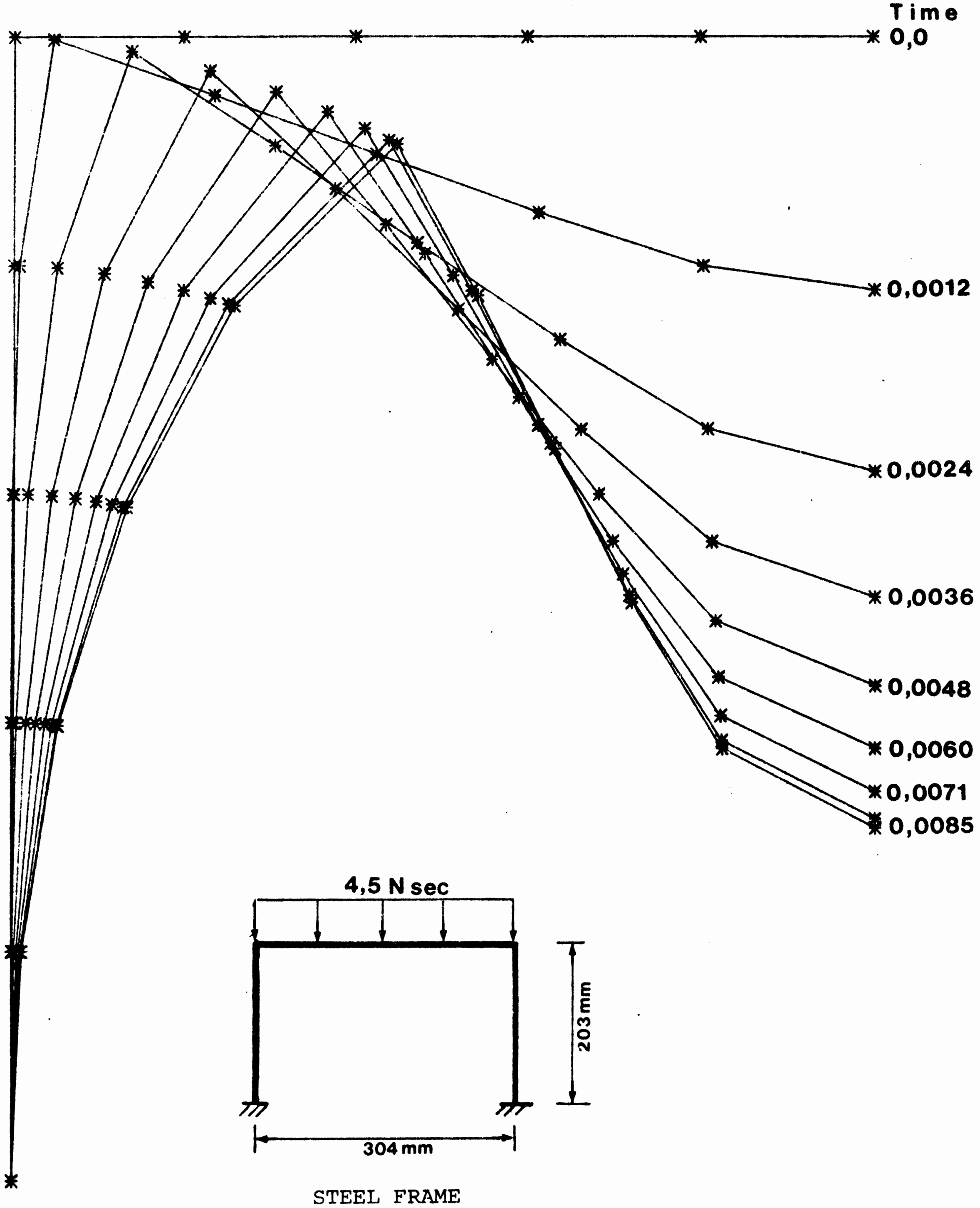
All analyses by Symonds and Raphanel [16]

$H = 3.1$ mm ; $b = 19{,}1$ mm

$\sigma_o = 228$ MPa ; $\dot{\varepsilon}_o = 40/\text{sec}$

$n = 5$ ; $\rho = 7850\ \text{kg/m}^3$

Figure 6.3 Plot of midspan transverse deflection vs uniform impulse along the beam for rectangular steel frame.




STEEL FRAME

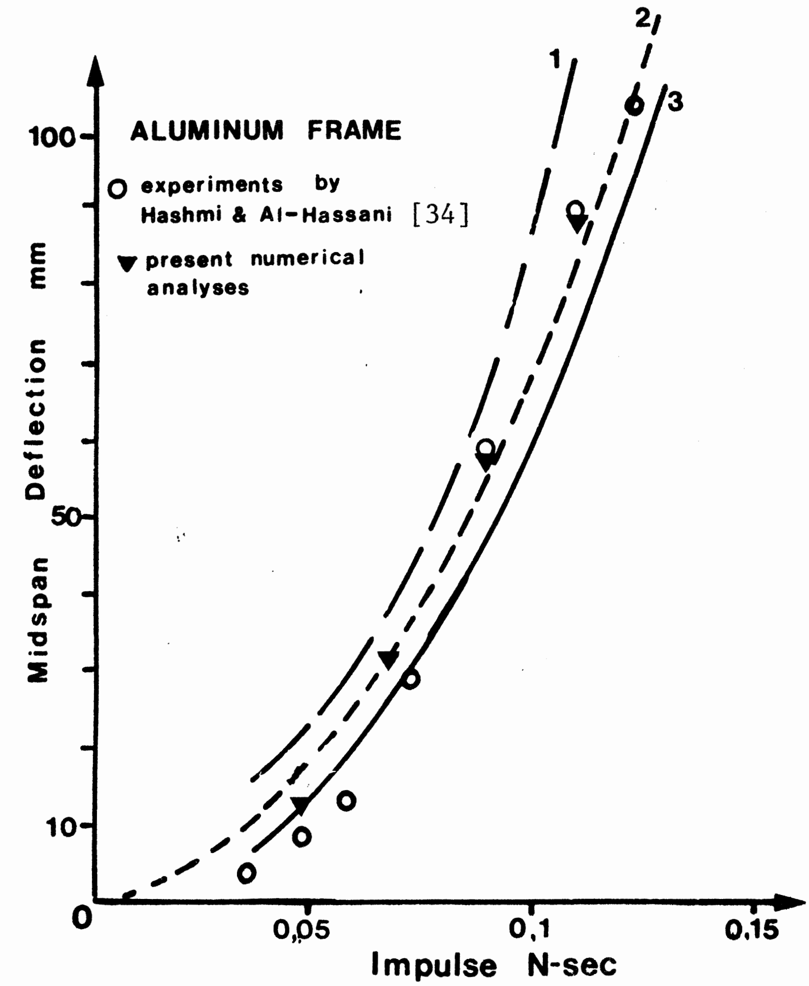
$H = 3{,}1$ mm $\quad b = 19{,}1$ mm

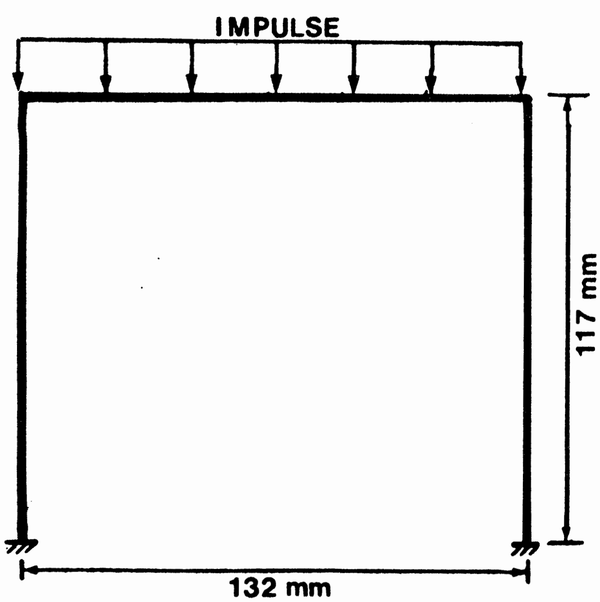
$\sigma_o = 228$ MPa $\quad \dot{\varepsilon}_o = 40$/sec

$n = 5 \quad \rho = 7850$ kg/m$^3$

CPU time is 1 min 47 sec

Figure 6.4 Displaced shape of half rectangular frame at successive time intervals subjected to uniform along the beam.




1 Rigid - perfectly plastic analysis by Hashmi & Al-Hassani [34]

2 Rigid - perfectly plastic analysis and,

3 Rigid - perfectly plastic analysis with strain rate and elastic corrections by Symonds and Raphanel [16]

$H = 0.91$ mm ; $b = 12{,}7$ mm

$\sigma_o = 83$ MPa ; $\dot{\varepsilon}_o = 6500$/sec

$n = 4$ ; $\rho = 2670$ kg/m$^3$

Figure 6.5 Plot of midspan transverse deflection vs uniform impulse along the beam for rectangular aluminum frame.

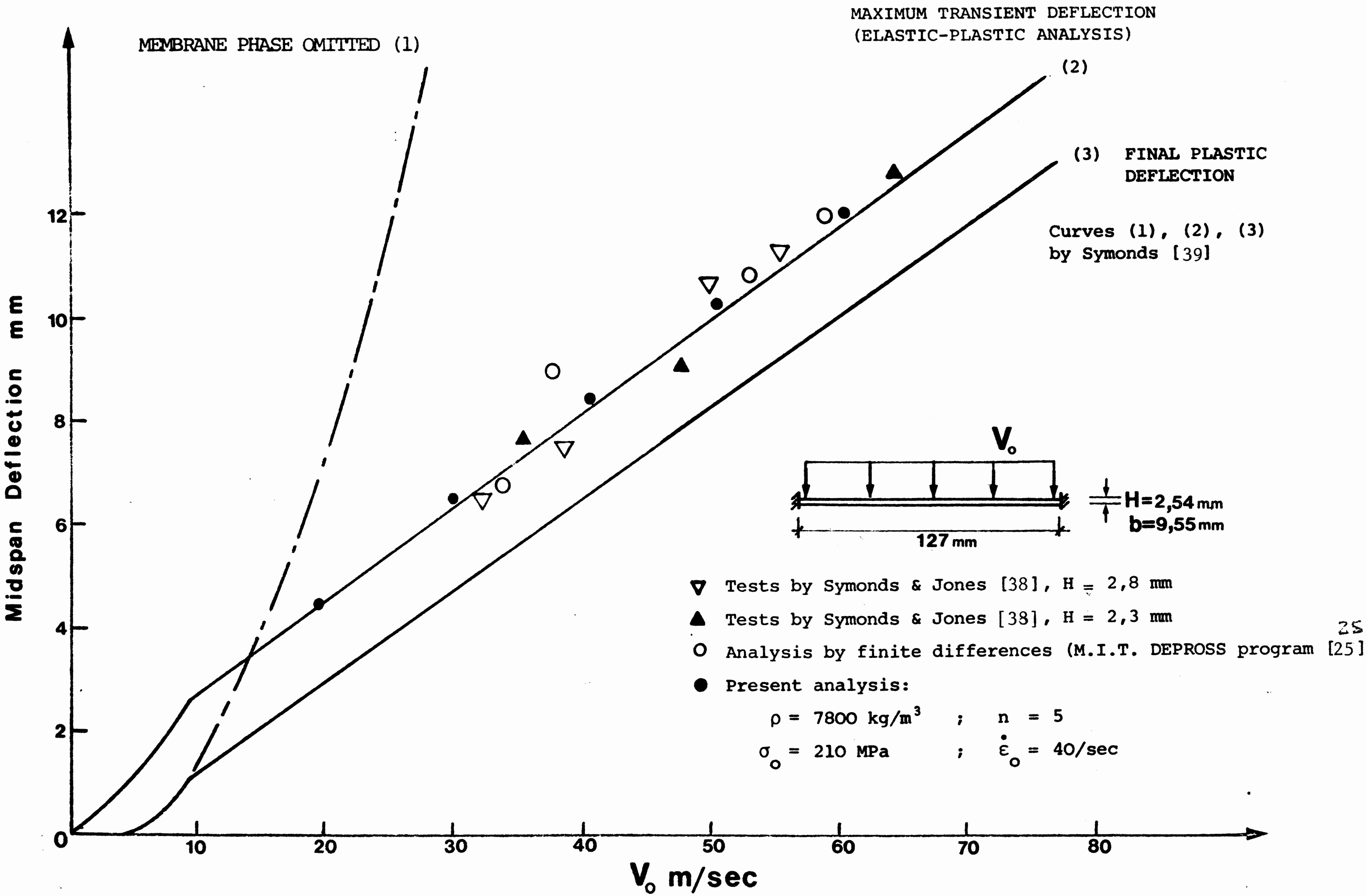



Figure 6.6 Plot of midspan transverse deflection vs uniform initial velocity for fixed end beam.

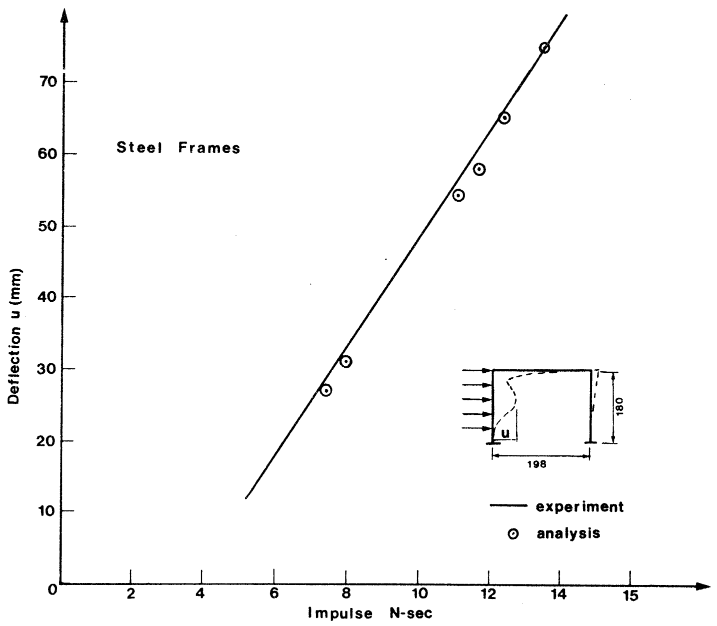



Figure 6.7 Plot of deflection versus impulse for rectangular steel frames subjected to uniform sideways impulse.

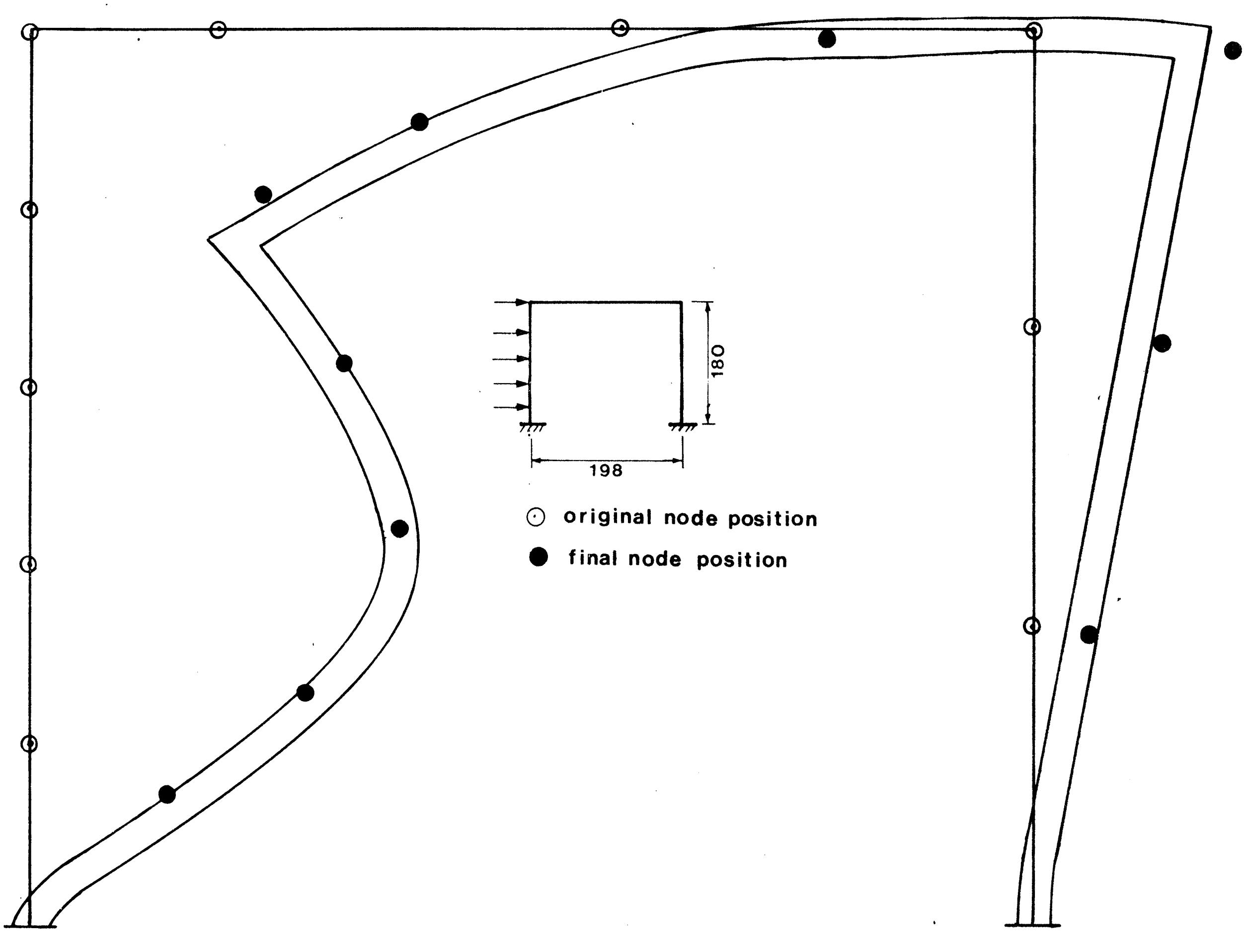



Figure 6.8 Deformed shape of a rectangular steel portal frame subjected to a uniform sideways impulse of 13.47 N-sec

mass density $\rho = 7859\ kg/m^3$ ; $\sigma_o = 350$ MPa

section depth H = 6 mm ; section breadth b = 20 mm

$\dot{\varepsilon}_o = 40/sec$ ; $n = n' = 5$ ; $\mu = 2,23$

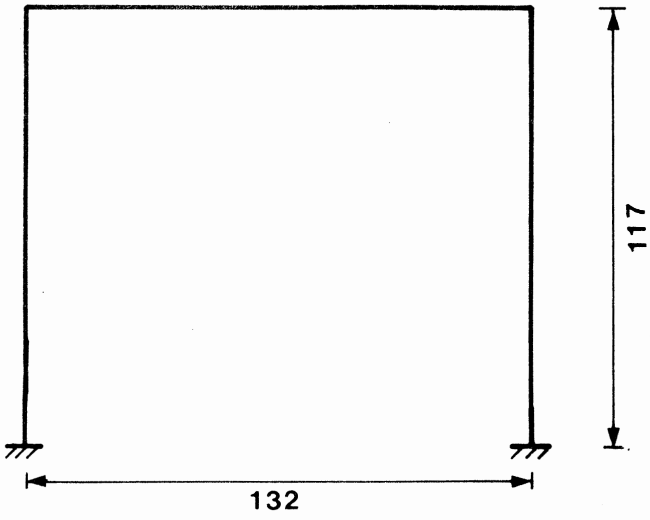



Figure 6.9 Aluminum rectangular portal frame.

mass density $\rho$ = 2670 kg/m ; $\sigma_o$ = 83 MPa

section depth H = 0,91 mm ; section breadth b = 12,7 mm

$\dot{\varepsilon}_o$ = 6500/sec ; n = 4

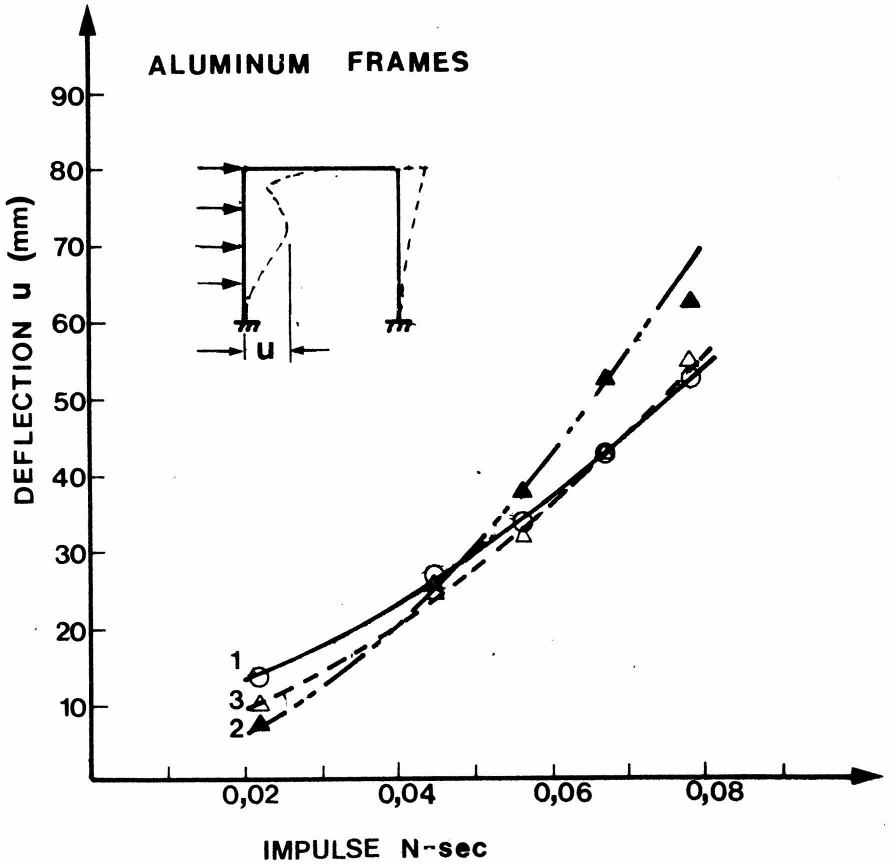



Figure 6.10 Plot of deflection vs impulse for rectangular aluminum portal frames subjected to a uniform sideways impulse.

Curve 1 : Test results by Hashmi and Al-Hassani [34].

Curve 2 : Analyses using homogeneous viscous relation matched on intercept alone with n = n' = 4.

Curve 3 : Analyses using homogeneous viscous relation matched on intercept alone but with n' between 6 and 8.

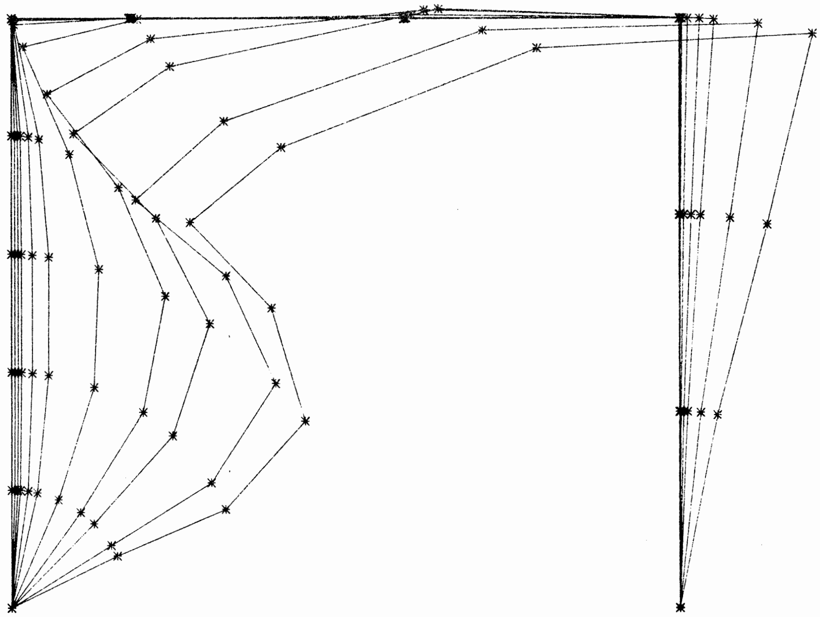


Figure 6.11 Deformed shape of rectangular aluminum portal frame subjected to a uniform sideways impulse of 0,0757 N-sec, at successive time intervals.

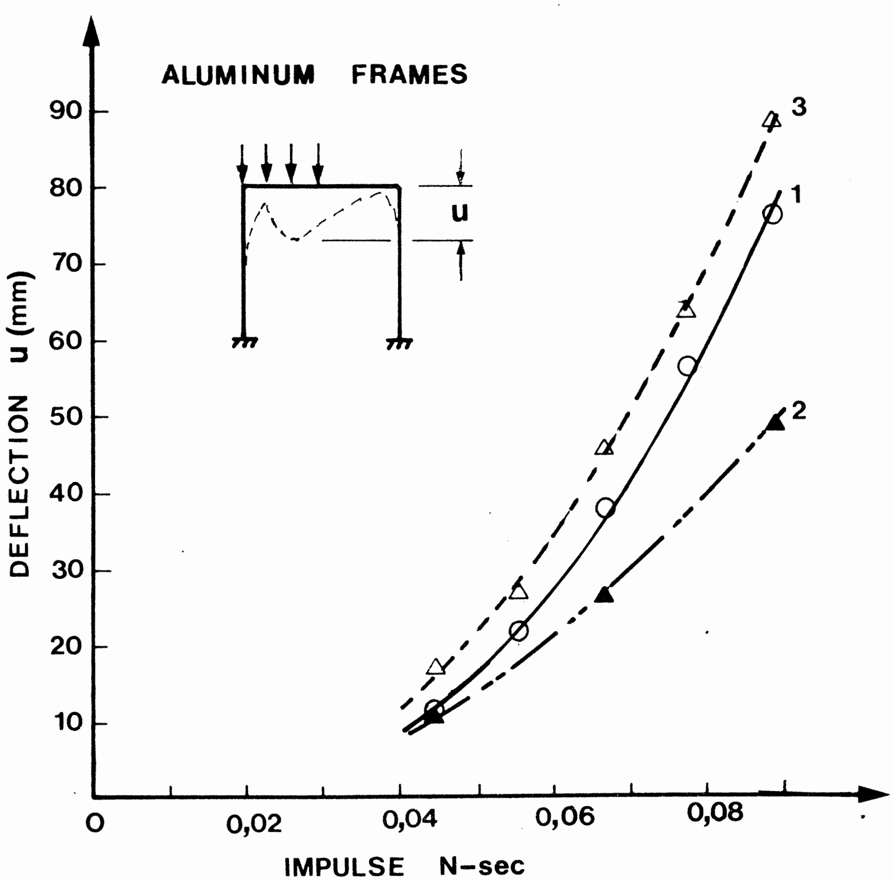



Figure 6.12 Plot of deflection vs impulse for rectangular aluminum portal frames subjected to a uniform impulse over half the length of the beam.

Curve 1 : Test results by Hashmi and Al-Hassani [34].

Curve 2 : Analyses using homogeneous viscous relation matched on intercept alone with n = n' = 4.

Curve 3 : Analyses using homogeneous viscous relation matched on intercept alone but with n' between 8 and 10.

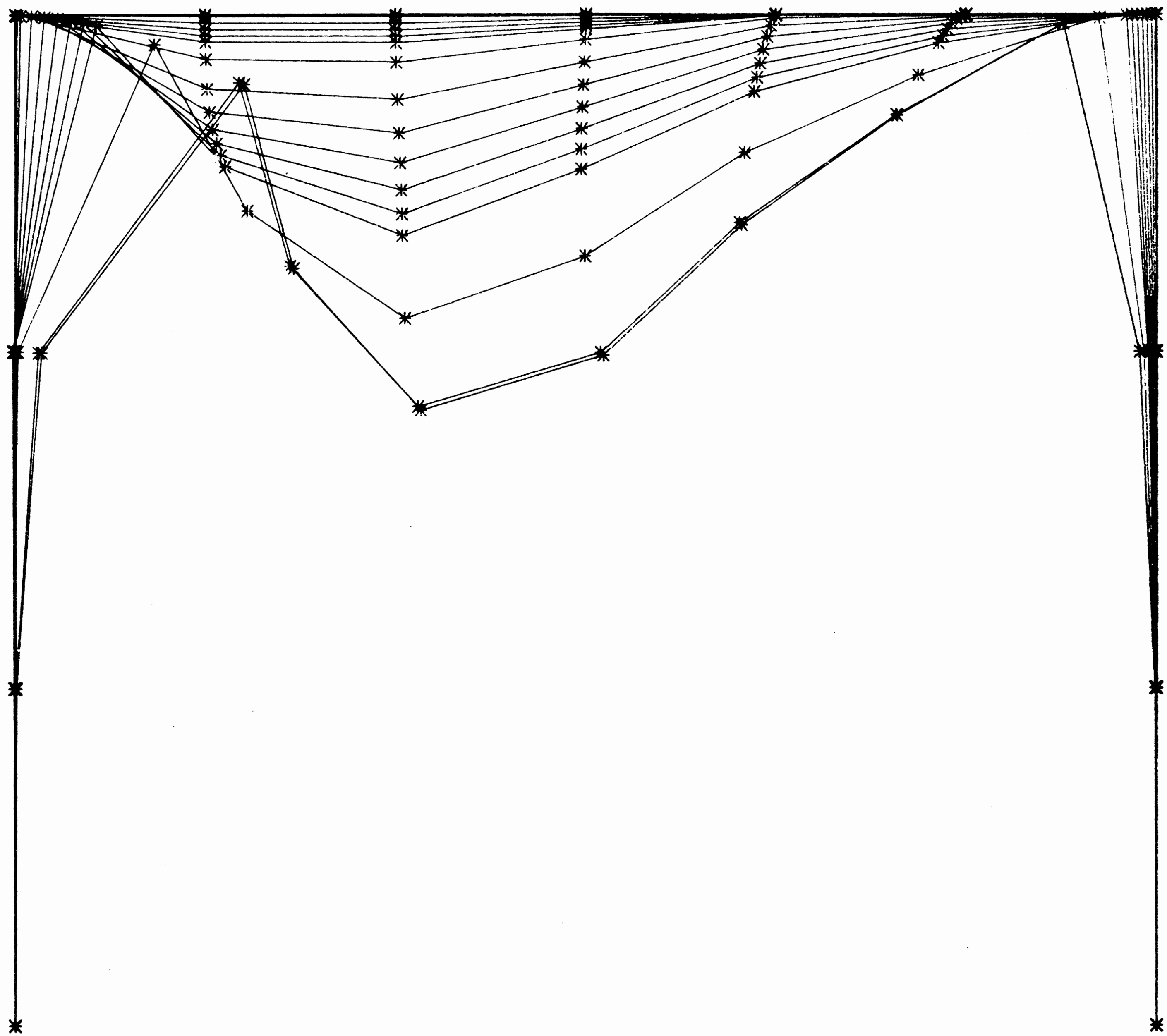


Figure 6.13 Deformed shape of a rectangular aluminum portal frame subjected to a uniform impulse of 0,066 N-sec along half the length of the beam, at successive time intervals.

# CHAPTER 7

# CONCLUSION

The numerical procedures outlined in this thesis provide computationally efficient and reasonably reliable methods for analysing ductile metal beam and frame structures subjected to large impulses, and form a useful aid in the conceptual understanding of the large displacement dynamic problem, which is often complex.

Good agreement with experimental results can be obtained if the homogeneous viscous relation is suitably matched to the rigid-viscoplastic constitutive equation. It is also clear that, if a solution cannot be obtained with matching on both intercept and slope at the maximum initial strain rate, a compromise is possible with $n'$ chosen as large as possible for numerical stability; the larger the value of $n'$ the better the correlation with experimental results. Solutions are sensitive to the choice of $\mu$ and $\nu$, and unambiguous methods for the choice of these factors should form the subject for further study.

The direct integration procedure presented here is far more costly than solutions obtained using the mode approximation technique, and shows conclusively that where the mode approximation technique is appropriate it should be used. Where the deformations are not modal, however, the use of the direct integration technique cannot be avoided.

For analyses where localised deformations are significant but where the dominant deformation pattern is modal, the direct method and the mode technique may be combined such that once the localised deformations have been quantified (by the direct method), subsequent deformations may be found using the instantaneous mode solution technique. This approach will lead to a more efficient solution procedure, but further research is required in order to determine methods for choosing which analysis technique is appropriate for a given structure and loading configuration.

REFERENCES

1. J.B. Martin and P.S. Symonds, "Mode Approximations for Impulsively Loaded Rigid-Plastic Structures", J. Eng. Mech. Div., Proc. ASCE, 92, (EM5), 43-66, 1966.

2. J.B. Martin, "Impulsively Loading Theorems for Rigid-Plastic Continua", J. Eng. Mech. Div., Proc. ASCE, 90, (EM5), 27-42, 1964.

3. M.J. Mangoine, "Influence of Rate of Strain and Temperature on Yield Stresses of Mild Steel", J. Appl. Mechanics, 11, A211-A218, 1944.

4. R.J. Aspden and J.D. Campbell, "The Effect of Loading Rate on the Elastic-Plastic Flexure of Steel Beams", Proc. Roy. Soc. A., 290, 266-285, 1966.

5. E.W. Parkes, "The Permanent Deformation of a Cantilever Struck Transversely at its Tip", Proc. Roy. Soc. A., 228, 462-476, 1955.

6. T.C.T. Ting and P.S. Symonds, "Impack of a Cantilever Beam with Strain Rate Sensitivity", Proc. of 4th U.S. Nat. Cong. Appl. Mech., ASME, 2, 1153-1165, 1962.

7. T.C.T. Ting, "The Plastic Deformation of a Cantilever Beam with Strain Rate Sensitivity under Impulsive Loading", J. Appl. Mech., 31, 38-42, 1964.

8. L.S.S. Lee and J.B. Martin, "Approximate Solutions of Impulsively Loaded Structures of a Rate Sensitive Material", J. Appl. Math. Physics (ZAMP), 21, 1011-1032, 1970.

9. P.S.Symonds, "Approximation Techniques for Impulsively Loaded Structures of Rate Sensitive Plastic Behaviour, SIAM J. Appl. Math., 25, 462-473, 1973.

10. S.R. Bodner and P.S. Symonds, "Experimental and Theoretical Investigation of the Plastic Deformation of Cantilever Beams Subjected to Impulsive Loading", J. App. Mech., 29, 719-727, 1962.

11. S.R. Bodner, "Deformation of Rate Sensitive Structures Under Impulsive Loading", Engineering Plasticity (J. Heyman and F.A. Leckie, Editors), Cambridge University Press, 77-91, 1968.

12. P.S. Symonds, "Viscoplastic Behaviour in Response of Structures to Dynamic Loading", Proc. of Colloq. on Behaviour of Materials under Dynamic Loading (N.J. Huffington Jr., Editor), ASME, N.Y., 106-129, 1965.

13. P.S. Symonds and C.T. Chon, "Approximation Techniques for Impulsive Loading of Structures of Time-Dependent Plastic Behaviour with Finite Deflections", Proc. Oxford Cong. on Mechanical Properties of Materials at High Strain Rates Inst. of Physics Conf. (J. Harding, Editor), 21, 299-315, 1974.

14. P.S. Symonds and C.T. Chon, "Large Viscoplate Deflections of Impulsively Loaded Plane Frames", Int. J. Solids Structures, 15, 15-31, 1979.

15. J.B. Martin, "The Determination of Mode Shapes for Dynamically Loaded Rigid-Plastic Structures". To appear in Meccanica.

16. P.S. Symonds and J.L. Raphanel, "Large Deflections of Impulsively Loaded Plane Frames - Extentions of the Mode Approximation Technique", Proc. Conf. on M-chanical Properties of Materials at High Rates of Strain, Inst. of Physics Conf. (J. Harding, Editor), 47, 277-287, 1979.

17. H.H. Bleich and M.G. Salvaderi, "Impulsive Motion of Elasto-Plastic Beams", Trans. ASCE, 120, 499-520, 1955.

18. P.E. Duwez, D.S. Clark and H.F. Bohenblust, "The Behaviour of Long Beams under Impact Loading", J. Appl. Mech., 17, 27-34, 1950.

19. J.B. Martin, "Extremum Principles for a Class of Dynamic Rigid-Plastic Problems", Int. J. Solids Structures, 8, 1185-1204, 1972.

20. E.H. Lee and P.S. Symonds, "Large Deformations of Beams Under Transverse Impact", J. Appl. Mech., ASME, 19, 308-315, 1952.

21. P.S. Symonds, "Dynamic Load Characteristics in Plastic Bending of Beams", J. Appl. Mech., ASME, 20, 475-482, 1953.

22. P.S. Symonds, "Large Deformations of Beams Under Blast Type Loading", Proc. 2nd U.S. Nat. Cong. on Appl. Mech., ASME, 505-515, 1954.

23. E.W. Parkes, "The Permanent Deformation of an Inelastic Beam Struck Transversely at any Point in its Span", Proc. Inst. of Civ. Engrs., London, 10, 277, 1958.

24. E.A. Witmer, H.A. Balmer, J.W. Leach and T.H.H. Pian, "Large Dynamic Deformation of Beams, Rings, Plates and Shells", AIAA, J. 1, 1848, 1963.

25. H.A. Balmer, "Improved Computer Programs- DEPROSS 1, 2 and 3 - to Calculate the Dynamic Elastic-Plastic Two-Dimensional Responses of Impulsively Loaded Beams, Rings, Plates and Shells of Revolution", Technical Report No. ASRL TR128-3, M.I.T. Aeroelastic and Structures Research Laboratory, Aug. 1965.

26. S.J. Hashmi, S.T.S. Al-Hassani and W. Johnson, "Large Deflection Elastic-Plastic Response of Certain Structures to Impulsive Load: Numerical Solutions and Experimental Results", Int. J. Mech. Sci., 14- 843-860, 1972.

27. M.P. Kamat, "Survey of Computer Programs for Prediction of Crash Response and its Experimental Validation". Proc. Conf. on Measurement and Prediction of Structural and Biodynamic Crash Impact, (K.J. Saczalski and W.D. Pilkey, Editors), ASME, 33-48, 1976.

28. M.P. Kamat, "Nonlinear Transient Analysis of Aircraft-like Structures - Theory and Validation", Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Rep. VPI-E-79-10, 1979.

29. R. Winter, A.B. Pifko and H. Armen, "Crash Simulation of Skin-frame Structures Using a Finite Element Code", SAE paper 770484, 1977

30. K.S. Yeung and R.C. Welsh, "Refinement of Finite Element Analysis of Automobile Structures under Crash Loading", Tech. Final Rep., Vol.11. Dept. of Transportation, National Highway Traffic Administration, Contract DOT-MS-6-01364, 1977.

31. A.B. Pifko and R. Winter, "Theory and Application of Finite Element Analysis to Structural Crash Simulation", Computers and Structures, 13, 277-285, 1981.

32. N.M. Newmark, "A Method of Computation for Structural Dynamics", J. Eng. Mech. Div., ASCE, 85, (EM3), Proc. Paper 2074, 67, 1959.

33. V.P. Tamuzh, "On the Minimum Principle in Dynamics of Rigid-Plastic Bodies", Prikladnaya Matematicka i Mekhanika, 25, 715-722, 1962.

34. S.J. Hashmi and S.T.S. Al-Hassani, "Large Deflection Response of Square Frames to Distributed Impulsive Loading", Int. J. Mech. Sci., 17, 513-523, 1975.

35. L.S.S. Lee, "Mode Response of Dynamically Loaded Structures", J. Appl. Mech., 39, 904-910, 1972.

36. P.S. Symonds, "Approximation Techniques for Impulsively Loaded Plastic Structures", Course on Dynamics of Plastic Structures, Centre Internationale Des Sciences Mecaniques, Udine, Italy, Oct. 1979.

37. S.R. Bodner and P.S. Symonds, "Experiments on Dynamic Plastic Loading of Frames", Int. J. Solids Structures, 15, 1-13, 1979.

38. P.S. Symonds and N. Jones, "Impulsive Loading of Fully Clamped Beams with Finite Plastic Deflections and Strain Rate Sensitivity", Int. J. Mech. Sci., 14, 49-69, 1972.

39. P.S. Symonds, "Finite Elastic and Plastic Deformations of Pulse Loaded Structures by an Extended Mode Technique", Tech. Report No. ARO.79/2, Div. Engineering, Brown University, Providence, R.I., Jan 1980.

40. R.B. Wegener, "Impulsive Loading Tests on Portal Frames", Tech. Report No. 17, Nonlinear Structural Mechanics Research Unit, University of Cape Town, April 1982.

41. C.F. Gerald, Applied Numerical Analysis, 2nd Ed., Addison-Wesley, Reading, Mass., 1978.

# APPENDIX A

## GNLIMST and DAGNVS User Manual

# APPENDIX A

## GNLIMST and DAGNVS User Manual

### Introduction

GNLIMST and DAGNVS are finite element programs for the dynamic large displacement analysis of rigid-viscoplastic beams and frames which lie in one plane, which are supported only at their ends, and which are subjected to large impulsive loading. The theoretical background to GNLIMST and DAGNVS is described in Chapter 2 and section 3.2 of this thesis, respectively and the programs' implementation is outlined in Chapter 5.

In the description of the data input each data card is presented in a rectangular block intended to make the data card stand out on the page. A mixed notation is used to denote each parameter on a card, the notation chosen being deemed the most meaningful in the context in which it is used. Hence in one situation the letter N might be used to denote a node number whereas in another the word node might be used.

All data is input in free format and the FORTRAN real/integer convention is employed. Those letters, variables or words beginning with the letters I, J, K. L, M, N stand for integer values and those beginning with other letters stand for real values. The data input for both GNLIMST and DAGNVS is identical in both format and type.

In Section A-1 the data input will be described, and in Section A-2 some guidelines for the efficient use of the two programs will be suggested.

## A-1 Data Input

### 1. Problem Title

The data deck begins with a single line title which identifies the problem to be solved.

| title |
|---|

title - an alphanumeric title which may occupy columns 1 through 72 inclusive

The title is printed at the start of the output for the problem.

### 2. Plotting Request

The user may request plots of the deformed shape at successive time intervals throughout the timespan of deformation.

| PICT |
|---|

PICT - PLOT plotting required

XXXX no plotting required

### 3. Matching factor

A matching factor $\nu$ must be specified to set the magnitude of the power n' in the matched viscous material model (Chapter 4)

| $\nu$ |
|---|

$\nu$ 1. match on intercept alone (that is $n' = n$, see Section A.2.1.)

>1. compromise matching scheme (See Section A.2.1.)

-1. match on slope and intercept. Note that the quantity -1 is merely a dummy variable signifying that $\nu$ must be automatically computed.

4. <u>Section Size and Material Properties.</u>

| | | | | |
|---|---|---|---|---|
| h | b | $\sigma_o$ | $\dot{\varepsilon}_o$ | n |

h - section depth in metres

b - section breadth in metres

$\sigma_o$ - yield stress of section in $N/m^2$

$\dot{\varepsilon}_o$ - strain rate constant

n - power n in rigid-viscoplastic constitutive relation

5. <u>Node Incidences</u>

The structures are discretised into elements, which are numbered sequentially from the origin of a cartesian coordinate system. Nodes define element ends; two nodes thus define an element. For each element, in turn, the following data is required

| | |
|---|---|
| $node_i(a)$ | $node_i(b)$ |

$node_i(a)$ - the node number of the "a" end of element i

$node_i(b)$ - the node number of the "b" end of element i

i from 1 to number of elements

6. Nodal Coordinates

For each node the global X and Y coordinates are input sequentially.

| | |
|---|---|
| $x_i$ | $y_i$ |

$x_i$ - the global X-coordinate of node i

$y_i$ - the global Y-coordinate of node i,

i from 1 to number of nodes.

7. Boundary Conditions

The boundary conditions must be defined at each node which is partially or wholly constrained. At each node where some constraint occurs, the following data is required.

| |
|---|
| node |

| | | |
|---|---|---|
| i | j | k |

node - node number where constraint is present

i - 0 no restraint in the global X-direction

1 restraint in the global X-direction

j - 0 no restraint in the global Y-direction

1 restraint in the global Y-direction

k - 0 free to rotate

1 rotational fixity

Note that boundary conditions may only be applied at the first and last nodes, as only chain type structures are considered.

To signify the end of boundary condition input, the following card is required:

| |
|---|
| -1 |

8. <u>Structural Type</u>

If the structure is statically determinate, the user must specify whether it is a cantilever type structure or simply supported.

If the structure is hyperstatic, the user must render it statically determinate by releasing relevant constraints so as to make the structure either a cantilever or simply supported. For both statically determinate and hyperstatic structures, the following data is required:

| |
|---|
| type |

type - CANT if the structure, or released structure, is a cantilever

SIMP if the structure, or released structure, is simply supported

For hyperstatic structures, the degrees of freedom which are to be released so that the structure becomes either a cantilever or simply supported, must be specified. There are three degrees of freedom per node : the n-th node thus has degrees of freedom (3n-3) + 1, (3n-3) + 2, (3n-3) + 3 in the X, Y and rotational directions respectively. The degrees of freedom which are to be released are specified as:

| |
|---|
| $k_i$ ...... $k_j$ |

$k_i$ - i-th degree of freedom to be released.

Note that i never exceeds 3 and that this card is omitted if the structure is statically determinate.

9. Lumped Mass Model

Half the mass of each element adjacent to a node is lumped at that node, though the user may use his discretion in the choice of mass distribution.

The mass at each node is entered in turn as follows:

| |
|---|
| $M_1$ $M_2$ ... $M_i$ .... $M_n$ |

$M_i$ - mass at node i in kilograms

Masses at fixities are set to large values, typically $10^{10}$.

10. Initial Velocities

The initial velocity in the X, Y and rotational degrees of freedom at each node where non-zero initial velocities occur must be defined. At each node $n_k$, or sequence of nodes $n_i$ to $n_j$ which has initial velocities $v_x$ $v_y$ $\dot{\theta}$ in the global X, Y and rotational degrees of freedom, respectively the following data is required:

$n_i$ $n_j$

$v_x$ $v_y$ $\dot{\theta}$

$n_i$ - node number i

$n_j$ - node number j

Note that i may equal j, and that the nodes i to j must be sequential.

$v_x$ $v_y$ $\dot{\theta}$ - the initial velocity at nodes $n_i$ through $n_j$ inclusive, in the global X, Y and rotational direction, in m/sec or radians/sec.

To signify the end of velocity in input, the following data is required:

| | |
|---|---|
| -1 | -1 |

11. Time Step Size and Output Requirements

A crude estimate of the total time of deformation $t_f$ is automatically calculated by both GNLIMST and DAGNVS. The user must decide into how many time steps $t_f$ is to be subdivided. The frequency of output must also be specified by requesting the output after every k time steps. The data input is:

| | |
|---|---|
| ISTEP | k |

ISTEP - the number of time steps into which $t_f$ is to be subdivided (see Section A2.2)

k - the number of time steps between each output. The final output, when the structure is at or near rest, is always printed.

12. Trial Values for Nodal Forces (DAGNVS) and Redundants (GNLIMST)

In Section 3.2.1, a method for determining the initial moments and axial forces in the structure resulting from an initial velocity field is described. In this scheme, nodal forces are calculated which when applied as static loads, lead to the initial velocity field. This is an iterative process which requires an initial trial estimate of the nodal forces. This trial estimate is input as a single number, and the program sets all values of the nodal forces to this quantity. In GNLMIST, this trial estimate refers to the initial estimate of the redundant forces, referred to in Section 2.4.

The data input is

| X |
|---|

X - initial trial estimate of all the nodal forces (See Section A2.3).

13. Number of Elements, Nodes and Degree of Redundancy

These quantities must be defined internally in the program element COMPROC contained in both GNLIMST and DAGNVS. The parameters NE, NN and NRED define the number of elements, the number of nodes and the degree of redundancy of the structure, respectively.

NE the number of elements

NN the number of nodes

NRED the degree of redundancy

1 statically indeterminate to degree 1

2 statically indeterminate to degree 2

3 statically indeterminate to degree 3

4 statically determinate

## A-2 Guidelines for the Use of GNLMST and DAGNVS

As with all materially and geometrically nonlinear programs, GNLIMST, and particularly DAGNVS require a certain level of experience of the user if efficient and meaningful solutions are to be obtained. Unwise choice of certain parameters may result in nonconvergence of the algorithms presented in this thesis, and no solution will be obtained.

Here, guidelines for the reasonable choice of magnitude of these parameters are given.

### A-2.1 The Matching Factor

In GNLIMST only matching on slope alone is permitted, so $\nu$ is set to 1.

In DAGNVS, the choice of three matching schemes is available; matching on slope alone ($\nu$=1), matching on slope and intercept ($\nu$=-1) and a compromise matching procedure where n' is set as high as will permit a solution ($\nu$>1).

For certain classes of problems, the $\nu$=1 option, whilst providing an economical solution as n' is as low as possible, does not give accurate results. For other problems, the full matching scheme ($\nu$=-1) cannot be implemented as it requires that n' exceeds 12, with the result that convergence cannot be obtained in the algorithm for the determination of the initial bending moments and axial forces (Section 3.2.1). The compromise matching procedure permits a solution to be obtained which improves on the results obtained with $\nu$=1, and circumvents the numerical difficulties encountered using the full matching scheme ($\nu$=-1). Trial values must be input to determine the highest $\nu$ for which a solution can be obtained. Typical values range between 1,8 and 2,2. Note, however, that the success of a solution for a particular $\nu$ depends on the choice of initial trial nodal forces (See section A-2.3).

A-2.2 Time Step Size

In both GNLIMST and DAGNVS, the choice of time step size depends on the complexity of the structure and the magnitude of the expected deformations; the greater the complexity and deformations, the smaller the timestep. In DAGNVS, the higher the value of n' used, the smaller the time step needed for convergence of the solution.

In GNLIMST, the time step parameter varies between 20 and 100, its choice being dictated more by the accuracy of the solution required rather than potential lack of convergence of the solution procedure.

In DAGNVS, a much larger time step parameter is required, typically in the range of 500 to 5000, depending on the complexity of the structure and the value of n'. A larger time step than the minimum is preferable, as fewer equilibrium iterations per time step are required which generally leads to a more efficient and accurate solution.

The critical phase from a convergence viewpoint of the direct analysis procedure are the first few time steps, which dictate the size of the time increment. In DAGNVS, if it is found that less than 5 equilibrium iterations are required per time step, the time increment is automatically increased by a factor of 1,5. The time step is thus automatically increased if it is found to be unnecessarily small.

### A-2.3 Choice of Initial Trial Node Forces and Redundants

In GNLIMST, it was found that a trial value of X=1. is suitable for all the problems considered in the scope of this thesis.

In DAGNVS, the choice of X depends on the choice of the matching factor $\nu$ (section A-2.1) and trial values must be input until a solution is obtained. If a solution cannot be found for any reasonable choice of X, then the choice of $\nu$ must be altered. No correlation is apparent between the value of $\nu$ and the choice of X. For the problems considered in this thesis, the value of X ranged between 0,1 and 100. In seeking solutions, the choice of this range was somewhat arbitrary, but serves as a rough guide. For most problems a solution was obtained for X=1. X=0 is not permitted.

# APPENDIX B

## GNLIMST program listing

```
C         ******************************
C
C          D R I V E R     R O U T I N E
C
C         ******************************
        INCLUDE GNLIMST.COMPROC
C       INPUT DATA
        CALL INPUT
C       DISPLAY DATA
        CALL DATA
 5      CONTINUE
C       DETERMINE STATICS OF STRUCTURE
        CALL STAT
 2      CONTINUE
C       DETERMINE CURRENT LOADING
        CALL LOAD
 1      CONTINUE
C        IF STRUCTURE STATICALLY DETERMINATE (IE IF NRED=4 )
C       THEN DON'T NEED TO ITERATE FOR REDUNDANTS
        IF(NRED.EQ.4) GO TO 3
C       ZERO ALL ARRAYS IN ITERATIVE PROCEDURE TO FOLLOW
        CALL ZERO
C       EVALUATE COMPATIBILITY EQUATIONS
        CALL COMEQU
C       CALCULATE PARTIAL DIRIVATIVES OF COMPATIBILITY EQUATIONS
        CALL PDIFF
C       INVERT MATRIX OF PARTIAL DIRIVATIVES
        CALL PIVOT
C       CHECK MAGNITUDE OF PERTURBATION IN VALUE OF REDUNDANTS
C       IF SMALL (IFLAG=1) GO TO VELOC : IF LARGE(IFLAG=0) ITERATE
        CALL DELTA
        IF(IFLAG.EQ.0) GO TO 1
 3      CONTINUE
C       DETERMINE VELOCITIES ASSOCIATED WITH
C       TOTAL STRESS STATE OF THE STRUCTURE
        CALL VELOC
C       CHECK IF PROCEDURE HAS CONVERGED ONTO MODE
        CALL MODECH
C        IF NOT THEN ITERATE
         IF(IDISIP.EQ.0) GO TO 2
 4      CONTINUE
C       ONCE PHI IS DETERMINED , MODE SOLUTION CAN COMMENCE
        CALL INMODE
C
C       DETERMINE MATCHING CONSTANT THEN STORE AND OUTPUT
C       INITIAL MODE CONFIGURATION
        IF(T.LT.1.E-9)THEN
        CALL MATCH
        CALL OUTPUT
        END IF
C
C
        IF(IRND.EQ.0)CALL STORE
C
        IF(IRND.EQ.0)T=T+DT
C
        CALL UPDATE
C
        IF(AMP(2)/AMP(1).LE.0.1)THEN
```

```
      IOUT=1
      ICOUNT=NDIV-1
      END IF
C
      IRND=IRND+1
C
C     IF MEMBRANE SOLUTION PREDOMINATES CALCULATE
C      NEW MATCHING FACTOR
       IF(MATCHA.EQ.1)CALL MATCH
C
      IF(IOUT.EQ.0)GO TO 5
C
      IRND=0
C
      ICOUNT=ICOUNT+1
C
      IF(IOUT.EQ.1.AND.ICOUNT.EQ.NDIV)CALL OUTPUT
C
      IOUT=0
C
C     CALL STORE
C
      IF(AMP(2)/AMP(1).GT.0.1)GO TO 5
C
C     IS PLOTTING OF DEFORMATION HISTORY REQUIRED ?
C
      IF(PICT.EQ.'PLOT')CALL PICTUR
C
      STOP
C
      END
C
C
C      ************************************
C
C                    I N P U T
C
C                 INPUT ALL DATA
C
C      ************************************
      SUBROUTINE INPUT
      INCLUDE GNLIMST.COMPROC
      DIMENSION IIBC(NF),VELDUM(NF)
  100 FORMAT( )
  101 FORMAT(A4)
  102 FORMAT(A80)
  103 FORMAT(A5)
      READ(IREAD,102)TITLE
      READ(IREAD,101)PICT
      READ(IREAD,100)PMATCH
      READ(IREAD,100)HH,BB,YSTRS,EPSIO,EN
C     NODE INCIDENCES (NUMBERING OF ELEMENT ENDS)
      DO 11 IE=1,NE
      READ(IREAD,100)(NBEAM(IE,I),I=1,2)
   11 CONTINUE
C     COORDINATES OF ORDERED NODES (X,Y)
      DO 12 I=1,NN
      READ(IREAD,100)COORDX(I),COORDY(I)
   12 CONTINUE
```

```
C       BOUNDARY CONDITIONS
C       FREEDOM : 0
C       FIXITY  : 1
        DO 2 I=1,NDF
        IBC(I)=0
2       CONTINUE
3       READ(IREAD,100)N1
        IF(N1.LT.0) GO TO 4
        READ(IREAD,100)(IIBC(J),J=1,NF)
        II=NF*(N1-1)
        DO 5 I=1,NF
        II=II+1
        IBC(II)=IIBC(I)
5       CONTINUE
        GO TO 3
4       CONTINUE
C        IS STRUCTURE DETERMINATE OR HYPERSTATIC ?
        IF(NRED.EQ.4) THEN
C        IF DETERMINATE , IS IT A CANTILEVER OR SIMPLY SUPPORTED ?
        READ(IREAD,101)STADET
        ELSE
C       IF HYPERSTATIC , NRED DEGREES OF FREEDOM MUST BE RELEASED
C       FOR A DETERMINATE STRUCTURE . IS RESULTING STUCTURE A
C       CANTILEVER OR A SIMPLY SUPPORTED STRUCTURE ?
        READ(IREAD,101)STADET
C       DEGREES OF FREEDOM WHICH ARE RELEASED
        READ(IREAD,100)(RELEAS(I),I=1,NRED)
        END IF
C       READ IN MASS VECTOR
        READ(IREAD,100)(RMASS(I),I=1,NN)
C       READ IN INITIAL VELOCITY
6       READ(IREAD,100)N1,N2
        IF(N1.LT.0)GO TO 7
        READ(IREAD,100)(VELDUM(I),I=1,NF)
        II=NF*(N1-1)
        JJ=N2-N1+1
        DO 8 J=1,JJ
        DO 8 I=1,NF
        II=II+1
        VEL(II,1)=VELDUM(I)
8       CONTINUE
        GO TO 6
7       CONTINUE
C       STATE HOW MANY TIME INTERVALS IN MODE SOLUTION ARE REQUIRED
C       AND NUMBER OF OUTPUTS REQUIRED
        READ(IREAD,100)RINT,NDIV
C        READ INITIAL ESTIMATE OF REDUNDANTS
C        READ(IREAD,100)XXDUM
C       CALCULATE YIELD MOMENT  AND AXIAL YIELD STRESS
        AA=HH*BB
        RM0=AA*HH*YSTRS/4.
        RN0=AA*YSTRS
C
C       SET INITIAL MODE SHAPE (GUESS) EQUAL TO INIYIAL VELOCITY
        DO 1 I=1,NDF
        PHI(I,1)=VEL(I,1)
1       CONTINUE
C        CALCULATE NORMALISATION CONSTANT AND INITIAL
C       DISSIPATION RATE .
```

```
         DO 10 I=1,NDF,3
         II=INT(FLOAT(I)/NF+0.7)
         IK=I+1
         DISIP(2)=DISIP(2)+VEL(I,1)*RMASS(II)*PHI(I,1)
         DISIP(2)=DISIP(2)+VEL(IK,1)*RMASS(II)*PHI(IK,1)
         RKINET=RKINET+PHI(I,1)*PHI(I,1)*RMASS(II)
         RKINET=RKINET+PHI(IK,1)*PHI(IK,1)*RMASS(II)
   10    CONTINUE
   C
         EN1=EN+1
         EN2=EN1+1
   C     IF STRUCTURE INDETERMINATE(NRED NOT EQUAL TO 4),DUMMY VALUES
   C     ARE ASSIGNED TO REACTANTS (X) TO AVOID POSSIBLE DIVISION BY
   C     ZERO IN FORMULATION OF COMPATIBILITY EQUATIONS .
         IF(NRED.NE.4)THEN
         DO 9 I=1,NRED
         X(I)=1.0
   9     CONTINUE
         END IF
         RETURN
         DEBUG SUBCHK
   C
         END
   C
   C
   C      **********************************
   C
   C                    D A T A
   C
   C      DISPLAYS ALL DATA FOR VERIFICATION
   C
   C      **********************************
         SUBROUTINE DATA
         INCLUDE GNLIMST.COMPROC
   C
         WRITE(IPRINT,1)TITLE
   1     FORMAT(1H1,5X,80('*'),/10X,A80,/,6X,80('*'),/)
         WRITE(IPRINT,17)
   17    FORMAT(1H ,/,20X,'LARGE DISPLACEMENT ANALYSIS',/)
         WRITE(IPRINT,13)EN,RMO,RNO,EPSIO,YSTRS
   13    FORMAT(1H ,5X,///,' MATERIAL ASSUMPTIONS',//,'  HOMOGENEOUS
        #VISCOUS WITH POWER N =',F7.3,/,2X,'YIELD MOMENT ='
        #,F11.6,/,2X,'AXIAL YIELD STRENGTH =',E15.4,/,2X,
        #'INITIAL STRAIN RATE =',E15.6,/,2X,
        #'YIELD  STRESS =',E15.8,/)
         WRITE(IPRINT,2)NE,NN
   2     FORMAT(1H ,/,3X,'NUMBER OF ELEMENTS  :',I3,/,3X,'NUMBER OF
        #NODES   :',I3,/)
         WRITE(IPRINT,3)
   3     FORMAT(//,6X,' COORDINATES OF NODES',//,
        #'NODE',10X,'X',12X,'Y')
         DO 110 I=1,NN
         WRITE(IPRINT,4)I,COORDX(I),COORDY(I)
   4     FORMAT(1H ,I3,1X,2(2X,F11.5))
   110   CONTINUE
         WRITE(IPRINT,5)
   5     FORMAT(//,' BOUNDARY CONDITIONS : 0=FREEDOM , 1=FIXITY',
        #//,' NODE',3X,'X',3X,'Y',3X,'ROTATION')
         WRITE(IPRINT,6)1,IBC(1),IBC(2),IBC(3)
```

```
      WRITE(IPRINT,6)NN,IBC(NDF-2),IBC(NDF-1),IBC(NDF)
6     FORMAT(1H ,I3,4X,I1,3X,I1,6X,I1)
      WRITE(IPRINT,11)
11    FORMAT(1H ,///,5X,'LUMPED MASS PER
     #NODE',//,6X,'NODE',10X,'MASS',/)
      DO 111 I=1,NN
      WRITE(IPRINT,12)I,RMASS(I)
12    FORMAT(1H ,5X,I2,6X,E11.4,/)
111   CONTINUE
      IF(NRED.NE.4)THEN
      IF(STADET.EQ.'CANT')WRITE(IPRINT,9)(RELEAS(I),I=1,NRED)
9     FORMAT(1H ,//,' STRUCTURE CANTILEVERED BY RELEASING
     #RESTRAINTS AT D.O.F. ',5(1X,I2))
      IF(STADET.EQ.'SIMP')WRITE(IPRINT,10)(RELEAS(I),I=1,NRED)
10    FORMAT(1H ,//,'  STRUCTURE MADE SIMPLY SUPPORTED BY
     #RELEASING RESTRAINTS AT D.O.F. ',5(3X,I2))
      END IF
      WRITE(IPRINT,14)
14    FORMAT(1H ,///,10X,'INITIAL VELOCITY',//,
     #'NODE',4X,'X',12X,'Y',9X,'ROTATION')
      DO 112 I=1,NDF,3
      II=INT(I/NF)+1
      WRITE(IPRINT,15)II,VEL(I,1),VEL(I+1,1),VEL(I+2,1)
15    FORMAT(1H ,I3,1X,3(2X,E11.3),/)
112   CONTINUE
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C
*************************************************************
C
C                         S T A T
C
C      IF THE STRUCTURE IS STATICALLY INDETERMINATE THEN
C      BOUNDARY CONDITIONS ARE RELEASED SUCH THAT IT
C      BECOMES EITHER A CANTILEVER ('CANT') OR SIMPLY
C      SUPPORTED ('SIMP').THE BENDING MOMENT DIAGRAM AND
C      AXIAL FORCE DIAGRAM DUE TO A UNIT LOAD APPLIED IN
C      TURN AT EACH DEGREE OF FREEDOM IS THEN DETERMINED
C      BENDING   : UNITM(I,J),I=D.O.F. WHERE LOAD APPLIED,
C                                          J=NODE NO.
C      AXIAL     : UNITN(I,J),I=D.O.F. WHERE LOAD APPLIED,
C                                          J=ELEMENT
C      FORCE(I,IE,J)  : SELF-STRESS AND STATICALLY ADMISSIBLE
C                                          SETS
C          I          : STATICALLY ADMISSIBLE SET
C          I=2,NRED   : SELF STRESS SYSTEMS
C          IE         : ELEMENT NO.
C          J=1        : MOMENT AT 'A' END OF BEAM
C          J=2        : MOMENT AT 'B' END OF BEAM
C          J=3        : AXIAL FORCE IN ELEMENT
C
C
*************************************************************
      SUBROUTINE STAT
      INCLUDE GNLIMST.COMPROC
```

```
C
      ITMODE=0
C     DETERMINE ORIENTATION OF ELEMENTS IN GLOBAL AXIS SYSTEM
      DO 55 IE=1,NE
      IL=NBEAM(IE,1) @NODE NO. OF A END OF ELEMENT IE
      IR=NBEAM(IE,2) @NODE NO. OF B END OF ELEMENT IE
C     CL=CURRENT LENGTH OF ELEMENT
      CL(IE)=SQRT((COORDX(IR)-COORDX(IL))**2.+
     #(COORDY(IR)-COORDY(IL))**2.)
      SSIN(IE)=(COORDY(IR)-COORDY(IL))/CL(IE)
      CCOS(IE)=(COORDX(IR)-COORDX(IL))/CL(IE)
 55   CONTINUE
C
C     FOR BENDING MOMENTS DUE TO UNIT LOAD AT D.O.F. I
C     FOR CANTILEVER
      IF(STADET.EQ.'CANT')THEN
C     HORISONTAL
C
      DO 1 I=4,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      DO 2 J=1,IR
      UNITM(I,J)=-(COORDY(IR+1)-COORDY(J))
 2    CONTINUE
 1    CONTINUE
C
C     VERTICAL
      DO 3 I=5,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      DO 4 J=1,IR
      UNITM(I,J)=-(COORDX(J)-COORDX(IR+1))
 4    CONTINUE
 3    CONTINUE
C     MOMENTS
      DO 26 I=6,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      DO 27 J=1,IR
      UNITM(I,J)=1.
 27   CONTINUE
 26   CONTINUE
      ELSE
C     FOR SIMPLY SUPPORTED STRUCTURE
C     HORISONTAL LOADING AT D.O.F. I
      DO 8 I=4,NDF-2,NF
      DO 9 J=2,NN-1
      R1=0.
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      IF(J.GT.IR+1)R1=1.
      UNITM(I,J)=-COORDX(J)*COORDY(IR+1)/COORDX(NN)
     #+(COORDY(J)-COORDY(1))-R1*(COORDY(J)-COORDY(IR+1))
 9    CONTINUE
 8    CONTINUE
C     VERTICAL UNIT LOADING AT D.O.F. I
      DO 10 I=5,NDF-1,NF
      DO 11 J=2,NN-1
      R1=0
```

```
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      IF(J.GT.IR+1)R1=1.
      UNITM(I,J)=(COORDX(IR+1)/COORDX(NN)-1.)*COORDX(J)
     #+R1*(COORDX(J)-COORDX(IR+1))
 11   CONTINUE
 10   CONTINUE
C      MOMENT AT D.O.F. I
      DO 30 I=3,NDF,NF
      DO 31 J=1,NN
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      R1=0.
      IF(J.GT.IR)R1=1.
      UNITM(I,J)=(COORDX(J)/COORDX(NN)-R1)
      IF(I.EQ.3)UNITM(I,1)=-1.
      IF(I.EQ.NDF)UNITM(I,NN)=1.
 31   CONTINUE
 30   CONTINUE
      END IF
C
C     FOR AXIAL FORCES DUE TO UNIT LOADS AT D.O.F. I
C
C      HORISONTAL UNIT LOADS
      DO 13 I=4,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      IRR=IR
      IF(STADET.EQ.'SIMP') IRR=NE
      DO 14 IE=1,IRR
      R1=0
      IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') R1=1.
      IF(STADET.EQ.'CANT')THEN
      UNITN(I,IE,1)=CCOS(IE)
      ELSE
      UNITN(I,IE,1)=CCOS(IE)+COORDY(IR+1)*SSIN(IE)/COORDX(NN)
     #-R1*CCOS(IE)
      END IF
 14   CONTINUE
 13   CONTINUE
C
C     VERTICAL UNIT LOADS
      DO 16 I=5,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      IRR=IR
      IF(STADET.EQ.'SIMP') IRR=NE
      DO 17 IE=1,IRR
      R1=0
      IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') R1=1.
      IF(STADET.EQ.'CANT')THEN
      UNITN(I,IE,1)=SSIN(IE)
      ELSE
      UNITN(I,IE,1)=SSIN(IE)*(1.-COORDX(IR+1)/COORDX(NN))
     #-R1*SSIN(IE)
      END IF
 17   CONTINUE
 16   CONTINUE
C
```

```
C       UNIT APPLIED MOMENT
C        NO AXIAL FORCES CAUSED BY UNIT MOMENTS IN CANTILEVER
        IF(STADET.EQ.'SIMP')THEN
        DO 33 I=3,NDF,NF
        DO 34 IE=1,NE
        UNITN(I,IE,1)=-SSIN(IE)/COORDX(NN)
34      CONTINUE
33      CONTINUE
        END IF
C
C
C       SET UP SELF STRESS SYSTEMS EQUAL TO MOMENTS AND AXAIL
C       FORCES ASSOCIATED WITH UNIT LOADS AT RELEASED D.O.F.'S
C       IF STATICALLY DETERMINATE THEN SKIP
        IF(NRED.NE.4)THEN
        DO 35 I=2,NRED+1
        DO 36 IE=1,NE
        FORCE(I,IE,1)=UNITM(RELEAS(I-1),NBEAM(IE,1))
        FORCE(I,IE,2)=UNITM(RELEAS(I-1),NBEAM(IE,2))
        FORCE(I,IE,3)=UNITN(RELEAS(I-1),IE,1)
36      CONTINUE
35      CONTINUE
        END IF
C
        RETURN
        DEBUG SUBCHK
        END
C
C
C        *********************************************
C
C                          L O A D
C
C        ASSEMBLE STATICALLY ADMISSIBLE BENDING MOMENT
C        AND AXIAL FORCE DIAGRAMS DUE TO LOADING GIVEN
C        BY PRODUCT (MASS/NODE*CURRENT MODE SHAPE)
C
C        *********************************************
        SUBROUTINE LOAD
        INCLUDE GNLIMST.COMPROC
C
        ITREAC=0
        DO 1 IE=1,NE
        DO 2 J=1,3
        FORCE(1,IE,J)=0.
2       CONTINUE
1       CONTINUE
        DO 18 I=1,NDF
        II=INT(FLOAT(I)/NF+0.7)
C       IF D.O.F. IS A ROTATION THEN NO LOAD ASSOCIATED WITH IT
        IJ=(INT(FLOAT(I)/NF+0.001))*NF
        IF(I.NE.IJ)THEN
        RLOAD(I)=RMASS(II)*PHI(I,1)
        END IF
        DO 19 IE=1,NE
        FORCE(1,IE,1)=FORCE(1,IE,1)+UNITM(I,NBEAM(IE,1))*RLOAD(I)
        FORCE(1,IE,2)=FORCE(1,IE,2)+UNITM(I,NBEAM(IE,2))*RLOAD(I)
C       AXIAL FORCES
        FORCE(1,IE,3)=FORCE(1,IE,3)+UNITN(I,IE,1)*RLOAD(I)
```

```
      IF(NRED.EQ.4)THEN
      FORCET(IE,1)=FORCE(1,IE,1)
      FORCET(IE,2)=FORCE(1,IE,2)
      FORCET(IE,3)=FORCE(1,IE,3)
      END IF
19    CONTINUE
18    CONTINUE
C
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C      ************************************
C
C                     Z E R O
C
C        INITIALISE ARRAYS USED IN ITERATIVE
C        PROCEDURE TO DETERMINE REDUNDANTS
C
C      ************************************
      SUBROUTINE ZERO
      INCLUDE GNLIMST.COMPROC
C
      DO 1 I=1,NRED
      FLEX(I,NRED+1)=0.
      COMPAT(I)=0.
      DO 2 J=1,NRED
      FLEX(I,J)=0.
      PARDIF(I,J)=0.
2     CONTINUE
1     CONTINUE
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C      ***********************************
C
C                 C O M E Q U E
C
C      SET UP NRED COMPATIBLITY EQUATIONS
C      WHERE NRED EQUALS NO. OF REDUNDANTS
C
C      ***********************************
      SUBROUTINE COMEQU
      INCLUDE GNLIMST.COMPROC
      DOUBLE PRECISION SUMA,SUMB,SUMC,POWA,POWB,POWC,POWE,POWF,
     #POWG,POWH,POWI,POWJ,POWK,POWL,PRODA,PRODB,PRODC
      ITREAC=ITREAC+1
C
C     SUM OVER ALL ELEMENTS IE
      DO 1 IE=1,NE
C     SET UP PARAMETERS FOR CALCULATIONS TO FOLLOW
      DO 2 I=1,NRED+1
C     MOMENTS AND AXIAL FORCES ARE NORMALISED
      AMOM(I)=(FORCE(I,IE,2)-FORCE(I,IE,1))/RMO
```

```
         ANORM1(I)=FORCE(I,IE,1)/RM0+FORCE(I,IE,3)/RN0
         ANORM2(I)=FORCE(I,IE,3)/RN0-FORCE(I,IE,1)/RM0
2        CONTINUE
         SUMA=ANORM1(1)
         SUMB=ANORM2(1)
         SUMC=AMOM(1)
         DO 3 I=1,NRED
         SUMA=SUMA+X(I)*ANORM1(I+1)
         SUMB=SUMB+X(I)*ANORM2(I+1)
         SUMC=SUMC+X(I)*AMOM(I+1)
3        CONTINUE
C        SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
         SIG=1.
         SIGA=1.
         SIGB=1.
         SIGC=1.
         SIGD=1.
         IF((SUMA+SUMC).LT.0) SIGA=-1.
         IF((SUMB-SUMC).LT.0) SIGB=-1.
         IF(SUMA.LT.0) SIGC=-1.
         IF(SUMB.LT.0) SIGD=-1.
         POWI=SIGA*(DABS(SUMA+SUMC)**EN)
         POWJ=SIGC*(DABS(SUMA)**EN)
         POWK=SIGB*(DABS(SUMB-SUMC)**EN)
         POWL=SIGD*(DABS(SUMB)**EN)
         POWA=POWI*(SUMA+SUMC)
         POWB=POWK*(SUMB-SUMC)
         POWC=POWA*(SUMA+SUMC)
         POWD=POWB*(SUMB-SUMC)
         POWE=POWJ*SUMA
         POWF=POWL*SUMB
         POWG=POWE*SUMA
         POWH=POWF*SUMB
         PRODA=EN1*SUMC/CL(IE)
         PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
         PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
C         SET UP COMPATIBILITY EQUATIONS,ONE FOR EACH
C        DEGREE OF REDUNDANCY .
         DO 4 I=1,NRED
         COMPAT(I)=COMPAT(I)+(FORCE(I+1,IE,3)*EPSI0*0.5)*
        #((POWA-POWE)/PRODA-(POWB-POWF)/PRODA)
        #+RN0*EPSI0/(2.*RM0)*((AMOM(I+1)*RM0/CL(IE))*
        #((POWC-POWG)/PRODB-SUMA*(POWA-POWE)/PRODC
        #-(POWD-POWH)/PRODB+SUMB*(POWB-POWF)/PRODC)
        #+FORCE(I+1,IE,1)*((POWA-POWE)/PRODA+(POWB-POWF)/PRODA))
4        CONTINUE
1        CONTINUE
C
         RETURN
         DEBUG SUBCHK
         END
C
C
```

```
C        ****************************************
C
C                        P D I F F
C
C        CALCULATE THE PARTIAL DIRIVATIVES OF
C        THE COMPATIBILITY EQUATIONS WITH RESPECT
C        TO THE REDUNDANT FORCES FOR USE IN THE
C        NEWTON-RAPHSON SOLUTION PRODEDURE
C
C        ****************************************
      SUBROUTINE PDIFF
      INCLUDE GNLIMST.COMPROC
      DOUBLE PRECISION SUMA,SUMB,SUMC,POWA,POWB,POWC,POWD,
     #POWE,POWF,POWG,POWH,POWI,POWJ,POWK,POWL,PRODA,PRODB,
     #PRODC,PARTA,PARTB,PART1PART2,PART3,PART4,PART5,PART6
C        TO OBTAIN THE UNKNOWN REACTIONS X(I) THE PARTIAL
C        DIRIVATIVES OF THE COMPATIBILITY EQUATIONS ARE
C        THE OBTAINED . THE NEWTON-RAPHSON METHOD IS USED
C     TO ITERATE ONTO A SOLUTION .
11    CONTINUE
      DO 1 IE=1,NE
C     SET UP PARAMETERS FOR CALCULATIONS TO FOLLOW
      DO 2 I=1,NRED+1
C     MOMENTS AND AXIAL FORCES ARE NORMALISED
      AMOM(I)=(FORCE(I,IE,2)-FORCE(I,IE,1))/RM0
      ANORM1(I)=FORCE(I,IE,1)/RM0+FORCE(I,IE,3)/RN0
      ANORM2(I)=(FORCE(I,IE,3)/RN0-FORCE(I,IE,1)/RM0)
2     CONTINUE
      SUMA=ANORM1(1)
      SUMB=ANORM2(1)
      SUMC=AMOM(1)
      DO 3 I=1,NRED
      SUMA=SUMA+X(I)*ANORM1(I+1)
      SUMB=SUMB+X(I)*ANORM2(I+1)
      SUMC=SUMC+X(I)*AMOM(I+1)
3     CONTINUE
C     SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
      SIG=1.
      SIGA=1.
      SIGB=1.
      SIGC=1.
      SIGD=1.
      IF((SUMA+SUMC).LT.0) SIGA=-1.
      IF((SUMB-SUMC).LT.0) SIGB=-1.
      IF(SUMA.LT.0) SIGC=-1.
      IF(SUMB.LT.0) SIGD=-1.
      POWI=SIGA*(DABS(SUMA+SUMC)**EN)
      POWJ=SIGC*(DABS(SUMA)**EN)
      POWK=SIGB*(DABS(SUMB-SUMC)**EN)
      POWL=SIGD*(DABS(SUMB)**EN)
      POWA=POWI*(SUMA+SUMC)
      POWB=POWK*(SUMB-SUMC)
      POWC=POWA*(SUMA+SUMC)
      POWD=POWB*(SUMB-SUMC)
      POWE=POWJ*SUMA
      POWF=POWL*SUMB
      POWG=POWE*SUMA
      POWH=POWF*SUMB
      PRODA=EN1*SUMC/CL(IE)
```

```
      PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
      PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
C     SET UP PARTIAL DIRIVATIVES
      DO 4 I=1,NRED
      DO 5 J=1,NRED
      PARTA=ANORM1(J+1)+AMOM(J+1)
      PARTB=ANORM2(J+1)-AMOM(J+1)
      PART1=((POWI*PARTA-POWJ*ANORM1(J+1))*EN1*PRODA
     #-EN1*AMOM(J+1)*(POWA-POWE)/CL(IE))/(PRODA*PRODA)
      PART2=((POWK*PARTB-POWL*ANORM2(J+1))*EN1*PRODA
     #-EN1*AMOM(J+1)*(POWB-POWF)/CL(IE))/(PRODA*PRODA)
      PART3=((POWA*PARTA-POWE*ANORM1(J+1))*EN2*EN2*PRODA*PRODA/
     #(EN1*EN1)-(POWC-POWG)*2.*EN2*PRODA*AMOM(J+1)/(EN1*CL(IE)))/
     #(PRODB*PRODB)
      PART4=(((POWA*ANORM1(J+1)+SUMA*EN1*POWI*PARTA)
     #-(POWE*ANORM1(J+1)
     #+SUMA*EN1*POWJ*ANORM1(J+1)))*PRODA*PRODA/EN1
     #-((POWA-POWE)*2.*PRODA*AMOM(J+1)*SUMA)/CL(IE))/
     #(PRODC*PRODC)
      PART5=((POWB*PARTB-POWF*ANORM2(J+1))*EN2*EN2*PRODA*PRODA/
     #(EN1*EN1)-(POWD-POWH)*2.*EN2*PRODA*AMOM(J+1)/(EN1*CL(IE)))/
     #(PRODB*PRODB)
      PART6=(((ANORM2(J+1)*POWB+SUMB*EN1*POWK*PARTB)-
     #(POWF*ANORM2(J+1)+SUMB*EN1*POWL*ANORM2(J+1)))
     #*PRODA*PRODA/EN1
     #-((POWB-POWF)*2.*PRODA*AMOM(J+1)*SUMB)/CL(IE))/
     #(PRODC*PRODC)
      PARDIF(I,J)=PARDIF(I,J)+FORCE(I+1,IE,3)*EPSI0*.5*
     #(PART1-PART2)+RN0*EPSI0/(2.*RM0)*((AMOM(I+1)*RM0/CL(IE))*
     #(PART3-PART4-PART5+PART6)+FORCE(I+1,IE,1)*(PART1+PART2))
    5 CONTINUE
    4 CONTINUE
    1 CONTINUE
C
C     HAVING OBTAINED THE PARTIAL DIRIVATIVES OF EQUILIBRIUM
C     EQUATIONS,MUST INVERT TO OBTAIN PERTURBATIONS ON REDUNDANTS
C     FIRST SET UP AUGMENTED MATRIX FOR INVERTION.
C
      DO 6 I=1,NRED
      FLEX(I,NRED+1)=-COMPAT(I)
      DO 7 J=1,NRED
      FLEX(I,J)=PARDIF(I,J)
    7 CONTINUE
    6 CONTINUE
      RETURN
      DEBUG SUBCHK
      END
C
C
```

```
C       ****************************************
C
C                       P I V O T
C
C        INVERSION ROUTINE WHICH INCLUDES PARTIAL
C                      PIVOTING
C
C       ****************************************
        SUBROUTINE PIVOT
        INCLUDE GNLIMST.COMPROC
        DOUBLE PRECISION RATIO,VALUE
C       FACTOR ALL ELEMENTS OF MATRIX BY LARGE NUMBER FOR
C       NUMERICAL STABILITY
        DO 60 I=1,NRED
        DO 61 J=1,NRED+1
        FLEX(I,J)=FLEX(I,J)*1.D9
61      CONTINUE
60      CONTINUE
        IF(NRED.NE.1)THEN
        NP=NRED+1
        NM1=NRED-1
        DO 1 I=1,NRED
        IF(DABS(FLEX(I,I)).LT.1.D-25)FLEX(I,I)=1.
1       CONTINUE
        DO 35 I=1,NM1
        IPVT=I
        IP1=I+1
        DO 10 J=IP1,NRED
        IF(DABS(FLEX(IPVT,I)).LT.DABS(FLEX(J,I))) IPVT=J
10      CONTINUE
        IF(DABS(FLEX(IPVT,I)).LT.1.D-32)GO TO 99
        IF(IPVT.EQ.I) GO TO 25
        DO 20 JCOL=I,NP
        FACT=FLEX(I,JCOL)
        FLEX(I,JCOL)=FLEX(IPVT,JCOL)
        FLEX(IPVT,JCOL)=FACT
20      CONTINUE
25      DO 32 JROW=IP1,NRED
        IF(DABS(FLEX(JROW,I)).LE.1.D-36) GO TO 32
        RATIO=FLEX(JROW,I)/FLEX(I,I)
        DO 30 KCOL=IP1,NP
        FLEX(JROW,KCOL)=FLEX(JROW,KCOL)-RATIO*FLEX(I,KCOL)
30      CONTINUE
32      CONTINUE
35      CONTINUE
        IF(DABS(FLEX(NRED,NRED)).LT.1.D-32)GO TO 99
        NP1=NP
        DO 50 KCOL=NP1,NP
        FLEX(NRED,KCOL)=FLEX(NRED,KCOL)/FLEX(NRED,NRED)
        DO 45 J=2,NRED
        NVBL=NP1-J
        L=NVBL+1
        VALUE=FLEX(NVBL,KCOL)
        DO 40 K=L,NRED
        VALUE=VALUE-FLEX(NVBL,K)*FLEX(K,KCOL)
40      CONTINUE
        FLEX(NVBL,KCOL)=VALUE/FLEX(NVBL,NVBL)
45      CONTINUE
50      CONTINUE
```

```
      ELSE
      FLEX(1,2)=FLEX(1,2)/FLEX(1,1)
      END IF
      IF(ITREAC.GT.50) WRITE(IPRINT,100)
100   FORMAT(/////,20X,'NO CONVERGENCE ONTO MOMENTS AFTER FIFTY
     #ITERATIONS      :     STOP',/)
      IF(ITREAC.GT.50) STOP
      RETURN
99    WRITE(IPRINT,101)
101   FORMAT(1H0,10X,'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')
      STOP
      DEBUG SUBCHK
      END
C
C
C      *****************************************
C
C                       D E L T A
C
C      CHECH IF CONVERGENCE ONTO REDUNDANT FORCES
C      HAS OCCURED.IF YES THEN ASSEMBLE FINAL
C      BENDING MOMENT AND AXIAL FORCE DIAGRAMS
C
C      *****************************************
      SUBROUTINE DELTA
      INCLUDE GNLIMST.COMPROC
      DOUBLE PRECISION COMDUM(NRED),XDUM(NRED)
      IFLAG1=0
      IFLAG2=0
      IFLAG=0
      IF(ITREAC.GT.1)THEN
      IFLAG1=1
      IFLAG2=1
      DO 9 I=1,NRED
      IF(DABS(X(I)).GT.1.D-6)THEN
      IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0
      END IF
9     CONTINUE
      END IF
      DO 99 I=1,NRED
      COMDUM(I)=COMPAT(I)
      XDUM(I)=X(I)
99    CONTINUE
      IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1
C      ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT
      DO 10 I=1,NRED
      X(I)=X(I)+FLEX(I,NRED+1)
10    CONTINUE
C      IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE
      IF(IFLAG.EQ.0) GO TO 15
C      CONSTRUCT FINAL BENDING MOMENT DIAGRAM
C
C     CHECK IF ANY REACTANT IS NEAR ZERO .
C     IF SO , SET EQUAL TO ZERO
      IF(NRED.GT.1)THEN
      DUM1=1.E16
      DUM2=1.D-16
C     FIND MINIMUM RECTANT
      DO 1 I=1,NRED
```

```
            IF(DABS(X(I)).LT.DUM1)DUM1=DABS(X(I))
            IF(DABS(X(I))-1.D-6.LE.DUM1)IMIN=I
      1     CONTINUE
      C     FIND MAXIMUM REACTANT
            DO 2 I=1,NRED
            IF(DABS(X(I)).GT.DUM2)DUM2=DABS(X(I))
            IF(DABS(X(I))+1.D-6.GT.DUM2)IMAX=I
      2     CONTINUE
      C     DETERMINE RATIO BETWEEN MAX AND MIN VALUES.
      C     IF MIN/MAX LESS THAN 1.D-9 THEN SET X(MIN) EQUAL TO ZERO.
            RATIO=X(IMIN)/X(IMAX)
            IF(DABS(RATIO).LE.1.D-9)X(IMIN)=0.
            END IF
            DO 12 IE=1,NE
            DO 13 IJ=1,3
            FORCET(IE,IJ)=0.
            FORCET(IE,IJ)=FORCE(1,IE,IJ)
            DO 14 IK=1,NRED
            FORCET(IE,IJ)=FORCET(IE,IJ)+X(IK)*FORCE(IK+1,IE,IJ)
      14    CONTINUE
      13    CONTINUE
      12    CONTINUE
      15    CONTINUE
            RETURN
            DEBUG SUBCHK
            END
      C
      C
      C       ****************************************
      C
      C                     V E L O C
      C
      C         CALCULATE VELOCITY CORRESPONDING TO THE
      C         BENDING MOMENT AND AXIAL FORCES IN THE
      C         STRUCTURE BY VIRTUAL VELOCITY CALC.
      C
      C         ***************************************
            SUBROUTINE VELOC
            INCLUDE GNLMIST.COMPROC
            DOUBLE PRECISION UNITMA(NDF,NN),SUMA,SUMB,SUMC,SUMD,
           #PROD1,PROD2,PROD3,PROD4,PROD5,PROD6,PROD7,PROD8,CONST1,
           #CONST2,CONST3,CONST4,CONST5,CONST6,CONST7,SIG1,SIG2,
           #SIG3,SIG4
            IJ=2
            DO 1 I=1,NDF
            VEL(I,2)=0.
      C     IF THE D.O.F. IS A BOUNDARY CONDITION THEN VELOCITY IS ZER
            IF(IBC(I).EQ.1) GO TO 1
      C      DETERMINE WHETHER D.O.F. IS A ROTATION
            IK=0
            IF(IJ*NF.EQ.I)IK=1
            IF(IK.EQ.1)IJ=IJ+1
            DO 2 IE=1,NE
      C     FOR AN APPLIED UNIT MOMENT , THERE IS A
      C     DISCONTINUITY OF MOMENT AT POINT OF APPLICATION
            IF(I.EQ.3) GO TO 3
            IF(I.EQ.NDF) GO TO 3
            UNITMA(I,NBEAM(IE,1))=0.
            IF(IK.EQ.1.AND.(IJ-1).EQ.NBEAM(IE,1))THEN
```

```
      UNITMA(I,NBEAM(IE,1))=UNITM(I,NBEAM(IE,1))-1.
      UNITM(I,NBEAM(IE,1))=UNITMA(I,NBEAM(IE,1))
      END IF
3     CONTINUE
      SUMA=UNITM(I,NBEAM(IE,2))-UNITM(I,NBEAM(IE,1))
      SUMB=(FORCET(IE,2)-FORCET(IE,1))/RMO
      SUMC=FORCET(IE,1)/RMO+FORCET(IE,3)/RNO
      SUMD=(FORCET(IE,3)/RNO-FORCET(IE,1)/RMO)
      SIG1=1.
      SIG2=1.
      SIG3=1.
      SIG4=1.
      IF((SUMB+SUMC).LT.0) SIG1=-1.
      IF((-SUMB+SUMD).LT.0) SIG2=-1.
      IF(SUMC.LT.0) SIG3=-1.
      IF(SUMD.LT.0) SIG4=-1.
      PROD1=SIG1*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)*(SUMB+SUMC)
      PROD2=SIG1*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)
      PROD3=SIG2*((DABS(-SUMB+SUMD))**EN)*(-SUMB+SUMD)
     #*(-SUMB+SUMD)
      PROD4=SIG2*((DABS(-SUMB+SUMD))**EN)*(-SUMB+SUMD)
      PROD5=SIG3*(DABS(SUMC)**EN)*SUMC*SUMC
      PROD6=SIG3*(DABS(SUMC)**EN)*SUMC
      PROD7=SIG4*(DABS(SUMD)**EN)*SUMD*SUMD
      PROD8=SIG4*(DABS(SUMD)**EN)*SUMD
      CONST1=SUMA/CL(IE)
      CONST2=CL(IE)*CL(IE)/(EN2*SUMB*SUMB)
      CONST3=(SUMC)*CL(IE)*CL(IE)/(EN1*SUMB*SUMB)
      CONST4=CONST3*(SUMD)/(SUMC)
      CONST5=CL(IE)/(EN1*(SUMB))
      CONST6=RNO*EPSI0/(2.*RMO)
      CONST7=EPSI0*0.5*UNITN(I,IE,1)
C      VELOCITY AT D.O.F. I
      VEL(I,2)=VEL(I,2)+CONST6*(CONST1*((PROD1-PROD5)*CONST2
     #-(PROD2-PROD6)*CONST3-(PROD3-PROD7)*CONST2+(PROD4-PROD8)
     #*CONST4)+UNITM(I,NBEAM(IE,1))*((PROD2-PROD6)*CONST5
     #+(PROD4-PROD8)*CONST5))+
     #CONST7*((PROD2-PROD6)*CONST5-(PROD4-PROD8)*CONST5)
      IF(IK.EQ.1.AND.(IJ-1).EQ.NBEAM(IE,1))THEN
      UNITM(I,NBEAM(IE,1))=UNITM(I,NBEAM(IE,1))+1.
      END IF
2     CONTINUE
1     CONTINUE
      RETURN
      DEBUG SUBCHK
      END
C
C
C       ****************************************
C
C                        M O D E C H
C
C       CHECH WHETHER CONVERGENCE ONTO MODE SHAPE
C       HAS OCCURED.IF NOT,THEN NORMALISE CURRENT
C       VELOCITY TO OBTAIN NEW TRIAL MODE SHAPE
C
C       ****************************************
      SUBROUTINE MODECH
      INCLUDE GNLMIST.COMPROC
```

```
C
C      NO. OF ITERATIONS TO DETERMINE MODE
C       IF MODE NOT OBTAINED AFTER FORTY ITERATIONS STOP
       IF(ITMODE.EQ.10)WRITE(IPRINT,100)
100    FORMAT(1H ,//,20X,'MODE NOT FOUND AFTER TEN ITERATIONS :
     # STOP',/)
       IF(ITMODE.EQ.10)STOP
       ITMODE=ITMODE+1
       A=0.
       B=0.
       DISIP(1)=DISIP(2)
       DISIP(2)=0.
C      CHECK FOR CONVERGENCE
C      CALCULATE CURRENT DISSIPATION RATE
       DO 1 I=1,NDF,3
       IK=I+1
       II=INT(FLOAT(I)/NF+0.7)
       DISIP(2)=DISIP(2)+VEL(I,2)*RMASS(II)*PHI(I,1)
       DISIP(2)=DISIP(2)+VEL(IK,2)*RMASS(II)*PHI(IK,1)
1      CONTINUE
C
C       NORMALISE VELOCITIES FOR NEW MODE SHAPE
C
       DUM=0.
       DO 5 I=1,NDF,3
       II=INT(FLOAT(I)/NF+0.7)
       DUM=DUM+VEL(I,2)*VEL(I,2)*RMASS(II)
       IK=I+1
       DUM=DUM+VEL(IK,2)*VEL(IK,2)*RMASS(II)
5      CONTINUE
C      ANORM=DSQRT(DUM/RKINET)
       RLAMDA=SQRT(DUM)
C
       DO 2 I=1,NDF
       PHI(I,2)=VEL(I,2)/RLAMDA
       PHI(I,1)=PHI(I,2)
2      CONTINUE
C       CHECK CHANGE IN DISSIPATION RATE
       IDISIP=0
       IF(ABS(DISIP(2)/DISIP(1)-1.D0).LT.5.D-2)IDISIP=1
C
C       OBTAIN AMPLITUDE OF VELOCITY BY PERFORMING MOMENTUM BALANC
       IF(IDISIP.EQ.1.AND.T.LT.1.D-9.AND.IRND.EQ.0)THEN
       DO 3 I=1,NDF
       II=INT(FLOAT(I)/NF+0.7)
       IJ=(INT(FLOAT(I)/NF+0.001))*NF
       IF(I.NE.IJ)THEN
       A=A+VEL(I,1)*RMASS(II)*PHI(I,1)
       B=B+PHI(I,1)*RMASS(II)*PHI(I,1)
       END IF
3      CONTINUE
       AMP(1)=A/B
       AMP(2)=AMP(1)
       DO 4 I=1,NDF
       VEL(I,2)=PHI(I,1)*AMP(2)
4      CONTINUE
       END IF
C
        IF(DISPL.EQ.'LARGE'.AND.IRND.EQ.0.AND.T.GT.0
```

```
     #.AND.IDISIP.EQ.1)THEN
      DO 6 I=1,NDF
      II=INT(FLOAT(I)/NF+0.7)
      IJ=(INT(FLOAT(I)/NF+0.001))*NF
      IF(I.NE.IJ)THEN
      A=A+VMODE(I,2)*RMASS(II)*PHI(I,1)
      B=B+PHI(I,1)*RMASS(II)*PHI(I,1)
      END IF
 6    CONTINUE
      AMP(2)=A/B
      END IF
C
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C      ******************************************
C
C                     I N M O D E
C
C      FORMULATE EXPLICIT EXPRESSIONS FOR THE TIME
C      FUNCTION , ITS DIRIVATIVE  , VELOCITY AND
C      DISPLACEMENT ONCE INSTANTANEOUS MODE HAS
C                  BEEN FOUND
C
C      ******************************************
      SUBROUTINE INMODE
      INCLUDE GNLMIST.COMPROC
      DOUBLE PRECISION A1,A2
C
C     CALCULATE FACTOR K IN EXPRESSION FOR T(T)
C
      TIME=DT
      IF(IRND.EQ.0)TIME=0.
      POWA=1/EN
      POWB=(EN-1.)/EN
      POWC=1./POWB
      POWD=(2.*EN-1.)/(EN-1.)
      RK=1./((ABS(RLAMDA)**POWA)*(AMP(2)**POWB))
C
C     CALCULATE EXPRESSION FOR T(T)
      FACT=POWB*RK*TIME
C     CHECK IF NEAR TOTAL TIME
      IF(FACT.GT.1)THEN
      FACT=1.
      END IF
C
      TT(2)=(1.-FACT)**POWC
C
C      DIRIVATIVE OF T(T)
C
      IF(IRND.EQ.0)DTTDT(2)=-(TT(2)**POWA)*RK
      IF(IRND.GT.0)THEN
      DTTDT(2)=(DTTDT(2)-(TT(2)**POWA)*RK)/2.
      END IF
C
      IF(IRND.GT.0)TT(2)=TT(1)+(DTTDT(1)+DTTDT(2))*DT/2.
```

```
C
C        CALCULATE EXPRESSION FOR MODE VELOCITIES,DISPLACEMENTS
C
         DO 3 I=1,NDF
         IF(IRND.EQ.0)VMODE(I,2)=AMP(2)*PHI(I,1)*TT(2)
         IF(IRND.GT.0)THEN
         VMODE(I,2)=(VMODE(I,2)+AMP(2)*PHI(I,1)*TT(2))/2.
         END IF
3        CONTINUE
C
C         CALCULATE EXPRESSION FOR MOMENTS AND AXIAL FORCES
C
         DO 4 IE=1,NE
         DO 5 J=1,3
         DUM=RMO
         IF(J.EQ.3)DUM=RNO
         FORMOD(IE,J)=(((AMP(2)*TT(2)/RLAMDA)**POWA)
        #*FORCET(IE,J))/DUM
5        CONTINUE
C
C        IF AXIAL FORCES LARGE SET FLAG TO REQUEST NEW MATCHING
C        FACTOR
C        IF MEMBRANE FACTOR ALREADY CALCULATED THEN SKIP
         IF(MATCHA.NE.-1.OR. MATCHA.EQ.0)THEN
         IF(DABS(FORMOD(IE,3)).GT.0.2)MATCHA=1
         END IF
C
4        CONTINUE
C
C
         RETURN
         DEBUG SUBCHK
         END
C
C
C         ******************************************
C
C                         M A T C H
C
C         DETERMINES MATCHING FACTOR ON SLOPE ALONE
C
C         ******************************************
         SUBROUTINE MATCH
         INCLUDE GNLMIST.COMPROC
         DMATCH=1.
         DUM=0.
C        LOCATE MAXIMUM BENDING MOMENT OR AXIAL FORCE IN STRUCTURE
C
         N1=1
         N2=2
         IF(MATCHA.EQ.1)THEN
         N1=3
         N2=3
         END IF
C
         DO 1 IE=1,NE
         DO 2 I=N1,N2
         IF(DABS(FORMOD(IE,I)).GE.DUM)DUM=ABS(FORMOD(IE,I))
2        CONTINUE
```

```
    1 CONTINUE
C
      AMPD=AMP(1)**(1./EN)
      RMMAX=DUM/AMPD
      IF(RMATCH.GT.1.01)THEN
      RMMAX=RMMAX*RMATCH
      DMATCH=RMATCH
      END IF
C
      RKMAX=AMP(1)*((RMMAX)**EN)
      RMATCH=(1.+(RKMAX)**(1./EN))/(RKMAX**(1./EN))
C
C     CALCULATE TOTAL TIME
C
      TF=(EN/(EN-1.))/(RK*RMATCH)
C
C     MATCH YIELD MOMENT AND AXIAL YIELD STRESS
      RMO=RMO*RMATCH/DMATCH
      RNO=RNO*RMATCH/DMATCH
C
      IF(MATCHA.EQ.1)THEN
      WRITE(IPRINT,3)RMATCH
    3 FORMAT(1H ,///,20X,'REVISED MATCHING FACTOR FOR MEMBRANE
     # ACTION IS : ',E11.6)
      MATCHA=-1
      END IF
C
      RETURN
      DEBUG SUBCHK
      END
C
C
```

```
C        ****************************************
C
C                    U P D A T E
C
C        UPDATES GEOMETRY OF THE STRUCTURE AFTER
C        EACH TIME INCREMENT AND STORES PREVIOUS
C        VELOCITIES AND DISPLACEMENTS.
C
C        ****************************************
         SUBROUTINE UPDATE
         INCLUDE GNLMIST.COMPROC
C
         DO 3 I=1,NDF
         U(I,2)=(VMODE(I,1)+VMODE(I,2))*DT/2.
 3       CONTINUE
         IOUT=1
         DO 5 I=1,NDF,3
         IJ=I+1
         IF(DABS(U(I,2)).GT.1.D-5)THEN
         IF(DABS(U(I,1)/U(I,2)-1.D0).GT.1.D-2)IOUT=0
         END IF
         IF(DABS(U(IJ,2)).GT.1.D-5)THEN
         IF(DABS(U(IJ,1)/U(IJ,2)-1.D0).GT.1.D-2)IOUT=0
         END IF
 5       CONTINUE
         DO 4 I=1,NDF,3
         II=INT(FLOAT(I)/NF+0.7)
         COORDX(II)=COORDX(II)-U(I,1)+U(I,2)
         U(I,1)=U(I,2)
         IJ=I+1
         COORDY(II)=COORDY(II)-U(IJ,1)+U(IJ,2)
         U(IJ,1)=U(IJ,2)
         IK=IJ+1
         U(IK,1)=U(IK,2)
 4       CONTINUE
C        IF CONVERGENCE ONTO MODE IN LARGE DISPLACEMENT ANALYSIS
C        HAS NOT OCCURRED AFTER FIVE ITERATIONS THEN STORE CURRENT
C        RESULT.
         IF(IRND.GT.5)WRITE(IPRINT,7)
 7       FORMAT(1H ,///,20X,'MODE IN LARGE DISPLACEMENT ANALYSIS NOT
        #FOUND AFTER FIVE ITERATIONS TO ONE PERCENT VARIATION :
        #CONTINUE')
         IF(IRND.GT.5)IOUT=1
C
         IF(IOUT.EQ.1)THEN
         DO 2 I=1,NDF
         UMODE(I)=UMODE(I)+U(I,1)
         U(I,1)=0.
 2       CONTINUE
         END IF
C
         RETURN
         DEBUG SUBCHK
         END
C
C
```

```
C      ****************************************
C
C                    S T O R E
C
C       STORES INTERMEDIATE VALUES OF THE TIME
C       FUNCTION,ITS DIRIVATIVE AND THE CURRENT
C       VELOCITY ESTIMATE IN THE INSTANTANEOUS
C                 MODE ALGORITHM
C
C       ***************************************
       SUBROUTINE STORE
       INCLUDE GNLMIST.COMPROC
C
C
       IF(T.LT.1.E-9)DT=TF/RINT
       TT(1)=TT(2)
       DTTDT(1)=DTTDT(2)
C
       DO 1 I=1,NDF
       VMODE(I,1)=VMODE(I,2)
 1     CONTINUE
C
       RETURN
        DEBUG SUBCHK
       END
C
C
C      *************************************
C
C                  O U T P U T
C
C       OUTPUTS RESULTS OF ANALYSIS AT
C        REQUESTED TIME INTERVALS
C
C      *************************************
       SUBROUTINE OUTPUT
       INCLUDE GNLMIST.COMPROC
       ICOUNT=0
       IF(T.LT.1.D-36)THEN
C
C      NO. OF ITERATIONS TO DETERMINE MODE SHAPE
       WRITE(IPRINT,5)ITMODE
 5     FORMAT(1H1,' NO OF ITERATIONS TO DETERMINE MODE :',I3)
       WRITE(IPRINT,77)TF,RMATCH
 77    FORMAT(1H ,/,'  INITIAL TOTAL TIME ESTIMATE :',
      #E17.7,10X,'MATCHING FACTOR IS :',E17.7,/)
      # FACTOR IS :',E17.7,/)
       WRITE(IPRINT,6)
 6     FORMAT(1H ,18X,'   MODE SHAPE (T=0)',//,
      #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
       DO 4 I=1,NDF-2,3
       II=INT(FLOAT(I)/NF+0.7)
       WRITE(IPRINT,3)II,PHI(I,1),PHI(I+1,1),PHI(I+2,1)
 4     CONTINUE
       WRITE(IPRINT,1)
 1     FORMAT(1H ,20X,'MODE VELOCITY (T=0)',//,
      #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
       DO 2 I=1,NDF-2,3
       II=INT(FLOAT(I)/NF+0.7)
```

```
      WRITE(IPRINT,3)II,VMODE(I,2),VMODE(I+1,2),VMODE(I+2,2)
3     FORMAT(1H ,I3,3X,E13.6,6X,E13.6,9X,E13.6,/)
2     CONTINUE
      WRITE(IPRINT,7)
7     FORMAT(1H ,/,20X,'MOMENTS AND AXIAL FORCES (T=0)',//
     #,'ELEMENT',8X,
     #'MOMENT(A)',8X,'MOMENT(B)',8X,'AXIAL FORCE',/)
      DO 8 IE=1,NE
      WRITE(IPRINT,9)IE,FORMOD(IE,1),FORMOD(IE,2),FORMOD(IE,3)
9     FORMAT(1H ,2X,I2,8X,E13.6,4X,E13.6,4X,E13.6,/)
8     CONTINUE
C     OUTPUT INITIAL ELEMENT LENGHTS
      WRITE(IPRINT,34)
34    FORMAT(1H ,//,' ELEMENT',8X,'INITIAL ELEMENT LENGTH',/)
      DO 33 IE=1,NE
      WRITE(IPRINT,32)IE,CL(IE)
32    FORMAT(1H ,2X,I2,8X,E13.6,/)
33    CONTINUE
C
      ELSE
      AMPERC=100.*AMP(2)/AMP(1)
      WRITE(IPRINT,11)T,DT,AMPERC
11    FORMAT(1H ,/,'TIME IS ',E11.6,15X,'TIME INCREMENT IS'
     #,EII.6,15X,'VELOCITY AMPLITUDE(PERCENT)  IS ',F7.3,/)
      IF(DISPL.EQ.'LARGE')THEN
      WRITE(IPRINT,60)
60    FORMAT(1H ,18X,'   MODE SHAPE',//,'
     #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
      DO 40 I=1,NDF-2,3
      II=INT(FLOAT(I)/NF+0.7)
      WRITE(IPRINT,3)II,PHI(I,1),PHI(I+1,1),PHI(I+2,1)
40    CONTINUE
      END IF
      WRITE(IPRINT,12)
12    FORMAT(1H ,20X,'MOMENTS AND AXIAL FORCES',//,'
     #'ELEMENT',8X,'MOMENT(A)',
     #8X,'MOMENT(B)',8X,'AXIAL FORCE',/)
      DO 13 IE=1,NE
      WRITE(IPRINT,9)IE,FORMOD(IE,1),FORMOD(IE,2),FORMOD(IE,3)
13    CONTINUE
      WRITE(IPRINT,14)
14    FORMAT(1H ,/,26X,'VELOCITY',//,
     #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
     #ON',/)
      DO 15 I=1,NDF-2,3
      II=INT(FLOAT(I)/NF+0.7)
      WRITE(IPRINT,3)II,VMODE(I,2),VMODE(I+1,2),VMODE(I+2,2)
15    CONTINUE
      WRITE(IPRINT,16)
16    FORMAT(1H ,/,26X,'DISPLACEMENTS',//,
     #'NODE',6X,'X',17X,'Y',19X,
     #'ROTATIONS',/)
      DO 17 I=1,NDF-2,3
      II=INT(FLOAT(I)/NF+0.7)
      WRITE(IPRINT,3)II,UMODE(I),UMODE(I+1),UMODE(I+2)
17    CONTINUE
C
      END IF
C
```

```
C       STORE CURRENT COORDINATES
        IF(PICT.EQ.'PLOT')THEN
        IPLTS=IPLTS+1
        DO 21 I=1,NN
        XCOORD(I,IPLTS)=COORDX(I)
        YCOORD(I,IPLTS)=COORDY(I)
 21     CONTINUE
        END IF
C
        RETURN
        DEBUG SUBCHK
        END
C
C
C       *****************************************
C
C                     P I C T U R
C
C        PLOTS INITIAL STRUCTURE AND SUBSEQUENT
C        DEFORMED SHAPE THROUGHOUT TIMESPAN OF
C                   DEFORMATION
C
C        ****************************************
        SUBROUTINE PICTUR
        INCLUDE GNLMIST.COMPROC
        DOUBLE PRECISION XMAX,YMAX
C
        CALL NEWPAG
        CALL PAGSIZ(20.5,29.)
        CALL PLOT(2.0,5.0,-3)
C
        XMAX=-999.D0
        YMAX=-999.D0
        DO 22 J=1,IPLTS
        DO 1 I=1,NN
        IF(DABS(XCOORD(I,J)).GE.XMAX)XMAX=DABS(XCOORD(I,J))
        IF(DABS(YCOORD(I,J)).GE.YMAX)YMAX=DABS(YCOORD(I,J))
 1      CONTINUE
 22     CONTINUE
        RMAX=DMAX1(XMAX,YMAX)
        FX=18.0/RMAX
        FY=18.0/RMAX
C
        DO 21 J=1,IPLTS
        DO 2 I=1,NN
        XCOORD(I,J)=XCOORD(I,J)*FX
        YCOORD(I,J)=YCOORD(I,J)*FY
 2      CONTINUE
        CALL PLOT(0.,0.,3)
        DO 3 I=1,NN
        CALL PLOT(XCOORD(I,J),YCOORD(I,J),2)
        CALL SYMBOL(XCOORD(I,J),YCOORD(I,J),0.2,11,0.,-1)
 3      CONTINUE
 21     CONTINUE
        RETURN
        DEBUG SUBCHK
        END
C
C
```

```
C       ***************************************
C
C            C O M M O N   B L O C K
C
C        CONTAINS ALL VARIABLES REQUIRED IN
C                 EACH SUBROUTINE
C
C       ***************************************
COMPROC PROC
      PARAMETER NE=12
      PARAMETER NN=13
      PARAMETER NRED=3
      PARAMETER INTER=3
      PARAMETER NF=3
      PARAMETER IREAD=8
      PARAMETER IPRINT=5
      PARAMETER NDF=NN*NF
      PARAMETER KINT=3*INTER
      COMMON/BLK1/NBEAM(NE,2),COORDX(NN),COORDY(NN),
     #PARDIF(NRED,NRED),RMASS(NN),IBC(NDF),RELEAS(NRED),
     #COMPAT(NRED),UNITM(NDF,NN),CL(NE),UNITN(NDF,NE,1),
     #SSIN(NE),CCOS(NE),FORCE(NRED+1,NE,3),DTTDT(2),
     #FLEX(NRED,NRED+1),PHI(NDF,2),DISIP(2),RLOAD(NDF),
     #VEL(NDF,2),TT(2),X(NRED),FORCET(NE,3),AMOM(NRED+1),
     #ANORM1(NRED+1),ANORM2(NRED+1),VMODE(NDF,2),TF,T,DT,
     #UMODE(NDF),FORMOD(NE,3),RINT,RLAMDA,U(NDF,2),
     #XCOORD(NN,15),YCOORD(NN,15),UMODD(NDF)
      COMMON/BLK2/RMO,RNO,EPSIO,EN,EN1,EN2,STADET,TITLE,
     #PICT,DISPL,RK,YSTRS,RMATCH,AMP(2),RKINET,POWB,POWC
      COMMON/BLK3/IFLAG,IDISIP,ITREAC,ITMODE,NNORM,IOUT,
     #IRND,NDIV,ICOUNT,NDIVA,IPLTS,MATCHA
      CHARACTER STADET*4
      CHARACTER TITLE*80
      CHARACTER DISPL*5
      CHARACTER PICT*4
      DOUBLE PRECISION COORDX,COORDY,PARDIF,RMASS,COMPAT,
     #FORCET,DUM,A,B,UNOTM,CL,UNITN,SSIN,CCOS,FORCE,FLEX,
     #PHI,RLOAD,VEL,X,AMOM,DTTDT,RK,T,TT,ANORM1,ANORM2,RMO,
     #RNO,EPSIO,EN,EN1,EN2,VMODE,UMODE,FORMOD,AMP,ANORM,FACT
      INTEGER RELEAS
 END
```

# APPENDIX C

## DAGNVS program listing

```
C       ****************************
C
C       D R I V E R    R O U T I N E
C
C       ****************************
        INCLUDE DAGNVS.COMPROC
C       INPUT DATA
        CALL INPUT
C       DISPLAY DATA
        CALL DATA
        CALL OUTPUT
C
  5     CONTINUE
        ITREAC=0
C
C       NNORM=1 INDICATES ITERATION ONTO NEW MODE SUCCESSFUL
        NNORM=0
C        MCOUNT COUNTS NO. OF ITERATIONS REQUIRED TO OBTAIN NEW MOD
        MCOUNT=0
C       ISTEP IS NO. OF TIME STEP
        ISTEP=ISTEP+1
C       ICOUNT COUNTS NO. OF TIME STEPS UNTIL NCOUNT=NDIV,THEN
C       THEN OUTPUT REQUESTED AND ICOUNT INITIALISED.
        ICOUNT=ICOUNT+1
C
C       INCREMENT TIME
        T=T+DT
C
  10    CONTINUE
C        COUNTER FOR LARGE DISPLACEMENT MODE ITERATIONS
        MCOUNT=MCOUNT+1
        IF(MCOUNT.EQ.10)THEN
        WRITE(IPRINT,11)
 11     FORMAT(1H ,///,20X,' NO CONVERGENCE ONTO NEW MODE IN LARGE
       #DISPLACEMENT MODE ANALYSIS : STOP')
        STOP
        END IF
C        DETERMINE STATICALLY ADMISSIBLE SETS (ONE FOR EACH D.O.F.)
C
         CALL STAT
C
C       IF NON-MODAL SOLUTION REQUIRED (ISYM=1) SKIP MODE SOLUTION
C       ALGORITHM EXCEPT AT T=0 SO AS TO DETERMINE TOTAL TIME
C        ESTIMATE
        IF(T.LT.1.D-9.AND.ISYM.EQ.1.OR.ISYM.EQ.0)THEN
C
C        SET UP SELF STRESS SYSTEMS IF STRUCTURE IS STATICALLY
C        INDETERMINATE
        IF(NRED.NE.4)THEN
C
        MODES=1
        CALL SLFSTR
C
        END IF
  2     CONTINUE
C
C        DETERMINE CURRENT LOADING (F=MASS*PHI)
C
        CALL LOAD
```

```
C
  1     CONTINUE
C
C       IF STRUCTURE IS STATICALLY DETERMINATE (NRED=4)
C       DON'T NEED TO ITERATE FOR REDUNDANTS
C
        IF(NRED.EQ.4)GO TO 3
C
C        ZERO ALL ARRAYS IN ITERATIVE PROCEDURE TO FOLLOW
C
        CALL ZERO
C
C       EVALUATE COMPATIBILITY EQUATIONS
C
        CALL COMEQU
C
C        CALCULATE PARTIAL DIRIVATIVES OF COMPATIBILITY EQUATIONS
C
        CALL PDIFF
C
C        INVERT MATRIX OF PARTIAL DIRIVATIVES FOR NEWTON-RAPHSON
C        METHOD
C
        CALL PIVOT
C
C        CHECK MAGNITUDE OF PERTURBATION IN VALUE OF REDUNDANTS
C
        CALL DELTA
C
C       IF SMALL(IFLAG=1)CALCULATE VELOCITIES
C       IF  LARGE(IFLAG=0) THEN ITERATE
C
        IF(IFLAG.EQ.0)GO TO 1
C
  3     CONTINUE
C
C       DETERMINE VELOCITIES ASSOCIATED WITH CURRENT STRESS STATE
C
        CALL VELOC
C
C       CHECK IF ALGORITHM HAS CONVERGED ONTO A MODE
C       IF IN MODE(IDISIP=1)             IF NOT(IDISIP=0)SO ITERATE
C
        CALL MODECH
C
        IF(IDISIP.EQ.0)GO TO 2
C
  4     CONTINUE
C
C        IF MEMBRANE SOLUTION PREDOMINATES CALCULATE NEW MATCHING
C        FACTOR
        IF(MATCHA.EQ.1)CALL MATCH
C
C        MODE SOLUTION MAY NOW COMMENCE
C
C
        CALL INMODE
C
C        IF IN MODE SOLUTION UPDATE CURRENT DISPLACEMENTS
```

```
      IF(INMSOL.EQ.1)THEN
      CALL STORE
C      IS ITERATION FOR NEW MODE REQUIRED ?
      IF(NNORM.EQ.0)THEN
      GO TO 10
      ELSE
      CALL UPDATE
C     CHECK IF OUTPUT REQUIRED
      GO TO 9
      END IF
      END IF
C
      END IF
C      IF T=0 , DETERMINE MATCHING FACTOR,ESTIMATE TOTAL TIME TF,
C      AND TIME INCREMENT DT. THEN DETERMINE INITIAL BODY FORCES.
C
      IF(T.LT.1.D-9)THEN
      IF(ISYM.EQ.0)CALL MATCH
C
C      CALCULATE INITIAL EQUILIBRATING FORCES
C
      MODES=0
C
      CALL SLFSTR
C
13     CONTINUE
      ITREAC=0
 6    CONTINUE
C
C      SET UP COMPATIBILITY EQUATIONS AND PARTIAL DIRIVATIVE
C
       CALL COMDIF
C
C     HAVE BODY FORCES BEEN FOUND ? YES=1  :  NO=0
C
      IF(IFLAG.EQ.1)GO TO 7
C
C      INVERT MATRIX OF PARTIAL DIRIVATIVES
C
      CALL PIVOT
C
      IF(IFLAG.EQ.0)GO TO 6
C
 7    CONTINUE
C
C     MATCH SYSTEM IF DIRECT ANALYSIS AND MATCHING FACTOR FKNOWN
      IF(ISYM.EQ.1)CALL VMATCH
      IF(IMATCH.EQ.0.AND.ISYM.EQ.1)GO TO 13
      END IF
C
C     ONCE BODY FORCES HAVE BEEN OBTAINED , ACCELERATIONS ARE
 C      CALCULATED
C
      CALL ACCAXC
C     COMMENCE IMPLICIT FORWARD INTEGRATION SCHEME
C    REVISE STATICALLY ADMISSIBLE SET
C
 8    CONTINUE
C
```

```
          ITREAC=0
          IRND=IRND+1
          IF(IRND.GT.1)THEN
C
          CALL STAT
C
          MODES=0
C
          CALL SLFSTR
C
          CALL COMDIF
C
          END IF
C
C          IMPLICIT TIME INTEGRATION SCHEME
C
          CALL IMPLIC
          CALL PIVOT
          CALL REVISE
C
C          IS DELTA X SUFFICIENTLY SMALL ?
C         IOUT=1 : YES                  IOUT=0 : NO
          IF(IRND.EQ.100)THEN
          WRITE(IPRINT,103)
 103      FORMAT(///,20X,'NO CONVERGENCE IN EQUILIBRIUM ITERATIONS :
         #DECREASE TIME STEP',/)
          STOP
          END IF
C
          IF(IOUT.EQ.0)GO TO 8
C
          IRND=0
C
C          CHECK IF DIRECT SOLUTION HAS CONVERGED INTO MODE SOLUTION
          IF(INMSOL.EQ.0)THEN
          CALL CHECK
          END IF
C          IF MODE REACHED OUTPUT RESULTS AT END OF DIRECT SOLUTION
C          PHASE
          IF(INMSOL.EQ.1.AND.NNORM.EQ.0)CALL OUTPUT
C
   9      CONTINUE
C
C          IS OUTPUT REQUIRED DURING ANALYSIS PHASE
          IF(ICOUNT.EQ.NDIV)THEN
C
          CALL OUTPUT
          ICOUNT=0
          END IF
C
C          CHECK IF VELOCITY NEAR ZERO (IS STRUCTURE AT REST ?)
C          THIS CHECK IS PERFORMED BY COMPARING THE CURRENT
C          MOMENTUM WITH THE ORIGINAL MOMENTUM (VEL*MASS)
C
          IFLAGA=1
          RMMO=0
          RMMC=0
C
          DO 12 I=2,NN
```

```
      IF(NRED.NE.4.AND.I.EQ.NN)GO TO 12
      II=3*I-2
      IJ=II+1
      RMMO=RMMO+(DABS(VINIT(II))+DABS(VINIT(IJ)))*RMASS(I)
      RMMC=RMMC+(DABS(VEL(II,1))+DABS(VEL(IJ,1)))*RMASS(I)
C
12    CONTINUE
C
      IF((RMMC/RMMO).GT.0.05)GO TO 5
C
      CALL OUTPUT
C
      IF(PICT.EQ.'PLOT')CALL PICTUR
      STOP
      END
C
C
C
C
C     **********
C
C      I N P U T
C
C     INPUT DATA
C
C     **********
      SUBROUTINE INPUT
      INCLUDE DAGNVS.COMPROC
      DIMENSION IIBC(NF),VELDUM(NF)
100   FORMAT( )
101   FORMAT(A4)
102   FORMAT(A80)
103   FORMAT(A5)
      READ(IREAD,102)TITLE
      READ(IREAD,101)PICT
      READ(IREAD,100)PMATCH
      READ(IREAD,100)HH,BB,YSTRS,EPSIO,EN
      SET PARAMETER TO REQUEST DIRECT ANALYSIS
      THIS PROGRAM HAS COMBINED MODE/DIRECT ANALYSIS FACILITY
      BUT ITS USE HAS PROVED INEFFICIENT.
      ISYM=1
C     NODE INCIDENCES (NUMBERING OF ELEMENT ENDS)
      DO 11 IE=1,NE
      READ(IREAD,100)(NBEAM(IE,I),I=1,2)
11    CONTINUE
C     COORDINATES OF ORDERED NODES (X,Y)
      DO 12 I=1,NN
      READ(IREAD,100)COORDX(I),COORDY(I)
12    CONTINUE
C     BOUNDARY CONDITIONS
C     FREEDOM : 0
C     FIXITY  : 1
      DO 2 I=1,NDF
      IBC(I)=0
2     CONTINUE
3     READ(IREAD,100)N1
      IF(N1.LT.0) GO TO 4
      READ(IREAD,100)(IIBC(J),J=1,NF)
      II=NF*(N1-1)
```

```
      DO 5 I=1,NF
      II=II+1
      IBC(II)=IIBC(I)
    5 CONTINUE
      GO TO 3
    4 CONTINUE
C      IS STRUCTURE DETERMINATE OR HYPERSTATIC ?
      IF(NRED.EQ.4) THEN
C      IF DETERMINATE , IS IT A CANTILEVER OR SIMPLY SUPPORTED ?
      READ(IREAD,101)STADET
      ELSE
C     IF HYPERSTATIC , NRED DEGREES OF FREEDOM MUST BE RELEASED
C     TO OBTAIN A DETERMINATE STRUCTURE . IS RESULTING STUCTURE
C     A CANTILEVER OR A SIMPLY SUPPORTED STRUCTURE ?
      READ(IREAD,101)STADET
C     DEGREES OF FREEDOM WHICH ARE RELEASED
      READ(IREAD,100)(RELEAS(I),I=1,NRED)
      END IF
C     READ IN MASS VECTOR
      READ(IREAD,100)(RMASS(I),I=1,NN)
C     READ IN INITIAL VELOCITY
    6 READ(IREAD,100)N1,N2
      IF(N1.LT.0)GO TO 7
      READ(IREAD,100)(VELDUM(I),I=1,NF)
      II=NF*(N1-1)
      JJ=N2-N1+1
      DO 8 J=1,JJ
      DO 8 I=1,NF
      II=II+1
      VEL(II,1)=VELDUM(I)
    8 CONTINUE
      GO TO 6
    7 CONTINUE
C     STATE HOW MANY TIME INTERVALS IN MODE SOLUTION ARE REQUIRED
C     AND NUMBER OF OUTPUTS REQUIRED
      READ(IREAD,100)RINT,NDIV
      READ(IREAD,100)XXDUM
C     CALCULATE YIELD MOMENT  AND AXIAL YIELD STRESS
      AREA=HH*BB
      RMO=AREA*HH*YSTRS/4.
      RNO=AREA*YSTRS
C
C     SET INITIAL MODE SHAPE (GUESS) EQUAL TO INITIAL VELOCITY
      DO 1 I=1,NDF
      PHI(I,1)=VEL(I,1)
      DUMVEL(I)=VEL(I,1)
    1 CONTINUE
C      CALCULATE NORMALISATION CONSTANT AND INITIAL DISSIPATION
 C       RATE
      DO 10 I=1,NDF,3
      II=INT(FLOAT(I)/NF+0.7)
      IK=I+1
      DISIP(2)=DISIP(2)+VEL(I,1)*RMASS(II)*PHI(I,1)
      DISIP(2)=DISIP(2)+VEL(IK,1)*RMASS(II)*PHI(IK,1)
   10 CONTINUE
C
C     IF STRUCTURE INDETERMINATE(NRED NOT EQUAL TO 4),DUMMY VALUES
C     ARE ASSIGNED TO REACTANTS (X) TO AVOID POSSIBLE DIVISION BY
C     ZERO IN FORMULATION OF COMPATIBILITY EQUATIONS . (THAT
```

```
C       SITUATION ARISES IF STATICALLY ADMISSIBLE B.M.D. IS ZERO)
        IF(NRED.NE.4)THEN
        DO 9 I=1,NRED
        X(I)=1.0
9       CONTINUE
        END IF
C
C       ASSIGN DUMMY VALUES TO BODY FORCES
C
        DO 13 I=1,2*NN
        XX(I)=XXDUM
13      CONTINUE
C
C       CALCULATE NO. OF BODY FORCES TO BE DETERMINED
C
        IF(STADET.EQ.'CANT'.AND.NRED.NE.4)NINV=2*NN-1
        IF(STADET.EQ.'SIMP'.AND.NRED.EQ.4)NINV=2*(NN-1)-1
        IF(STADET.EQ.'CANT'.AND.NRED.EQ.4)NINV=2*(NN-1)
C
C       SET UP ARRAY CONTAINING DEGREE OF FREEDOM OF REACTIONS
        IF(STADET.EQ.'CANT')THEN
        DO 14 I=1,2*(NN-1)
        I1=INT(FLOAT(I)/2+0.55)
        I2=NF+I+I1-1
        ISTAT(I)=I2
14      CONTINUE
        END IF
C
        IF(STADET.EQ.'CANT'.AND.NRED.NE.4)ISTAT(2*NN-1)=NDF
        IF(STADET.EQ.'SIMP')THEN
        DO 15 I=1,2*NN-3
        I1=INT(FLOAT(I)/2+0.55)
        I2=NF+I1+I-1
        IJ=I
        IF(NRED.NE.4)IJ=I+1
        ISTAT(IJ)=I2
15      CONTINUE
        IF(NRED.NE.4)THEN
        ISTAT(1)=3
        ISTAT(2*NN-1)=NDF
        END IF
        END IF
C
C        STORE INITIAL COORDINATES
C
        DO 66 I=1,NN
        XCOORD(I)=COORDX(I)
        YCOORD(I)=COORDY(I)
66      CONTINUE
C
        EN1=EN+1
        EN2=EN1+1
C
        RETURN
        DEBUG SUBCHK
        END
C
C
```

```
C       ****************************************
C
C                         D A T A
C
C       DISPLAYS ALL INPUT DATA FOR VERIFICATION
C
C       ****************************************
        SUBROUTINE DATA
        INCLUDE DAGNVS.COMPROC
        WRITE(IPRINT,1)TITLE
1       FORMAT(1H1,5X,80('*'),/10X,A80,/,6X,80('*'),/)
        WRITE(IPRINT,17)
17      FORMAT(1H ,/,20X,'LARGE DISPLACEMENT ANALYSIS',/)
        WRITE(IPRINT,13)EN,RMO,RNO,EPSIO,YSTRS,HH,BB
13      FORMAT(1H ,5X,///,' MATERIAL ASSUMPTIONS',//,'  HOMOGENEOUS
       #VISCOUS WITH POWER N =',F7.3,/,2X,'YIELD MOMENT =',
       #F11.6,/,2X,'AXIAL YIELD STRENGTH =',E10.4,/,
       #2X,'INITIAL STRAIN RATE =',E10.4,/,2X,'YIELD
       # STRESS =',E10.4,/,2X, 'SECTION DEPTH =',E15.6,/,2X,
       #'SECTION WIDTH =',E15.6,/)
        WRITE(IPRINT,2)NE,NN
2       FORMAT(1H ,/,3X,'NUMBER OF ELEMENTS  :',I3,/,3X,'NUMBER OF
       #NODES    :',I3,/)
        WRITE(IPRINT,3)
3       FORMAT(//,6X,' COORDINATES OF NODES',//,
       #' NODE',10X,'X',12X,'Y')
        DO 110 I=1,NN
        WRITE(IPRINT,4)I,COORDX(I),COORDY(I)
4       FORMAT(1H ,I3,1X,2(2X,F11.5))
110     CONTINUE
        WRITE(IPRINT,5)
5       FORMAT(//,' BOUNDARY CONDITIONS : 0=FREEDOM , 1=FIXITY',
       #//,' NODE',3X,'X',3X,'Y',3X,'ROTATION')
        WRITE(IPRINT,6)1,IBC(1),IBC(2),IBC(3)
C        IF STRUCTURE STATICALLY DETERMINATE NO BOUNDARY
C        CONDITIONS AT LAST NODE.
        IF(NRED.NE.4)THEN
        WRITE(IPRINT,6)NN,IBC(NDF-2),IBC(NDF-1),IBC(NDF)
        END IF
6       FORMAT(1H ,I3,4X,I1,3X,I1,6X,I1)
        WRITE(IPRINT,11)
11      FORMAT(1H ,///,5X,'LUMPED MASS PER
       #NODE',//,6X,'NODE',10X,'MASS',/)
        DO 111 I=1,NN
        WRITE(IPRINT,12)I,RMASS(I)
12      FORMAT(1H ,5X,I2,6X,E11.4,/)
111     CONTINUE
        IF(NRED.NE.4)THEN
        IF(STADET.EQ.'CANT')WRITE(IPRINT,9)(RELEAS(I),I=1,NRED)
9       FORMAT(1H ,//,' STRUCTURE CANTILEVERED BY RELEASING
       #RESTRAINTS AT D.O.F. ',5(1X,I2))
        IF(STADET.EQ.'SIMP')WRITE(IPRINT,10)(RELEAS(I),I=1,NRED)
10      FORMAT(1H ,//,'  STRUCTURE MADE SIMPLY SUPPORTED BY
       #RELEASED RESTRAINTS AT D.O.F. ',5(3X,I2))
        END IF
        WRITE(IPRINT,14)
14      FORMAT(1H ,///,10X,'INITIAL VELOCITY',//,'
       #' NODE',4X,'X',12X,'Y',9X,'ROTATION')
C
```

```
C
         DO 112 I=1,NDF,3
         II=INT(I/NF)+1
C         STORE INITIAL VELOCITY
         VINIT(I)=VEL(I,1)
         VINIT(I+1)=VEL(I+1,1)
         WRITE(IPRINT,15)II,VEL(I,1),VEL(I+1,1),VEL(I+2,1)
15       FORMAT(1H ,I3,1X,3(2X,E11.3),/)
112      CONTINUE
C
         RETURN
         DEBUG SUBCHK
         END
C
C
C       ******************************************************************
C
C                                  S T A T
C
C        THE STRUCTURE IS MADE STATICALLY DETERMINATE BY RELEASING
C        BOUNDARY CONDITIONS SUCH THAT IT BECOMES EITHER A CANTILEVE
C        ('CANT') OR SIMPLY SUPPORTED ('SIMP') . THE BENDING MOMENT
C        AND AXIAL FORCE IN THE RELEASED STRUCTURE DUE TO A UNIT LOA
C        APPLIED IN TURN TO EACH DEGREE OF FREEDOM IS THEN
C         DETERMINED.
C        BENDING   : UNITM(I,J),I=D.O.F. WHERE LOAD APPLIED
C                                J = NODE NO.
C        AXIAL     : UNITN(I,J),I=D.O.F. WHERE LOAD APPLIED
C                                J = ELEMENT NO.
C        FORCE(I,IE,J)  : SELF-STRESS , STATICALLY ADMISSIBLE SETS
C            I=1         : STATICALLY ADMISSIBLE SET
C            I=2,NRED+1: SELF STRESS SYSTEMS
C            IE          : 1,NO. OF ELEMENTS
C            J=1         :  MOMENT AT 'A' END OF ELEMENT
C            J=2         :  MOMENT AT 'B' END OF ELEMENT
C            J=3         :  AXIAL FORCE IN ELEMENT
C
C
C       ******************************************************************
         SUBROUTINE STAT
         INCLUDE DAGNVS.COMPROC
C
         ITMODE=0
C        DETERMINE ORIENTATION OF ELEMENTS IN GLOBAL AXIS SYSTEM
         DO 55 IE=1,NE
         IL=NBEAM(IE,1) @NODE NO. OF A END OF ELEMENT IE
         IR=NBEAM(IE,2) @NODE NO. OF B END OF ELEMENT IE
C        CL=CURRENT LENGTH OF ELEMENT
         CL(IE)=SQRT((COORDX(IR)-COORDX(IL))**2.D0+(COORDY(IR))
        #-COORDY(IL))**2.D0
C
         IF(T.LT.1.D-9)RLO(IE)=CL(IE)
C
         SSIN(IE)=(COORDY(IR)-COORDY(IL))/CL(IE)
         CCOS(IE)=(COORDX(IR)-COORDX(IL))/CL(IE)
55       CONTINUE
C
C        FOR BENDING MOMENTS DUE TO UNIT LOAD AT D.O.F. I
C        FOR CANTILEVER
```

```
      IF(STADET.EQ.'CANT')THEN
C     HORISONTAL
C
      DO 1 I=4,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      DO 2 J=1,IR
      UNITM(I,J)=-(COORDY(IR+1)-COORDY(J))
2     CONTINUE
1     CONTINUE
C
C     VERTICAL
      DO 3 I=5,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      DO 4 J=1,IR
      UNITM(I,J)=-(COORDX(J)-COORDX(IR+1))
4     CONTINUE
3     CONTINUE
C     MOMENTS
      DO 26 I=6,NDF,NF
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      DO 27 J=1,IR
      UNITM(I,J)=1.
27    CONTINUE
26    CONTINUE
      ELSE
C     FOR SIMPLY SUPPORTED STRUCTURE
C     HORISONTAL LOADING AT D.O.F. I
      DO 8 I=4,NDF-2,NF
      DO 9 J=2,NN-1
      R1=0.
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      IF(J.GT.IR+1)R1=1.
      UNITM(I,J)=-COORDX(J)*COORDY(IR+1)/COORDX(NN)
     #+(COORDY(J)-COORDY(1))-R1*(COORDY(J)-COORDY(IR+1))
9     CONTINUE
8     CONTINUE
C     VERTICAL UNIT LOADING AT D.O.F. I
      DO 10 I=5,NDF-1,NF
      DO 11 J=2,NN-1
      R1=0
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      IF(J.GT.IR+1)R1=1.
      UNITM(I,J)=(COORDX(IR+1)/COORDX(NN)-1.)*COORDX(J)
     #+R1*(COORDX(J)-COORDX(IR+1))
11    CONTINUE
10    CONTINUE
C      MOMENT AT D.O.F. I
      DO 30 I=3,NDF,NF
      DO 31 J=1,NN
      RI=FLOAT(I)/NF+0.1
      IR=INT(RI)
      R1=0.
      IF(J.GT.IR)R1=1.
      UNITM(I,J)=(COORDX(J)/COORDX(NN)-R1)
```

```
            IF(I.EQ.3)UNITM(I,1)=-1.
            IF(I.EQ.NDF)UNITM(I,NN)=1.
   31       CONTINUE
   30       CONTINUE
            END IF
C
C           FOR AXIAL FORCES DUE TO UNIT LOADS AT D.O.F. I
C
C            HORISONTAL UNIT LOADS
            DO 13 I=4,NDF,NF
            RI=FLOAT(I)/NF+0.1
            IR=INT(RI)
            IRR=IR
            IF(STADET.EQ.'SIMP') IRR=NE
            DO 14 IE=1,IRR
            R1=0
            IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') R1=1.
            IF(STADET.EQ.'CANT')THEN
            UNITN(I,IE)=CCOS(IE)
            ELSE
            UNITN(I,IE)=CCOS(IE)+COORDY(IR+1)*SSIN(IE)/COORDX(NN)
           #-R1*CCOS(IE)
            END IF
   14       CONTINUE
   13       CONTINUE
C
C           VERTICAL UNIT LOADS
            DO 16 I=5,NDF,NF
            RI=FLOAT(I)/NF+0.1
            IR=INT(RI)
            IRR=IR
            IF(STADET.EQ.'SIMP') IRR=NE
            DO 17 IE=1,IRR
            R1=0
            IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') R1=1.
            IF(STADET.EQ.'CANT')THEN
            UNITN(I,IE)=SSIN(IE)
            ELSE
            UNITN(I,IE)=SSIN(IE)*(1.-COORDX(IR+1)/COORDX(NN))
           #-R1*SSIN(IE)
            END IF
   17       CONTINUE
   16       CONTINUE
C
C           UNIT APPLIED MOMENT
C            NO AXIAL FORCES CAUSED BY UNIT MOMENTS IN CANTILEVER
            IF(STADET.EQ.'SIMP')THEN
            DO 33 I=3,NDF,NF
            DO 34 IE=1,NE
            UNITN(I,IE)=-SSIN(IE)/COORDX(NN)
   34       CONTINUE
   33       CONTINUE
            END IF
C
            RETURN
            DEBUG SUBCHK
            END
C
C
```

```
C         ******************************************
C
C                        S L F S T R
C
C         FOR BOTH MODE SOLUTION (MODES=1) AND DIRECT
C         ANALYSES (MODES=0) SELF-STRESS SYSTEMS
C        CORRESPONDING TO RELEASED DEGREES OF FREEDOM
C         (MODES=1) AND EACH DEGREE OF FREEDOM AT A
C         NODE (MODES=0) ARE ASSEMBLED .
C
C         ******************************************
          SUBROUTINE SLFSTR
          INCLUDE DAGNVS.COMPROC
C
C          IF MODAL SOLUTION PHASE SET UP SELF STRESS SYSTEMS
C         EQUAL TO MOMENTS AND AXIAL FORCES ASSOCIATED WITH
C          UNIT LOADS AT RELEASED D.O.F.'S
          IF(MODES.EQ.1)THEN
          DO 1 I=2,NRED+1
          DO 2 IE=1,NE
          FORCE(I,IE,1)=UNITM(RELEAS(I-1),NBEAM(IE,1))
          FORCE(I,IE,2)=UNITM(RELEAS(I-1),NBEAM(IE,2))
          FORCE(I,IE,3)=UNITN(RELEAS(I-1),IE)
2         CONTINUE
1         CONTINUE
C
          ELSE
C
C          FOR DIRECT ANALYSIS SET UP SELF STRESS SYSTEMS
C          CORRESPONDING TO EACH DEGREE OF FREEDOM IN THE
C          STRUCTURE .
C
          DO 3 IE=1,NE
          DO 4 I=1,NINV
          FORCE(I,IE,1)=UNITM(ISTAT(I),NBEAM(IE,1))
          FORCE(I,IE,2)=UNITM(ISTAT(I),NBEAM(IE,2))
          FORCE(I,IE,3)=UNITN(ISTAT(I),IE)
4         CONTINUE
3         CONTINUE
C
          END IF
          RETURN
          DEBUG SUBCHK
          END
C
C
C         *******************************************************
C
C                              L O A D
C
C         ASSEMBLE THE STATICALLY ADMISSIBLE MOMENT AND AXIAL FORCE
C         DIAGRAM BY FACTORING THE UNIT DIAGRAMS BY THE LOAD VECTOR
C         PER D.O.F. AND SUM ALL CONTRIBUTIONS AT A NODE.
C         THE 'LOADS' ARE GIVEN BY (MASS*CURRENT MODE SHAPE)
C
C         *******************************************************
          SUBROUTINE LOAD
          INCLUDE DAGNVS.COMPROC
C
```

```
      ITREAC=0
      DO 1 IE=1,NE
      DO 2 J=1,3
      FORCE(1,IE,J)=0.
 2    CONTINUE
 1    CONTINUE
      DO 18 I=1,NDF
      II=INT(FLOAT(I)/NF+0.7)
C     IF D.O.F. IS A ROTATION THEN NO LOAD ASSOCIATED WITH IT
      IJ=(INT(FLOAT(I)/NF+0.001))*NF
      IF(I.NE.IJ)THEN
      RLOAD(I)=RMASS(II)*PHI(I,1)
      END IF
      DO 19 IE=1,NE
      FORCE(1,IE,1)=FORCE(1,IE,1)+UNITM(I,NBEAM(IE,1))*RLOAD(I)
      FORCE(1,IE,2)=FORCE(1,IE,2)+UNITM(I,NBEAM(IE,2))*RLOAD(I)
C     AXIAL FORCES
      FORCE(1,IE,3)=FORCE(1,IE,3)+UNITN(I,IE)*RLOAD(I)
      IF(NRED.EQ.4)THEN
      FORCET(IE,1)=FORCE(1,IE,1)
      FORCET(IE,2)=FORCE(1,IE,2)
      FORCET(IE,3)=FORCE(1,IE,3)
      END IF
 19   CONTINUE
 18   CONTINUE
C
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C     ******************************************
C
C                        Z E R O
C
C     INITIALISE ALL RELEVENT ARRAYS USED IN THE
C     ITERATIVE PROCEDURE TO DETERMINE REDUNDANTS.
C     THIS PROCEDURE APPLIES ONLY WHEN OPTION
C     OF COMBINED MODE AND DIRECT ANALSIS
C     PROCEDURE (ISYM = 0) IS APPLIED.
C
C     ******************************************
      SUBROUTINE ZERO
      INCLUDE DAGNVS.COMPROC
C
      DO 1 I=1,NRED
      FLEX(I,NRED+1)=0.
      COMPAT(I)=0.
      DO 2 J=1,NRED
      FLEX(I,J)=0.
      PARDIF(I,J)=0.
 2    CONTINUE
 1    CONTINUE
C
      RETURN
      DEBUG SUBCHK
      END
C
```

```
C
C     ************************************************
C
C                      C O M E Q U
C
C     ASSEMBLE NRED COMPATIBILITY EQUATIONS (NRED
C     = NO. OF REDUNDANTS) IF STRUCTURE IS STATICALLY
C     INDETERMINATE IN TERMS OF UNKNOWN REDUNDANT
C                          FORCES
C
C     ************************************************
      SUBROUTINE COMEQU
      INCLUDE DAGNVS.COMPROC
      ITREAC=ITREAC+1
C     SUM OVER ALL ELEMENTS IE
      DO 1 IE=1,NE
C     SET UP PARAMETERS FOR CALCULATIONS TO FOLLOW
      DO 2 I=1,NRED+1
C     MOMENTS AND AXIAL FORCES ARE NORMALISED
      AMOM(I)=(FORCE(I,IE,2)-FORCE(I,IE,1))/RMO
      ANORM1(I)=FORCE(I,IE,1)/RMO+FORCE(I,IE,3)/RNO
      ANORM2(I)=FORCE(I,IE,3)/RNO-FORCE(I,IE,1)/RMO
2     CONTINUE
      SUMA=ANORM1(1)
      SUMB=ANORM2(1)
      SUMC=AMOM(1)
      DO 3 I=1,NRED
      SUMA=SUMA+X(I)*ANORM1(I+1)
      SUMB=SUMB+X(I)*ANORM2(I+1)
      SUMC=SUMC+X(I)*AMOM(I+1)
3     CONTINUE
C     SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
      SIG=1.
      SIGA=1.
      SIGB=1.
      SIGC=1.
      SIGD=1.
      IF((SUMA+SUMC).LT.0) SIGA=-1.
      IF((SUMB-SUMC).LT.0) SIGB=-1.
      IF(SUMA.LT.0) SIGC=-1.
      IF(SUMB.LT.0) SIGD=-1.
      POWI=SIGA*(DABS(SUMA+SUMC)**EN)
      POWJ=SIGC*(DABS(SUMA)**EN)
      POWK=SIGB*(DABS(SUMB-SUMC)**EN)
      POWL=SIGD*(DABS(SUMB)**EN)
      POWA=POWI*(SUMA+SUMC)
      POWB=POWK*(SUMB-SUMC)
      POWC=POWA*(SUMA+SUMC)
      POWD=POWB*(SUMB-SUMC)
      POWE=POWJ*SUMA
      POWF=POWL*SUMB
      POWG=POWE*SUMA
      POWH=POWF*SUMB
      PRODA=EN1*SUMC/CL(IE)
      PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
      PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
C      SET UP COMPATIBILITY EQUATIONS,ONE FOR EACH REDUNDANCY
      DO 4 I=1,NRED
      COMPAT(I)=COMPAT(I)+(FORCE(I+1,IE,3)*EPSIO*0.5)*
```

```
     #((POWA-POWE)/PRODA-(POWB-POWF)/PRODA)
     #+RNO*EPSIO/(2.*RMO)*((AMOM(I+1)*RMO/CL(IE))*
     #((POWC-POWG)/PRODB-SUMA*(POWA-POWE)/PRODC
     #-(POWD-POWH)/PRODB+SUMB*(POWB-POWF)/PRODC)
     #+FORCE(I+1,IE,1)*((POWA-POWE)/PRODA+(POWB-POWF)/PRODA))
4     CONTINUE
1     CONTINUE
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C     *************************************************
C
C                          P D I F F
C
C     TO OBTAIN UNKNOWN REDUNDANTS X THE PARTIAL
C     DIRIVATIVES OF THE COMPATIBILITY EQUATIONS ARE
C     CALCULATED IN TERMS OF REDUNDANTS . THE NEWTON-
C     RAPHSON METHOD IS EMPLOYED TO ITERATE ONTO A
C                    SOLUTION
C
C     *************************************************
      SUBROUTINE PDIFF
      INCLUDE DAGNVS.COMPROC
      DO 1 IE=1,NE
C     SET UP PARAMETERS FOR CALCULATIONS TO FOLLOW
      DO 2 I=1,NRED+1
C     MOMENTS AND AXIAL FORCES ARE NORMALISED
      AMOM(I)=(FORCE(I,IE,2)-FORCE(I,IE,1))/RMO
      ANORM1(I)=FORCE(I,IE,1)/RMO+FORCE(I,IE,3)/RNO
      ANORM2(I)=(FORCE(I,IE,3)/RNO-FORCE(I,IE,1)/RMO)
2     CONTINUE
      SUMA=ANORM1(1)
      SUMB=ANORM2(1)
      SUMC=AMOM(1)
      DO 3 I=1,NRED
      SUMA=SUMA+X(I)*ANORM1(I+1)
      SUMB=SUMB+X(I)*ANORM2(I+1)
      SUMC=SUMC+X(I)*AMOM(I+1)
3     CONTINUE
C     SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
      SIG=1.
      SIGA=1.
      SIGB=1.
      SIGC=1.
      SIGD=1.
      IF((SUMA+SUMC).LT.0) SIGA=-1.
      IF((SUMB-SUMC).LT.0) SIGB=-1.
      IF(SUMA.LT.0) SIGC=-1.
      IF(SUMB.LT.0) SIGD=-1.
      POWI=SIGA*(DABS(SUMA+SUMC)**EN)
      POWJ=SIGC*(DABS(SUMA)**EN)
      POWK=SIGB*(DABS(SUMB-SUMC)**EN)
      POWL=SIGD*(DABS(SUMB)**EN)
      POWA=POWI*(SUMA+SUMC)
      POWB=POWK*(SUMB-SUMC)
      POWC=POWA*(SUMA+SUMC)
```

```
      POWD=POWB*(SUMB-SUMC)
      POWE=POWJ*SUMA
      POWF=POWL*SUMB
      POWG=POWE*SUMA
      POWH=POWF*SUMB
      PRODA=EN1*SUMC/CL(IE)
      PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
      PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
C     SET UP PARTIAL DIRIVATIVES
      DO 4 I=1,NRED
      DO 5 J=I,NRED
      PARTA=ANORM1(J+1)+AMOM(J+1)
      PARTB=ANORM2(J+1)-AMOM(J+1)
      PART1=((POWI*PARTA-POWJ*ANORM1(J+1))*EN1*PRODA
     #-EN1*AMOM(J+1)*(POWA-POWE)/CL(IE))/(PRODA*PRODA)
      PART2=((POWK*PARTB-POWL*ANORM2(J+1))*EN1*PRODA
     #-EN1*AMOM(J+1)*(POWB-POWF)/CL(IE))/(PRODA*PRODA)
      PART3=((POWA*PARTA-POWE*ANORM1(J+1))*EN2*EN2*PRODA*PRODA/
     #(EN1*EN1)-(POWC-POWG)*2.*EN2*PRODA*AMOM(J+1)/(EN1*CL(IE)))/
     #(PRODB*PRODB)
      PART4=(((POWA*ANORM1(J+1)+SUMA*EN1*POWI*PARTA)-
     #(POWE*ANORM1(J+1)
     #+SUMA*EN1*POWJ*ANORM1(J+1)))*PRODA*PRODA/EN1
     #-((POWA-POWE)*2.*PRODA*AMOM(J+1)*SUMA)/CL(IE))/(PRODC*PRODC)
      PART5=((POWB*PARTB-POWF*ANORM2(J+1))*EN2*EN2*PRODA*PRODA/
     #(EN1*EN1)-(POWD-POWH)*2.*EN2*PRODA*AMOM(J+1)/(EN1*CL(IE)))/
     #(PRODB*PRODB)
      PART6=(((ANORM2(J+1)*POWB+SUMB*EN1*POWK*PARTB)-
     #(POWF*ANORM2(J+1)
     #+SUMB*EN1*POWL*ANORM2(J+1)))*PRODA*PRODA/EN1
     #-((POWB-POWF)*2.*PRODA*AMOM(J+1)*SUMB)/CL(IE))/(PRODC*PRODC)
      PARDIF(I,J)=PARDIF(I,J)+FORCE(I+1,IE,3)*EPSI0*
     #.5*(PART1-PART2)+RN0*EPSI0/(2.*RM0)*((AMOM(I+1)*RM0/CL(IE))
     #*(PART3-PART4-PART5+PART6)+FORCE(I+1,IE,1)*(PART1+PART2))
    5 CONTINUE
    4 CONTINUE
    1 CONTINUE
C
C     HAVING OBTAINED THE PARTIAL DIRIVATIVES OF EQUILIBRIUM
C     EQUATIONS,MUST INVERT TO OBTAIN PERTURBATIONS ON REDUNDANTS
C     FIRST SET UP AUGMENTED MATRIX FOR INVERTION.
C
      DO 6 I=1,NRED
      FLEX(I,NRED+1)=-COMPAT(I)
      DO 7 J=I,NRED
      FLEX(I,J)=PARDIF(I,J)
      FLEX(J,I)=PARDIF(I,J)
    7 CONTINUE
    6 CONTINUE
      RETURN
      DEBUG SUBCHK
      END
C
C
```

```
C      ***********************************************
C
C                         P I V O T
C
C      INVERSION ROUTINE     :    C.F. GERALD
C      APPLIED NUMERICAL ANALYSIS , SECOND EDITION P.132
C
C      ***********************************************
       SUBROUTINE PIVOT
       INCLUDE DAGNVS.COMPROC
       DOUBLE PRECISION RATIO,VALUE
C
       NEQU=NINV
       IF(MODES.EQ.1)THEN
       NEQU=NRED
       END IF
C      FACTOR ALL ELEMENTS OF MATRIX BY LARGE NUMBER FOR NUMERICAL
C       STABILITY
       DO 60 I=1,NEQU
       DO 61 J=1,NEQU+1
       FLEX(I,J)=FLEX(I,J)*1.D12
       IF(ITREAC.LE.2)THEN
       END IF
61     CONTINUE
60     CONTINUE
       IF(NEQU.NE.1)THEN
       NP=NEQU+1
       NM1=NEQU-1
       DO 35 I=1,NM1
       IPVT=I
       IP1=I+1
       DO 10 J=IP1,NEQU
       IF(DABS(FLEX(IPVT,I)).LT.DABS(FLEX(J,I))) IPVT=J
10     CONTINUE
       IF(DABS(FLEX(IPVT,I)).LT.1.D-64)GO TO 99
       IF(IPVT.EQ.I) GO TO 25
       DO 20 JCOL=I,NP
       FACT=FLEX(I,JCOL)
       FLEX(I,JCOL)=FLEX(IPVT,JCOL)
       FLEX(IPVT,JCOL)=FACT
20     CONTINUE
25     DO 32 JROW=IP1,NEQU
       IF(DABS(FLEX(JROW,I)).LE.1.D-64) GO TO 32
       RATIO=FLEX(JROW,I)/FLEX(I,I)
       DO 30 KCOL=IP1,NP
       FLEX(JROW,KCOL)=FLEX(JROW,KCOL)-RATIO*FLEX(I,KCOL)
30     CONTINUE
32     CONTINUE
35     CONTINUE
       IF(DABS(FLEX(NEQU,NEQU)).LT.1.D-64)GO TO 99
       NP1=NP
       DO 50 KCOL=NP1,NP
       FLEX(NEQU,KCOL)=FLEX(NEQU,KCOL)/FLEX(NEQU,NEQU)
       DO 45 J=2,NEQU
       NVBL=NP1-J
       L=NVBL+1
       VALUE=FLEX(NVBL,KCOL)
       DO 40 K=L,NEQU
       VALUE=VALUE-FLEX(NVBL,K)*FLEX(K,KCOL)
```

```
40    CONTINUE
      FLEX(NVBL,KCOL)=VALUE/FLEX(NVBL,NVBL)
45    CONTINUE
50    CONTINUE
      ELSE
      FLEX(1,2)=FLEX(1,2)/FLEX(1,1)
      END IF
      RETURN
99    WRITE(IPRINT,101)
101   FORMAT(1H0,10X,'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')
      STOP
      DEBUG SUBCHK
      END
C
C
C
C     ******************************************
C
C                    D E L T A
C
C     IF STRUCTURE IS STATICALLY INDETERMINATE
C     AND MODE ANALYSIS IS REQUIRED,REDUNDANT
C     FORCES ARE DETERMINED BY NEWTON-RAPHSON
C     PROCEDURE.RESIDUAL REDUNDANT FORCES ARE
C     CHECKED HERE TO DETERMINE IF CONVERGENCE
C     HAS OCCURED(IFLAG=1).IF NOT THEN
C     MORE EQUILIBRIUM ITERATIONS REQUIRED.
C
C     ******************************************
      SUBROUTINE DELTA
      INCLUDE DAGNVS.COMPROC
C     AND THAT THE COMPATIBILITY EQUATIONS ARE NEAR ZERO
      DOUBLE PRECISION XDUM(2*NN)
      IFLAG1=0
      IFLAG2=0
      IFLAG=0
      IF(ITREAC.GT.1)THEN
      IFLAG1=1
      IFLAG2=1
      DO 9 I=1,NRED
      IF(DABS(X(I)).GT.1.D-6)THEN
      IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0
      END IF
9     CONTINUE
      END IF
      DO 99 I=1,NRED
      XDUM(I)=X(I)
99    CONTINUE
      IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1
C      ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT
      DO 10 I=1,NRED
      X(I)=X(I)+FLEX(I,NRED+1)
10    CONTINUE
C      IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE
      IF(IFLAG.EQ.0) GO TO 15
C      CONSTRUCT FINAL BENDING MOMENT DIAGRAM
C
C     CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO
C      ZERO
```

```
      IF(NRED.GT.1)THEN
      DUM1=1.E16
      DUM2=1.D-16
C     FIND MINIMUM REACTANT
      DO 1 I=1,NRED
      IF(DABS(X(I)).LT.DUM1)DUM1=DABS(X(I))
      IF(DABS(X(I))-1.D-6.LE.DUM1)IMIN=I
1     CONTINUE
C     FIND MAXIMUM REACTANT
      DO 2 I=1,NRED
      IF(DABS(X(I)).GT.DUM2)DUM2=DABS(X(I))
      IF(DABS(X(I))+1.D-6.GT.DUM2)IMAX=I
2     CONTINUE
C     DETERMINE RATIO BETWEEN MAX AND MIN VALUES.
C     IF MIN/MAX LESS THAN 1.D-9 THEN SET X(MIN) EQUAL TO ZERO.
      RATIO=X(IMIN)/X(IMAX)
      IF(DABS(RATIO).LE.1.D-9)X(IMIN)=0.
      END IF
      DO 12 IE=1,NE
      DO 13 IJ=1,3
      FORCET(IE,IJ)=0.
      FORCET(IE,IJ)=FORCE(1,IE,IJ)
      DO 14 IK=1,NRED
      FORCET(IE,IJ)=FORCET(IE,IJ)+X(IK)*FORCE(IK+1,IE,IJ)
14    CONTINUE
13    CONTINUE
12    CONTINUE
15    CONTINUE
      RETURN
      DEBUG SUBCHK
      END
C
C
C     *******************************************
C
C                    V E L O C
C
C     HAVING OBTAINED THE TOTAL BENDING MOMENT
C     AND AXIAL FORCE DIAGRAMS , VELOCITIES ARE
C     DETERMINED USING THE PRINCIPLE OF VIRTUAL
C                    VELOCITIES
C
C     *******************************************
      SUBROUTINE VELOC
      INCLUDE DAGNVS.COMPROC
      DOUBLE PRECISION UNITMA(NDF,NN)
      IJ=2
      DO 1 I=1,NDF
      VEL(I,2)=0.
C     IF THE D.O.F. IS A BOUNDARY CONDITION THEN VELOCITY IS ZERO
      IF(IBC(I).EQ.1) GO TO 1
C      DETERMINE WHETHER D.O.F. IS A ROTATION
      IK=0
      IF(IJ*NF.EQ.I)IK=1
      IF(IK.EQ.1)IJ=IJ+1
      DO 2 IE=1,NE
C     FOR AN APPLIED UNIT MOMENT , THERE IS A
C     DISCONTINUITY OF MOMENT AT POINT OF APPLICATION
      IF(I.EQ.3) GO TO 3
```

```
      IF(I.EQ.NDF) GO TO 3
      UNITMA(I,NBEAM(IE,1))=0.
      IF(IK.EQ.1.AND.(IJ-1).EQ.NBEAM(IE,1))THEN
      UNITMA(I,NBEAM(IE,1))=UNITM(I,NBEAM(IE,1))-1.
      UNITM(I,NBEAM(IE,1))=UNITMA(I,NBEAM(IE,1))
      END IF
    3 CONTINUE
      SUMA=UNITM(I,NBEAM(IE,2))-UNITM(I,NBEAM(IE,1))
      SUMB=(FORCET(IE,2)-FORCET(IE,1))/RMO
      SUMC=FORCET(IE,1)/RMO+FORCET(IE,3)/RNO
      SUMD=(FORCET(IE,3)/RNO-FORCET(IE,1)/RMO)
      SIG1=1.
      SIG2=1.
      SIG3=1.
      SIG4=1.
      IF((SUMB+SUMC).LT.0) SIG1=-1.
      IF((-SUMB+SUMD).LT.0) SIG2=-1.
      IF(SUMC.LT.0) SIG3=-1.
      IF(SUMD.LT.0) SIG4=-1.
      PROD1=SIG1*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)*(SUMB+SUMC)
      PROD2=SIG1*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)
      PROD3=SIG2*((DABS(-SUMB+SUMD))**EN)*(-SUMB+SUMD)
     #*(-SUMB+SUMD)
      PROD4=SIG2*((DABS(-SUMB+SUMD))**EN)*(-SUMB+SUMD)
      PROD5=SIG3*(DABS(SUMC)**EN)*SUMC*SUMC
      PROD6=SIG3*(DABS(SUMC)**EN)*SUMC
      PROD7=SIG4*(DABS(SUMD)**EN)*SUMD*SUMD
      PROD8=SIG4*(DABS(SUMD)**EN)*SUMD
      CONST1=SUMA/CL(IE)
      CONST2=CL(IE)*CL(IE)/(EN2*SUMB*SUMB)
      CONST3=(SUMC)*CL(IE)*CL(IE)/(EN1*SUMB*SUMB)
      CONST4=CONST3*(SUMD)/(SUMC)
      CONST5=CL(IE)/(EN1*(SUMB))
      CONST6=RNO*EPSI0/(2.*RMO)
      CONST7=EPSI0*0.5*UNITN(I,IE)
C      VELOCITY AT D.O.F. I
      VEL(I,2)=VEL(I,2)+CONST6*(CONST1*((PROD1-PROD5)
     #*CONST2-(PROD2-PROD6)*CONST3-(PROD3-PROD7)*CONST2+
     #(PROD4-PROD8)*CONST4)+UNITM(I,NBEAM(IE,1))*
     #((PROD2-PROD6)*CONST5+(PROD4-PROD8)*CONST5)-)+
     #CONST7*((PROD2-PROD6)*CONST5-(PROD4-PROD8)*CONST5)
    2 CONTINUE
    1 CONTINUE
      RETURN
      DEBUG SUBCHK
      END
```

```
C     *******************************
C
C                  M O D E C H
C
C     DETERMINES WHETHER CONVERGENCE ONTO
C     A MODE SHAPE HAS OCCURED (INMSOL=1)
C
C     *********************************
      SUBROUTINE MODECH
      INCLUDE DAGNVS.COMPROC
C
C     NO. OF ITERATIONS TO DETERMINE MODE
C      IF MODE NOT OBTAINED AFTER FORTY ITERATIONS STOP
      IF(ITMODE.EQ.10)WRITE(IPRINT,100)
100   FORMAT(1H ,//,20X,'MODE NOT FOUND AFTER TEN ITERATIONS :
     # STOP',/)
      IF(ITMODE.EQ.10)STOP
      ITMODE=ITMODE+1
      A=0.
      B=0.
      DISIP(1)=DISIP(2)
      DISIP(2)=0.
C     CHECK FOR CONVERGENCE
C     CALCULATE CURRENT DISSIPATION RATE
      DO 1 I=1,NDF,3
      IK=I+1
      II=INT(FLOAT(I)/NF+0.7)
      DISIP(2)=DISIP(2)+VEL(I,2)*RMASS(II)*PHI(I,1)
      DISIP(2)=DISIP(2)+VEL(IK,2)*RMASS(II)*PHI(IK,1)
1     CONTINUE
C
C      NORMALISE VELOCITIES FOR NEW MODE SHAPE
C
      DUM=0.
      DO 5 I=1,NDF,3
      II=INT(FLOAT(I)/NF+0.7)
      DUM=DUM+VEL(I,2)*VEL(I,2)*RMASS(II)
      IK=I+1
      DUM=DUM+VEL(IK,2)*VEL(IK,2)*RMASS(II)
5     CONTINUE
      RLAMDA=SQRT(DUM)
C
      DO 2 I=1,NDF
      PHI(I,2)=VEL(I,2)/RLAMDA
      PHI(I,1)=PHI(I,2)
2     CONTINUE
C      CHECK CHANGE IN DISSIPATION RATE
C     RLAMDA=ANORM
      IDISIP=0
      IF(ABS(DISIP(2)/DISIP(1)-1.D0).LT.5.D-2)IDISIP=1
C
C      OBTAIN AMPLITUDE OF VELOCITY BY PERFORMING MOMENTUM BALANCE
      IF(IDISIP.EQ.1.AND.T.LT.1.D-9)THEN
      DO 3 I=1,NDF
      II=INT(FLOAT(I)/NF+0.7)
      IJ=(INT(FLOAT(I)/NF+0.001))*NF
      IF(I.NE.IJ)THEN
      A=A+VEL(I,1)*RMASS(II)*PHI(I,1)
      B=B+PHI(I,1)*RMASS(II)*PHI(I,1)
```

```
      END IF
 3    CONTINUE
      AMP(1)=A/B
      AMP(2)=AMP(1)
      DO 4 I=1,NDF
      VEL(I,2)=PHI(I,1)*AMP(2)
 4    CONTINUE
      END IF
C
      IF(T.GT.0.AND.IDISIP.EQ.1.AND.INMSOL.EQ.0)THEN
      DO 6 I=1,NDF
      II=INT(FLOAT(I)/NF+0.7)
      IJ=(INT(FLOAT(I)/NF+0.001))*NF
      IF(I.NE.IJ)THEN
      A=A+VEL(I,1)*RMASS(II)*PHI(I,1)
      B=B+PHI(I,1)*RMASS(II)*PHI(I,1)
      END IF
 6    CONTINUE
      AMP(2)=A/B
      END IF
C
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C     *****************************************
C
C                    I N M O D E
C
C     SETS UP TIME FUNCTION T AND AMPLITUDE
C     OF MODE SOLUTION,ESTIMATES TOTAL TIME AND
C     IN INSTANTANEOUS MODE SOLUTION TECHNIQUE
C     CHECKS IF NEW MODE HAS BEEN FOUND.LATTER
C     ONLY PERFORMED IF COMBINED MODE AND DIRECT
C     ANALYSIS (ISYM=0) IS REQUESTED .
C
C     *****************************************
      SUBROUTINE INMODE
      INCLUDE DAGNVS.COMPROC
C
      MODES=0
      A=0.D0
      B=0.D0
      ITMODE=0
      POWA=1.D0/EN
      POWB=(EN-1.D0)/EN
      POWC=1.D0/POWB
      POWD=1.D0/(EN-1.D0)
C
      RK=1.D0/((DABS(RLAMDA)**POWA)*(AMP(2)**POWB))
      TIME=DT
      FACT=POWB*RK*TIME
C
C      CALCULATE EXPRESSION FOR T(T)
      TT(2)=(1.D0-FACT)**POWC
C      DIRIVATIVE OF T(T)
      IF(MCOUNT.EQ.1)DTTDT(2)=-RK*(TT(2)**POWA)
```

```
C      CONSIDER CASES :A) BEFORE MODE SOLN (INMSOL=0)
C      SEPARATELY      B) AFTER MODE SOLN. (INMSOL=1)
      IF(INMSOL.EQ.0)THEN
C      MODE VELOCITIES
      DO 1 I=1,NDF
      VMODE(I,2)=AMP(2)*PHI(I,1)*TT(2)
1     CONTINUE
C
      ELSE
      IF(MCOUNT.GT.1)THEN
      DTTDT(2)=(DTTDT(2)-RK*(TT(2)**POWA))/2.D0
C
      TT(2)=TT(1)+(DTTDT(1)+DTTDT(2))*DT/2.D0
      END IF
      DO 2 I=1,NDF
      IF(MCOUNT.EQ.1)U(I,1)=0.D0
      VEL(I,2)=VMODE(I,2)
      IF(MCOUNT.EQ.1)VMODE(I,2)=AMP(2)*PHI(I,1)*TT(2)
C
      IF(MCOUNT.GT.1)VMODE(I,2)=(VMODE(I,2)+AMP(2)*PHI(I,1)
     #*TT(2))/2.D0
C
2     CONTINUE
C      CHECK IF NEW MODE HAS BEEN REACHED
C
      DO 5 I=1,NDF,3
      II=I+1
      A=A+(VMODE(I,2)-VEL(I,2))*(VMODE(I,2)-VEL(I,2))
      B=B+(VMODE(I,2)*VMODE(I,2))
      A=A+(VMODE(II,2)-VEL(II,2))*(VMODE(II,2)-VEL(II,2))
      B=B+(VMODE(II,2)*VMODE(II,2))
5     CONTINUE
C
C      IF NEW MODE REACHED SET NNORM = 1
C
      IF(DSQRT(A)/DSQRT(B).LT.0.05D0.AND.MCOUNT.GT.1)NNORM=1
C
      END IF
C
C      CALCULATE MODE MOMENTS AND AXIAL FORCES
C
      DO 3 IE=1,NE
      DO 4 J=1,3
      DUM=RM0
      IF(J.EQ.3)DUM=RN0
      FORMOD(IE,J)=(((AMP(2)*TT(2)/RLAMDA)**POWA)*
     #FORCET(IE,J))/DUM
4     CONTINUE
C
C      IF AXIAL FORCES ARE LARGE SET FLAG TO REQUEST NEW MATCHING
C      FACTOR.
C      IF NEW FACTOR HAS BEEN PREVIOUSLY CALCULATED THEN SKIP.
      IF(MATCHA.NE.-1.OR.MATCHA.EQ.0)THEN
      IF(DABS(FORMOD(IE,3)).GT.0.2)MATCHA=1
      END IF
3      CONTINUE
      RETURN
      DEBUG SUBCHK
      END
```

```
C
C      ***************************************
C
                      M A T C H
C
C      MATCHING PROCEDURE (SLOPE ALONE) FOR
C      MODE SOLUTION TECHNIQUE . THIS MATCHING
C      IS ONLY OPERATIVE IF COMBINED MODE AND
C      DIRECT ANALYSIS IS REQUESTED (ISYM=0)
C
C      ***************************************
      SUBROUTINE MATCH
      INCLUDE DAGNVS.COMPROC
      DMATCH=1.
      DUM=0.
C     LOCATE MAXIMUM BENDING MOMENT OR AXIAL FORCE IN STRUCTURE
      N1=1
      N2=2
      IF(MATCHA.EQ.1)THEN
      N1=3
      N2=3
      END IF
C
      DO 1 IE=1,NE
      DO 2 I=N1,N2
      IF(DABS(FORMOD(IE,I)).GE.DUM)DUM=ABS(FORMOD(IE,I))
2     CONTINUE
1     CONTINUE
C
      AMPD=AMP(1)**(1./EN)
      RMMAX=DUM/AMPD
      IF(RMATCH.GT.1.01)THEN
      RMMAX=RMMAX*RMATCH
      DMATCH=RMATCH
      END IF
C
      RKMAX=AMP(1)*((RMMAX)**EN)
      RMATCH=(1.+(RKMAX)**(1./EN))/(RKMAX**(1./EN))
C
C     CALCULATE TOTAL TIME
C
      TF=(EN/(EN-1.))/(RK*RMATCH)
      DT=TF/RINT
C
C     MATCH YIELD MOMENT AND AXIAL YIELD STRESS
      RM0=RM0*RMATCH/DMATCH
      RN0=RN0*RMATCH/DMATCH
      IF(MATCHA.EQ.0)WRITE(IPRINT,4)RMATCH
4     FORMAT(1H ,///,20X,'MATCHING FACTOR IS :',E11.6,///)
C
      IF(MATCHA.EQ.1)THEN
      WRITE(IPRINT,3)RMATCH
3     FORMAT(1H ,///,20X,'REVISED MATCHING FACTOR FOR MEMBRANE
     #ACTION IS : ',E11.6,///)
      MATCHA=-1
      END IF
      RETURN
      DEBUG SUBCHK
      END
```

```
C
C
C       ***********************************
C
C                    V M A T C H
C
C       CALCULATE MATCHING FACTOR(S) IN
C       DIRECT SOLUTION PROCEDURE . MATCHING
C       PROCEDURE IS CHOSEN BY DATA INPUT
C
C       ***********************************
        SUBROUTINE VMATCH
        INCLUDE DAGNVS.COMPROC
        DOUBLE PRECISION RMOMT(NE,2),DMATCH,DPMACH
C
        IMATCH=0
        DUM=0.D0
        DMATCH=RMATCH
        DPMACH=PMATCH
        ITMACH=ITMACH+1
        IF(ITMACH.EQ.1)THEN
        DMATCH=1.D0
        DPMACH=1.D0
        END IF
C
C        DETERMINE BENDING MOMENT DIAGRAM
C
        DO 6 IE=1,NE
        RMOMT(IE,1)=0.D0
        RMOMT(IE,2)=0.D0
6       CONTINUE
        DO 1 IE=1,NE
        DO 2 I=1,NINV
        RMOMT(IE,1)=RMOMT(IE,1)+FORCE(I,IE,1)*XX(I)
        RMOMT(IE,2)=RMOMT(IE,2)+FORCE(I,IE,2)*XX(I)
2       CONTINUE
1       CONTINUE
C
        IF(T.LT.1.D-9)THEN
C
C        FIND MAXIMUM B.M.
C
        DO 3 IE=1,NE
        DO 4 I=1,2
        IF(DABS(RMOMT(IE,I)).GT.DUM)DUM=DABS(RMOMT(IE,I))
4       CONTINUE
3       CONTINUE
        IF(PAMTCH.LT.1)THEN
        PMATCH=(1.D0+(DUM/RM0))/((DUM/RM0))
        END IF
        RMATCH=(1.D0+DUM/RM0)/((DUM/RM0)**(1.D0/PMATCH))
        RM0=RM0*RMATCH/DMATCH
        RN0=RN0*RMATCH/DMATCH
        EN=EN*PMATCH/DPMACH
        EN1=EN+1.D0
        EN2=EN1+1.D0
        END IF
C
C       RESET REACTIONS TO GET BETTER ESTIMATE OF REACTIONS AT
```

```
 C       MATCHED VALUES
        IF(ITMACH.EQ.1)THEN
        DO 11 I=1,NINV
        XX(I)=XXDUM
 11     CONTINUE
        END IF
C
        IF(T.LT.1.D-9)TF=(EN/(EN-1.))/(RK*RMATCH*20.D0)
C
        WRITE(IPRINT,5)ITMACH,RMATCH,PMATCH
5       FORMAT(1H ,//,20X,'ITERATION NO. ',1X,I2,5X,'STRESS FACTOR
       #IS :',2X,E11.6,10X,'POWER FACTOR IS :',2X,E11.6,//)
        DT=TF/RINT
C
C        HAS MATCHING FACTOR CONVERGED
C
        IF(DABS((RMATCH/DMATCH)-1.D0).LT.0.05D0.AND.
       #DABS((PMATCH/DPMACH)-1.D0).LT.0.05D0)IMATCH=1
C
        IF(ITMACH.GT.10)THEN
        WRITE(IPRINT,7)
7       FORMAT(1H ,///,30X,' MATCHING FACTOR HAS NOT CONVERGED AFTER
       #TEN ITERATIONS   :  STOP',///)
        STOP
        END IF
        IF(IMATCH.EQ.1)ITMACH=0
C
        RETURN
        DEBUG SUBCHK
        END
C
C
C       *******************************************
C
C                      C O M D I F
C
C       SET UP COMPATIBILITY EQUATIONS , ONE FOR
C       EACH DEGREE OF FREEDOM OF THE RELEASED
C       STRUCTURE AS FUNCTIONS OF NODAL FORCES.
C       TAKE PARTIAL DIRIVATIVES OF EQUATIONS W.R.T.
C       EACH NODAL FORCE IN TURN (MATRIX PARDIF).
C
C       *******************************************
        SUBROUTINE COMDIF
        INCLUDE DAGNVS.COMPROC
        DOUBLE PRECISION XDUM(2*NN)
        IFLAG=1
        ITREAC=ITREAC+1
C
        DO 55 I=1,NINV
C        DONT NEED TO TEST FOR CONVERGENCE IF IN IMPLICIT MODE
C
        IF(IRND.EQ.0)THEN
        XDUM(I)=XX(I)
        IF(ITREAC.GT.1)THEN
        XX(I)=XX(I)+FLEX(I,NINV+1)
        IF(DABS(XX(I)).GT.1.D-3)THEN
        IF(DABS(XX(I)/XDUM(I)-1.D0).GT.1.D-3)IFLAG=0
        END IF
```

```
      END IF
      END IF
      FLEX(I,NINV+1)=0.
      COMPAT(I)=0.
55    CONTINUE
C
      IF(ITREAC.EQ.1)IFLAG=0
C
      DO 6 I=1,NINV
      DO 9 J=1,NINV
      PARDIF(I,J)=0.
9     CONTINUE
6     CONTINUE
C
      IFLAG=1
C
C     SET UP NINV COMPATIBILITY EQUATIONS (NINV=NO. OF NODAL
C     FORCES)
C     SUM OVER ALL ELEMENTS IE
      DO 1 IE=1,NE
      DO 2 I=1,NINV
C     MOMENTS AND AXIAL FORCES ARE NORMALISED
      AMOM(I)=(FORCE(I,IE,2)-FORCE(I,IE,1))/RMO
      ANORM1(I)=FORCE(I,IE,1)/RMO+FORCE(I,IE,3)/RNO
      ANORM2(I)=FORCE(I,IE,3)/RNO-FORCE(I,IE,1)/RMO
2     CONTINUE
      SUMA=0.
      SUMB=0.
      SUMC=0.
      DO 3 I=1,NINV
      SUMA=SUMA+XX(I)*ANORM1(I)
      SUMB=SUMB+XX(I)*ANORM2(I)
      SUMC=SUMC+XX(I)*AMOM(I)
3     CONTINUE
C
      SUM1=SUMA+SUMC
      SUM2=SUMB-SUMC
      IF(DABS(SUM1).LT.1.D-12)SUM1=0.
      IF(DABS(SUM2).LT.1.D-12)SUM2=0.
C
C     SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
      SIG=1.D0
      SIGA=1.D0
      SIGB=1.D0
      SIGC=1.D0
      SIGD=1.D0
      IF((SUM1).LT.0) SIGA=-1.D0
      IF((SUM2).LT.0) SIGB=-1.D0
      IF(SUMA.LT.0) SIGC=-1.D0
      IF(SUMB.LT.0) SIGD=-1.D0
      POWI=SIGA*(DABS(SUM1)**EN)
      POWJ=SIGC*(DABS(SUMA)**EN)
      POWK=SIGB*(DABS(SUM2)**EN)
      POWL=SIGD*(DABS(SUMB)**EN)
      POWA=POWI*(SUM1)
      POWB=POWK*(SUM2)
      POWC=POWA*(SUM1)
      POWD=POWB*(SUM2)
      POWE=POWJ*SUMA
```

```
      POWF=POWL*SUMB
      POWG=POWE*SUMA
      POWH=POWF*SUMB
      PRODA=EN1*SUMC/CL(IE)
      PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
      PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
C      SET UP COMPATIBILITY EQUATIONS,ONE FOR EACH COMPONENT OF
C     VELOCITY
      DO 4 I=1,NINV
      COMPAT(I)=COMPAT(I)+(FORCE(I,IE,3)*EPSI0*0.5)*
     #((POWA-POWE)/PRODA-(POWB-POWF)/PRODA)
     #+RN0*EPSI0/(2.*RM0)*((AMOM(I)*RM0/CL(IE))*((POWC-POWG)/PRODB
     #-SUMA*(POWA-POWE)/PRODC
     #-(POWD-POWH)/PRODB+SUMB*(POWB-POWF)/PRODC)
     #+FORCE(I,IE,1)*((POWA-POWE)/PRODA+(POWB-POWF)/PRODA))
C
C     SET UP PARTIAL DIRIVATIVES
      DO 5 J=I,NINV
      PARTA=ANORM1(J)+AMOM(J)
      PARTB=ANORM2(J)-AMOM(J)
      IF(DABS(PARTA).LT.1.D-12)PARTA=0.
      IF(DABS(PARTB).LT.1.D-12)PARTB=0.
      PART1=((POWI*PARTA-POWJ*ANORM1(J))*EN1*PRODA
     #-EN1*AMOM(J)*(POWA-POWE)/CL(IE))/(PRODA*PRODA)
      PART2=((POWK*PARTB-POWL*ANORM2(J))*EN1*PRODA
     #-EN1*AMOM(J)*(POWB-POWF)/CL(IE))/(PRODA*PRODA)
      PART3=((POWA*PARTA-POWE*ANORM1(J))*EN2*EN2*PRODA*PRODA/
     #(EN1*EN1)-(POWC-POWG)*2.*EN2*PRODA*AMOM(J)/(EN1*CL(IE)))/
     #(PRODB*PRODB)
      PART4=(((POWA*ANORM1(J)+SUMA*EN1*POWI*PARTA)-(POWE*ANORM1(J)
     #+SUMA*EN1*POWJ*ANORM1(J)))*PRODA*PRODA/EN1
     #-((POWA-POWE)*2.*PRODA*AMOM(J)*SUMA)/CL(IE))/(PRODC*PRODC)
      PART5=((POWB*PARTB-POWF*ANORM2(J))*EN2*EN2*PRODA*PRODA/
     #(EN1*EN1)-(POWD-POWH)*2.*EN2*PRODA*AMOM(J)/(EN1*CL(IE)))/
     #(PRODB*PRODB)
      PART6=(((ANORM2(J)*POWB+SUMB*EN1*POWK*PARTB)-(POWF*ANORM2(J)
     #+SUMB*EN1*POWL*ANORM2(J)))*PRODA*PRODA/EN1
     #-((POWB-POWF)*2.*PRODA*AMOM(J)*SUMB)/CL(IE))/(PRODC*PRODC)
      PARDIF(I,J)=PARDIF(I,J)+FORCE(I,IE,3)*EPSI0*.5*(PART1-PART2)
     #+RN0*EPSI0/(2.*RM0)*((AMOM(I)*RM0/CL(IE))*(PART3
     #-PART4-PART5+PART6)+FORCE(I,IE,1)*(PART1+PART2))
5     CONTINUE
4     CONTINUE
C
C      CHECK IF NODAL FORCES HAVE BEEN FOUND
C      (ARE VIRTUAL VELOCITIES EQUAL TO REAL VELOCITIES ?)
C
      DO 11 I=1,NE
      II=2*I-2
      IJ=3*I
      DO 12 J=1,2
      IK=II+J
      IL=IJ+J
      IF(DABS(DUMVEL(IL)).GT.1.D-3)THEN
      IF(DABS(COMPAT(IK)/DUMVEL(IL)-1.D0).GT.1.D-3)IFLAG=0
      ELSE
      IF(DABS(COMPAT(IK)-DUMVEL(IL)).GT.1.D-4)IFLAG=0
      END IF
12    CONTINUE
```

```
 11     CONTINUE
        IF(NRED.NE.4)THEN
        IF(DABS(COMPAT(NINV)).GT.1.D-4)IFLAG=0
        END IF
C
C
        IF(ITREAC.EQ.100)THEN
        WRITE(IPRINT,30)
 30     FORMAT(///,3X,' NO CONVERGENCE ONTO NODAL FORCES : CHANGE N
       #OR INITIAL NODAL FORCE ESTIMATE ')
        STOP
        END IF
        IF(IFLAG.EQ.1)ITREAC=0
C
C
C       HAVING OBTAINED THE PARTIAL DIRIVATIVES OF COMPATIBILITY
C       EQUATIONS,MUST INVERT TO OBTAIN PERTURBATIONS ON NODAL
C       FORCES . FIRST SET UP AUGMENTED MATRIX FOR INVERSION.
C
        DO 66 I=1,NINV
        IE=INT(FLOAT(I)/2+0.6)
C        FOR STATICALLY INDETERMINATE STRUCTURES
        IF(IE.GT.NE)IE=NE
        II=I+NF+IE-1
        FLEX(I,NINV+1)=-COMPAT(I)+DUMVEL(II)
        DO 77 J=I,NINV
        FLEX(I,J)=PARDIF(I,J)
        FLEX(J,I)=PARDIF(I,J)
        AA(I,J)=PARDIF(I,J)
        AA(J,I)=PARDIF(I,J)
C
 77     CONTINUE
 66     CONTINUE
C
        RETURN
        DEBUG SUBCHK
        END
C
C
C        *******************************************
C
C                         A C C A X C
C
C        CALCULATE ACCELERATIONS FROM THE NODAL FORCES
C        USING THE EQUILIBRIUM EQUATIONS AT EACH D.O.F.
C
C        ********************************************
        SUBROUTINE ACCAXC
        INCLUDE DAGNVS.COMPROC
C
        DO 1 IE=1,NE
        IA=NBEAM(IE,1)
        IB=NBEAM(IE,2)
        IJ=3*IA-2
        IK=3*IB-2
        I1A=IJ
        I2A=I1A+1
        I1B=IK
        I2B=I1B+1
```

```
      IIJ=I1B-IE-2
      IIK=I2B-IE-2
      II=INT(FLOAT(I1B)/3+0.7)
C
C     CALCULATE ACCELERATIONS FROM NODAL FORCES
C
      ACC(I1B,1)=0.D0
      ACC(I2B,1)=0.D0
      ACC(I1B,1)=-XX(IIJ)/RMASS(II)
      ACC(I2B,1)=-XX(IIK)/RMASS(II)
C
1     CONTINUE
C
      RETURN
      DEBUG SUBCHK
      END
C
C
C
C     ****************************************
C
C                   I M P L I C
C
C     ASSEMBLES MATRICES AND VECTORS REQUIRED
C     TO SOLVE FOR NODAL FORCE RESIDUALS IN
C     IMPLICIT FORWARD INTEGRATION PROCEDURE
C
C     ****************************************
      SUBROUTINE IMPLIC
      INCLUDE DAGNVS.COMPROC
C
      SIG=1.D0
      IF(IRND.GT.1)SIG=-1.D0
      DO 1 IE=1,NE
      IA=NBEAM(IE,1)
      IB=NBEAM(IE,2)
      IJ=3*IA-2
      IK=3*IB-2
      I1B=IK
      I2B=IK+1
      IIJ=I1B-IE-2
      IIK=I2B-IE-2
      II=INT(FLOAT(I1B)/3+0.7)
C
      FLEX(IIJ,NINV+1)=-(COMPAT(IIJ)-DUMVEL(I1B))
     #+(SIG*ACC(I1B,1)-XX(IIJ)/RMASS(II))*DT/2.D0
      FLEX(IIK,NINV+1)=-(COMPAT(IIK)-DUMVEL(I2B))
     #+(SIG*ACC(I2B,1)-XX(IIK)/RMASS(II))*DT/2.D0
C
1     CONTINUE
C
C     FOR INDETERMINATE STRUCTURES ADD ROTATIONAL CONTRIBUTION
C
      IF(NRED.NE.4)THEN
      FLEX(NINV,NINV+1)=-(COMPAT(NINV)-DUMVEL(NDF))
     #+(SIG*ACC(NDF,1)-XX(NINV)/RMASS(NN))*DT/2.D0
      END IF
C
      IOUT=0
```

```
      DO 2 I=1,NINV
      II=INT(FLOAT(I+2)/2+0.6)
      DO 3 J=1,NINV
      SIG=0.
      IF(I.EQ.J)SIG=1.
      IF(II.GT.NN)II=NN
      FLEX(I,J)=AA(I,J)+DT*SIG/(2.D0*RMASS(II))
3     CONTINUE
2     CONTINUE
      RETURN
      DEBUG SUBCHK
      END
C
C
C
C     ************************************
C
                  R E V I S E
C
C      UPDATES OR REVISES ACCELERATIONS,
C      VELOCITIES , DISPLACEMENTS AND NODAL
C      FORCES BY RESIDUAL QUANTITIES.
C
C      ************************************
      SUBROUTINE REVISE
      INCLUDE DAGNVS.COMPROC
      DOUBLE PRECISION DELTAV(NDF),IMAXM(NDF)
      ITREAC=0
C
      DO 9 I=1,NINV
      ANORM1(I)=0.D0
      DO 10 J=1,NINV
      ANORM1(I)=ANORM1(I)+AA(I,J)*FLEX(J,NINV+1)
10    CONTINUE
9     CONTINUE
      SIG=1.D0
      IF(IRND.EQ.1)SIG=-1.D0
      DO 1 IE=1,NE
      IB=NBEAM(IE,2)
      IK=3*IB-2
      I1B=IK
      I2B=IK+1
      IIJ=I1B-IE-2
      IIK=I2B-IE-2
      II=INT(FLOAT(I1B)/3+0.7)
      IF(II.GT.NN)II=NN
      DELTAV(I1B)=-(RMASS(II)*SIG*ACC(I1B,1)+XX(IIJ)+
     #FLEX(IIJ,NINV+1))
      DELTAV(I2B)=-(RMASS(II)*SIG*ACC(I2B,1)+XX(IIK)+
     #FLEX(IIK,NINV+1))
      DELTAV(I1B)=DELTAV(I1B)*DT/(2.D0*RMASS(II))
      DELTAV(I2B)=DELTAV(I2B)*DT/(2.D0*RMASS(II))
1     CONTINUE
      DELTAV(NDF)=-(RMASS(NN)*SIG*ACC(NDF,1)+XX(NINV)+
     #FLEX(NINV,NINV+1))
      DELTAV(NDF)=DELTAV(NDF)*DT/(2.D0*RMASS(NN))
C
C     UPDATE ACCELERATIONS,VELOCITIES AND DISPLACEMENTS
      DO 4 I=1,NDF
```

```
      IF(IRND.EQ.1)DUMVEL(I)=VEL(I,1)
      ACC(I,2)=ACC(I,1)
      ACC(I,1)=2.D0*DELTAV(I)/DT+SIG*ACC(I,1)
      VEL(I,2)=DUMVEL(I)+DELTAV(I)
      DUMVEL(I)=VEL(I,2)
      U(I,2)=U(I,1)+(VEL(I,1)+VEL(I,2))*DT/2.D0
4     CONTINUE
C
      DO 2 I=1,NINV
      IE=INT(FLOAT(I)/2+0.6)
      II=I+3+IE-1
      IF(II.GT.NDF)II=NDF
      XX(I)=XX(I)+FLEX(I,NINV+1)
C
2     CONTINUE
C
C      CHECK FOR CONVERGENCE
C
      IOUT=1
      DO 5 I=1,NINV
      IE=INT(FLOAT(I)/2+0.6)
      II=I+IE+2
      IF(II.GT.NDF)II=NDF
      IF(DABS(VEL(II,2)).GT.1.D-1)THEN
      IF(DABS(COMPAT(I)/VEL(II,2)-1.D0).GT.1.D-2)IOUT=0
      END IF
5     CONTINUE
C
      IF(IRND.EQ.1)IOUT=0
C
C      SET UP VELOCITY FOR MODE DETERMINATION
      IF(IOUT.EQ.1)THEN
      IF(IRND.LE.5)THEN
      DT=DT*1.5
      WRITE(IPRINT,123)
123   FORMAT(///,15X,'STABLE SOLUTION : INCREASE TIME STEP BY
     #FACTOR OF 1.5  ',///)
      END IF
      DO 6 I=1,NDF
      VEL(I,1)=VEL(I,2)
6     CONTINUE
      END IF
C
      DO 3 IE=1,NE
      I=NBEAM(IE,2)
      II=2*I+IE-1
      IJ=II+1
      COORDX(I)=XCOORD(I)+U(II,2)
      COORDY(I)=YCOORD(I)+U(IJ,2)
 3    CONTINUE
C
      RETURN
      DEBUG SUBCHK
      END
C
```

```
C       **************************************
C
C                     C H E C K
C
C       CHECKS IF RESULTS OBTAINED FROM THE
C       DIRECT METHOD OF ANALYSIS HAVE CONVERGED
C       ONTO THE INSTANTANEOUS MODE SOLUTIONS.
C       THIS CHECK IS ONLY PERFORMED IF THE
C       COMBINED DIRECT AND MODE SOLUTION OPTION
C                     IS EMPLOYED.
C
C       **************************************
        SUBROUTINE CHECK
        INCLUDE DAGNVS.COMPROC
C       FOR MODE STRUCTURES ONLY
        IF(ISYM.EQ.0)THEN
        A=0.D0
        B=0.D0
        DO 1 I=1,NDF,3
        II=I+1
        A=A+(VMODE(I,2)-VEL(I,2))*(VMODE(I,2)-VEL(I,2))
        B=B+VMODE(I,2)*VMODE(I,2)
        A=A+(VMODE(II,2)-VEL(II,2))*(VMODE(II,2)-VEL(II,2))
        B=B+VMODE(II,2)*VMODE(II,2)
1       CONTINUE
        A=DSQRT(A)/DSQRT(B)
        IF(A.LT.0.05D0)INMSOL=1
        IF(INMSOL.EQ.1)THEN
        WRITE(IPRINT,100)
100     FORMAT(1H ,///,20X,'MODE SOLUTION REACHED : TREBLE TIME
       #STEP',///)
        DT=DT*3.D0
        END IF
        END IF
C        SET UP VELOCITY ARRAYS FOR DIRECT ANALYSIS PROCEDURE
C        RESET DISPLACEMENT ARRAY.
        DO 2 I=1,NDF
        U(I,1)=U(I,2)
C       IF MODE FOUND STORE CURRENT DISPLACEMENTS
        IF(INMSOL.EQ.1)THEN
        UMODE(I)=U(I,2)
        VMODE(I,1)=VMODE(I,2)
        VEL(I,1)=VMODE(I,2)
        VEL(I,2)=VMODE(I,2)
        U(I,1)=0.D0
        U(I,2)=0.D0
        END IF
        VEL(I,1)=VEL(I,2)
2       CONTINUE
C
C       STORE INITIAL VALUES OF T(T) AND DT(T)/DT
C
        IF(INMSOL.EQ.1)THEN
        DTTDT(1)=DTTDT(2)
        TT(1)=TT(2)
        END IF
        RETURN
        DEBUG SUBCHK
        END
```

```
C
C       ************************************
C
C                    S T O R E
C
C       REVISES AND STORES DISPLACEMENTS
C       AND CURRENT GEOMETRY OF STRUCTURE IN
C       INSTANTANEOUS MODE SOLUTION TECHNIQUE
C       NOTE THAT THIS TECHNIQUE IS ONLY
C       EMPLOYED WHEN COMBINED MODE AND
C       DIRECT ANALYS IS REQUESTED (ISYM=0)
C
C       ************************************
        SUBROUTINE STORE
        INCLUDE DAGNVS.COMPROC
C
C        UPDATES DISPLACEMENTS IN LARGE DISPLACEMENT
C       MODE SOLUTION PROCEDURE
C
        DO 1 I=1,NDF
        U(I,2)=(VMODE(I,1)+VMODE(I,2))*DT/2.D0
C
1       CONTINUE
C
        DO 2 IE=1,NE
        I=NBEAM(IE,2)
        II=2*I+IE-1
        IJ=II+1
        COORDX(I)=COORDX(I)+U(II,2)-U(II,1)
        COORDY(I)=COORDY(I)+U(IJ,2)-U(IJ,1)
        UMODE(II)=COORDX(I)-XCOORD(I)
        UMODE(IJ)=COORDY(I)-YCOORD(I)
        U(II,1)=U(II,2)
        U(IJ,1)=U(IJ,2)
2       CONTINUE
C
        RETURN
        DEBUG SUBCHK
        END
C
C
C       ********************************
C                U P D A T E
C
C       UPDATES VELOCITY AND AMPLITUDE
C       QUANTITIES IN INSTANTANEOUS MODE
C       SOLUTION TECHNIQUE .NOTE THAT
C       THIS TECHNIQUE IS ONLY USED WHEN
C       COMBINED MODE AND DIRECT SOLUTION
C       PROCEDURE IS EMPLOYED (ISYM=0)
C
C       ********************************
        SUBROUTINE UPDATE
        INCLUDE DAGNVS.COMPROC
C
C        SET UP INITIAL VELOCITY ARRAY FOR NEW MODE DETERMINATION
C
        A=0.D0
        B=0.D0
```

```
      DO 1 I=1,NDF
      VMODE(I,1)=VMODE(I,2)
C
C      DETERMINE NEW AMPLITUDE
C
      II=INT(FLOAT(I)/NF+0.7)
      IJ=(INT(FLOAT(I)/NF+0.001))*NF
      IF(I.NE.IJ)THEN
      A=A+VMODE(I,1)*RMASS(II)*PHI(I,1)
      B=B+PHI(I,1)*RMASS(II)*PHI(I,1)
      END IF
C
 1    CONTINUE
      AMP(2)=A/B
      TT(1)=1.D0
      DTTDT(1)=DTTDT(2)
C
      RETURN
      DEBUG SUBCHK
      END
C
C     **********************************
C
C              P I C T U R
C
C     IF PICT EQUALS 'PLOT' THEN PLOTS
C     ARE CREATED OF DEFORMED SHAPE OF
C     THE STRUCTURE CORRESPONDING TO
C     TIME STEP WHEN RESULTS ARE OUTPUT
C
C     **********************************
      SUBROUTINE PICTUR
      INCLUDE DAGNVS.COMPROC
      DOUBLE PRECISION XMAX,YMAX
C
      CALL NEWPAG
      CALL PAGSIZ(20.5,29.)
      CALL PLOT(2.0,5.0,-3)
C
      XMAX=-999.D0
      YMAX=-999.D0
      DO 22 J=1,IPLTS
      DO 1 I=1,NN
      IF(DABS(XCOPLT(I,J)).GE.XMAX)XMAX=DABS(XCOPLT(I,J))
      IF(DABS(YCOPLT(I,J)).GE.YMAX)YMAX=DABS(YCOPLT(I,J))
 1    CONTINUE
 22   CONTINUE
      RMAX=DMAX1(XMAX,YMAX)
      FX=18.0/RMAX
      FY=18.0/RMAX
C
      DO 21 J=1,IPLTS
      DO 2 I=1,NN
      XCOPLT(I,J)=XCOPLT(I,J)*FX
      YCOPLT(I,J)=YCOPLT(I,J)*FY
 2    CONTINUE
      CALL PLOT(0.,0.,3)
      DO 3 I=1,NN
      CALL PLOT(XCOPLT(I,J),YCOPLT(I,J),2)
```

```
      CALL SYMBOL(XCOPLT(I,J),YCOPLT(I,J),0.2,11,0.,-1)
3     CONTINUE
21    CONTINUE
      RETURN
      DEBUG SUBCHK
      END
C
C     **********************************
C
C                O U T P U T
C
C     OUTPUTS RESULTS OF ANALYSIS AFTER
C     NDIV TIME STEPS AND WHEN STRUCTURE
C         IS AT OR NEAR REST
C
C     **********************************
C
      SUBROUTINE OUTPUT
      INCLUDE DAGNVS.COMPROC
C
      IF(T.LT.1.D-9)GO TO 23
C
C      OUTPUT RESULTS OF MODE SOLUTION
3     FORMAT(1H ,I3,3X,E13.6,6X,E13.6,9X,E13.6,/)
9     FORMAT(1H ,2X,I2,8X,E13.6,4X,E13.6,4X,E13.6,/)
4     FORMAT(1H ,2X,I2,8X,E13.6,4X,E13.6,/)
      AMPERC=100.*AMP(2)/AMP(1)
      WRITE(IPRINT,11)T,DT,AMPERC
11    FORMAT(1H ,/,' TIME  ',E11.6,15X,'TIME INCREMENT IS',
     #E11.6,15X,'VELOCITY AMPLITUDE (PERCENT)  IS ',F7.3,/)
      IF(INMSOL.EQ.1)THEN
      IF(DISPL.EQ.'LARGE')THEN
      WRITE(IPRINT,60)
60    FORMAT(1H ,18X,'    MODE SHAPE',//,
     #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
      DO 40 I=1,NDF-2,3
      II=INT(FLOAT(I)/NF+0.7)
      WRITE(IPRINT,3)II,PHI(I,1),PHI(I+1,1),PHI(I+2,1)
40    CONTINUE
      END IF
      WRITE(IPRINT,12)
12    FORMAT(1H ,20X,'MOMENTS AND AXIAL FORCES',//,
     #'ELEMENT',8X,'MOMENT(A)',
     #8X,'MOMENT(B)',8X,'AXIAL FORCE',/)
      DO 13 IE=1,NE
      WRITE(IPRINT,9)IE,FORMOD(IE,1),FORMOD(IE,2),FORMOD(IE,3)
13    CONTINUE
      WRITE(IPRINT,14)
14    FORMAT(1H ,/,26X,'VELOCITY',//,
     #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
      DO 15 I=1,NDF-2,3
      II=INT(FLOAT(I)/NF+0.7)
      WRITE(IPRINT,3)II,VMODE(I,2),VMODE(I+1,2),VMODE(I+2,2)
15    CONTINUE
      WRITE(IPRINT,16)
16    FORMAT(1H ,/,21X,'DISPLACEMENTS',//,
     #'NODE',11X,'X',17X,'Y',/)
      DO 17 I=1,NDF-2,3
      II=INT(FLOAT(I)/NF+0.7)
```

```
          WRITE(IPRINT,4)II,UMODE(I),UMODE(I+1)
   17     CONTINUE
C
          END IF
          IF(INMSOL.EQ.0.AND.ISYM.EQ.0)THEN
C
          WRITE(IPRINT,18)
   18     FORMAT(1H ,/,33X,'VELOCITIES',//,'NODE',
         #11X,'X(DIRECT)',10X,'X(MODE)',
         #16X,'Y(DIRECT)',12X,'Y(MODE)',/)
          DO 19 I=1,NDF-2,3
          II=INT(FLOAT(I)/NF+0.7)
          WRITE(IPRINT,20)II,VEL(I,1),VMODE(I,2),VEL(I+1,1),
         #VMODE(I+1,2)
   20     FORMAT(1H ,2X,I2,7X,E13.6,6X,E13.6,9X,E13.6,9X,E13.6,/)
   19     CONTINUE
C
          WRITE(IPRINT,16)
          DO 22 I=1,NDF-2,3
          II=INT(FLOAT(I)/NF+0.7)
          WRITE(IPRINT,4)II,U(I,1),U(I+1,1)
   22     CONTINUE
          END IF
C
          IF(ISYM.EQ.1)THEN
          WRITE(IPRINT,24)
   24     FORMAT(1H ,/,26X,'VELOCITY',//,'   NODE',6X,'X',17X,'Y',/)
          DO 25 I=1,NDF-2,3
          II=INT(FLOAT(I)/NF+0.7)
          WRITE(IPRINT,4)II,VEL(I,1),VEL(I+1,1)
   25     CONTINUE
          WRITE(IPRINT,16)
          DO 26 I=1,NDF-2,3
          II=INT(FLOAT(I)/NF+0.7)
          WRITE(IPRINT,4)II,U(I,1),U(I+1,1)
   26     CONTINUE
          END IF
C
   23     CONTINUE
C         STORE CURRENT COORDINATES
          IF(PICT.EQ.'PLOT')THEN
          IPLTS=IPLTS+1
          DO 21 I=1,NN
          XCOPLT(I,IPLTS)=COORDX(I)
          YCOPLT(I,IPLTS)=COORDY(I)
   21     CONTINUE
          END IF
C
          RETURN
          DEBUG SUBCHK
          END
C
C
```

```
C         ******************************
C
C                  C O M P R O C
C
C         COMMON BLOCK OF ALL ARRAYS AND
C           PARAMETERS USED IN DAGNVS.
C
C         ******************************
COMPROC PROC
      PARAMETER NE=5
      PARAMETER NN=6
      PARAMETER NRED=4
      PARAMETER NF=3
      PARAMETER IREAD=8
      PARAMETER IPRINT=5
      PARAMETER NDF=NN*NF
      COMMON/BLK1/NBEAM(NE,2),COORDX(NN),COORDY(NN),
     #RMASS(NN),IBC(NDF),RELEAS(NRED),COMPAT(2*NN),
     #UNITN(NDF,NE),SSIN(NE),CCOS(NE),FORCE(2*NN,NE,3),
     #FLEX(2*NN,2*NN+1),PHI(NDF,2),DISIP(2),RLOAD(NDF),
     #XX(2*NN),X(NRED),FORCET(NE,3),AMOM(2*NN),ANORM1(2*NN),
     #VMODE(NDF,2),UMODE(NDF),FORMOD(NE,3),TF,RINT,
     #XCOPLT(NN,20),YCOPLT(NN,20),XCOORD(NN),YCOORD(NN),
     #U2(NDF),EPSI2(NE,2),RLO(NE),ACC(NDF,2),ISTAT(2*NN),
     #VINIT(NDF),TIME,PARDIF(2*NN,2*NN),UNITM(NDF,NN),CL(NE),
     #DTTDT(2),HH,BB,VEL(NDF,2),TT(2),ANORM2(2*NN),DT,
     #RLAMDA,U(NDF,2),AA(2*NN,2*NN),DUMVEL(NDF)
      COMMON/BLK2/RM0,RN0,EPSI0,EN,EN1,EN2,STADET,TITLE,
     #DISPL,PICT,RK,YSTRS,RMATCH,PMATCH,AMP(2),XXDUM
      COMMON/BLK3/IFLAG,IDISIP,ITREAC,ITMODE,NNORM,IOUT,
     #ICOUNT,NDIVA,IPLTS,MATCHA,INMSOL,MODES,NINV,
     #IRND,NDIV,INEWT,MCOUNT,ISYM,IMATCH
      COMMON/BLK4/PARTA,PARTB,SUMA,SUMB,SUMC,SUMD,POWA,
     #POWB,POWC,POWD,POWE,POWF,POWG,POWH,POWI,POWJ,POWK,
     #POWL,PRODA,PRODB,PRODC,PART1,PART2,PART3,PART4,PART5,
     #PART6,PROD1,PROD2,PROD3,PROD4,PROD5,PROD6,PROD7,PROD8,
     #CONST1,CONST2,CONST3,CONST4,CONST5,CONST6,CONST7,
     #SIG1,SIG2,SIG3,SIG4,SIG,SIGA,SIGB,SIGC,SIGD,SUM1,SUM2
      CHARACTER STADET*4
      CHARACTER TITLE*80
      CHARACTER DISPL*5
      CHARACTER PICT*4
      DOUBLE PRECISION COORDX,COORDY,PARDIF,RMASS,COMPAT,
     #UNITM,CL,UNITN,SSIN,CCOS,FORCE,FLEX,PHI,RLOAD,VEL,
     #TT,ANORM1,ANORM2,RM0,RN0,EPSI0,EN,EN1,EN2,VMODE,
     #T,X,AMOM,DTTDT,RK,UMODE,FORMOD,XX,TIME,ACC,
     #AMP,ANORM,FACT,EPSI2,RL0,AA,U2,DUMVEL,U2MODE,PARTA,
     #PARTB,SUMA,SUMB,SUMC,SUMD,POWA,POWB,POWC,POWD,POWE,
     #POWF,POWG,POWH,POWI,POWJ,POWK,POWL,PRODA,PRODB,PRODC,
     #PART1,PART2,PART3,PART4,PART5,PART6,PROD1,PROD2,PROD3,
     #PROD4,PROD5,PROD6,PROD7,PROD8,CONST1,CONST2,CONST3,
     #CONST4,CONST5,CONST6,CONST7,SIG1,SIG2,SIG3,SIG4,SIG,
     #SIGA,SIGB,SIGC,SIGD,SUM1,SUM2,RMATCH,PMATCH,XXDUM
      INTEGER RELEAS
  END
```

# APPENDIX D

## Published Work

The following works, co-authored with Professor J.B. Martin, have been accepted for publication.

i) P.D. Griffin and J.B. Martin, "Finite Element Analysis of Dynamically Loaded Homogeneous Viscous Beams". To appear in the Journal of Structural Mechanics.

ii) P.D. Griffin and J.B. Martin, "Geometrically Nonlinear Mode Approximations for Impulsively Loaded Homogeneous Beams and Frames". To appear in the International Journal of Mechanical Sciences.

iii) P.D. Griffin and J.B. Martin, "The Prediction of Large Permanent Deformations in Rigid-Plastic Impulsively Loaded Frames". To appear in the D.C. Drucker Anniversary Volume, and to be presented at the Conference of Mechanics of Material Behaviour, Urbana-Champaign, Illinois, June 1983.