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. The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

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 rigidviscoplastic 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.

(ii)

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

TITLE PAGE		(i)
ABSTRACT		(ii)
DECLARAT	ION	(iii)
ACKNOWLE	DGEMENTS	(iv)
CONTENTS		(v)
NOMENCLA	FURE	(vii)
1.	INTRODUCTION	1
2.	THE MODE APPROXIMATION TECHNIQUE	13
2.1	The basis of the Mode Approximation Technique	13
2.2	An Algorithm for the Determination of the Mode Shape	16
2.3	Proof of Convergence of the Mode Algorithm	18
2.4	Numerical Implementation of the Mode Approximation Technique	22
2.5	The Instantaneous Mode Technique	28
3.	DIRECT METHODS OF ANALYSIS	32
3.1.1	General Formulation of Geometrically Linear Problems Based on the Tamuzh Principle	33
3.1.2	Numerical Formulation of the Problem	35
3.2	Direct Solutions Using An Implicit Forward Integration Scheme	43
3.2.1	Initial Moments and Axial Forces in the Structure	45
3.2.2	An Implicit Time Integration Scheme	47
4.	THE MATCHING PROCEDURE	54
5.	COMPUTER IMPLEMENTATION OF THE MODE AND THE CONVENTIONAL DIRECT SOLUTION TECHNIQUE	58

1			١.
C	v	l)

5.1	Numerical Implementation of the Instantaneous Mode Technique Using GNLIMST	58
5.2	Numerical Implementation of the Conventional Direct Solution Technique Using DAGNVS	64
6	ILLUSTRATIVE EXAMPLES	72
7	CONCLUSIONS	89
	REFERENCES	90
APPENDIX A	GNLIMST and DAGNVS User Manual	A.1 - A.12
APPENDIX B	Program Listing of GNLIMST	B.1 - B.26
APPENDIX C	Program Listing of DAGNVS	C.1 - C.39
APPENDIX D	Published Work	

NOMENCLATURE

SPECIAL SYMBOLS

[]	a matrix
c ~	a vector c
• c	the differential of c with respect to time
c	the absolute value of c
T(superscript)	the transpose of a matrix
-1(superscript)	the inverse of a matrix
d	differentiation with respect to
9	partial differentiation with respect to

LOWER CASE CHARACTERS

h	rectangular section depth
l	length of an element
n	power in constitutive relation
p	generalised loads
• q	generalised strain rate
s	spacial variable (two-dimensional)
t	time variable
u	displacement
ů	velocity
ü	acceleration
x	spacial variable (one-dimensional)

.

UPPER CASE CHARACTERS

А	rectangular section area
D	dissipation rate
M	moments
M	moment rate
Mo	yield moment
N	axial force
No	axial yield force
Q	generalised internal forces
Т	time function in mode analysis
Х	global cartesian X–axis
Y	global cartesian Y – axis

MATRICES AND VECTORS

[B]	deformation matrix
[G]	lumped mass matrix
ĭ	impulse vector
M	moment vector
™ ∼	moment rate vector
[m]	influence matrix of nodal moments
N ~	axial force vector
[n]	influence matrix of element axial forces
P ~	load vector
u ~	displacement vector
• u ~	velocity vector
ü	acceleration vector
X ~	nodal force vector
ε ~	strain rate vector
• ~	curvature rate vector
ф ~	mode shape vector

GREEK CHARACTERS

γ	specific mass
Δ	increment in
• ε	axial strain rate
ε _o	strain rate material constant
θ	rotation
$\dot{\Theta}$	rotation rate
 Ө	rotational acceleration
• K	curvature rate
Ко	curvature rate material constant
μ	stress matching factor
ν	power matching factor
σο	yield stress
φ	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
0	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

$$\frac{\varepsilon}{\varepsilon_{o}} = \left(\frac{\sigma}{\sigma_{o}} - 1\right)^{n} \quad \text{for } \sigma \ge \sigma_{o} \quad ;$$

$$\varepsilon = 0 \quad \text{for } \sigma \le \sigma \le \sigma_{o} \quad .$$
(1.1)

In this equation $\tilde{\epsilon}$, σ are strain rate and stress respectively and $\tilde{\epsilon}_{o}$, σ_{o} are material constants with the dimensions of strain rate and

stress respectively. The constants ε_0 , σ_0 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 rigidplastic structures. For such structures in which displacements are small, mode solutions are admitted in which the velocity field $\underbrace{u^m}_{x}(s,t)$ is given by the product of a function \oint_{∞} of the spatial variable s and a function T of time t;

$$u^{m}(s,t) = \phi(s)T(t)$$
 (1.2)

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

$$\Delta^{o} = \int \frac{\gamma(s)}{2} \left(\underbrace{\overset{o}}{\overset{u}}{\overset{o}}{(s)} - T(o) \phi(s) \right) \left(\underbrace{\overset{o}}{\overset{u}}{\overset{o}}{(s)} - T(o) \phi(s) \right) ds \quad , \quad (1.3)$$

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 \triangle 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

$$\frac{\varepsilon}{\varepsilon_{o}} = \left(\frac{\sigma}{\mu\sigma_{o}}\right)^{\nu n} , \qquad (1.4)$$

where μ , ν 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 The response time of the structure is divided into a Chon [13]). number of small intervals Δt , 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 rigidplastic 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 $\dot{u}(s,o) = \dot{u}^{o}(s) = I(s)/\gamma(s)$ 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 $\dot{\epsilon}$ (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)





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

$$N_{o}\dot{\varepsilon} = \frac{\partial \Psi}{\partial N/N_{o}}$$
; $M_{o}\dot{\kappa} = \frac{\partial \Psi}{\partial M/M_{o}}$ (1.5a)

where

$$\Psi = \frac{N \tilde{\epsilon}_{o}}{2(n+1)} \left[\left| \frac{N}{N_{o}} + \frac{M}{M_{o}} \right| - 1 \right| + \left| \frac{N}{N_{o}} - \frac{M}{M_{o}} \right| - 1 \right] (1.5b)$$

and where

$$N_o = A\sigma_o$$
, $M_o = \frac{1}{2}Ah\sigma_o$ (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

$$N_{o}^{\dagger}\dot{\varepsilon} = \frac{\partial \Psi}{\partial N/N_{o}^{\dagger}}$$
; $M_{o}^{\dagger}\kappa = \frac{\partial \Psi}{\partial M/M_{o}^{\dagger}}$ (1.6a)

where

$$\Psi = \frac{\underset{o}{N} \overset{\circ}{\overset{\circ}{\epsilon}} _{0}}{2(n'+1)} \left[\left| \frac{\underset{o}{N}}{\underset{o}{N'}} + \frac{\underset{M}{M}}{\underset{o}{M'}} \right|^{n'+1} + \left| \frac{\underset{N}{N}}{\underset{o}{N'}} - \frac{\underset{M}{M'}}{\underset{o}{M'}} \right|^{n'+1} \right] (1.6b)$$

and where

$$\sigma'_{o} = \mu \sigma_{o}$$
, $n' = \nu n$, $N'_{o} = A\sigma'_{o}$ and $M'_{o} = \frac{1}{2}Ah\sigma'_{o}$ (1.6c)

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



Figure 1.2 Constitutive laws for rate-dependent sandwich beam sections

For a suitable choice of μ , ν , the equns. (1.6) can provide a close approximation to the rigid-viscoplastic relations of equns. (1.5). The choice of μ and ν 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 Mass will be lumped at the nodes in the convenkinematic variables. tional 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 ϕ , 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 $\dot{\epsilon}$, so that

$$N_{o}\dot{\varepsilon} = \frac{\partial \Psi}{\partial N/N_{o}}$$
; $M_{o}\dot{\kappa} = \frac{\partial \Psi}{\partial M/M_{o}}$, (2.1a)

where

$$\Psi = \frac{N_{o} \varepsilon_{o}}{2(n+1)} \left[\left| \frac{N_{o}}{N_{o}} + \frac{M}{M_{o}} \right|^{n+1} + \left| \frac{N_{o}}{N_{o}} - \frac{M}{M_{o}} \right|^{n+1} \right]$$
(2.1b)

and where

$$N_{o} = A\sigma_{o}, \quad M_{o} = \frac{1}{2}Ah\sigma_{o}$$
 (2.1c)

Equations (2.1) are identical to equns. (1.6) with the prime on σ_{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 $\dot{u}^{0}(s)$, with no external loads for t > 0. The solution of the problem will be denoted by the velocity field $\dot{u}(s,t)$, with $\dot{u}(s,o) = \dot{u}^{0}(s)$.

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

$$u^{*m}(s,t) = \phi(s)T(t)$$
, (2.2)

where $\phi(s)$ 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 ϕ and T are properly chosen, is the transient behaviour of the structure before the velocity field adopts the mode shape.

If $\phi(s)$ 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 $\Delta^{O}(\overset{\bullet}{u}(s), \overset{\bullet}{u}^{m}(s,o))$ with respect to T(o), where

$$\Delta^{o} = \frac{1}{2} \int \gamma(s) \left(\underbrace{\overset{\bullet}{u}}_{\sim}^{o}(s) - \underbrace{\overset{\bullet}{u}}_{\sim}^{m}(s, o) \right) \left(\underbrace{\overset{\bullet}{u}}_{\sim}^{o}(s) - \underbrace{\overset{\bullet}{u}}_{\sim}^{m}(s, o) \right) ds \qquad (2.3)$$

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

$$T(o) = \frac{\int \dot{\gamma}(s) \phi(s) \dot{u}^{0}(s) ds}{\int \gamma(s) \phi(s) \dot{\phi}(s) \dot{\phi}(s) ds} \qquad (2.4)$$

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 $\kappa^{m}(s,t)$ and middle surface axial strain rates $\hat{\epsilon}^{m}(s,t)$, respectively, then

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

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

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

where $D(\dot{\kappa}(s,t), \dot{\epsilon}(s,t))$ is a homogeneous dissipation function of degree $\left(\frac{n+1}{n}\right)$ in the components of the strain rate quantities $\dot{\kappa}, \dot{\epsilon}$. Since strain rates and velocities $\dot{u}(s,t)$ 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 $\dot{u}(s,t)$, so that

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

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

$$- T \frac{dT}{dt} \int \gamma(s) \phi(s) \phi(s) ds = T \int D(\phi(s)) ds , \qquad (2.7a)$$

hence

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

The solution of this differential equation can be written as

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

where

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

It can be seen from these expressions that the mode shape $\phi(s)$ must be known, together with the dissipation rate associated with the mode $D(\phi(s))$ and the energy of the mode $\int \gamma(s)\phi(s)\phi(s)ds$, in order to determine T(t). All this information is provided by a new algorithm for the determination of the mode shape $\phi(s)$, 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 $\oint_{\infty}(s)$. 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 $\oint_{\infty}(s)$ 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 $\phi(s)$ 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 $\phi^i(s)$ the given initial velocity \dot{u}_{α}^{o} .
- Step 1 : Apply 'loads' $\gamma(s) \phi^i(s)$ to the structure, and determine corresponding moments M and axial forces N in the structure.
- Step 2 : Compute velocities uⁱ(s) resulting from this loading, using the principle of virtual velocities.
- Step 3 : Normalise the velocities $\dot{u}^{i}(s)$ by dividing by the product $\int \gamma(s) \phi^{i}(s) \phi^{i}(s) ds$, to give a new trial mode shape,

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

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

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, ... n) and associated generalised strain rates \dot{q}_j in the form

$$Q_{j} = \frac{\partial}{\partial \dot{q}_{j}} \left\{ \frac{n}{n+1} D(\dot{q}_{j}) \right\} , \qquad (2.10)$$

where $D(q_j)$ is the dissipation function,

$$D(\dot{q}_{j}) = Q_{j} \dot{q}_{j}$$
(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

$$\int Q_{j} dq_{j} = \frac{n}{n+1} Q_{j} q_{j} = \frac{n}{n+1} D(q_{j}) . \qquad (2.12)$$

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

$$J(\overset{\bullet}{u}) = \int \frac{n}{n+1} D(\overset{\bullet}{u}) ds - \lambda \int \gamma \overset{\bullet}{u} \overset{\bullet}{u} ds , \qquad (2.13)$$

where the generalised strain rates q_j and velocities \dot{u} are related through a linear set of strain rate-displacement rate equations and λ 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 $\phi^i(s)$, and seek to minimise

$$J^{*}(\overset{\bullet}{u}) = \int \frac{n}{n+1} D(\overset{\bullet}{u}) ds - \int \gamma \phi^{i} \overset{\bullet}{u} ds \qquad (2.14)$$

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

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

 $\int \gamma \phi \phi ds = A$ (2.15)

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

$$\phi_{\alpha}^{i+1} = \alpha \, \psi_{\alpha}^{i} , \qquad (2.16)$$

where

$$\alpha^{2} \int \gamma \underbrace{\overset{\bullet}{u}^{i}}_{\sim} \underbrace{\overset{\bullet}{u}^{i}}_{\sim} ds = \int \gamma \underbrace{\phi^{i}}_{\sim} \underbrace{\phi^{i}}_{\sim} ds = A \qquad (2.17)$$

Note that α is positive definite. We also define a parameter β such that

$$\beta \int \gamma \phi^{i} \phi^{i} ds = \int \gamma \phi^{i} \dot{u} ds . \qquad (2.18)$$

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

$$\int \gamma \phi^{i} \psi^{i} ds > 0 , \qquad (2.19)$$

and since

$$\int \gamma \phi^{i} \phi^{i} ds > 0 , \qquad (2.20)$$

it further follows from equn. (2.18) that β is positive definite.

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

$$\int \frac{n}{n+1} D (\phi) ds \qquad (2.21)$$

If we can show that

$$\int \frac{n}{n+1} D(\phi^{i+1}) ds \leq \int \frac{n}{n+1} D(\phi^{i}) ds , \qquad (2.22)$$

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 $\beta \varphi^{i}$. In view of the requirement that $\overset{i}{u}^{i}$ minimises J* (equn. (2.14)), we have

$$\int \frac{n}{n+1} D \left(\beta \phi^{i}\right) ds - \beta \int \gamma \phi^{i} \phi^{i} ds \qquad (2.23)$$

$$\geq \int \frac{n}{n+1} D \left(\psi^{i}\right) ds - \int \gamma \phi^{i} \psi^{i} ds \quad .$$

Using equn. (2.18), this reduces to

$$\int \frac{n}{n+1} D \left(\beta \phi^{i}\right) ds \geq \int \frac{n}{n+1} \left(\psi^{i}\right) ds \qquad (2.24)$$

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

$$(\alpha\beta)^{\frac{n+1}{n}}\int_{\frac{n}{n+1}}^{\frac{n}{n}} D(\phi^{i}) ds \geq \int_{\frac{n}{n+1}}^{\frac{n}{n+1}} D(\phi^{i+1}) ds \quad . \qquad (2.25)$$

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

$$(\alpha\beta) < 1$$
 . (2.26)

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

$$\int \gamma \phi^{i} \psi^{i} ds = \beta \int \gamma \phi^{i} \phi^{i} ds = \frac{1}{\alpha} \int \gamma \phi^{i} \phi^{i+1} ds ,$$

and hence

$$(\alpha\beta) \int \gamma \phi^{i} \phi^{i} ds = \int \gamma \phi^{i} \phi^{i+1} ds \qquad (2.27)$$

Further, it follows that

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

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

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

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

It follows from this that equn. (2.26) holds, and that ϕ^{i+1} is a better approximation to the mode shape than ϕ^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 $\phi(s)$ 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 u(t), u(t) and ü(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

$$K = \frac{1}{2} \stackrel{\bullet}{u}^{T} [G] \stackrel{\bullet}{u} , \qquad (2.30)$$

at any instant, where superscript T denotes the transpose. The mass terms corresponding to constrained velocity components of $\frac{1}{2}$ 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

 ^{l}e , and ^{M}a , ^{M}b are the nodal moments, the bending moment at distance s from node a is given by

$$M(s) = M_{a}(1 - \frac{s}{\ell_{e}}) + M_{b}(\frac{s}{\ell_{e}}) . \qquad (2.31)$$

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 $\dot{\epsilon}$ 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 M, N.

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

$$I = [G]_{u}^{\bullet 0} = [G]_{u}^{\bullet 0}(0) \qquad (2.32)$$

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

 $\phi^{0} = \dot{u}^{0} \qquad (2.33)$

Subsequent trial mode shapes will be denoted by ϕ^i , where superscript i denotes the i-th iteration.

In Step 1 of the algorithm, bending moments M and axial forces N which result from loading $[G]\phi^i$ on the structure must be calculated. For statically determinate structures, M and 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 $\dot{u}(t)$ be denoted by X. The nodal moments M may be expressed as

$$M = M^{S} + [m] X$$
, (2.34)

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

$$N = N^{S} + [n] X$$
, (2.35)

where $N_{\tilde{n}}^{s}$ are the axial forces in the statically determinate released structure resulting from loading $[G]\phi^{i}$, and each row of the influence matrix [n] is the set of axial forces due to a unit value of some component of 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_i of the vector 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 $\dot{\epsilon}$ at each point on the structure in terms of X. We denote the components of u which correspond to constrained but "released" nodes as \dot{u}_j . With the curvature rates $\dot{\kappa}$, axial strain rates $\dot{\epsilon}$ and \dot{u}_j as the kinematic system, and a unit value of the j-th component of 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{u} as

$$\dot{u}_{j} = \sum_{\text{elements}} \left\{ \int_{e} m_{j} \dot{\kappa} ds + n_{j} \dot{\epsilon} \ell_{e} \right\}$$
 (2.36a)

$$F(X)$$
 (2.36b)

where ℓ_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}_i conjugate to components of X, giving

$$F(X) = 0$$
 . (2.37)

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

$$[A(\underline{X})] = \frac{\partial F}{\partial \underline{X}} \qquad (2.38)$$

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

$$x_{\tilde{k}}^{k+1} = x_{\tilde{k}}^{k} - [A^{k}]^{-1} \{ F(x_{\tilde{k}}^{k}) \} .$$
 (2.39)

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

In Step 2, the velocities u^{i} in the structure which result from loading $[G]\phi^{i}$ 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 $[G]\phi^{i}$. The bending moments m_j then become the moments along each element resulting from a unit value of the j-th component of $[G]\phi^i$. 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 $[G]\phi^i$. The axial forces n_j then become the forces in each element resulting from a unit of the j-th component of $[G]\phi^i$. Applying the principle of virtual velocities, the j-th velocity component is

$$\dot{u}_{j} = \sum_{\text{elements}} \left(\int_{\ell} m_{j} \dot{K} ds + n_{j} \dot{\epsilon} \ell_{e} \right)$$
(2.40)

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

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

$$\phi^{i+1} = \frac{\psi^{i}}{\phi^{i}[G]\phi^{i}}$$
(2.41)

A new load $[G]\phi^{i+1}$ is then formed, and the resultant bending moments M, axial forces N and velocities \dot{u}^{i+1} are calculated. Equation (2.41) is reapplied to revise ϕ , and the iterative procedure is repeated until satisfactory convergence of ϕ 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 $[G]\phi$ are applied to the structure, a
work rate balance for this static problem is

$$\phi^{\mathrm{T}}[G]_{\underline{u}} = \int D(\overset{\bullet}{\underline{u}}) \mathrm{ds} \quad , \qquad (2.42)$$

where $D(\overset{\bullet}{u})$ is a homogeneous dissipation function of degree $\frac{n+1}{n}$. If α^* is a factor such that

$$\oint_{\alpha} = \alpha * \underbrace{\mathbf{u}}_{\alpha} , \qquad (2.43)$$

we have from equn. (2.42)

$$\frac{1}{\alpha^{\star}} \phi^{\mathrm{T}}[G] \phi = \frac{1}{(\alpha^{\star})} \int D(\phi) \, \mathrm{ds} \quad , \qquad (2.44a)$$

hence

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

$$\mathbf{u}(\mathbf{s},\mathbf{t}) = \phi^{\mathbf{t}} \mathbf{T}^{\mathbf{t}} , \qquad (2.45)$$

where $\phi_{\tilde{L}}^{t}$ 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))

$$\dot{\mathbf{T}}^{\mathsf{t}} = - (\mathbf{T}^{\mathsf{t}})^{1/n} \frac{\int D(\phi^{\mathsf{t}}) ds}{\int \gamma \phi^{\mathsf{t}} \phi^{\mathsf{t}} ds} \qquad (2.46)$$

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 predictor-

corrector method with an average rate of change is used. Hence with t + 1 denoting the instant $t + \Delta t$, we put

$$T^{t+1} = T^{t} + \frac{\Delta t}{2} (T^{t} + T^{t+1}) , \qquad (2.47a)$$

$$u_{\mu}^{t+1} = u_{\mu}^{t} + \frac{\Delta t}{2} (T^{t} \phi^{t} + T^{t+1} \phi^{t+1}) . \qquad (2.47b)$$

In applying these equations we assume that T^{t} , ϕ^{t} and \dot{u}^{t} are known. This is not sufficient information to compute T^{t+1} , 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

$$T_{i+1}^{t+1} = T^{t} + \frac{\Delta t}{2} (\dot{1}^{t} + \dot{T}_{i}^{t+1}) , \qquad (2.48a)$$

$$u_{i+1}^{t+1} = u_{i}^{t} + \frac{\Delta t}{2} (T\phi_{i}^{t} + T_{i}^{t+1}\phi_{i}^{t+1}) . \qquad (2.48b)$$

The initial values of \dot{T}_{i}^{t+1} , ϕ_{i}^{t+1} are taken as \dot{T}^{t} , ϕ^{t} ; thereafter the updated configuration (obtained using u^{t+1}) is used to recompute ϕ^{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} , 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 A fundamental assumption implicit in the technique is that the shape. predominant pattern of behaviour of the structure throughout the timespan 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 $\dot{u}^{0}(x,t) = \dot{u}(x,o)$ 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

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

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

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

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

The constitutive equation will be assumed to take the form

$$\frac{\kappa}{\kappa_{o}} = \left(\frac{M}{M_{o}'}\right)^{n'}, \qquad (3.3a)$$

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

$$\frac{\kappa}{\kappa_{o}} = \left(\frac{M}{M_{o}} - 1\right)^{n} \quad \text{for } n \geq M_{o} \quad ,$$

$$\kappa = 0 \quad \text{for } 0 \leq M \leq M_{o} \quad .$$

$$(3.3b)$$

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 ü as those which provide an unconstrained minimum of the functional

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

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 $\dot{\mathbf{u}}$ and a rotation $\dot{\boldsymbol{\theta}}$. 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



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{u} and acceleration vector \ddot{u} comprising only the unconstrained transverse velocity and acceleration components, respectively at each node,

$$\mathbf{u}_{\sim}^{\bullet} = (\mathbf{u}_{1}^{\bullet} \mathbf{u}_{2}^{\bullet} \dots)^{\mathrm{T}}$$
(3.5a)

and

$$\ddot{u}_{\sim} = (\ddot{u}_1 \ \ddot{u}_2 \ \dots \)^{\mathrm{T}}$$
 (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 P, conjugate to 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

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

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

$$\int_{S} M \ddot{\kappa} dx = \ddot{u}^{T} [B] M_{\tilde{\kappa}} , \qquad (3.7)$$

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

$$[B]_{e} = \begin{bmatrix} 1/_{\ell} & -1/_{\ell} \\ e & e \\ -1/_{\ell} & 1/_{\ell} \\ e & e \end{bmatrix}$$
(3.8)

Tamuzh's functional now becomes

$$J = \frac{1}{2} \ddot{u}^{T}[G]\ddot{u} - \ddot{u}^{T}P + \ddot{u}^{T}[B] M , \qquad (3.9)$$

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

$$[G]\ddot{u} = P - [B] M$$
 . (3.10a)

The solution

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

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

In order that the procedure may move forward in time, however, we must be able to determine the moments M at $(t+\Delta t)$ given the velocities \ddot{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

$$-\frac{1}{\ell_{e}}\dot{\mathbf{u}}_{a} - \dot{\theta}_{a} + \frac{1}{\ell_{e}}\dot{\mathbf{u}}_{b} = \int_{ab}^{M_{1}}\dot{\mathbf{k}}d\mathbf{x} , \qquad (3.11a)$$

$$\frac{1}{\ell_{e}}\dot{\mathbf{u}}_{a} - \frac{1}{\ell_{e}}\dot{\mathbf{u}}_{b} + \dot{\theta}_{b} = \int_{ab}^{M_{2}}\dot{\mathbf{k}}d\mathbf{x} , \qquad (3.11b)$$

where

$$M_{1} = 1 - x/\ell_{e}$$

$$M_{2} = x/\ell_{e}$$

$$\kappa_{o} \left(\frac{M}{M_{o}^{\dagger}}\right)^{n'}$$

$$= \kappa_{o} \left\{\frac{M_{a}(1-x/\ell_{e}) + M_{b}(x/\ell_{e})}{M_{o}^{\dagger}}\right\}^{n'}$$
(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

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

In this equation

$$C_{1} = \frac{n'\kappa_{o}}{M_{o}'n'} \int_{ab}^{n'-1} (1-x/\ell_{e})^{2} \{M_{a}(1-x/\ell_{e}) + M_{b}(x/\ell_{e})\} ds , \qquad (3.13a)$$

$$C_{2} = \frac{n'\kappa_{o}}{M_{o}'n'} \int_{ab} (1-x/\ell_{e}) (x/\ell_{e}) \{ M_{a}(1-x/\ell_{e}) + M_{b}(x/\ell_{e}) \}^{n'-1} ds , \quad (3.13b)$$

and

$$C_{3} = \frac{n'_{o}}{M_{o}'^{n'}} \int_{ab} (x/\ell_{e})^{2} \{ M_{a}(1-x/\ell_{e}) + M_{b}(x/\ell_{e}) \}^{n'-1} ds .$$
(3.13c)

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

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

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

$$\begin{bmatrix} D \end{bmatrix} \begin{bmatrix} \mathbf{\dot{M}} \\ -\frac{\mathbf{\ddot{C}}}{\mathbf{\dot{\theta}}} \\ -\frac{\mathbf{\ddot{\theta}}}{\mathbf{\dot{\theta}}} \end{bmatrix} = - \begin{bmatrix} B \end{bmatrix} \mathbf{\ddot{u}} \quad . \tag{3.15}$$

Once \ddot{u} is known, equn. (3.15) enables \dot{h} and the rotational accelerations at the unconstrained nodes $\ddot{\theta}$ 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+ Δ 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{u}_t and \dot{M}_t , the rates of change of velocity and moment. If t+1 denotes the time step (t+ Δ t), a first estimate of \dot{u}_{t+1} and M_{t+1} is

$$\overset{\bullet}{\underset{t+1}{\overset{\bullet}{_{t+1}}} = \overset{\bullet}{\underset{t}{\overset{\bullet}{_{t}}} + \Delta t \overset{"}{\underset{t}{\overset{"}{_{t}}} , \qquad (3.16a)$$

and

$$M_{t+1} = M_{t} + \Delta t \dot{M}_{t} . \qquad (3.16b)$$

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

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

$$\underset{\sim t+1}{\overset{u}{}} = \underset{\sim t}{\overset{u}{}} + \frac{\Delta t}{2} (\underset{\sim t}{\overset{u}{}} + \underset{\sim t+1}{\overset{u}{}}) ,$$
 (3/17a)

and

$$M_{t+1} = M_{t} + \frac{\Delta t}{2} \left(M_{t+1} + M_{t+1} \right)$$

This process of refining the estimate of \dot{u}_{t+1} and M_{t+1} and recomputing \ddot{u}_{t+1} , \dot{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{u}_t is known, and we can compute the mode velocity \dot{u}_t^m from equn. (2.2). Using equn. (3.10b) and from equn. (2.2), we can find \ddot{u}_t and \ddot{u}_t^m , respectively. The difference $(\ddot{u}_t - \ddot{u}_{t+1}^m)$ is then formed. We use this rate of change, by the same Euler method described above, to determine $(\dot{u}_{t+1} - \dot{u}_{t+1}^m)$ and hence find \dot{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 formu-

lation 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 u(t), u(t) and 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

$$K = \frac{1}{2} \underbrace{\stackrel{\circ}{u}}_{u}^{T} [G] \underbrace{\stackrel{\circ}{u}}_{u}, \qquad (3.18)$$

at any instant. The mass terms corresponding to constrained velocity components of \dot{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 I. The impulsive load imparts an initial velocity to each node, given by

$$I = [G]\dot{u}^{0} = [G]\dot{u}(0) . (3.19)$$

We wish to determine the resulting motion of the structure, with initial displacements u(o) = 0 and initial velocities $\dot{u}(o)$ 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 $\dot{u}^{0} = \dot{u}(0)$ completely prescribed. Note that $\dot{u}(0)$ 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 u(0) = 0.

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

$$M = [m]X$$
 . (3.20)

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

$$M(s) = M_a (1 - \frac{s}{\ell_e}) + M_b (\frac{s}{\ell_e}) \qquad (3.21)$$

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 X.

The axial forces are constant along each element, and are represented by an element axial force vector N, given by

$$N = [n]X$$
 (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 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 X_i.

Using the constitutive equn. (2.1) we can write the curvature rate $\dot{\kappa}$ and the strain rate $\dot{\epsilon}$ at each point on the structure in terms of X. With the curvature rates $\dot{\kappa}$, axial strain rates $\dot{\epsilon}$ and velocities $\ddot{\dot{\nu}}$ as the kinematic system, and a unit value of the j-th component of 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}_i of \dot{u} as

$$u_{j} = \sum_{\text{elements } l_{e}} \int_{e}^{m_{j} \star ds} + n_{j} \dot{\epsilon} l_{e} \qquad (3.23)$$

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

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

It is a straight forward computational problem to determine u given X; we require, however, X given 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

$$\dot{u} - F(X) = 0$$
, (3.25)

and a matrix of partial derivatives of F with respect to X is defined,

$$[A(X)] = \frac{\partial F}{\partial X} \qquad (3.26)$$

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

$$x_{x}^{k+1} = x_{x}^{k} - [A^{k}]^{-1} \{ F(x_{x}^{k}) - \dot{u} \} .$$
 (3.27)

The process is repeated until an estimate of X of acceptable accuracy is obtained. At this point, the moment and axial force vectors $M_{\tilde{e}}$, $N_{\tilde{e}}$ 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 $\dot{u}(t)$ and the configuration, described by $\dot{u}(t)$, 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+ Δ t respectively, and let superscript i denote the i-th iteration in the algorithm which will be outlined below. At time t velocities \dot{u}_{t} and displacements u_{t} are known, as are the nodal forces X_{t} . The nodal forces at time t = 0 are calculated by the procedure set out in the previous section; thereafter 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,

$$[G]\ddot{u}_{t} + X_{t} = 0 , \qquad (3.28a)$$

or

$$\ddot{u}_{t} = - [G]^{-1} \chi_{t}$$
 (3.28b)

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

$$u_{t+1} = F(X_{t+1}) = F_{t+1}$$
 (3.29)

It is implicitly assumed that the function $F_{,}$ evaluated according to equns. (3.23) and (3.24), refers to the geometry of the structure at time t+1. Thus $F_{,}$ can be found only when $u_{,}$ (or an estimate of $u_{,}$ 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{u} , \dot{u} , u and χ are defined by the equations

$$\ddot{u}_{t+1} = \ddot{u}_{t} + \Delta \ddot{u} ,$$

$$\dot{u}_{t+1} = \dot{u}_{t} + \Delta \dot{u} ,$$

$$\dot{u}_{t+1} = u_{t} + \Delta u ,$$

$$\dot{u}_{t+1} = u_{t} + \Delta u ,$$

$$\dot{u}_{t+1} = \dot{u}_{t} + \Delta \dot{u} ,$$

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

$$[G](\ddot{u}_{t} + \Delta \ddot{u}) + (\chi_{t} + \Delta \chi) = 0 , \qquad (3.31a)$$

hence

and

$$[G] \Delta \ddot{u}_{t} + \Delta \chi = - ([G] \ddot{u}_{t} + \chi_{t}) \qquad (3.31b)$$

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

$$\mathbf{u}_{\sim t} + \Delta \mathbf{u}_{\sim} = \mathbf{F}_{\sim t} + [\mathbf{A}_{t}] \Delta \mathbf{X}, \qquad (3.32a)$$

where

$$\begin{bmatrix} A_t \end{bmatrix} = \begin{bmatrix} A(X_t) \end{bmatrix} = \frac{\partial F}{\partial X} \begin{vmatrix} t \\ 0 \end{vmatrix} t , \qquad (3.32b)$$

and is given by the last evaluation of equn. (3.26) in the iterative procedure to determine $X_{\sim t}$, described in the previous section. Note that as the constitutive relation used is homogeneous, the partial

derivatives of F with respect to 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) ,

$$\Delta \dot{\mathbf{u}} = [\mathbf{A}_{t}] \Delta \mathbf{x} + (\mathbf{F}_{t} - \dot{\mathbf{u}}_{t}) \qquad (3.32c)$$

Using the trapezoidal rule, we put

$$\mathbf{u}_{t+1} = \mathbf{u}_{t} + \frac{\Delta t}{2} \left(\mathbf{u}_{t} + \mathbf{u}_{t+1} \right) , \qquad (3.33)$$

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

$$\Delta \ddot{\mathbf{u}} = \ddot{\mathbf{u}}_{t+1} - \ddot{\mathbf{u}}_{t} = \frac{2}{\Delta t} \Delta \dot{\mathbf{u}}_{t} - 2 \ddot{\mathbf{u}}_{t} . \qquad (3.34)$$

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

$$\frac{2}{\Delta t} [G] \Delta \dot{u} + \Delta x = - (-[G] \ddot{u}_{t} + x_{t}) \qquad (3.35)$$

Finally, substituting for Δu from equn. (3.32c), and rearranging, we have

$$\{ [A_t] + \frac{\Delta t}{2} [G^{-1}] \} \Delta x = - (F_t - \dot{u}_t) + \frac{\Delta t}{2} (\ddot{u}_t - [G]^{-1} x_t)$$
(3.36)

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

$$u_{t+1} = u_{t} + \frac{\Delta t}{2} \left(\underbrace{u}_{t} + \underbrace{u}_{t+1} \right) \qquad (3.37)$$

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 $\Delta \dot{\mathbf{u}}$, $\Delta \mathbf{X}$ 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

$$[G]_{t+1}^{u+1} + \chi_{t+1}^{i+1} = 0 , \qquad (3.38)$$

and

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

$$\ddot{u}_{t+1}^{i+1} = \ddot{u}_{t+1}^{i} + \Delta \ddot{u}_{t+1}^{i}$$
, (3.40a)

$$\underset{t+1}{\overset{i+1}{\underset{t+1}{u}}} = \underset{t+1}{\overset{i}{\underset{t+1}{u}}} + \underset{t+1}{\overset{i}{\underset{t+1}{u}}}, \qquad (3.40b)$$

$$u_{t+1}^{i+1} = u_{t+1}^{i} + \Delta u_{t+1}^{i}$$
, (3.40c)

$$x_{t+1}^{i+1} = x_{t+1}^{i} + \Delta x_{t+1}^{i} .$$
 (3.40d)

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

$$\underset{\sim t+1}{\overset{\bullet}{u}} = \underset{\sim t}{\overset{\bullet}{u}} + \frac{\Delta t}{2} (\underset{\sim t}{\overset{\bullet}{u}} + \underset{t+1}{\overset{\bullet}{u}})$$
(3.41a)

and

$$\underbrace{ \overset{i+1}{u}_{t+1}}_{t+1} = \underbrace{ \overset{i}{u}}_{t} + \frac{\Delta t}{2} (\underbrace{ \overset{i}{u}}_{t} + \underbrace{ \overset{i+1}{u}}_{t+1})$$
 (3.41b)

and hence

$$\Delta_{t+1}^{\bullet i} = \frac{\Delta t}{2} \Delta_{t+1}^{i} \qquad (3.41c)$$

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

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

$$\frac{2}{\Delta t} [G] \Delta_{t+1}^{i} + \Delta_{t+1}^{i} = - ([G]_{t+1}^{i} + x_{t+1}^{i}) \qquad (3.43)$$

From equn. (3.32a), we write

where

$$\begin{bmatrix} A_{t+1}^{i} \end{bmatrix} = \frac{\partial F}{\partial X} \begin{vmatrix} x_{t+1}^{i} \\ z \\ t + 1 \end{vmatrix}$$
 (3.44b)

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

In order to find the final expression for Δx_{t+1}^i , we substitute equn. (3.44a), into equn. (3.43), and rearrange so that

$$\{[A_{t+1}^{i}] + \frac{\Delta t}{2} [G]^{-1}\} \Delta x_{t+1}^{i} = -(F_{t+1}^{i} - \tilde{u}_{t+1}^{i}) - \frac{\Delta t}{2} (\tilde{u}_{t+1}^{i} - [G]^{-1} x_{t+1}^{i})$$
(3.45)

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

$$\Delta \dot{u}_{t+1}^{i} = \frac{\Delta t}{2} \Delta \ddot{u}_{t+1}^{i} . \qquad (3.46)$$

We may thus find revised estimates $\overset{\bullet}{\underset{t+1}{\text{u}}} + 1$, $\overset{\bullet}{\underset{t+1}{\text{u}}} + 1$, $\overset{\bullet}{\underset{t+1}{\text{v}}} + 1$, $\overset{\bullet}{\underset{t+1}{\text{v}} + 1$, $\overset{\bullet}$

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 $\sigma^{\text{\prime}}_{\text{o}}$ 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-viscoplastic relation

$$\frac{\varepsilon}{\varepsilon_{o}} = \left(\frac{\sigma}{\sigma_{o}} - 1\right)^{n} \quad \text{for } \sigma \ge \sigma_{o} \quad ,$$

$$\frac{\varepsilon}{\varepsilon_{o}} = 0 \quad \text{for } \sigma < \sigma_{o} \quad ,$$
(4.1)

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

$$\frac{\varepsilon}{\varepsilon_{0}} = \left(\frac{\sigma}{\sigma_{0}'}\right)^{n'}, \qquad (4.2a)$$

with

$$\sigma'_{2} = \mu \sigma_{2}$$
 , (4.2b)

 $n' = vn \qquad (4.2c)$

A strategy for choosing μ or ν is thus an essential part of the application of the homogeneous viscous material in dynamic problems.

Symonds [9] suggested that the factors μ and ν 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 $\hat{\epsilon}_{max}$, this strategy gives

$$\nu = \frac{1 + \left(\frac{\varepsilon_{max}}{\varepsilon_{o}}\right)^{1/n}}{\left(\frac{\varepsilon_{max}}{\varepsilon_{o}}\right)^{1/n}}, \qquad (4.3a)$$

$$\mu = \frac{1 + \left(\frac{\varepsilon_{max}}{\varepsilon_{o}}\right)^{1/n}}{\left(\frac{\varepsilon_{max}}{\varepsilon_{o}}\right)^{1/\nu n}} \qquad (4.3b)$$

This matching is shown diagrammatically in Fig. 4.1; the rigidviscoplastic 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 $\epsilon/\epsilon_0 = 0,05; \ \mu = 1,904; \ n' = 14,52$
Curve	3	:	homogeneous viscous curve matched on intercept alone at $\epsilon/\epsilon_o = 0,04$; $\mu = 2,904$; $n = n' = 5$
Curve	4	:	homogeneous viscous curve matched on intercept alone at $\varepsilon/\varepsilon_0 = 0,04$ 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 μ so that the homogeneous viscous curve intersects the rigid-viscoplastic curve at $\hat{\epsilon}_{max}$. 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 $\hat{\epsilon}_{max}$ 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 μ is chosen so that the rigid-viscoplastic relation and the homogeneous relation intersect at $\hat{\epsilon}_{max}$. 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 κ_{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 $N/N_{o} \ge 0.1$. 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 (<u>G</u>eometrically <u>N</u>on-linear <u>Instantaneous Mode Solution Technique</u>) and DAGNVS (<u>Direct Analysis of <u>G</u>eometrically <u>N</u>on-linear <u>V</u>iscous <u>S</u>tructures) 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.</u>

The data input for each program is identical, comprising material constants ($\dot{\epsilon}_0$, σ_0 , 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 <u>Numerical Implementation of the Instantaneous Mode Technique</u> 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

Compatibility equations are formed, corresponding the redundants. 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-viscoplastic 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 μ 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 Δt .

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+ Δ t). The time function, velocities and displacements are evaluated using equns. (2.48a), (2.45) and (2.48b) at t+ Δ 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+ Δ 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+ Δ 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 <u>Numerical Implementation of the Conventional Direct Solution</u> 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-viscoplastic 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+ Δ 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 (sub-Equations (3.45) are formulated in subroutine IMPLIC. routine COMDIF). 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.



Figure 5.1







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 In order to analyse beam and frame structures and incorthesis. porate 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 capebilities 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 For n = n' = 5 the maximum transverse tip displacement was result. 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 Using the direct method of analysis, the of deformation as 0.065s. 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 v = 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 v = 2,813 and μ = 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 rigidperfectly 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 μ = 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, $\dot{\epsilon}_{0}$ 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 μ was found to range 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 rigidviscoplastic 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.



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







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



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



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











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.

between 8 and 10.

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'



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 μ and ν , 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

- 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.
- J.B. Martin, "Impulsively Loading Theorems for Rigid-Plastic Continua", J. Eng. Mech. Div., Proc. ASCE, <u>90</u>, (EM5), 27-42, 1964.
- M.J. Mangoine, "Influence of Rate of Strain and Temperature on Yield Stresses of Mild Steel", J. Appl. Mechanics, <u>11</u>, A211-A218, 1944.
- 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.
- E.W. Parkes, "The Permanent Deformation of a Cantilever Struck Transversely at its Tip", Proc. Roy. Soc. A., <u>228</u>, 462-476, 1955.
- T.C.T. Ting and P.S. Symonds, "Impack of a CantileverBeam with Strain Rate Sensitivity", Proc. of 4th U.S. Nat. Cong. Appl. Mech., ASME, 2, 1153-1165, 1962.
- 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.

- 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.
- 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., <u>29</u>, 719-727, 1962.
- S.R. Bodner, "Deformation of Rate Sensitive Structures Under Impulsive Loading", <u>Engineering Plasticity</u> (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), <u>21</u>, 299-315, 1974.
- P.S. Symonds and C.T. Chon, "Large Viscoplate Deflections of Impulsively Loaded Plane Frames", Int. J. Solids Structures, 15, 15-31, 1979.

•

- 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), <u>47</u>, 277-287, 1979.
- H.H. Bleich and M.G. Salvaderi, "Impulsive Motion of Elasto-Plastic Beams", Trans. ASCE, <u>120</u>, 499-520, 1955.
- P.E. Duwez, D.S. Clark and H.F. Bohenblust, "The Behaviour of Long Beams under Impact Loading", J. Appl. Mech., <u>17</u>, 27-34, 1950.
- 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.
- P.S. Symonds, "Dynamic Load Characteristics in Plastic Bending of Beams", J. Appl. Mech., ASME, <u>20</u>, 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.
- E.W. Parkes, "The Permanent Deformation of an Inelastic Beam Struck Transversely at any Point in its Span", Proc. Inst. of Civ. Engrs., London, <u>10</u>, 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., <u>14</u>- 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 Skinframe 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, <u>25</u>, 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.
- L.S.S. Lee, "Mode Response of Dynamically Loaded Structures",
 J. Appl. Mech., <u>39</u>, 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. AR0.79/2, 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.
- C.F. Gerald, <u>Applied Numerical Analysis</u>, 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 v must be specified to set the magnitude of the power n' in the matched viscous material model (Chapter 4)



ν

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 ∨ must be automatically computed.

4. Section Size and Material Properties.

h b σ $\dot{\varepsilon}$ n

h	-	section depth in metres
b	-	section breadth in metres
σ	-	yield stress of section in N/m^2
έ	-	strain rate constant
n	-	power n in rigid-viscoplastic constitutive relation

5. Node Incidences

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.



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	-	node number where constraint is present	esent	
i) no restraint in the global X-direction	rection	
		l restraint in the global X-direction	tion	
j	-) no restraint in the global Y-direction	rection	
		l restraint in the global Y-direction	tion	
k	-) free to rotate		
	`	l 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:



8. Structural Type

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:



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-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:

 $M_1 \quad M_2 \quad \dots \quad M_i \quad \dots \quad M_n$

 $\rm M_{i}$ - 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 - node number i i

n_i - 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 through n 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:

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 - 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 $\boldsymbol{\nu}$ is set to 1.

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

For certain classes of problems, the v=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 (v=-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 ν 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 ν 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

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 v (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 v must be altered. No correlation is apparent between the value of v 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

3	С	* * * * * * * * * * * * * * * * * * * *
4	C	
т Б	C	סזאדס סקאדס ס
c	0	DRIVER ROOTINE
6	C	
/	C	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
8		INCLUDE GNLIMST.COMPROC
9	С	INPUT DATA
10		CALL INPUT
11	С	DISPLAY DATA
12		CALL DATA
13	5	CONTINUE
14	С	DETERMINE STATICS OF STRUCTURE
15		CALL STAT
16	2	CONTINUE
17	2	
10	C	CILL LOND
18		
19	1	CONTINUE
20	С	IF STRUCTURE STATICALLY DETERMINATE (IE IF NRED=4)
21	С	THEN DON'T NEED TO ITERATE FOR REDUNDANTS
22		IF(NRED.EQ.4) GO TO 3
23	С	ZERO ALL ARRAYS IN ITERATIVE PROCEDURE TO FOLLOW
24		CALL ZERO
25	С	EVALUATE COMPATIBILITY EQUATIONS
26		CALL COMEQU
27	С	CALCULATE PARTIAL DIRIVATIVES OF COMPATIBILITY EQUATIONS
28		CALL PDIFF
29	С	INVERT MATRIX OF PARTIAL DIRIVATIVES
30	•	CALL PINOT
31	C	CHECK MAGNITUDE OF PERTURBATION IN VALUE OF REDUNDANTS
3.7	C	TE CMALL (IELAG-1) CO TO VELOC . IE LADGE(IELAG=0) ITERATE
22	C	TE SMALL (TELAG-I) GO TO VELOC : TE LARGE(TELAG-O) TIERATE
22		
34	•	IF(IFLAG.EQ.U) GO TO I
35	3	CONTINUE
36	C	DETERMINE VELOCITIES ASSOCIATED WITH
37	C	TOTAL STRESS STATE OF THE STRUCTURE
38		CALL VELOC
39	С	CHECK IF PROCEDURE HAS CONVERGED ONTO MODE
40		CALL MODECH
41	С	IF NOT THEN ITERATE
42		IF(IDISIP.EQ.0) GO TO 2
43	4	CONTINUE
44	С	ONCE PHI IS DETERMINED, MODE SOLUTION CAN COMMENCE
45		CALL INMODE
46	С	
47	C	DETERMINE MATCHING CONSTANT THEN STORE AND OUTPUT
18	Č	INITIAL MODE CONFIGURATION
40	C	
50		
50		
51		CALL OUTPUT
52	-	END IF
53	C	
54	C	
55		IF(IRND.EQ.0)CALL STORE
56	С	
57		IF(IRND.EQ.O)T=T+DT
58	С	
59		CALL UPDATE
60	С	
61		IF(AMP(2)/AMP(1).LE.O.1)THEN

в.2

62		IOUT=1
63		ICOUNT=NDIV-1
64		END IF
65	С	
66		IRND=IRND+1
67	С	
68	С	IF MEMBRANE SOLUTION PREDOMINATES CALCULATE
69	С	NEW MATCHING FACTOR
70		IF(MATCHA.EQ.1)CALL MATCH
71	С	
72		IF(IOUT.EQ.0)GO TO 5
73	С	
74		IRND=0
75	С	
76		ICOUNT=ICOUNT+1
77	С	
78		IF(IOUT.EQ.1.AND.ICOUNT.EQ.NDIV)CALL OUTPUT
79	С	
80		IOUT=0
81	С	
82	С	CALL STORE
83	С	
84		IF(AMP(2)/AMP(1).GT.0.1)GO TO 5
85	С	
86	С	IS PLOTTING OF DEFORMATION HISTORY REQUIRED ?
87	С	
88		IF(PICT.EQ.'PLOT')CALL PICTUR
89	С	
90		STOP
91	С	
92		END
93	С	
94	С	
95	С	* * * * * * * * * * * * * * * * * * * *
96	С	
97	С	INPUT
98	С	
99	С	INPUT ALL DATA
100	С	
101	С	*******
102		SUBROUTINE INPUT
103		INCLUDE GNLIMST.COMPROC
104		DIMENSION IIBC(NF),VELDUM(NF)
105	100	FORMAT()
106	101	FORMAT (A4)
107	102	FORMAT(A80)
108	103	FORMAT(A5)
109		READ(IREAD,102)TITLE
110		READ(IREAD, 101)PICT
111		READ(IREAD, 100)PMATCH
112		READ(IREAD, 100)HH, BB, YSTRS, EPSIO, EN
113	С	NODE INCIDENCES (NUMBERING OF ELEMENT ENDS)
114		DO 11 IE=1,NE
115		READ(IREAD, 100) (NBEAM(IE, I), $I=1, 2$)
116	11	CONTINUE
117	С	COORDINATES OF ORDERED NODES (X,Y)
118		DO 12 I=1, NN
119		READ(IREAD,100)COORDX(I),COORDY(I)
120	12	CONTINUE

121	С	BOUNDARY CONDITIONS
122	С	FREEDOM : O
123	С	FIXITY : 1
1 24		DO 2 I=1,NDF
125		IBC(I)=0
126	2	CONTINUE
127	3	READ(IREAD, 100)N1
128		IF(N1.LT.0) GO TO 4
129		READ(IREAD, 100)(IIBC(J), J=1, NF)
130		II=NF*(N1-1)
131		DO 5 $I=1, NF$
132		II=II+1
133		<pre>IBC(II)=IIBC(I)</pre>
134	5	CONTINUE
135		GO TO 3
136	4	CONTINUE
137	C	IS STRUCTURE DETERMINATE OR HYPERSTATIC ?
138	-	IF(NRED.EO.4) THEN
139	С	IF DETERMINATE . IS IT A CANTILEVER OR SIMPLY SUPPORTED ?
140	°,	READ(IREAD, 101)STADET
141		ELSE
142	С	IF HYPERSTATIC . NRED DEGREES OF FREEDOM MUST BE RELEASED
143	Č	FOR A DETERMINATE STRUCTURE . IS RESULTING STUCTURE A
144	č	CANTILEVER OF A SIMPLY SUPPORTED STRUCTURE ?
145	C	READ(IREAD, 101)STADET
146	C	DEGREES OF ERFEDOM WHICH ARE RELEASED
1 4 7	C	PFAD(IPFAD OO)(PFLFAS(I) I=1 NPFD)
1/0		END IF
140	C	END IN MACS VECTOR
150	C	READ IN MASS VECTOR $PEAD(TPEAD 100)(PMACC(T) T-1 NN)$
150	0	$ \begin{array}{c} \text{READ}(\text{IREAD}, 100)(\text{RMASS}(1), 1-1, \text{NN}) \\ \text{DEAD} \text{IN INITIAL VELOCITY} \end{array} $
121	Ċ	READ IN INITIAL VELOCITY
152	6	READ(IREAD, 100)NI, NZ
153		IF(NI,LT,U)GUTU / PROV(T) = 1 PROV
154		READ(IREAD, 100)(VELDUM(I), I=I, NF)
155		11 = NF * (NI - 1)
156		JJ = N2 - N1 + 1
157		DO 8 J=1, JJ
158		DO 8 $I=1, NF$
159		II=II+1
160		VEL(II,1)=VELDUM(I)
161	8	CONTINUE
162		GO TO 6
163	7	CONTINUE
164	С	STATE HOW MANY TIME INTERVALS IN MODE SOLUTION ARE REQUIRED
165	С	AND NUMBER OF OUTPUTS REQUIRED
166		READ(IREAD, 100)RINT, NDIV
167	С	READ INITIAL ESTIMATE OF REDUNDANTS
168	С	READ(IREAD, 100)XXDUM
169	С	CALCULATE YIELD MOMENT AND AXIAL YIELD STRESS
170		AA=HH*BB
171		RMO=AA*HH*YSTRS/4.
172		RNO=AA*YSTRS
173	С	
174	С	SET INITIAL MODE SHAPE (GUESS) EQUAL TO INIYIAL VELOCITY
175	2	DO 1 $I=1, NDF$
176		PHI(I,1) = VEL(I,1)
177	1	CONTINUE
178	Ĉ	CALCULATE NORMALISATION CONSTANT AND INITIAL
179	č	DISSIPATION RATE .
	•	

в.4

180		DO 10 $I=1, NDF, 3$
181		II = INT(FLOAT(I)/NF+0.7)
182		IK=I+1
183		DISTP(2) = DISTP(2) + VEL(T, 1) * RMASS(TT) * PHT(T, 1)
194		DISIP(2) = DISIP(2) + VEL(IK 1) * PMASS(II) * DHI(IVI)
195		DISIT(2) = DISIT(2) + OBC(R, I) ACASS(II) INI(IR, I) $DVINFT = DVINFT + DHI(I) + DHI(I) + DMACC(II)$
100		$\mathbf{R}_{\mathbf{T}} = \mathbf{R}_{\mathbf{T}} = $
186	1.0	$RKINET=RKINET+PHI(IK, I) ^{PHI}(IK, I) ^{RMASS(II)}$
187	10	CONTINUE
188	C	
189		EN1=EN+1
190		EN2=EN1+1
191	С	IF STRUCTURE INDETERMINATE (NRED NOT EQUAL TO 4), DUMMY VALUES
1 92	С	ARE ASSIGNED TO REACTANTS (X) TO AVOID POSSIBLE DIVISION BY
193	С	ZERO IN FORMULATION OF COMPATIBILITY EQUATIONS .
194		IF(NRED.NE.4)THEN
195		DO 9 I=1,NRED
196		X(I) = 1.0
197	9	CONTINUE
198	-	ENDIF
199		RETURN
200		DEBIG SUBCHK
200	C	DEDOG SOBELIK
201	C	END
202	0	END
203		
204	C	
205	C	*****
206	С	
207	С	DATA
208	С	
209	С	DISPLAYS ALL DATA FOR VERIFICATION
210	С	
211	С	* * * * * * * * * * * * * * * * * * * *
212		SUBROUTINE DATA
213		INCLUDE GNLIMST.COMPROC
214	С	
215	•	WRTTE(TPRINT, 1) TITE
216	1	FORMAT(1H1 5X 80('*') /10X A80 / 6X 80('*') /)
210	-	WDTME (INF, 5X, 60(), / IOX, A00, / / OX, 60(/ / / / /
217	17	$\frac{1}{10} \frac{1}{10} \frac$
218	1/	FORMAT(IH ,/, 20X, LARGE DISPLACEMENT ANALISIS ,/)
219		WRITE (IPRINT, I3) EN, RMO, RNO, EPSIO, YSTRS
220	13	FORMAT(1H, 5X, ///, 'MATERIAL ASSUMPTIONS', //, 'HOMOGENEOUS
221		#VISCOUS WITH POWER N =', F7.3, /, 2X, 'YIELD MOMENT ='
222		#,Fll.6,/,2X,'AXIAL YIELD STRENGTH =',El5.4,/,2X,
223		<pre>#'INITIAL STRAIN RATE =',E15.6,/,2X,</pre>
224		#'YIELD STRESS =',E15.8,/)
225		WRITE(IPRINT,2)NE,NN
226	2	FORMAT(1H ,/, 3X, 'NUMBER OF ELEMENTS :', I3, /, 3X, 'NUMBER OF
227		#NODES : ', I3, /)
228		WRTTE (TPRINT. 3)
229	3	FORMAT(//.6X.' COORDINATES OF NODES'.//.
230	5	# 'NODE' 10X 'X' 12X 'Y')
220		$= \frac{110}{10} \frac{110}{10} \frac{1}{10} \frac{1}$
231		DO IIO I-I, NN
232		WRITE(IPRINT, 4)I, COORDX(I), COORDI(I)
233	4	FORMAT(1H, 13, 1X, 2(2X, F11.5))
234	110	CONTINUE
235		WRITE(IPRINT, 5)
236	5	FORMAT(//, ' BOUNDARY CONDITIONS : 0=FREEDOM , 1=FIXITY',
237		#//,' NODE',3X,'X',3X,'Y',3X,'ROTATION')
238		WRITE(IPRINT, 6)1, IBC(1), IBC(2), IBC(3)

```
239
             WRITE(IPRINT, 6)NN, IBC(NDF-2), IBC(NDF-1), IBC(NDF)
             FORMAT(1H , I3, 4X, I1, 3X, I1, 6X, I1)
       6
240
             WRITE (IPRINT, 11)
241
             FORMAT(1H ,///,5X,'LUMPED MASS PER
242
       11
            #NODE',//,6X, 'NODE',10X, 'MASS',/)
243
244
             DO 111 I=1,NN
             WRITE(IPRINT, 12)I, RMASS(I)
245
246
       12
             FORMAT(1H , 5X, 12, 6X, E11.4, /)
247
       111
             CONTINUE
             IF(NRED.NE.4)THEN
248
             IF(STADET.EQ. 'CANT')WRITE(IPRINT,9)(RELEAS(I),I=1,NRED)
249
             FORMAT(1H ,//, ' STRUCTURE CANTILEVERED BY RELEASING
250
       9
            #RESTRAINTS AT D.O.F. ',5(1X,I2))
251
             IF(STADET.EQ.'SIMP')WRITE(IPRINT,10)(RELEAS(I),I=1,NRED)
FORMAT(1H,//,' STRUCTURE MADE SIMPLY SUPPORTED BY
252
       10
253
            #RELEASING RESTRAINTS AT D.O.F. ',5(3X,I2))
254
255
             END IF
256
             WRITE(IPRINT, 14)
             FORMAT(1H ,///,10X, 'INITIAL VELOCITY',//,
257
       14
            # 'NODE', 4X, 'X', 12X, 'Y', 9X, 'ROTATION')
258
             DO 112 I=1,NDF,3
259
             II=INT(I/NF)+1
260
             WRITE(IPRINT, 15)II, VEL(I, 1), VEL(I+1, 1), VEL(I+2, 1)
261
       15
             FORMAT(1H , I3, 1X, 3(2X, E11.3), /)
262
       112
263
             CONTINUE
264
       С
265
             RETURN
266
             DEBUG SUBCHK
267
             END
268
       С
       С
269
270
       С
       271
272
       С
273
       С
                                   STAT
       С
274
              IF THE STRUCTURE IS STATICALLY INDETERMINATE THEN
       С
275
              BOUNDARY CONDITIONS ARE RELEASED SUCH THAT IT
       С
276
       С
              BECOMES EITHER A CANTILEVER ('CANT') OR SIMPLY
277
       С
              SUPPORTED ('SIMP'). THE BENDING MOMENT DIAGRAM AND
278
       С
              AXIAL FORCE DIAGRAM DUE TO A UNIT LOAD APPLIED IN
279
              TURN AT EACH DEGREE OF FREEDOM IS THEN DETERMINED
       С
280
       С
                         : UNITM(I,J), I=D.O.F. WHERE LOAD APPLIED,
281
              BENDING
       Ċ
                                                    J=NODE NO.
282
       С
                         : UNITN(I,J), I=D.O.F. WHERE LOAD APPLIED,
283
              AXIAL
       С
                                                    J=ELEMENT
284
                              : SELF-STRESS AND STATICALLY ADMISSIBLE
       С
285
              FORCE(I,IE,J)
286
       С
                                                    SETS
       С
                              : STATICALLY ADMISSIBLE SET
287
                   Ι
       С
                   I=2, NRED
                              : SELF STRESS SYSTEMS
288
       С
289
                   ΙE
                              : ELEMENT NO.
       С
                              : MOMENT AT 'A' END OF BEAM
290
                   J=1
       С
                              : MOMENT AT 'B' END OF BEAM
                   J=2
291
       С
                               : AXIAL FORCE IN ELEMENT
292
                   J=3
293
       С
       С
294
       295
296
              SUBROUTINE STAT
              INCLUDE GNLIMST.COMPROC
297
```

298	С	
299		ITMODE=0
300	С	DETERMINE ORIENTATION OF ELEMENTS IN GLOBAL AXIS SYSTEM
301		DO 55 IE=1,NE
302		IL=NBEAM(IE,1) @NODE NO. OF A END OF ELEMENT IE
303		IR=NBEAM(IE,2) @NODE NO. OF B END OF ELEMENT IE
304	С	CL=CURRENT LENGTH OF ELEMENT
305	Ŭ	CL(TE) = SORT((COORDX(TR) - COORDX(TL)) **2.+
306		#(COORDY(IR) - COORDY(IL)) **2.)
307		SSIN(IF) = (COOPDY(IP) - COOPDY(IL)) / CL(IF)
200		CCOS(IE) = (COORDY(IE) = COORDY(IE))/CL(IE)
308		COS(IE) = (COORDA(IR) = COORDA(IE))/CE(IE)
309	55	CONTINUE
310	C	
311	C	FOR BENDING MOMENTS DUE TO UNIT LOAD AT D.O.F. I
312	С	FOR CANTILEVER
313		IF (STADET.EQ. 'CANT') THEN
314	С	HORISONTAL
315	С	
316		DO 1 $I=4$, NDF, NF
317		RI=FLOAT(I)/NF+0.1
318		IR=INT(RI)
319		DO 2 $J=1$, IR
320		UNITM(I,J) = -(COORDY(IR+1) - COORDY(J))
321	2	CONTINUE
322	1	CONTINUE
322	Ċ	CONTINUE
323	c	νερωταλι
324	C	DO 2 I-5 NDE NE
325		
320		RI = FLOAT(I) / NF + 0.1
327		
328		DO 4 $J=1,1R$
329		UNITM(I,J) = -(COORDX(J) - COORDX(IR+1))
330	4	CONTINUE
331	3	CONTINUE
332	С	MOMENTS
333		DO 26 $I=6, NDF, NF$
334		RI=FLOAT(I)/NF+0.1
335		IR=INT(RI)
336		DO 27 J=1, IR
337		UNTTM(I,J)=1.
338	27	CONTINUE
330	26	CONTINUE
340	20	FLGF
241	'n	
341		HORICONTAL LOADING AT DO E T
342	C	HORISONIAL LOADING AT D.O.F. I
343		$DO \ 8 \ I=4, NDF-2, NF$
344		DO 9 J=2, NN-1
345		
346		RI = FLOAT(I)/NF+0.1
347		IR=INT(RI)
348		IF(J.GT.IR+1)R1=1.
349		UNITM(I,J) = -COORDX(J) * COORDY(IR+1) / COORDX(NN)
350		#+(COORDY(J)-COORDY(1))-R1*(COORDY(J)-COORDY(IR+1))
351	9	CONTINUE
352	8	CONTINUE
353	С	VERTICAL UNIT LOADING AT D.O.F. I
354		DO 10 $I=5, NDF-1, NF$
355		DO 11 $J=2, NN-1$
356		R1=0

в.7

.

357 358 359		RI=FLOAT(I)/NF+0.1 IR=INT(RI) IF(J.GT.IR+1)RI=1. INITER(I) = (COODDY(ID+1))(COODDY(ND)) = (COODDY(ID))
360		$\frac{1}{1} = (COORDX(1R+1)/COORDX(NN)-1.) COORDX(0)$
362	11	$\frac{1}{2} + \frac{1}{2} + \frac{1}$
262	10	CONTINUE
261		
304	C	MOMENT AT D.0.F. I
305		DO SO $I=3$, NDF, NF
366		DO 31 $J=1, NN$
367		RI=FLOAT(1)/NF+0.1
368		IR=INT(RI)
369		R1=0.
370		IF(J.GT.IR)RI=1.
371		UNITM(I,J) = (COORDX(J) / COORDX(NN) - R1)
372		IF(I.EQ.3)UNITM(I,1) = -1.
373		IF(I.EQ.NDF)UNITM(I,NN)=1.
374	31	CONTINUE
375	30	CONTINUE
376		ENDIF
377	С	
378	C	FOR AXIAL FORCES DUE TO UNIT LOADS AT D.O.F. I
370	č	TOR IMINE FORCED BOL TO ONLY DOUDD IN DIOTE 1
300	č	
201	C	DO 12 I = 4 NDE NE
202		DU IS I=4, NDF, NF
382		RI = FLOAT(I)/NF = 0.1
383		
384		
385		IF(STADET.EQ.'SIMP') IRR=NE
386		DO 14 IE=1,IRR
387		R1=0
388		IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') Rl=1.
389		IF(STADET.EQ. 'CANT')THEN
390		UNITN(I, IE, 1) = CCOS(IE)
391		ELSE
392		UNITN(I,IE,1)=CCOS(IE)+COORDY(IR+1)*SSIN(IE)/COORDX(NN)
393		#-R1*CCOS(TE)
394		END IF
395	14	CONTINUE
306	12	CONTINUE
207	15	CONTINUE
391	C	
398	C	VERTICAL UNIT LOADS
399		DO 16 $I=5$, NDF, NF
400		RI = FLOAT(I)/NF+0.1
401		IR=INT(RI)
402		IRR=IR
403		IF(STADET.EQ.'SIMP') IRR=NE
404		DO 17 IE=1,IRR
405		R1=0
406		IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') Rl=1.
407		IF(STADET.EQ. 'CANT')THEN
408		UNITN(I,IE,1)=SSIN(IE)
409		ELSE
410		UNITN(I,IE,1)=SSIN(IE)*(1COORDX(IR+1)/COORDX(NN))
411		#-R1*SSIN(IE)
412		FND TF
113	17	
413	16	
414 /15	10	
TTJ	0	

в.8

416	С	UNIT APPLIED MOMENT
417	C	NO AXIAL FORCES CAUSED BY UNIT MOMENTS IN CANTILEVER
418	Ŭ	TE (STADET FO 'SIMP')THEN
410		
419		DO SS I=S, NDF, NF
420		DO 34 IE=1, NE
421		UNITN(I, IE, 1) = -SSIN(IE)/COORDX(NN)
422	34	CONTINUE
423	33	CONTINUE
424		END TE
425	C	
425		
426	C	
427	C	SET UP SELF STRESS SYSTEMS EQUAL TO MOMENTS AND AXAIL
428	C	FORCES ASSOCIATED WITH UNIT LOADS AT RELEASED D.O.F.'S
429	С	IF STATICALLY DETERMINATE THEN SKIP
430		IF (NRED, NE. 4) THEN
431		$DO_{35} I=2$ NRED+1
432		
432		DO SO IE-1, NE DO S
433		FORCE (1,1E,1) = UNITM (RELEAS (1-1), NBEAM (1E,1))
434		FORCE $(1, 1E, 2) = UNITM (RELEAS (1-1), NBEAM (1E, 2))$
435		<pre>FORCE(I,IE,3)=UNITN(RELEAS(I-1),IE,1)</pre>
436	36	CONTINUE
437	35	CONTINUE
438		END IF
130	C	
440	C	וארנושא
440		RETORN
441		DEBOG SUBCHK
442		END
443	С	
444	С	
445	С	********
116	Ĉ	
440		
447	C	
448	С	
449	С	ASSEMBLE STATICALLY ADMISSIBLE BENDING MOMENT
450	С	AND AXIAL FORCE DIAGRAMS DUE TO LOADING GIVEN
451	С	BY PRODUCT (MASS/NODE*CURRENT MODE SHAPE)
452	C	
452	Č	******
455	C	
454		SUBROUTINE LOAD
455		INCLUDE GNLIMST.COMPROC
456	С	
457		ITREAC=0
458		DO 1 IE=1,NE
459		DO 2 I = 1.3
460		
460	•	
461	2	CONTINUE
462	1	CONTINUE
463		DO 18 I=1,NDF
464		II=INT(FLOAT(I)/NF+0.7)
465	С	IF D.O.F. IS A ROTATION THEN NO LOAD ASSOCIATED WITH IT
466	•	$I_{J} = (INT(FLOAT(I)/NF+0.001))*NF$
400		
467		F(I, NE, IJ) Then
468		$KLOAD(1) = KMASS(11) \circ PHI(1,1)$
469		END IF
470		DO 19 IE=1,NE
471		<pre>FORCE(1,IE,1)=FORCE(1,IE,1)+UNITM(I,NBEAM(IE,1))*RLOAD(I)</pre>
472		<pre>FORCE(1, IE, 2) = FORCE(1, IE, 2) + UNITM(I, NBEAM(IE, 2)) * RLOAD(I)</pre>
473	C	AXIAL FORCES
171	C	$E \cap PCE(1 \text{ TE } 3) = E \cap PCE(1 \text{ TE } 3) + IINT TN(T \text{ TE } 1) + PI \cap AD(T)$
4/4		

475		IF (NRED.EQ.4) THEN
476		FORCET(IE, 1) = FORCE(1, IE, 1)
477		FORCET(IE, 2) = FORCE(1, IE, 2)
478		FORCEP(TE 3) - FORCE(1) TE 3)
470		FND IF
490	10	
400	10	
401	10	CONTINUE
482	C	
483	C	
484		RETURN
485		DEBUG SUBCHK
486		END
487	С	
488	С	
489	С	******
490	С	
491	C	ZERO
492	Ċ	
493	Ĉ	TNITIALISE ARRAVS USED IN ITERATIVE
191	Ċ	
494	C	PROCEDORE TO DETERMINE REDUNDANIS
495		
496	C	
497		SUBROUTINE ZERO
498		INCLUDE GNLIMST.COMPROC
499	C	
500		DO 1 $I=1$, NRED
501		FLEX(I,NRED+1)=0.
502		COMPAT(I)=0.
503		DO 2 J=1, NRED
504		FLEX(I,J) = 0.
505		PARDTF(T,J)=0.
506	2	CONTINUE
507	ĩ	CONTINUE
508	Ċ	CONTINUE
500	C	סדייויסא
510		DEBUC CUDCUV
510		
511	0	
512	C	
513	C	
514	С	*****
515	С	
516	С	COMEQUE
517	· C	
518	С	SET UP NRED COMPATIBLITY EQUATIONS
519	С	WHERE NRED EQUALS NO. OF REDUNDANTS
520	С	
521	C	* * * * * * * * * * * * * * * * * * * *
522	-	SUBROUTINE COMEOU
523		INCLIDE COMPROC
520		DOUBLE DEFCISION SUMA SUME SUME DOWN DOWN DOWN DOWF DOWF
525		#DOWC DOWL DOWL DOWL DOWL DOWL DOWL DOWL DOWL
525		TUDERC-TUDERCH
520	9	ITREAC=ITREAC+I
527	C	
528	C	SUM OVER ALL ELEMENTS IE
529		DO $I IE=I, NE$
530	С	SET UP PARAMETERS FOR CALCULATIONS TO FOLLOW
531		DO 2 $I=1, NRED+1$
532	С	MOMENTS AND AXIAL FORCES ARE NORMALISED
533		AMOM(I) = (FORCE(I, IE, 2) - FORCE(I, IE, 1))/RMO

534		ANORM1(I)=FORCE(I,IE,1)/RMO+FORCE(I,IE,3)/RNO
535	-	ANORM2(I)=FORCE(I,IE,3)/RNO-FORCE(I,IE,1)/RMO
536	2	CONTINUE
537		SUMA=ANORM1(1)
538		SUMB=ANORM2(1)
539		SUMC=AMOM(I)
540		DO 3 I=1, NRED
541		SUMA=SUMA+X(1) *ANORMI(1+1)
542		$SUMB=SUMB+X(I)^{ANORM2}(I+I)$
545	2	
544	C C	SET UD STANUM FUNCTIONS FOR DOWERED TERMS
546	C	SIG=1.
547		SIG-1.
548		SIGR=1.
549		SIGC=1.
550		SIGD=1.
551		IF((SUMA+SUMC).LT.0) SIGA=-1.
552		IF((SUMB-SUMC).LT.0) SIGB=-1.
553		IF(SUMA.LT.O) SIGC=-1.
554		IF(SUMB.LT.O) SIGD=-1.
555		POWI=SIGA*(DABS(SUMA+SUMC)**EN)
556		POWJ=SIGC*(DABS(SUMA)**EN)
557		POWK=SIGB*(DABS(SUMB-SUMC)**EN)
558		POWL=SIGD*(DABS(SUMB)**EN)
559		POWA=POW1*(SUMA+SUMC)
560		POWB=POWK*(SUMB-SUMC)
561		POWC=POWA*(SUMA+SUMC)
562		
563		POWE-POWE SUMP
565		POWE = POWE * SUMA
566		POWH=POWF*SUMB
567		PRODA=EN1*SUMC/CL(TE)
568		PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
569		PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
570	С	SET UP COMPATIBILITY EQUATIONS, ONE FOR EACH
571	С	DEGREE OF REDUNDANCY .
572		DO 4 I=1,NRED
573		COMPAT(I)=COMPAT(I)+(FORCE(I+1,IE,3)*EPSI0*0.5)*
574		#((POWA-POWE)/PRODA-(POWB-POWF)/PRODA)
5 75		#+RN0*EPSI0/(2.*RM0)*((AMOM(I+1)*RM0/CL(IE))*
576	:	#((POWC-POWG)/PRODB-SUMA*(POWA-POWE)/PRODC
577		#-(POWD-POWH)/PRODB+SUMB*(POWB-POWF)/PRODC)
578		#+FORCE(I+1,IE,1)*((POWA-POWE)/PRODA+(POWB-POWF)/PRODA))
579	4	CONTINUE
580		CONTINUE
501	C	סניתונסא
502		VETORN
584		END
585	C	
586	č	

588	С	***************************************
589	С	
590	С	PDIFF
591	С	
592	С	CALCULATE THE PARTIAL DIRIVATIVES OF
593	С	THE COMPATIBILITY EQUATIONS WITH RESPECT
594	Ċ	TO THE REDUNDANT FORCES FOR USE IN THE
595	č	NEWTON-RAPHSON SOLUTION PRODEDURE
595		NEWTON ARTISON SOLUTION TRODEDORE
590		******
597	C.	
598		SUBROUTINE PDIFF
599		INCLUDE GNLIMST.COMPROC
600		DOUBLE PRECISION SUMA, SUMB, SUMC, POWA, POWB, POWC, POWD,
601		<pre>#POWE,POWF,POWG,POWH,POWI,POWJ,POWK,POWL,PRODA,PRODB,</pre>
602		<pre>#PRODC,PARTA,PARTB,PART1PART2,PART3,PART4,PART5,PART6</pre>
603	С	TO OBTAIN THE UNKNOWN REACTIONS X(I) THE PARTIAL
604	С	DIRIVATIVES OF THE COMPATIBILITY EQUATIONS ARE
605	С	THE OBTAINED . THE NEWTON-RAPHSON METHOD IS USED
606	Ċ	TO ITERATE ONTO A SOLUTION .
607	11	CONTINUE
608	T T	DO 1 IF=1 NF
600	0	
609	C	SET UP PARAMETERS FOR CALCULATIONS TO FOLLOW
610		DU Z I=I,NRED+I
611	C	MOMENTS AND AXIAL FORCES ARE NORMALISED
612		AMOM(I) = (FORCE(I, IE, 2) - FORCE(I, IE, I)) / RMO
613		ANORM1(I)=FORCE(I,IE,1)/RMO+FORCE(I,IE,3)/RNO
614		ANORM2(I) = (FORCE(I, IE, 3) / RNO - FORCE(I, IE, 1) / RMO)
615	2	CONTINUE
616		SUMA=ANORM1(1)
617		SUMB=ANORM2(1)
618		SUMC=AMOM(1)
619		$DO_3 I=1.NRED$
620		SUMA=SUMA+X(T) *ANORMI(T+1)
621		SIMB=SIMB+Y(T) *ANORM2(T+1)
622		SUMC - SUMC + Y(T) * AMOM(T+1)
622	2	
623	3	CONTINUE
624	C	SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
625		SIG=1.
626		SIGA=1.
627		SIGB=1.
628		SIGC=1.
629		SIGD=1.
630		IF((SUMA+SUMC).LT.0) SIGA=-1.
631		IF((SUMB-SUMC).LT.0) SIGB=-1.
632		IF(SUMA.LT.O) SIGC=-1.
633		IF(SUMB,LT,0) SIGD=-1.
634		POWT=SIGA*(DABS(SUMA+SUMC)**EN)
635		POWI=SIGC*(DABS(SUMA)**EN)
636		POWV = GICP * (DABG(GUMP = GUMC) * * FNI)
630		$POWR = SIGB^{(DABS(SOMB = SOMC))}$
637		POWL=SIGD (DADS(SOMD) (DADS(SOMD)) (DADS(S
038		POWA=POWI " (SUMA+SUMC)
639		POWB=POWK*(SUMB-SUMC)
640		POWC=POWA*(SUMA+SUMC)
641		POWD=POWB*(SUMB-SUMC)
642		POWE=POWJ*SUMA
643		POWF=POWL*SUMB
644		POWG=POWE*SUMA
645		POWH=POWF*SUMB
646		PRODA=EN1*SUMC/CL(IE)

~

647		PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
648		PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
649	С	SET UP PARTIAL DIRIVATIVES
650		DO 4 I=1, NRED
651		DO 5 $J=1$, NRED
652		PARTA = ANORM1(J+1) + AMOM(J+1)
653		PARTB = ANORM2(J+1) - AMOM(J+1)
654		PARTI = ((POWI * PARTA - POWJ * ANORM1 (J+1)) * EN1 * PRODA
655		#-EN1*AMOM(J+1)*(POWA-POWE)/CL(IE))/(PRODA*PRODA)
656		PART2=((POWK*PARTB-POWL*ANORM2(J+1))*EN1*PRODA
657		#-EN1*AMOM(J+1)*(POWB-POWF)/CL(IE))/(PRODA*PRODA)
658		PART3=((POWA*PARTA-POWE*ANORM1(J+1))*EN2*EN2*PRODA*PRODA/
659		#(EN1*EN1)-(POWC-POWG)*2.*EN2*PRODA*AMOM(J+1)/(EN1*CL(IE)))/
660		#(PRODB*PRODB)
661		PART4 = ((POWA * ANORM1 (J+1) + SUMA * EN1 * POWI * PARTA))
662		#-(POWE*ANORM1(J+1))
663		#+SUMA*EN1*POWJ*ANORM1(J+1)))*PRODA*PRODA/EN1
664		#-((POWA-POWE)*2.*PRODA*AMOM(J+1)*SUMA)/CL(IE))/
665		#(PRODC*PRODC)
666		PART5=((POWB*PARTB-POWF*ANORM2(J+1))*EN2*EN2*PRODA*PRODA/
667		#(EN1*EN1)-(POWD-POWH)*2.*EN2*PRODA*AMOM(J+1)/(EN1*CL(IE)))/
668		#(PRODB*PRODB)
669		PART6=(((ANORM2(J+1)*POWB+SUMB*EN1*POWK*PARTB)-
670		#(POWF*ANORM2(J+1)+SUMB*EN1*POWL*ANORM2(J+1)))
671		#*PRODA*PRODA/EN1
672		<pre>#-((POWB-POWF)*2.*PRODA*AMOM(J+1)*SUMB)/CL(IE))/</pre>
673		#(PRODC*PRODC)
674		<pre>PARDIF(I,J)=PARDIF(I,J)+FORCE(I+1,IE,3)*EPSIO*.5*</pre>
675		#(PART1-PART2)+RN0*EPSI0/(2.*RM0)*((AMOM(I+1)*RM0/CL(IE))*
676		#(PART3-PART4-PART5+PART6)+FORCE(I+1,IE,1)*(PART1+PART2))
677	5	CONTINUE
678	4	CONTINUE
679	1	CONTINUE
680	С	
681	С	HAVING OBTAINED THE PARTIAL DIRIVATIVES OF EQUILIBRIUM
682	С	EQUATIONS, MUST INVERT TO OBTAIN PERTURBATIONS ON REDUNDANTS
683	С	FIRST SET UP AUGMENTED MATRIX FOR INVERTION.
684	С	
685		DO 6 $I=1, NRED$
686		FLEX(I, NRED+1) = -COMPAT(I)
687		DO 7 $J=1, NRED$
688		FLEX(I,J) = PARDIF(I,J)
689	7	CONTINUE
690	6	CONTINUE
691		RETURN
692		DEBUG SUBCHK
693		END
694	С	
695	С	

697	С	*******
698	C	
699	c	
700	C	FIVOI
700	C	
701	C	INVERSION ROUTINE WHICH INCLUDES PARTIAL
702	C	PIVOTING
703	С	
704	С	*******
705		SUBROUTINE PIVOT
706		INCLUDE GNLIMST.COMPROC
707		DOUBLE PRECISION RATIO, VALUE
708	С	FACTOR ALL ELEMENTS OF MATRIX BY LARGE NUMBER FOR
709	Ċ	NUMERICAL STABILITY
710	•	$DO_{60} I = 1. NRED$
711		DO 61 J=1 NRED+1
712		$\mathbf{F} = \mathbf{F} \cdot $
712	61	CONTINUE
713	60	
714	60	
/15		IF (NRED.NE.I) THEN
/16		NP=NRED+1
717		NM1 =N RED-1
718		DO 1 I=1,NRED
719		IF(DABS(FLEX(I,I)).LT.1.D-25)FLEX(I,I)=1.
720	1	CONTINUE
721		DO 35 I=1,NM1
722		IPVT=I
723		IP1=I+1
724		DO 10 J=IP1.NRED
725		IF(DABS(FLEX(IPVT,I)).LT.DABS(FLEX(J,I))) IPVT=J
726	10	CONTINUE
727	10	TF(DABS(FLEX(TPVT,T)), LT, 1, D=32)GO TO 99
729		II(DABD(IDDA(IIVI,I)),DI(IDD(22)00,I0,0))
720		$\frac{11}{11} \frac{11}{11} 11$
729		
730		FACT=FLEX(1, JCOL)
731		FLEX(1, JCOL) = FLEX(1PVT, JCOL)
132		FLEX(IPVT, JCOL)=FACT
/33	20	CONTINUE
734	25	DO 32 JROW=IP1, NRED
735		IF(DABS(FLEX(JROW,I)).LE.1.D-36) GO TO 32
736		RATIO=FLEX(JROW,I)/FLEX(I,I)
737		DO 30 KCOL=IP1,NP
738		<pre>FLEX(JROW,KCOL)=FLEX(JROW,KCOL)-RATIO*FLEX(I,KCOL)</pre>
739	.30	CONTINUE
740	32	CONTINUE
741	35	CONTINUE
742		IF(DABS(FLEX(NRED, NRED)).LT.1.D-32)GO TO 99
743		NP1=NP
744		DO 50 KCOL=NP1, NP
745		FLEX(NRED, KCOL) = FLEX(NRED, KCOL)/FLEX(NRED, NRED)
746		DO 45 J=2 NRED
740		NVBL=NP1-T
740		
740		UNTUR-FIFY (NURL KCOL)
749		VALUE-FLEA(NVBL, COL)
150		UU 4U K=L, NKLU
/51		VALUE=VALUE-FLEX(NVBL,K)*FLEX(K,KCOL)
752	40	CONTINUE
753		<pre>FLEX(NVBL,KCOL)=VALUE/FLEX(NVBL,NVBL)</pre>
754	45	CONTINUE
755	50	CONTINUE

•

756		
757		FLEX(1,2) = FLEX(1,2) / FLEX(1,1)
758		END IF
759	100	FORMAR(//// 20X INC CONVERCENCE ONTO MOMENTIC AFTER FLETRY
760	100	#UTERAT(////,20X, NO CONVERGENCE ONTO MOMENTS AFTER FIFTY
762		$\frac{11111111005}{11111005} = \frac{11111005}{1111005} = \frac{1111005}{111005} = \frac{111005}{11005} = 111$
763		RETURN
764	99	WRITE (IPRINT, 101)
765	101	FORMAT(1H0, 10X, 'SOLUTION NOT FEASIBLE, NEAR ZERO ON PIVOT')
766	101	STOP
767		DEBUG SUBCHK
768		END
769	С	
770	С	
771	С	* * * * * * * * * * * * * * * * * * * *
772	С	
773	С	DELTA
774	С	
775	С	CHECH IF CONVERGENCE ONTO REDUNDANT FORCES
776	С	HAS OCCURED.IF YES THEN ASSEMBLE FINAL
777	C	BENDING MOMENT AND AXIAL FORCE DIAGRAMS
778	C	
779	C	
780		SUBROUTINE DELTA
702		DOUDLE DRECTSION COMPROC
702		TELACI-O
701		
785		IFLAG=0
786		IF UNC = 0 $IF (ITREAC GT 1) THEN$
787		IF(IIREC.GI.I) IIER
788		IFLAG2=1
789		DO 9 I=1.NRED
790		IF(DABS(X(I)), GT, 1, D-6)THEN
791		IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0
792		END IF
793	9	CONTINUE
794		END IF
795		DO 99 $I=1, NRED$
796		COMDUM(I)=COMPAT(I)
797		XDUM(I) = X(I)
798	99	CONTINUE
799		IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1
800	С	ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT
801		DO 10 $I=1$, NRED
802		X(I) = X(I) + FLEX(I, NRED+1)
803	10	CONTINUE
804	C	IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE
805	0	IF (IFLAG, EQ. U) GO TO IS
806	C	CONSTRUCT FINAL BENDING MOMENT DIAGRAM
807	C	CUECK TE ANY DEACTANT TO NEAD 7EDA
000	C	TE CO SET FOUNT TO ZEDO
810	C	TE(NEED GT 1)THEN
811 811		
812		DIIM2 = 1. D - 16
813	C	FIND MINIMUM RECTANT
814	C	DO 1 I=1.NRED

815 IF(DABS(X(I)).LT.DUM1)DUM1=DABS(X(I))816 IF(DABS(X(I))-1.D-6.LE.DUM1)IMIN=I 1 817 CONTINUE С FIND MAXIMUM REACTANT 818 819 DO 2 I=1, NRED 820 IF(DABS(X(I)).GT.DUM2)DUM2=DABS(X(I))821 IF(DABS(X(I))+1.D-6.GT.DUM2)IMAX=I2 822 CONTINUE 823 С DETERMINE RATIO BETWEEN MAX AND MIN VALUES. С IF MIN/MAX LESS THAN 1.D-9 THEN SET X(MIN) EQUAL TO ZERO. 824 825 RATIO=X(IMIN)/X(IMAX) 826 IF(DABS(RATIO).LE.1.D-9)X(IMIN)=0. 827 END IF DO 12 IE=1,NE 828 829 DO 13 IJ=1,3 830 FORCET(IE, IJ) = 0.831 FORCET(IE,IJ)=FORCE(1,IE,IJ) 832 DO 14 IK=1,NRED 833 FORCET(IE,IJ)=FORCET(IE,IJ)+X(IK)*FORCE(IK+1,IE,IJ) 834 14 CONTINUE 835 13 CONTINUE 836 12 CONTINUE 837 15 CONTINUE 838 RETURN 839 DEBUG SUBCHK 840 END 841 С С 842 843 С С 844 С 845 VELOC 846 С 847 С CALCULATE VELOCITY CORRESPONDING TO THE С BENDING MOMENT AND AXIAL FORCES IN THE 848 С STRUCTURE BY VIRTUAL VELOCITY CALC. 849 С 850 851 С SUBROUTINE VELOC 852 853 INCLUDE GNLMIST.COMPROC 854 DOUBLE PRECISION UNITMA (NDF, NN), SUMA, SUMB, SUMC, SUMD, 855 #PROD1, PROD2, PROD3, PROD4, PROD5, PROD6, PROD7, PROD8, CONST1, 856 #CONST2, CONST3, CONST4, CONST5, CONST6, CONST7, SIG1, SIG2, 857 #SIG3,SIG4 IJ=2858 859 DO 1 I=1, NDF860 VEL(1,2) = 0.IF THE D.O.F. IS A BOUNDARY CONDITION THEN VELOCITY IS ZERO 861 С IF(IBC(I).EQ.1) GO TO 1 862 DETERMINE WHETHER D.O.F. IS A ROTATION 863 С 864 IK=0865 IF(IJ*NF.EQ.I)IK=1 IF(IK.EQ.1)IJ=IJ+1 866 867 DO 2 IE=1, NEFOR AN APPLIED UNIT MOMENT , THERE IS A 868 С DISCONTINUITY OF MOMENT AT POINT OF APPLICATION С 869 IF(I.EQ.3) GO TO 3 870 IF(I.EQ.NDF) GO TO 3 871 UNITMA(I, NBEAM(IE, 1)) = 0.872 IF(IK.EQ.1.AND.(IJ-1).EQ.NBEAM(IE,1))THEN 873

874		UNITMA(I,NBEAM(IE,1))=UNITM(I,NBEAM(IE,1))-1.
875		UNITM(I, NBEAM(IE, 1))=UNITMA(I, NBEAM(IE, 1))
876		END IF
877	3	CONTINUE
070	0	GIMA-INTTM(T NDEAM(TE 2)) - INTTM(T NDEAM(TE 1))
070		SUMP = (BODORT(IE, 2)) = ONIM(I, NDEAM(IE, I))
8/9		SUMB = (FORCET(IE, 2) - FORCET(IE, 1))/RMO
880		SUMC=FORCET(IE,I)/RMO+FORCET(IE,3)/RNO
881		SUMD=(FORCET(IE,3)/RNO-FORCET(IE,1)/RMO)
882		SIG1=1.
883		SIG2=1.
884		SIG3=1.
885		SIG4=1
005		TE((SIMP+SIMC) TT O) STC] = 1
000		IF((SUMD+SUMD) IM O) SIGI-I.
887		IF((-SUMB+SUMD).LT.0) $SIG2=-1.$
888		IF(SUMC.LT.0) $SIG3=-1.$
889		IF(SUMD.LT.O) SIG4=-1.
890		<pre>PROD1=SIG1*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)*(SUMB+SUMC)</pre>
891		PROD2=SIG1*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)
892		PROD3 = SIG2*((DABS(-SUMB+SUMD))**EN)*(-SUMB+SUMD)
893		#*(-SIMB+SIMD)
801		PRODA=SIG2*((DARS(-SUMB+SUMD))**FN)*(-SUMB+SUMD))
005		PROD = CIC2 * (PABC COMPTONE) + CIMC + CIMC
895		PRODS=SIGS*(DABS(SOMC)**EN)*SOMC*SOMC
896		PROD6=SIG3*(DABS(SUMC)**EN)*SUMC
897		PROD7=SIG4*(DABS(SUMD)**EN)*SUMD*SUMD
898		PROD8=SIG4*(DABS(SUMD)**EN)*SUMD
899		CONST1=SUMA/CL(IE)
900		CONST2=CL(IE)*CL(IE)/(EN2*SUMB*SUMB)
901		CONST3 = (SUMC) * CL(IE) * CL(IE) / (EN1 * SUMB * SUMB)
902		CONST4 = CONST3 * (SUMD) / (SUMC)
002		CONGETE - CI (IF) / (FN1 * (CIMP))
903		CONSTS=CD(TE)/(ENT (SOMB))
904		CONST6=RNU^EPSIU/(2. ^RMU)
905		CONST/=EPSI0*0.5*UNITN(I,IE,I)
906	С	VELOCITY AT D.O.F. I
907		<pre>VEL(I,2)=VEL(I,2)+CONST6*(CONST1*((PROD1-PROD5)*CONST2</pre>
908		#-(PROD2-PROD6)*CONST3-(PROD3-PROD7)*CONST2+(PROD4-PROD8)
909		<pre>#*CONST4)+UNITM(I,NBEAM(IE,1))*((PROD2-PROD6)*CONST5</pre>
910		#+(PROD4-PROD8)*CONST5))+
011		$\#CONST7*((BBOD)^2 - BBOD()*CONST5 - (BBOD)^2 - BBOD()*CONST5)$
911		$\frac{1}{10000000000000000000000000000000000$
912		IF(IK, EQ. 1. AND. (IJ-1), EQ. NBEAM(IE, 1))THEN
913		UNITM(I,NBEAM(IE,I))=UNITM(I,NBEAM(IE,I))+1.
914		END IF
915	2	CONTINUE
916	·1	CONTINUE
917		RETURN
918		DEBUG SUBCHK
010		FND
020	C	
920	č	
921	C	
922	C	*********
923	С	
924	С	MODECH
925	С	
926	С	CHECH WHETHER CONVERGENCE ONTO MODE SHAPE
927	Ċ	HAS OCCURED. IF NOT, THEN NORMALISE CURRENT
0.20	ĉ	VELOCITY TO OBTAIN NEW TRIAL MODE SHAPE
920	c	VEROCITI TO OBTAIN NEW INITH MODE OWARD
929	C	* * * * * * * * * * * * * * * * * * * *
930	C	
931		SUBROUTINE MODECH
932		INCLUDE GNLMIST.COMPROC

933 С С NO. OF ITERATIONS TO DETERMINE MODE 934 935 С IF MODE NOT OBTAINED AFTER FORTY ITERATIONS STOP 936 IF(ITMODE.EQ.10)WRITE(IPRINT,100) 100 FORMAT(1H ,//,20X, 'MODE NOT FOUND AFTER TEN ITERATIONS : 937 938 # STOP',/) IF(ITMODE.EQ.10)STOP 939 ITMODE=ITMODE+1 940 941 A=0. 942 B=0. DISIP(1)=DISIP(2) 943 DISIP(2)=0.944 945 С CHECK FOR CONVERGENCE С CALCULATE CURRENT DISSIPATION RATE 946 947 DO 1 I=1,NDF,3 IK=I+1948 II=INT(FLOAT(I)/NF+0.7) 949 950 DISIP(2) = DISIP(2) + VEL(1, 2) * RMASS(II) * PHI(1, 1)DISIP(2)=DISIP(2)+VEL(IK, 2)*RMASS(II)*PHI(IK, 1) 951 952 1 CONTINUE 953 С 954 С NORMALISE VELOCITIES FOR NEW MODE SHAPE 955 С DUM=0. 956 DO 5 I=1, NDF, 3957 958 II = INT(FLOAT(I)/NF+0.7)DUM=DUM+VEL(I,2)*VEL(I,2)*RMASS(II) 959 960 IK=I+1DUM=DUM+VEL(IK, 2) *VEL(IK, 2) *RMASS(II) 961 962 5 CONTINUE ANORM=DSORT(DUM/RKINET) 963 С RLAMDA=SQRT(DUM) 964 С 965 DO 2 I=1, NDF966 967 PHI(I,2) = VEL(I,2) / RLAMDAPHI(I,1) = PHI(I,2)968 969 2 CONTINUE 970 С CHECK CHANGE IN DISSIPATION RATE 971 IDISIP=0 972 IF(ABS(DISIP(2)/DISIP(1)-1.D0).LT.5.D-2)IDISIP=1 973 С OBTAIN AMPLITUDE OF VELOCITY BY PERFORMING MOMENTUM BALANC С 974 975 IF (IDISIP.EO.1. AND.T.LT.1.D-9. AND. IRND.EQ.0) THEN DO 3 I=1, NDF976 II=INT(FLOAT(I)/NF+0.7) 977 IJ = (INT(FLOAT(I)/NF+0.001))*NF978 IF(I.NE.IJ)THEN 979 A=A+VEL(I,1)*RMASS(II)*PHI(I,1) 980 B=B+PHI(I,1)*RMASS(II)*PHI(I,1)981 END IF 982 3 CONTINUE 983 984 AMP(1) = A/BAMP(2) = AMP(1)985 DO 4 I=1, NDF986 VEL(I,2) = PHI(I,1) * AMP(2)987 988 4 CONTINUE END IF 989 990 С IF(DISPL.EQ. 'LARGE'.AND.IRND.EQ.O.AND.T.GT.O 991

992		#.AND.IDISIP.EQ.1)THEN
993		DO 6 $I=1, NDF$
994		II = INT(FLOAT(I)/NF+0.7)
995		IJ = (INT(FLOAT(I)/NF+0.001))*NF
996		IF(I.NE.IJ)THEN
997		A=A+VMODE(I,2)*RMASS(II)*PHI(I,1)
998		B=B+PHI(I,1)*RMASS(II)*PHI(I,1)
999		END IF
1000	6	CONTINUE
1001		AMP(2) = A/B
1002		END. IF
1.003	С	
1004	С	
1005		RETURN
1006		DEBUG SUBCHK
1007		END
1008	С	
1009	Ċ	
1010	C	******
1011	Ċ	
1012	č	INMODE
1013	Ċ	
1014	C	FORMULATE EXPLICIT EXPRESSIONS FOR THE TIME
1015	Č	FUNCTION . ITS DIRIVATIVE . VELOCITY AND
1016	Ċ	DISPLACEMENT ONCE INSTANTANEOUS MODE HAS
1017	Č	BEEN FOUND
1018	C	
1019	Č	******
1020	C	SUBROUTINE INMODE
1021		INCLUDE GNIMIST, COMPROC
1021		DOUBLE DEFCISION AL A2
1022	C	DOUBLE PRECISION RI, RZ
1023	Ċ	CALCULATE FACTOR K IN EXPRESSION FOR T(T)
1025	C	CALCOLATE FACTOR & IN EXPRESSION FOR I(I)
1025	C	
1020		
1027		$\frac{1}{100} = \frac{1}{100} = \frac{1}$
1020		POWA-1/EN
1029		POWB = (EN - 1.) / EN
1030		POWC=1./POWB
1031		POWD = (2. EN - 1.) / (EN - 1.)
1032	6	RK=1./((ABS(RLAMDA)**POWA)*(AMP(2)**POWB))
1033	C	CALCULANTE EXEDERCION BOD (M)
1034	Ċ	CALCULATE EXPRESSION FOR T(T)
1035	0	FACT=POWB^RK^TIME
1036	C	CHECK IF NEAR TOTAL TIME
1037		IF (FACT.GT.I) THEN
1038		FACT=1.
1039	-	END TE
1040	C	
1041	_	TT(2) = (1 - FACT) * * POWC
1042	С	
1043	C	DIRIVATIVE OF T(T)
1044	C	
1045		IF(IRND.EQ.0)DTTDT(2) = -(TT(2) * POWA) * RK
1046		IF(IRND.GT.O)THEN
1047		DTTDT(2) = (DTTDT(2) - (TT(2) * *POWA) *RK) / 2.
1048		END IF
1049	С	
1050		IF(IRND.GT.0)TT(2)=TT(1)+(DTTDT(1)+DTTDT(2))*DT/2.

1051	C	ALAULARE ENDERGION FOR MORE UPLOATETER PLANT ARVENTA
1052	C	CALCULATE EXPRESSION FOR MODE VELOCITIES, DISPLACEMENTS
1053	C	
1054		DO 3 $1=1$, NDF TR(TRND, RO, O) TR(T, O) -3 ND(O) $+7$ DUT(T, D) $+7$ MD(O)
1055		$IF(IRND.EQ.0)VMODE(I,2)=AMP(2)^{PHI}(I,I)^{TT}(2)$ $IF(IRND.CT.0)TUEN$
1056		IF(IRND,GT,O)THEN $IRODR(T,O) = (IRODR(T,O) + NRD(O) + DUT(T,O) + TRR(O)) / O$
1057		VMODE(1, 2) = (VMODE(1, 2) + AMP(2) * PH1(1, 1) * TT(2))/2.
1058	С	
1060	S C	CONTINUE
1061	C	CALCULATE EVEDERCION FOR MOMENTE AND AVIAL FORCES
1061	C	CALCULATE EXPRESSION FOR MOMENTS AND AXIAL FORCES
1062	C	DO 4 IE-1 NE
1064		DO 4 IE-I, NE DO 5 I-1 3
1065		DUM=RM0
1066		TE(TEO 3) DUM-DNO
1067		FORMOD(IF, I) = (((AMP(2) * TT(2) / PLAMDA) * POWA))
1068		$\# * FORCET(IE_J) / DUM$
1069	5	CONTINUE
1070	Č	CONTINUE
1071	Č	IF AXIAL FORCES LARGE SET FLAG TO REOUEST NEW MATCHING
1072	C	FACTOR
1073	č	IF MEMBRANE FACTOR ALREADY CALCULATED THEN SKIP
1074	•	IF (MATCHA.NE1.OR. MATCHA.EO.O) THEN
075		IF(DABS(FORMOD(IE.3)), GT.0.2)MATCHA=1
1076		END IF
1077	С	
1078	4	CONTINUE
.079	С	
1080	C	
1081		RETURN
1082		DEBUG SUBCHK
1083		END
1084	С	
1085	С	
1086	С	*******
108 7	С	
208 8	С	МАТСН
108 9	С	
1090	С	DETERMINES MATCHING FACTOR ON SLOPE ALONE
1091	С	
1092	С	***************************************
1093		SUBROUTINE MATCH
1094		INCLUDE GNLMIST.COMPROC
1095		DMATCH=1.
1096		DUM=0.
1097	С	LOCATE MAXIMUM BENDING MOMENT OR AXIAL FORCE IN STRUCTURE
10 98	C	
109 9		Nl=1
1100		N2=2
1101		IF (MATCHA.EQ.1) THEN
1102		N1=3
1103		N2=3
1104		END IF
1105	C	
1106		DO 1 $IE=1, NE$
1107		DO 2 $I=N1, N2$
1108		IF(DABS(FORMOD(IE,I)).GE.DUM)DUM=ABS(FORMOD(IE,I))
1109	2	CONTINUE

1110	1	CONTINUE
1111	С	
1112		AMPD=AMP(1)**(1./EN)
1113		RMMAX=DUM/AMPD
1114		IF(RMATCH.GT.1.01)THEN
1115		RMMAX=RMMAX*RMATCH
1116		DMATCH=RMATCH
1117		END IF
1118	С	
1119		RKMAX = AMP(1)*((RMMAX)**EN)
1120	•	RMATCH=(1.+(RKMAX)**(1./EN))/(RKMAX**(1./EN))
1121	С	
1122	С	CALCULATE TOTAL TIME
1123	С	
1124		TF = (EN/(EN-1.))/(RK*RMATCH)
1125	С	
1126	С	MATCH YIELD MOMENT AND AXIAL YIELD STRESS
1127		RMO=RMO*RMATCH/DMATCH
1128		RN0=RN0*RMATCH/DMATCH
1129	С	
1130		IF (MATCHA.EQ.1) THEN
1131		WRITE(IPRINT, 3)RMATCH
1132	3	FORMAT(1H ,///,20X, 'REVISED MATCHING FACTOR FOR MEMBRANE
1133		# ACTION IS : ',Ell.6)
1134		MATCHA=-1
1135	_	END IF
1136	C	
1137		RETURN
1138		DEBUG SUBCHK
1139		END
1140	C	
1141	С	

1143	С	******
1144	С	
1145	C	UPDATE
1146	С	
1147	C	UPDATES GEOMETRY OF THE STRUCTURE AFTER
1148	С	EACH TIME INCREMENT AND STORES PREVIOUS
1149	C	VELOCITIES AND DISPLACEMENTS.
1150	Č	
1151	č	*****
1152	•	SUBROUTINE UPDATE
1153		INCLUDE GNLMIST.COMPROC
1154	С	
1155	Ũ	$DO_3 T=1.NDF$
1156		$U(T_2) = (VMODE(T_1) + VMODE(T_2)) * DT/2$
1157	З	
1159	5	
1150		
1160		10^{-1}
1161		T = (T + T)
1162		IF(DADS(U(1,2)),GI(1,D-3)) $IE(DADS(U(1,1))/U(1,2) = 1, DO) Cm (1, D-2) IOUM-0$
1162		$\frac{1}{1} \frac{1}{1} \frac{1}$
1103		END IF $EEDDEC(U(TT 2)) Cm = D = E MUEN$
1165		IF(DADS(U(13,2)),GI(1,D-3)] HEN
1166		$\frac{1}{2} \frac{1}{2} \frac{1}$
1167	F	
1160	5	DO 4 I-1 NDE 2
1160		DO 4 I - I, NDF, S
1170		$\frac{11-1}{1} \frac{1}{1} - \frac{1}{1} \frac{1}{1} - \frac{1}{1} \frac{1}{$
1170		U(I, I) - U(I, 2)
11/1		U(1,1)=U(1,2)
11/2		$1^{j}=1+1$
11/3		COORDY(11) = COORDY(11) = U(13, 1) + U(13, 2)
11/4		U(1J, 1) = U(1J, 2)
11/5		1K=10+1
11/6	4	U(1K, 1) = U(1K, 2)
11//	4	CONTINUE
11/8	C	IF CONVERGENCE ONTO MODE IN LARGE DISPLACEMENT ANALYSIS
1179	C	HAS NOT OCCURRED AFTER FIVE ITERATIONS THEN STORE CURRENT
1180	C	RESULT.
1181	_	IF(IRND.GT.5)WRITE(IPRINT,/)
1182	7	FORMAT(IH ,//, 20X, MODE IN LARGE DISPLACEMENT ANALYSIS NO
1183		#FOUND AFTER FIVE ITERATIONS TO ONE PERCENT VARIATION :
1184		#CONTINUE')
1185	_	IF(IRND.GT.5)IOUT=1
1186	C	
1187		IF(IOUT.EQ.1)THEN
1188		DO 2 I=1, NDF
1189		UMODE(I) = UMODE(I) + U(I, 1)
1190		U(I, 1) = 0.
1191	2	CONTINUE
1192		END IF
1193	C	
1194		RETURN
1195		DEBUG SUBCHK
1196		END
1197	С	
1198	С	

1200	С	*****************
1201	С	
1202	С	STORE
1203	С	
1204	С	STORES INTERMEDIATE VALUES OF THE TIME
1205	Ċ	FUNCTION.ITS DIRIVATIVE AND THE CURRENT
1206	Č	VELOCITY ESTIMATE IN THE INSTANTANEOUS
1200	Ċ	
1207		MODE ALGORITHM
1200		* * * * * * * * * * * * * * * * * * * *
1209	C	
1210		SUBROUTINE STORE
1211		INCLUDE GNLMIST.COMPROC
1212	С	
1213	С	
1214		IF(T.LT.1.E-9)DT=TF/RINT
1215		TT(1)=TT(2)
1216		DTTDT(1) = DTTDT(2)
1217	С	
1218		DO 1 I=1,NDF
1219		VMODE(I,1) = VMODE(I,2)
1220	1	CONTINUE
1 2 2 1	Ĉ	
1222	Č	RETIIRN
1222		DFBUG SUBCHK
1 2 2 3		FND
1 2 2 4	C	
1225		
1220		
1227	C	
1228	C	
1229	C	Ουτρυτ
1230	C	
1231	С	OUTPUTS RESULTS OF ANALYSIS AT
1232	С	REQUESTED TIME INTERVALS
1233	С	
1234	C	* * * * * * * * * * * * * * * * * * * *
1235		SUBROUTINE OUTPUT
1236		INCLUDE GNLMIST.COMPROC
1237		ICOUNT=0
1238		TF(T, LT, 1, D-36) THEN
1 2 3 9	C	
12/0	Ċ	NO OF TURBAUTONS TO DETERMINE MODE SHADE
1240	C	WOTTE (TODINT 5) TTMODE
1040	· 5	FORMAM(101 ' NO OF IMPRATIONS TO DETERMINE MODE . ' 12)
1242	5	WDIME (IDDING 77) ME DNAMOU
1243		WRITE (IPRINT, //)TF, RMATCH
1244	//	FORMAT(IH ,/, INITIAL TOTAL TIME ESTIMATE : ,
1245		#E1/./, IUX, 'MATCHING FACTOR IS :', E1/./,/)
1246		# FACTOR IS :', E17.7,/)
1247		WRITE(IPRINT,6)
1248	6	FORMAT(1H ,18X,' MODE SHAPE (T=0)',//,
1249		#'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
1250		DO 4 $I=1, NDF-2, 3$
1251		II=INT(FLOAT(I)/NF+0.7)
1252		WRITE(IPRINT,3)II,PHI(I,1),PHI(I+1,1),PHI(I+2,1)
1253	4	CONTINUE
1254		WRITE(IPRINT,1)
1255	1	FORMAT(1H .20X, 'MODE VELOCITY (T=0)'.//.
1256	-	#'NODE', 6X, 'X', 17X, 'Y', 19X, 'ROTATION'./)
1257		DO 2 I=1.NDF-2.3
1250		T = T N T (FLOAT (T) / NF + 0.7)
1410		TT-THILL DOUT (T) / HL + 0 + 1 /

```
1259
               WRITE(IPRINT, 3)II, VMODE(I, 2), VMODE(I+1, 2), VMODE(I+2, 2)
        3
1260
               FORMAT(1H , I3, 3X, E13.6, 6X, E13.6, 9X, E13.6, /)
        2
1261
               CONTINUE
               WRITE(IPRINT,7)
1262
        7
1263
               FORMAT(1H ,/,20X, 'MOMENTS AND AXIAL FORCES (T=0)',//
              #, 'ELEMENT',8X,
1264
              #'MOMENT(A)',8X, 'MOMENT(B)',8X, 'AXIAL FORCE',/)
1265
1266
               DO 8 IE=1, NE
1267
               WRITE(IPRINT,9)IE,FORMOD(IE,1),FORMOD(IE,2),FORMOD(IE,3)
        9
               FORMAT(1H, 2X, 12, 8X, E13.6, 4X, E13.6, 4X, E13.6, /)
1268
1269
        8
               CONTINUE
        С
1270
               OUTPUT INITIAL ELEMENT LENGHTS
1271
               WRITE(IPRINT, 34)
               FORMAT(1H ,//, ' ELEMENT',8X, 'INITIAL ELEMENT LENGTH',/)
1272
         34
1273
               DO 33 IE=1,NE
1274
               WRITE(IPRINT, 32)IE, CL(IE)
1275
         32
               FORMAT(1H ,2X,12,8X,E13.6,/)
1276
         33
               CONTINUE
1277
        С
1278
               ELSE
1279
               AMPERC=100.*AMP(2)/AMP(1)
1280
               WRITE(IPRINT, 11)T, DT, AMPERC
        11
               FORMAT(1H ,/, 'TIME IS ', E11.6, 15X, 'TIME INCREMENT IS'
1281
              #,EII.6,15X, 'VELOCITY AMPLITUDE(PERCENT) IS ',F7.3,/)
1282
1283
               IF (DISPL.EQ. 'LARGE') THEN
1284
               WRITE (IPRINT, 60)
1285
        60
               FORMAT(1H ,18X,'
                                   MODE SHAPE',//,'
              # 'NODE', 6X, 'X', 17X, 'Y', 19X, 'ROTATION', /)
1286
               DO 40 I=1,NDF-2,3
1287
1288
               II=INT(FLOAT(I)/NF+0.7)
               WRITE(IPRINT,3)II,PHI(I,1),PHI(I+1,1),PHI(I+2,1)
1289
        40
1290
               CONTINUE
1291
               END IF
1292
               WRITE (IPRINT, 12)
         12
               FORMAT(1H, 20X, 'MOMENTS AND AXIAL FORCES', //, '
1293
              #'ELEMENT',8X, 'MOMENT(A)',
1294
              #8X, 'MOMENT(B)',8X, 'AXIAL FORCE',/)
1295
1296
               DO 13 IE=1,NE
1297
               WRITE(IPRINT,9)IE, FORMOD(IE,1), FORMOD(IE,2), FORMOD(IE,3)
1298
        13
               CONTINUE
               WRITE (IPRINT, 14)
1299
               FORMAT(1H ,/,26X, 'VELOCITY',//,
1300
        .14
              #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/)
1301
              #ON',/)
DO 15 I=1,NDF-2,3
1302
1303
               II=INT(FLOAT(I)/NF+0.7)
1304
1305
               WRITE (IPRINT, 3) II, VMODE (I, 2), VMODE (I+1, 2), VMODE (I+2, 2)
         15
1306
               CONTINUE
1307
               WRITE(IPRINT, 16)
               FORMAT(1H ,/,26X, 'DISPLACEMENTS',//,
         16
1308
              #'NODE',6X,'X',17X,'Y',19X,
1309
              #'ROTATIONS',/)
1310
1311
               DO 17 I=1,NDF-2,3
               II=INT(FLOAT(I)/NF+0.7)
1312
1313
               WRITE(IPRINT, 3)II, UMODE(I), UMODE(I+1), UMODE(I+2)
         17
1314
               CONTINUE
1315
         С
1316
               END IF
1317
         С
```

1318	С	STORE CURRENT COORDINATES
1319		IF (PICT.EQ. 'PLOT') THEN
1320		IPLTS=IPLTS+1
1321		DO 21 I=1,NN
1322		XCOORD(I, IPLTS)=COORDX(I)
1323		YCOORD(I, IPLTS)=COORDY(I)
1324	21	CONTINUE
1325		END IF
1326	С	
1327	Ŭ	RETURN
1328		DEBUG SUBCHK
1329		FND
1320	C	
1221	Č	
1001	C	*******
1222		
1224		
1005		PICIUR
1335		DI ANG INIMINI CODUCOUDE NND CUDCEOUEND
1336	C	PLOTS INITIAL STRUCTURE AND SUBSEQUENT
133/	C	DEFORMED SHAPE THROUGHOUT TIMESPAN OF
1338	C	DEFORMATION
1339	C	
1340	C	
1341		SUBROUTINE PICTUR
1342		INCLUDE GNLMIST.COMPROC
1343	0	DOUBLE PRECISION XMAX, YMAX
1344	C	
1345		CALL NEWPAG
1346		CALL PAGS12(20.5,29.)
1347	-	CALL PLOT $(2.0, 5.0, -3)$
1348	C	
1349		XMAX = -999.D0
1350		YMAX=-999.D0
1351		DO 22 J=1, IPLTS
1352		DO I I=1, NN E_{1} E_{2}
1353		1F(DABS(XCOORD(I,J)).GE.XMAX)XMAX=DABS(XCOORD(I,J))
1354	_	IF(DABS(YCOORD(I,J)).GE.YMAX)YMAX=DABS(YCOORD(I,J))
1355	1	CONTINUE
1356	22	CONTINUE
1357		RMAX=DMAX1(XMAX,YMAX)
1358		FX=18.0/RMAX
1359	·	FY=18.0/RMAX
1360	С	
1361		DO 21 J=1, IPLTS
1362		DO 2 $I=1, NN$
1363		XCOORD(I,J) = XCOORD(I,J) * FX
1364		YCOORD(I,J)=YCOORD(I,J)*FY
1365	2	CONTINUE
1366		CALL $PLOT(0.,0.,3)$
1367		DO 3 I=1,NN
1368		CALL PLOT(XCOORD(I,J),YCOORD(I,J),2)
1369		CALL SYMBOL(XCOORD(I,J),YCOORD(I,J),0.2,11,0.,-1)
1370	3	CONTINUE
1371	21	CONTINUE
1372		RETURN
1373		DEBUG SUBCHK
1374		END
1375	С	
1376	С	·

1378	С	******
1379	C	
1380	C	соммол вгоск
1381	C	
1382	Ċ	CONTAINS ALL VARIABLES REQUIRED IN
1383	C	EACH SUBROUTINE
1384	C	
1385	Č	*****
1386	COMPR	ROC PROC
1387		PARAMETER NE=12
1388		PARAMETER NN=13
1389		PARAMETER NRED=3
1390		PARAMETER INTER=3
1391		PARAMETER NF=3
1392		PARAMETER IREAD=8
1393		PARAMETER IPRINT=5
1394		PARAMETER NDF=NN*NF
1395		PARAMETER KINT=3*INTER
1396		COMMON/BLK1/NBEAM(NE,2),COORDX(NN),COORDY(NN),
1397		<pre>#PARDIF(NRED, NRED), RMASS(NN), IBC(NDF), RELEAS(NRED),</pre>
1398		#COMPAT (NRED), UNITM (NDF, NN), CL (NE), UNITN (NDF, NE, 1),
1399		#SSIN(NE), CCOS(NE), FORCE(NRED+1, NE, 3), DTTDT(2),
1400		<pre>#FLEX(NRED, NRED+1), PHI(NDF, 2), DISIP(2), RLOAD(NDF),</pre>
1401		<pre>#VEL(NDF,2),TT(2),X(NRED),FORCET(NE,3),AMOM(NRED+1),</pre>
1402		#ANORM1 (NRED+1), ANORM2 (NRED+1), VMODE (NDF, 2), TF, T, DT,
1403		<pre>#UMODE(NDF),FORMOD(NE,3),RINT,RLAMDA,U(NDF,2),</pre>
1404		<pre>#XCOORD(NN,15),YCOORD(NN,15),UMODD(NDF)</pre>
1405		COMMON/BLK2/RM0, RN0, EPSI0, EN, EN1, EN2, STADET, TITLE,
1406		<pre>#PICT, DISPL, RK, YSTRS, RMATCH, AMP(2), RKINET, POWB, POWC</pre>
1407		COMMON/BLK3/IFLAG, IDISIP, ITREAC, ITMODE, NNORM, IOUT,
1408		#IRND, NDIV, ICOUNT, NDIVA, IPLTS, MATCHA
1409		CHARACTER STADET*4
1410		CHARACTER TITLE*80
1411		CHARACTER DISPL*5
1412		CHARACTER PICT*4
1413		DOUBLE PRECISION COORDX, COORDY, PARDIF, RMASS, COMPAT,
1414		<pre>#FORCET,DUM,A,B,UNOTM,CL,UNITN,SSIN,CCOS,FORCE,FLEX,</pre>
1415		<pre>#PHI, RLOAD, VEL, X, AMOM, DTTDT, RK, T, TT, ANORM1, ANORM2, RMO,</pre>
1416		<pre>#RNO, EPSIO, EN, EN1, EN2, VMODE, UMODE, FORMOD, AMP, ANORM, FACT</pre>
1417		INTEGER RELEAS
1418	END	

.

APPENDIX C

DAGNVS program listing

3	C	* * * * * * * * * * * * * * * * * * * *
1	C	
4 E		
5	C	DRIVER ROUTINE
6	C	
7	С	******************
8		INCLUDE DAGNVS.COMPROC
9	С	INPUT DATA
10		CALL INPUT
11	C	
10	U	
1 2		
13	~	CALL OUTPUT
14	C	b.
15	5	CONTINUE
16		ITREAC=0
17	С	
18	С	NNORM=1 INDICATES ITERATION ONTO NEW MODE SUCCESSFUL
19	-	$NN \cap BM = 0$
20	C	
20	C	MCOUNT COUNTS NO. OF THERATIONS REQUIRED TO OBTAIN NEW MOD
21		MCOUNTED
22	С	ISTEP IS NO. OF TIME STEP
23		ISTEP=ISTEP+1
24	С	ICOUNT COUNTS NO. OF TIME STEPS UNTIL NCOUNT=NDIV, THEN
25	С	THEN OUTPUT REQUESTED AND ICOUNT INITIALISED.
26	•	TCOUNT=TCOUNT+1
27	C	1000M1 1000M111
27		
28	C	
29		T = T + D T
30	С	
31	10	CONTINUE
32	С	COUNTER FOR LARGE DISPLACEMENT MODE ITERATIONS
33		MCOUNT=MCOUNT+1
34		TE (MCOUNT, EO, 10) THEN
35		WRITE (TORINT 11)
35	11	EDMANTIN /// DOY ! NO CONVERCENCE ONTO NEW MODE IN LARCE
20	ΤΤ	FORMAT(IT, ///, ZOX, NO CONVERGENCE ONTO NEW MODE IN LARGE
31		#DISPLACEMENT MODE ANALYSIS : STOP)
38		STOP
39		END IF
40	С	DETERMINE STATICALLY ADMISSIBLE SETS (ONE FOR EACH D.O.F.)
41	С	
12	Ŭ	
42	0	CALL STAT
43	C	TE NOW MODIL COLUMNON PROVIDER (TOUR 1) OUTB MODE COLUMNON
44	· C	IF NON-MODAL SOLUTION REQUIRED (ISYM=I) SKIP MODE SOLUTION
45	С	ALGORITHM EXCEPT AT T=0 SO AS TO DETERMINE TOTAL TIME
46	С	ESTIMATE
47		IF(T.LT.1.D-9.AND.ISYM.EQ.1.OR.ISYM.EQ.0)THEN
48	С	
49	Ċ	SET UP SELF STRESS SYSTEMS IF STRUCTURE IS STATICALLY
50	Č	TNDETERMINATE
50	C	
DT		IF (NRED-NE-4)ITEN
52	C	
53		MODES=1
54		CALL SLFSTR
55	С	
56		END IF
57	2	CONTINUE
58	Ċ	
50	C	DETERMINE CURRENT LOADING (T-MACC*DUI)
59		COLUEI TOADING (L-MUSSILLI)
60	C	
61		CALL LOAD

-

С.2
62 63	C	CONTINUE
64	C	CONTINUE
65 66	C C	IF STRUCTURE IS STATICALLY DETERMINATE (NRED=4) DON'T NEED TO ITERATE FOR REDUNDANTS
67 68	C	IF(NRED.EQ.4)GO TO 3
69 70 71	C C	ZERO ALL ARRAYS IN ITERATIVE PROCEDURE TO FOLLOW
71 72 72	C	CALL ZERO
73 74 75	C	EVALUATE COMPATIBILITY EQUATIONS
75 76	C	CALL COMEQU
77 78 79	C C C	CALCULATE PARTIAL DIRIVATIVES OF COMPATIBILITY EQUATIONS
80 81	C	CALL PDIFF
82 83	C C	INVERT MATRIX OF PARTIAL DIRIVATIVES FOR NEWTON-RAPHSON METHOD
84 85	C	CALL PIVOT
86 87	C C	CHECK MAGNITUDE OF PERTURBATION IN VALUE OF REDUNDANTS
88 89	C	CALL DELTA
90 91 92	C	IF SMALL(IFLAG=1)CALCULATE VELOCITIES
92 93	C	IF LARGE (IFLAG=0) THEN ITERATE
94 95	С	IF (IFLAG.EQ.0)GO TO I
96 97	C C	CONTINUE
98 99	C C	DETERMINE VELOCITIES ASSOCIATED WITH CURRENT STRESS STATE
100	С	CALL VELOC
102 103 104	C C	CHECK IF ALGORITHM HAS CONVERGED ONTO A MODE IF IN MODE(IDISIP=1) IF NOT(IDISIP=0)SO ITERATE
105 106	C	CALL MODECH
107 108	С	IF(IDISIP.EQ.0)GO TO 2
109 110	4 C	CONTINUE
111 112	C C	IF MEMBRANE SOLUTION PREDOMINATES CALCULATE NEW MATCHING FACTOR IF (MATCHA FO 1) CALL MATCH
114115	C C	MODE SOLUTION MAY NOW COMMENCE
116 117	C C	
118 119	С	CALL INMODE
120	С	IF IN MODE SOLUTION UPDATE CURRENT DISPLACEMENTS

121		IF(INMSOL.EO.1)THEN
1 2 2		CALL STOPE
122		
123	C	IS ITERATION FOR NEW MODE REQUIRED ?
124		IF (NNORM.EQ.O) THEN
125		GO TO 10
125		
126		ELSE
127		CALL HPDATE
1 2 0	C	
128	C	CHECK IF OUTPUT REQUIRED
129		GO TO 9
130		FND IF
130		
131		END IF
132	C	
1 2 2	Č	
133		END IF
134	С	IF T=0, DETERMINE MATCHING FACTOR, ESTIMATE TOTAL TIME TF.
135	C	AND TIME INCREMENT OF THEN DETERMINE INTTIAL BODY ECOCIES
155	C	AND TIME INCREMENT DI. THEN DETERMINE INITIAL BODT FORCES
136	C ·	
137		TF(T, I, T, 1, D-9) THEN
100		
138		IF (ISYM.EQ.0) CALL MATCH
139	С	
140	C	CALCULATE INITIAL FOULLIPPATING FORCES
140	C	CALCULATE INITIAL EQUILIBRATING FORCES
141	С	
142		MODES=0
140	9	
143	C	
144		CALL SLFSTR
145	C	
145		
146	13	CONTINUE
147		ITREAC=0
1 4 0	C	
148	6	CONTINUE
149	С	
150	C	
150	C	SET OF COMPATIBILITY EQUATIONS AND PARTIAL DIRIVATIVE
151	С	
152		CALL COMDIF
152	6	
153	C	
154	С	HAVE BODY FORCES BEEN FOUND ? YES=1 : NO=0
155	C	
155	C	
156		IF(IFLAG.EQ.1)GO TO 7
157	C	
157	C	
158	С	INVERT MATRIX OF PARTIAL DIRIVATIVES
159	C	
1.00	C	
160		CALL PIVOT
161	С	
162		
102		IF(IFLAG.EQ.0)GO IO 6
163	С	
164	7	CONTINUE
165	,	CONTINOL
105	C	
166	С	MATCH SYSTEM IF DIRECT ANALYSIS AND MATCHING FACTOR FKNOWN
167		
107		IF (ISIM.EQ.I) CALL VMAICH
168		IF(IMATCH.EQ.O.AND.ISYM.EQ.1)GO TO 13
169		FND IF
170	0	
T 10	C	
171	С	ONCE BODY FORCES HAVE BEEN OBTAINED . ACCELERATIONS ARE
170	C	
1/2	C	CALCULATED
173	С	
174		CALL ACCAYC
1/7	~	
175	C	COMMENCE IMPLICIT FORWARD INTEGRATION SCHEME
176	С	REVISE STATICALLY ADMISSIBLE SET
177	č	NEVIOL SIMICALLI ADMOSTBLE SEI
T / /	C	
178	8	CONTINUE
179	C	
± , /	~	

180		ITREAC=0
181		IRND=IRND+1
182		TF(TRND, GT, 1) THEN
183	C	
184	C	CALL STAT
195	C	CALL STAT
105	C	NODER
186	a	MODES=0
187	C	
188	_	CALL SLFSTR
189	C	
190		CALL COMDIF
191	С	
192		END IF
193	С	
194	С	IMPLICIT TIME INTEGRATION SCHEME
195	С	
196		CALL IMPLIC
197		CALL PIVOT
198		CALL REVISE
199	C	
200	C	IS DELTA X SHEFT CIENTLY SMALL 2
200	C	
201	C	$\frac{1001-0}{100} = 100$
202		IF (IRND, EQ, IOU) THEN
203	102	WRITE (IPRINT, 103) $PORTURE IN PORTURE IN TERMS$
204	103	FORMAT(///,20X, 'NO CONVERGENCE IN EQUILIBRIUM ITERATIONS :
205		#DECREASE TIME STEP',/)
206		STOP
207		END IF
208	С	
209		IF(IOUT.EQ.0)GO TO 8
210	С	
211		IRND=0
212	С	
213	С	CHECK IF DIRECT SOLUTION HAS CONVERGED INTO MODE SOLUTION
214	-	IF (INMSOL, EO, O) THEN
215		CALL CHECK
216		FND TE
210	C	
217	C	DUACE
210	C	THADE
219	9	IF (INMSOL.EQ.I.AND.NNORM.EQ.U) CALL OUTPUT
220	C	
221	9	CONTINUE
222	C	
223	С	IS OUTPUT REQUIRED DURING ANALYSIS PHASE
224		IF(ICOUNT.EQ.NDIV)THEN
225	С	
226		CALL OUTPUT
227		ICOUNT=0
228		END IF
229	С	
230	С	CHECK IF VELOCITY NEAR ZERO (IS STRUCTURE AT REST 2)
231	C	THIS CHECK IS PERFORMED BY COMPARING THE CURRENT
2.32	Č	MOMENTIM WITH THE ORIGINAL MOMENTUM (VEL*MASS)
232	C	HOLESTON WITH THE OKIGINAL HOLESTON (VED MADD)
231	C	
234		
200		
236		RMMC=0
237	С	
238		DO 12 $1=2$, NN

239 240 241 242 243		<pre>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)</pre>
244 245	C 12	CONTINUE
246 247 248	С	IF((RMMC/RMMO).GT.0.05)GO TO 5
248	C	CALL OUTPUT
250	С	
251 252 253		IF(PICT.EQ.'PLOT')CALL PICTUR STOP END
254	С	
255	C	
256	С	
257	С	
258	С	* * * * * * * *
259	С	
260	C	INPUT
261	C	
262	C	INPUT DATA
263	C	* * * * * * * *
265	C	SUBROUTTINE INPUT
266		INCLUDE DAGNVS.COMPROC
267		DIMENSION IIBC(NF), VELDUM(NF)
268	100	FORMAT()
269	101	FORMAT(A4)
270	102	FORMAT (A80)
271	103	FORMAT(A5)
272		READ(IREAD, 102)TITLE
273		READ(IREAD, 101)PICT
274		READ(IREAD, 100)PMATCH
275		READ(IREAD, 100)HH, BB, YSTRS, EPSID, EN
270		THIS PROGRAM HAS COMBINED MODE/DIRECT ANALYSIS FACILITY
278		BUT ITS USE HAS PROVED INEFFICIENT.
279		ISYM=1
280	С	NODE INCIDENCES (NUMBERING OF ELEMENT ENDS)
281		DO 11 IE=1,NE
282		READ(IREAD, 100)(NBEAM(IE, I), $I=1, 2$)
283	11	CONTINUE
284	С	COORDINATES OF ORDERED NODES (X,Y)
285		DO 12 I=1,NN $PPPP(T) = PPP(T)$
280	10	CONTINUE
207	12	CONTINUE BOUNDARY CONDITIONS
289	C	FREEDOM • O
290	C	FIXITY : 1
291	Ũ	DO 2 I=1,NDF
292		IBC(I)=0
293	2	CONTINUE
294	3	READ(IREAD, 100)N1
295		IF(N1.LT.O) GO TO 4
296		READ(IREAD, 100)(IIBC(J), J=1, NF)
297		II = NF*(N1-1)

298		DO 5 $I=1, NF$
299		II=II+1
300		<pre>IBC(II)=IIBC(I)</pre>
301	5	CONTINUE
302		GO TO 3
303	4	CONTINUE
304	С	IS STRUCTURE DETERMINATE OR HYPERSTATIC ?
305	-	IF(NRED.EQ.4) THEN
306	C	IF DETERMINATE , IS IT A CANTILEVER OR SIMPLY SUPPORTED ?
307		READ(IREAD, 101)STADET
308	0	ELSE
210	C	IF HIPERSTATIC , NRED DEGREES OF FREEDOM MOST BE RELEASED
310	C	A CANTILEVER OR A SIMPLY SUPPORTED STRUCTURE ?
312	C	READ(IREAD, 101)STADET
313	С	DEGREES OF FREEDOM WHICH ARE RELEASED
314	Ŭ	READ(IREAD, 100) (RELEAS(I), $I=1$, NRED)
315		END IF
316	С	READ IN MASS VECTOR
317		READ(IREAD, 100)(RMASS(I), I=1, NN)
318	С	READ IN INITIAL VELOCITY
319	6	READ(IREAD, 100)N1, N2
320		IF(N1.LT.O)GO TO 7
321		READ(IREAD, 100)(VELDUM(I), I=1, NF)
322		II=NF*(N1-1)
323		JJ=N2-N1+1
324		DO 8 $J=1, JJ$
325		DO 8 I=1,NF
326		
327	0	VEL(II,I)=VELDUM(I)
328	8	CONTINUE
329	7	GO TO O
331	ć	STATE HOW MANY TIME INTERNALS IN MODE SOLUTION ARE REGULERD
332	C	AND NUMBER OF OUTPUTS REQUIRED
333	C	READ (IREAD, 100) RINT, NDIV
334		READ (IREAD, 100) XXDUM
335	С	CALCULATE YIELD MOMENT AND AXIAL YIELD STRESS
336	Ū	AREA=HH*BB
337		RMO=AREA*HH*YSTRS/4.
338		RNO=AREA*YSTRS
339	.C	
340	С	SET INITIAL MODE SHAPE (GUESS) EQUAL TO INITIAL VELOCITY
341		DO 1 I=1,NDF
342		PHI(I,1)=VEL(I,1)
343		DUMVEL(I) = VEL(I, 1)
344	1	CONTINUE
345	С	CALCULATE NORMALISATION CONSTANT AND INITIAL DISSIPATION
346	С	RATE
347		DO 10 I=1,NDF,3
348		II = INT(FLOAT(I)/NF+0.7)
349		
350		DISIP(2) = DISIP(2) + VEL(1,1) * RMASS(11) * PHI(1,1)
351	10	DISIP(2) = DISIP(2) + VEL(1K, 1) * RMASS(11) * PHI(1K, 1)
352	10	CONTINUE
303	C	
354	C	LE SIKUCTUKE INDETERMINATE(NKED NOT EQUAL TO 4), DUMMI VALUES
356	C	ZERO IN FORMULATION OF COMPATIBILITY EOUATIONS . (THAT

с.7

357 358 359	С	SITUATION ARISES IF STATICALLY ADMISSIBLE B.M.D. IS ZERO) IF(NRED.NE.4)THEN DO 9 I=1,NRED
360		X(I) = 1.0
361 362	9	CONTINUE END IF
363	С	
364	C	ASSIGN DUMMY VALUES TO BODY FORCES
365	C	MEDICA DOINT VIHOLD TO BODI TOKCHD
366		DO 13 I=1,2*NN
367		XX(I) = XXDUM
368	13	CONTINUE
369	Ċ	CONTINUE
370	C	CALCULATE NO OF BODY FORCES TO BE DETERMINED
271	C	CALCOLATE NO. OF BODI FORCES TO BE DETERMINED
371	C	T = (C = A) = A = A = A = A = A = A
372		IF (STADET.EQ. CANT .AND.NRED.NE.4)NINV=2^NN-1
373		1F(STADET.EQ.SIMP'.AND.NRED.EQ.4)NINV=2*(NN-1)-1
374		IF(STADET.EQ.'CANT'.AND.NRED.EQ.4)NINV=2*(NN-1)
375	С	
376	С	SET UP ARRAY CONTAINING DEGREE OF FREEDOM OF REACTIONS
377		IF(STADET.EQ. 'CANT')THEN
378		DO 14 $I=1, 2*(NN-1)$
379		I = INT(FLOAT(I)/2+0.55)
380		12 = NF + T + T - 1
381		TSTAT(T) = T2
202	14	
202	14	
303	0	
384	C	
385		1F(STADET.EQ.'CANT'.AND.NRED.NE.4) ISTAT(2*NN-1) = NDF
386		IF(STADET.EQ.'SIMP')THEN
387		DO 15 I=1,2*NN-3
388		I1=INT(FLOAT(I)/2+0.55)
389		12=NF+11+1-1
390		IJ=I
391		IF(NRED.NE.4)IJ=I+1
392		TSTAT(TJ) = T2
393	15	CONTINUE
394	15	TE (NDED NE /) THEN
205		
395		1 STAT(1) = 3
396		1 STAT(2 *NN - 1) = NDF
397		END IF
398		END IF
399	С	
400	С	STORE INITIAL COORDINATES
401	С	
402		DO 66 I=1,NN
403		XCOORD(I) = COORDX(I)
404		YCOORD(I) = COORDY(I)
405	66	CONTINUE
406	C	CONTINUE
400	C	FN1-FN11
407		
408	0	ENZ = ENI + I
409	C	
410		RETURN
411		DEBUG SUBCHK
412		END
413	С	
414	С	

116	C	****
410	C	
41/	C	
418	С	DATA
419	С	
420	С	DISPLAYS ALL INPUT DATA FOR VERIFICATION
421	C	
422	č	****
422	C	
423		SUBROUTINE DATA
424		INCLUDE DAGNVS.COMPROC
425		WRITE(IPRINT,1)TITLE
426	1	FORMAT(1H1,5X,80('*'),/10X,A80,/,6X,80('*'),/)
427		WRITE(IPRINT, 17)
428	17	FORMAT(1H ,/,20X,'LARGE DISPLACEMENT ANALYSIS',/)
429		WRITE (IPRINT, 13) EN, RMO, RNO, EPSIO, YSTRS, HH, BB
430	13	FORMAT(1H .5X.///. 'MATERIAL ASSUMPTIONS' // ' HOMOGENEOUS
431		\pm VISCOUS WITH POWER N =' F7 3 / 2Y 'VIELD MOMENT -'
432		# VISCOUS WITH FOWER N = , F7.5, /, 2X, FIELD MOMENT = , # RIL 6 / 2X 'AXIAL VIELD CODENCOUL - , RIC 4 /
432		$\frac{1}{2} + \frac{1}{2} + \frac{1}$
433		#2X, INITIAL STRAIN RATE = , E10.4, /, 2X, YIELD
434		# STRESS =', E10.4, /, 2X, 'SECTION DEPTH =', E15.6, /, 2X,
435		#'SECTION WIDTH =',E15.6,/)
436		WRITE(IPRINT,2)NE,NN
437	2	FORMAT(1H ,/, 3X, 'NUMBER OF ELEMENTS :', I3, /, 3X, 'NUMBER OF
438		#NODES :', I3, /)
439		WRITE(IPRINT.3)
440	3	FORMAT(// 6X ' COORDINATES OF NODES' //
110	0	# NODE' 10X 'X' 12X 'V')
441		= 1001 , 100, x , 120, 1
442		DO IIO I=1, NN (T) COOPDU(T) COOPDU(T)
443		WRITE(IPRINT, 4)I, COORDX(I), COORDY(I)
444	4	FORMAT(1H ,13,1X,2(2X,F11.5))
445	110	CONTINUE
446		WRITE(IPRINT,5)
447	5	FORMAT(//, ' BOUNDARY CONDITIONS : 0=FREEDOM , 1=FIXITY',
448		#//, ' NODE', 3X, 'X', 3X, 'Y', 3X, 'ROTATION')
449		WRITE(IPRINT, 6)1, IBC(1), IBC(2), IBC(3)
450	С	IF STRUCTURE STATICALLY DETERMINATE NO BOUNDARY
451	C	CONDITIONS AT LAST NODE.
452	Ũ	TE(NPED NE A)THEN
452		$\frac{1}{1} \left(\frac{1}{1} \frac$
455		WRITE(IPRINT, 6)NN, IBC(NDF-2), IBC(NDF-1), IBC(NDF)
454	~	END IF
455	6	FORMAT(1H, 13, 4X, 11, 3X, 11, 6X, 11)
456		WRITE(IPRINT,11)
457 ·	11	FORMAT(1H ,///,5X,'LUMPED MASS PER
458		#NODE',//,6X,'NODE',10X,'MASS',/)
459		DO 111 $I=1, NN$
460		WRITE(IPRINT, 12)I, RMASS(I)
461	12	FORMAT(1H.5X, I2, 6X, E11, 4, /)
462	111	CONTINUE
463		TF(NRED_NE_4)THEN
160		IE (EE ADE E E C CANEL WEIGE (IDEING O) (DEIEAC(I) I I NDED)
404	0	EOPMAT(1H // ! CTRUCTURE CANTUREDED DY DELEAS(1), 1=1, NRED)
405	9	FORMAI(IN ,//, SIRUCTURE CANTILEVERED BY RELEASING
466		$\frac{1}{2} RESTRAINTS AT D.O.F. (1X, 12)$
467		IF(STADET.EQ, SIMP) WRITE(IPRINT, IU)(RELEAS(I), I=1, NRED)
468	10	FORMAT(1H ,//,' STRUCTURE MADE SIMPLY SUPPORTED BY
469		<pre>#RELEASED RESTRAINTS AT D.O.F. ',5(3X,12))</pre>
470		END IF
471		WRITE(IPRINT,14)
472	14	FORMAT(1H .///.10X. 'INITIAL VELOCITY' //.'
473		#' NODE', 4X, 'X', 12X, 'Y', 9X, 'ROTATION')
474	С	

с.9

475	C	
476	C	20 112 T=1 אתוא I
477		112 1-1, NDF, 5
470	0	II-INI(I/NF)+I
478	C	STORE INITIAL VELOCITY
479		VINIT(1) = VEL(1,1)
480		VINIT(I+1) = VEL(I+1,1)
481		WRITE(IPRINT,15)II,VEL(I,1),VEL(I+1,1),VEL(I+2,1)
482	15	FORMAT(1H ,I3,1X,3(2X,E11.3),/)
483	112	CONTINUE
484	С	
485		RETURN
486		DEBUG SUBCHK
487		END
488	С	
489	С	
490	С	***************************************
491	С	
492	Č	STAT
493	Ċ	51/11
494	Ċ	THE STRUCTURE IS MADE STATICALLY DETERMINATE BY DELEASING
405	C	POUNDARY CONDUCTORS CHORE BIAITCALLI DETERMINATE BI RELEASING
495	C	(CONTRACT CONDITIONS SUCH THAT IT BECOMES EITHER A CANTILEVE
490	C	(CANT) OR SIMPLY SUPPORTED (SIMP') . THE BENDING MOMENT
497	C	AND AXIAL FORCE IN THE RELEASED STRUCTURE DUE TO A UNIT LOA
498	C	APPLIED IN TURN TO EACH DEGREE OF FREEDOM IS THEN
499	C	DETERMINED.
500	C	BENDING : UNITM(I,J), I=D.O.F. WHERE LOAD APPLIED
501	C	J = NODE NO.
502	С	AXIAL : UNITN(I,J),I=D.O.F. WHERE LOAD APPLIED
503	C	J = ELEMENT NO.
504	С	FORCE(I,IE,J) : SELF-STRESS , STATICALLY ADMISSIBLE SETS
505	С	I=1 : STATICALLY ADMISSIBLE SET
506	С	I=2,NRED+1: SELF STRESS SYSTEMS
507	С	IE : 1, NO. OF ELEMENTS
508	С	J=1 : MOMENT AT 'A' END OF ELEMENT
509	C	J=2 : MOMENT AT 'B' END OF ELEMENT
510	Č	J=3 • AXIAL FORCE IN ELEMENT
511	Ċ	
512	C	
512	C	*****
515	C	
514		SUBROUTINE STAT
515	0	INCLUDE DAGNVS.COMPROC
516	Ċ	
517	_	I TMODE=0
518	C	DETERMINE ORIENTATION OF ELEMENTS IN GLOBAL AXIS SYSTEM
519		DO 55 IE=1,NE
520		IL=NBEAM(IE,1) @NODE NO. OF A END OF ELEMENT IE
521		IR=NBEAM(IE,2) @NODE NO. OF B END OF ELEMENT IE
522	C	CL=CURRENT LENGTH OF ELEMENT
523		CL(IE)=SQRT((COORDX(IR)-COORDX(IL))**2.DO+(COORDY(IR))
524		#-COORDY(IL))**2.D0
525	С	
526		IF(T.LT.1.D-9)RLO(IE) = CL(IE)
527	С	
528	-	SSIN(TE) = (COORDY(TE) - COORDY(TL))/CL(TE)
529		CCOS(IE) = (COORDX(IR) - COORDX(IL))/CL(IE)
520	55	
530	55 C	
222	C	FOR DENDING MOMENTE DUE TO UNITE LOND AT DO T
552	C	FOR CANULLEVED
533	C	FOR CANTILEVER

534 IF (STADET.EQ. 'CANT') THEN 535 С HORISONTAL С 536 537 DO 1 I=4, NDF, NFRI=FLOAT(I)/NF+0.1 538 539 IR=INT(RI) 540 DO 2 J=1,IR UNITM(I,J) = -(COORDY(IR+1) - COORDY(J))541 542 2 CONTINUE 543 1 CONTINUE 544 С С 545 VERTICAL 546 DO 3 I=5, NDF, NF547 RI = FLOAT(I)/NF+0.1548 IR=INT(RI) 549 DO 4 J=1, IR550 UNITM(I,J) = -(COORDX(J) - COORDX(IR+1))551 4 CONTINUE 552 3 CONTINUE С 553 MOMENTS 554 DO 26 I=6, NDF, NF555 RI=FLOAT(I)/NF+0.1 556 IR=INT(RI) 557 DO 27 J=1,IR UNITM(I,J)=1. 558 559 27 CONTINUE 560 26 CONTINUE 561 ELSE 562 С FOR SIMPLY SUPPORTED STRUCTURE С 563 HORISONTAL LOADING AT D.O.F. I 564 DO 8 I=4, NDF-2, NF565 DO 9 J=2,NN-1 566 R1 = 0. 567 RI = FLOAT(I)/NF+0.1568 IR=INT(RI) 569 IF(J.GT.IR+1)R1=1.UNITM(I,J) =-COORDX(J) *COORDY(IR+1)/COORDX(NN) 570 #+(COORDY(J)-COORDY(1))-R1*(COORDY(J)-COORDY(IR+1)) 571 572 9 CONTINUE 573 8 CONTINUE 574 С VERTICAL UNIT LOADING AT D.O.F. I 575 DO 10 I=5,NDF-1,NF 576 DO 11 J=2,NN-1 577 R1=0RI=FLOAT(I)/NF+0.1 578 IR=INT(RI) 579 580 IF(J.GT.IR+1)R1=1.581 UNITM(I,J) = (COORDX(IR+1)/COORDX(NN)-1.)*COORDX(J)#+R1*(COORDX(J)-COORDX(IR+1))582 11 CONTINUE 583 584 10 CONTINUE 585 С MOMENT AT D.O.F. I 586 DO 30 I=3, NDF, NFDO 31 J=1,NN 587 RI = FLOAT(I)/NF+0.1588 589 IR=INT(RI) R1=0.590 591 IF(J.GT.IR)R1=1.UNITM(I, J) = (COORDX(J) / COORDX(NN) - R1)

592

	C.12

593 IF(I.EQ.3)UNITM(I,1) = -1.594 IF(I.EQ.NDF)UNITM(I,NN)=1. 595 31 CONTINUE 596 30 CONTINUE 597 END IF 598 С 599 С FOR AXIAL FORCES DUE TO UNIT LOADS AT D.O.F. I С 600 HORISONTAL UNIT LOADS 601 С 602 DO 13 I=4, NDF, NF 603 RI = FLOAT(I)/NF + 0.1604 IR=INT(RI) 605 IRR=IR 606 IF(STADET.EQ.'SIMP') IRR=NE 607 DO 14 IE=1, IRR 608 Rl=0609 IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') R1=1. 610 IF (STADET.EQ. 'CANT') THEN 611 UNITN(I,IE)=CCOS(IE) 612 ELSE UNITN(I,IE)=CCOS(IE)+COORDY(IR+1)*SSIN(IE)/COORDX(NN) 613 614 #-R1*CCOS(IE)END IF 615 616 14 CONTINUE 13 617 CONTINUE 618 С С 619 VERTICAL UNIT LOADS 620 DO 16 I=5, NDF, NF621 RI=FLOAT(I)/NF+0.1 622 IR=INT(RI) 623 IRR=IR624 IF(STADET.EQ.'SIMP') IRR=NE 625 DO 17 IE=1, IRR626 R1=0627 IF((NF*IE).GT.I.AND.STADET.EQ.'SIMP') R1=1. 628 IF (STADET.EQ. 'CANT') THEN 629 UNITN(I,IE)=SSIN(IE) 630 ELSE 631 UNITN(I,IE)=SSIN(IE)*(1.-COORDX(IR+1)/COORDX(NN)) 632 #-R1*SSIN(IE)633 END IF . 17 634 CONTINUE 16 635 CONTINUE 636 С 637 С UNIT APPLIED MOMENT С 638 NO AXIAL FORCES CAUSED BY UNIT MOMENTS IN CANTILEVER 639 IF (STADET.EQ. 'SIMP') THEN 640 DO 33 I=3, NDF, NF641 DO 34 IE=1,NE 642 UNITN(I,IE)=-SSIN(IE)/COORDX(NN) 643 34 CONTINUE 644 33 CONTINUE 645 END IF 646 С 647 RETURN 648 DEBUG SUBCHK 649 END 650 С С 651

652	С	*******
653	С	
654	С	SLFSTR
655	С	
656	С	FOR BOTH MODE SOLUTION (MODES=1) AND DIRECT
657	С	ANALYSES (MODES=0) SELF-STRESS SYSTEMS
658	С	CORRESPONDING TO RELEASED DEGREES OF FREEDOM
659	С	(MODES=1) AND EACH DEGREE OF FREEDOM AT A
660	С	NODE (MODES=0) ARE ASSEMBLED .
661	С	
662	С	* * * * * * * * * * * * * * * * * * * *
663		SUBROUTINE SLFSTR
664		INCLUDE DAGNVS.COMPROC
665	С	
666	C	IF MODAL SOLUTION PHASE SET UP SELF STRESS SYSTEMS
667	C	EQUAL TO MOMENTS AND AXIAL FORCES ASSOCIATED WITH
668	Č	UNIT LOADS AT RELEASED D.O.F.'S
669		IF (MODES, EO, 1) THEN
670		$DO = 1 = 2 \cdot NRED + 1$
671		DO 2 IE=1.NE
672		FORCE(I, IE, I) = UNITM(RELEAS(I-I), NBEAM(IE, I))
673		FORCE(I, IE, 2) = UNITM(RELEAS(I-1), NBEAM(IE, 2))
674		FORCE(I, IE, 3) = UNITN(RELEAS(I-1), IE)
675	2	CONTINUE
676	1	CONTINUE
677	Ĉ	
678	U	ELSE
679	С	
680	C	FOR DIRECT ANALYSIS SET UP SELF STRESS SYSTEMS
681	Č	CORRESPONDING TO EACH DEGREE OF FREEDOM IN THE
682	C	STRUCTURE .
683	Č	
684	Ũ	DO 3 $IE=1.NE$
685		DO 4 T=1.NTNV
686		FORCE(I, IE, I) = UNITM(ISTAT(I), NBEAM(IE, I))
687		FORCE(I, IE, 2) = UNITM(ISTAT(I), NBEAM(IE, 2))
688		FORCE(T, TE, 3) = UNITN(TSTAT(T), TE)
689	4	CONTINUE
690	3	CONTINUE
691	Ċ	
692	Ŭ	END TF
693		RETURN
694		DEBUG SUBCHK
695		END
696	С	
697	C	
698	C	* * * * * * * * * * * * * * * * * * * *
699	C	
700	C	LOAD
701	Ċ	
702	С	ASSEMBLE THE STATICALLY ADMISSIBLE MOMENT AND AXIAL FORCE
703	С	DIAGRAM BY FACTORING THE UNIT DIAGRAMS BY THE LOAD VECTOR
704	C	PER D.O.F. AND SUM ALL CONTRIBUTIONS AT A NODE.
705	C	THE 'LOADS' ARE GIVEN BY (MASS*CURRENT MODE SHAPE)
706	č	
707	C	*****
708	-	SUBROUTINE LOAD
709		INCLUDE DAGNVS.COMPROC
710	С	

711 ITREAC=0 712 DO 1 IE=1,NE 713 DO 2 J=1,3 714 FORCE(1, IE, J) = 0.715 2 CONTINUE 716 1 CONTINUE 717 DO 18 I=1.NDF 718 II=INT(FLOAT(I)/NF+0.7) С 719 IF D.O.F. IS A ROTATION THEN NO LOAD ASSOCIATED WITH IT 720 IJ = (INT(FLOAT(I)/NF+0.001))*NF721 IF(I.NE.IJ)THEN 722 RLOAD(I)=RMASS(II)*PHI(I,1) 723 END IF 724 DO 19 IE=1,NE 725 FORCE(1,IE,1)=FORCE(1,IE,1)+UNITM(I,NBEAM(IE,1))*RLOAD(I) 726 FORCE(1,IE,2)=FORCE(1,IE,2)+UNITM(I,NBEAM(IE,2))*RLOAD(I) С 727 AXIAL FORCES 728 FORCE(1, IE, 3) = FORCE(1, IE, 3) + UNITN(I, IE) * RLOAD(I)729 IF (NRED.EO.4) THEN 730 FORCET(IE, 1) = FORCE(1, IE, 1)731 FORCET(IE, 2) = FORCE(1, IE, 2)732 FORCET(IE,3) = FORCE(1,IE,3)733 END IF 19 734 CONTINUE 735 18 CONTINUE 736 С С 737 738 RETURN 739 DEBUG SUBCHK 740 END 741 С С 742 С 743 744 С 745 С ZERO 746 С С 747 INITIALISE ALL RELEVENT ARRAYS USED IN THE 748 С ITERATIVE PROCEDURE TO DETERMINE REDUNDANTS. 749 С THIS PROCEDURE APPLIES ONLY WHEN OPTION С 750 OF COMBINED MODE AND DIRECT ANALSIS 751 С PROCEDURE (ISYM = 0) IS APPLIED. 752 С 753 С 754 SUBROUTINE ZERO 755 INCLUDE DAGNVS.COMPROC 756 С 757 DO 1 I=1, NRED758 FLEX(I, NRED+1) = 0.759 COMPAT(I)=0.760 DO 2 J=1,NRED FLEX(I,J)=0.761 762 PARDIF(I,J)=0.763 2 CONTINUE 764 1 CONTINUE 765 С 766 RETURN 767 DEBUG SUBCHK 768 END 769 С

770	С	
771	С	* * * * * * * * * * * * * * * * * * * *
772	С	
773	C	COMEOU
774	C	
775	Ċ	ASSEMBLE NRED COMPATIBILITY FOUATIONS (NRED
776	C	- NO OF DEDINDANTE) IF CODUCTIONS (NED
770		- NO. OF REDUNDANTS) IF STRUCTURE IS STATICALLY
770	C	INDETERMINATE IN TERMS OF UNKNOWN REDUNDANT
770	C	FORCES
779	C	
780	C	~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
781		SUBROUTINE COMEQU
782		INCLUDE DAGNVS.COMPROC
783		ITREAC=ITREAC+1
784	С	SUM OVER ALL ELEMENTS IE
785		DO 1 IE=1,NE
786	С	SET UP PARAMETERS FOR CALCULATIONS TO FOLLOW
787		DO 2 $I=1, NRED+1$
788	С	MOMENTS AND AXIAL FORCES ARE NORMALISED
789		AMOM(I) = (FORCE(I, IE, 2) - FORCE(I, IE, 1)) / RMO
790		ANORM1(I)=FORCE(I,IE,1)/RMO+FORCE(I,IE,3)/RNO
791		ANORM2(I) = FORCE(I, IE, 3)/RNO - FORCE(I, IE, 1)/RMO
792	2	CONTINUE
793	-	SUMA=ANORM1(1)
794		SUMB=ANORM2(1)
795		$SUMC = \lambda MOM(1)$
796		DO 3 I - 1 NPED
790		DO S I = I, NED
797		SUMA=SUMA+X(1) *ANORMI(1+1)
798		SUMB=SUMB+X(1) ANORM2(1+1)
799	•	SUMC=SUMC+X(1) *AMOM(1+1)
800	3	CONTINUE
801	C	SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
802		SIG=1.
803		SIGA=1.
804		SIGB=1.
805		SIGC=1.
806		SIGD=1.
807		IF((SUMA+SUMC).LT.O) SIGA=-1.
808		IF((SUMB-SUMC).LT.0) SIGB=-1.
809		IF(SUMA.LT.O) SIGC=-1.
810		IF(SUMB.LT.0) SIGD=-1.
811		POWI=SIGA*(DABS(SUMA+SUMC)**EN)
812		POWJ = SIGC * (DABS(SUMA) * * EN)
813		POWK=SIGB*(DABS(SUMB-SUMC)**EN)
814		POWL=SIGD*(DABS(SUMB)**EN)
815		POWA = POWT * (SIMA + SIMC)
816		$POWP = POWF (SUMP_SUMC)$
917		POWC = POWR (SUMD - SUMC)
017		POWD-POWR*(SUMP CUMC)
010		POWD=POWB" (SUMB-SUMC)
019		POWE-POWD SUMA
820		POWE POWE SUMB
821		POWG=POWE^SUMA
822		POWH=POWF SUMB
823		PRODA=EN1*SUMC/CL(IE)
824		PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
825		PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
826	С	SET UP COMPATIBILITY EQUATIONS, ONE FOR EACH REDUNDANCY
827		DO 4 I=1,NRED
828		COMPAT(I)=COMPAT(I)+(FORCE(I+1,IE,3)*EPSI0*0.5)*

829 830 831 832 833 834 834	4	<pre>#((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)) CONTINUE CONTINUE</pre>
836	Ċ	CONTINUE
837	C	DEMION
030		DEDIC CUDCUK
030		FND
039	C	END
040	C	
041	C	*****
042	C	
843	C	
044	C	
845	C	
040	C	TO OBTAIN UNKNOWN REDUNDANTS & THE PARTIAL
847	C	DIRIVATIVES OF THE COMPATIBILITY EQUATIONS ARE
848	C	CALCULATED IN TERMS OF REDUNDANTS . THE NEWTON-
849	C	RAPHSON METHOD IS EMPLOYED TO ITERATE ONTO A
850	C	SOLUTION
051 801	C	******
052	C	
853		SUBROUTINE PDIFF
054		DO 1 JE-1 NE
055	C	DO I IE-I,NE SEM UD DADAMEMEDS FOD CALCULATIONS TO FOLLOW
050	C	DO 2 I-1 NDEDL1
057	C	DO 2 I=I,NREDTI Momenues and avial eodoes are normalised
050	C	MOMENTS AND AXIAL FORCES ARE NORMALISED
859		AMOM(I) = (FORCE(I, IE, 2) - FORCE(I, IE, I)) / RMO
860		ANORMI(1)=FORCE(1,1E,1)/RM0+FORCE(1,1E,3)/RN0 NORM2(1) (RODGE(1,1E,2)/RM0+FORCE(1,1E,3)/RN0
861	0	ANORM2(1) = (FORCE(1, 1E, 3) / RNO - FORCE(1, 1E, 1) / RMO)
862	2	CONTINUE
863		SUMA=ANORMI(I)
864		SUMB=ANORM2(1)
865		SUMC=AMOM(1)
866		DO 3 $I=1, NRED$
867		SUMA=SUMA+X(I)*ANORM1(I+1)
868		SUMB=SUMB+X(I)*ANORM2(I+1)
869		SUMC=SUMC+X(I)*AMOM(I+1)
870	3	CONTINUE
871	C	SET UP SIGNUM FUNCTIONS FOR POWERED TERMS
872		SIG=1.
873		SIGA=1.
874		SIGB=1.
875		SIGC=1.
876		SIGD=1.
877		IF((SUMA+SUMC).LT.O) SIGA=-1.
878		IF((SUMB-SUMC).LT.0) SIGB=-1.
879		IF(SUMA.LT.O) SIGC=-1.
880		IF(SUMB.LT.O) SIGD=-1.
881		POWI=SIGA*(DABS(SUMA+SUMC)**EN)
882		POWJ=S1GC*(DABS(SUMA)**EN)
883		POWK=SIGB*(DABS(SUMB-SUMC)**EN)
884		POWL=SIGD*(DABS(SUMB)**EN)
885		POWA=POWI*(SUMA+SUMC)
886		POWB=POWK* (SUMB-SUMC)
887		POWC=POWA*(SUMA+SUMC)

888		POWD=POWB*(SUMB-SUMC)
889		POWE=POWJ*SUMA
890		POWF=POWL*SUMB
891		POWG=POWE*SUMA
892		POWH=POWF*SUMB
893		PRODA=EN1*SUMC/CL(TE)
201		DPODR-FN2*SUMC*SUMC/(CL(TE)*CL(TE))
005		
006	C	
0.70	C	DO 4 I-1 NDED
897		DO 4 I=1, NRED
898		DO S J=1, NRED
899		PARTA=ANORMI(J+I)+AMOM(J+I)
900		PARTB = ANORM2 (J+1) - AMOM (J+1)
901		PART1=((POWI*PARTA-POWJ*ANORM1(J+1))*EN1*PRODA
902		#-EN1*AMOM(J+1)*(POWA-POWE)/CL(IE))/(PRODA*PRODA)
903		PART2=((POWK*PARTB-POWL*ANORM2(J+1))*EN1*PRODA
904		#-EN1*AMOM(J+1)*(POWB-POWF)/CL(IE))/(PRODA*PRODA)
905		PART3=((POWA*PARTA-POWE*ANORM1(J+1))*EN2*EN2*PRODA*PRODA/
906		#(EN1*EN1) - (POWC - POWG)*2.*EN2*PRODA*AMOM(J+1)/(EN1*CL(IE)))/
907		#(PRODB*PRODB)
908		PART4 = ((POWA * ANORM1 (J+1) + SUMA * EN1 * POWI * PARTA) -
909		#(POWE*ANORM1(J+1)
910		$\#+SUMA \times EN1 \times POWJ \times ANORM1 (J+1)) \times PRODA \times PRODA / EN1$
911		#-((POWA-POWE)*2.*PRODA*AMOM(J+1)*SUMA)/CL(IE))/(PRODC*PRODC)
912		PART5 = ((POWB*PARTB - POWF*ANORM2(J+1))*EN2*EN2*PRODA*PRODA/
013		#(EN1*EN1) - (POWD - POWH) *2 *EN2*PRODA*AMOM(.T+1) / (EN1*CL(TE)))/
913		#(ENT ENT) = (TOWD = TOWN) 2. EN2 TROOM AMON(0, T) / (ENT OB(TB))
015		$\frac{1}{2} \left(\frac{1}{2} \sqrt{2} \sqrt{2} \sqrt{2} \sqrt{2} \sqrt{2} \sqrt{2} \sqrt{2} $
915		$PARTO = ((ANORM2(0+1))^{POWB+SOMB} \cdot ENI^{POWR} \cdot PARTD)^{-1}$
910		$# (POWF "ANORM2 (J+1)) \\ # (CUMP + EN1 + DOWL + NOPM2 (J+1)) + DDODA + DDODA / EN1$
917		#+SUMB*EN1*POWL*ANORM2(J+1)) *PRODA*PRODA/EN1
918		#-((POWB-POWF)*2.*PRODA*AMOM(J+1)*SUMB)/CL(IE))/(PRODC*PRODC)
919		$PARDIF(1, J) = PARDIF(1, J) + FORCE(1+1, 1E, 3) * EPS10^{\circ}$
920		#.5*(PART1-PART2)+RN0*EPSI0/(2.*RM0)*((AMOM(1+1)*RM0/CL(1E))
921		<pre>#*(PART3-PART4-PART5+PART6)+FORCE(I+1,IE,1)*(PART1+PART2))</pre>
922	5	CONTINUE
923	4	CONTINUE
924	1	CONTINUE
925	С	
926	С	HAVING OBTAINED THE PARTIAL DIRIVATIVES OF EQUILIBRIUM
927	С	EQUATIONS, MUST INVERT TO OBTAIN PERTURBATIONS ON REDUNDANTS
928	C	FIRST SET UP AUGMENTED MATRIX FOR INVERTION.
929	Ċ	
030	C	DO 6 I-1 NRED
930		EU = 1, RED EU = 2, RED = 2, COMPAT(T)
931		P = DEX(1, NEE) + 1 = -COMPART(1)
932		DU / U - I, NRED
933		FLEX(I,J) = PARDIF(I,J)
934	7	$\Gamma LLLA(J', I) = \Gamma AKDIF(I, J)$
935	1	CONTINUE
936	6	CONTINUE
937		RETURN
938		DEBUG SUBCHK
939		END
940	С	
941	С	

943	С	* * * * * * * * * * * * * * * * * * * *
011	Ċ	
944	C	
945	C	
940	C	
947	C	INVERSION ROUTINE : C.F. GERALD
948	C	APPLIED NUMERICAL ANALYSIS , SECOND EDITION P.132
949	С	
950	С	***************************************
951		SUBROUTINE PIVOT
952		INCLUDE DAGNVS.COMPROC
953		DOUBLE PRECISION RATIO, VALUE
954	С	
955		NEQU=NINV
956		IF (MODES.EQ.1) THEN
957		NEQU=NRED
958		END IF
959	С	FACTOR ALL ELEMENTS OF MATRIX BY LARGE NUMBER FOR NUMERICAL
960	Ċ	STABILITY
961	•	$DO_{0} = 1$ NEOU
962		DO 61 J=1. NEOU+1
963		FLFX(T, T) = FLFX(T, T) * 1
961		$\frac{1}{1} \frac{1}{1} \frac{1}$
965		FND IF
905	61	
967	60	CONTINUE
907	00	
900		IF (NEQUINEII)ITEN
909		
970		NMI=NEQU-I
9/1		DO 35 I=I, NMI
972		
9/3		
974		DO 10 J=IP1,NEQU
975		IF(DABS(FLEX(IPVT,I)).LT.DABS(FLEX(J,I))) IPVT=J
976	10	CONTINUE
977		IF(DABS(FLEX(IPVT,I)).LT.1.D-64)GO TO 99
978		IF(IPVT.EQ.I) GO TO 25
979		DO 20 JCOL=I,NP
980		FACT=FLEX(I,JCOL)
981		FLEX(I,JCOL)=FLEX(IPVT,JCOL)
982		FLEX(IPVT,JCOL)=FACT
983	20	CONTINUE
984	25	DO 32 JROW=IP1,NEQU
985	•	IF(DABS(FLEX(JROW,I)).LE.1.D-64) GO TO 32
986		RATIO=FLEX(JROW, I)/FLEX(I,I)
987		DO 30 KCOL=IP1.NP
988		FLEX(JROW, KCOL) = FLEX(JROW, KCOL) - RATTO*FLEX(I, KCOL)
989	30	CONTINUE
990	32	CONTINUE
991	35	CONTINUE
992	55	E(DARS(FLEX(NEOU NEOU))) LT 1 D-64)GO TO 99
002		$\frac{11}{10} \frac{10}{10} 10$
001		DO 50 KCOL-NDI ND
994 00F		DO JO ROUL-MELINE
222		FLEA(NEQU, ACOL) - FLEA(NEQU, ACOL) / FLEA(NEQU, NEQU)
996		DU 45 J=Z,NEQU
997		N ART=N KT-O
998		T=NART+T
999		VALUE=FLEX(NVBL,KCOL)
1000		DO 40 K=L, NEQU
1001		VALUE=VALUE-FLEX(NVBL,K)*FLEX(K,KCOL)

1003 FLEX(NVBL,KCC)=VALUE/FLEX(NVBL,NVBL) 1004 45 CONTINUE 1005 50 CONTINUE 1006 ELSE 1007 FLEX(1,2)=FLEX(1,2)/FLEX(1,1) 1008 END IF 1009 RETURN 1009 RETURN 1010 99 WRITE(TPRINT,101) 1011 101 FORMAT(1H0,10X,'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT') 1013 DEBUG SUBCHK 1014 END 1015 C 1016 C 1017 C 1018 C 1017 C 1019 C 1020 C D E L T A 1021 C 1022 C IF STRUCTURE IS STATICALLY INDETERMINATE 1023 C AND MODE ANALYSIS IS REQUIRED, REDUNDANT 1024 C FORCES ARE DETERMINED BY NEWTON-RAPHSON 1025 C PROCEDURE.RESIDUAL REDUNDANT FORCES ARE 1026 C CHECKED HERE TO DETERMINE IF CONVERGENCE 1027 C HAS OCCURED(TELGS].1F NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1030 C **********************************	1002	40	CONTINUE
100445CONTINUE100550CONTINUE1006ELSEFLEX(1, 2)=FLEX(1, 2)/FLEX(1, 1)1007FLEX[1, 2]=FLEX(1, 2)/FLEX(1, 1)1008END IF1009RETURN101099WRITE(IPENT, 101)1011101FORMAT(1H0, 10X, 'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')1013DEBUG SUBCHK1014END1015C1016C1017C1018C1020C1019C1021C1022C1032C1041END1052C1052C1053C1054C1054C1055C1056C1057C1056C1056C1057C1058C1050C1050C1050C1051SUBROUTINE DELTA1031SUBROUTINE DELTA1031SUBROUTINE DELTA1032INCLUDE DARONS.COMPROC1033C1034IF(LAG1=0)1035IFLAG2=01036IFLAG2=01037IFLAG2=11040IFLAG2=11041DO 9 1=1,NRED1042IF(DABS(X(1)),GT.1.D-6)THEN1043IF(DABS(X(1))/XDUM(1)-1.).GT.1.D-2)IFLAG2=01044END IF1045Y1046 <td>1003</td> <td></td> <td>FLEX(NVBL,KCOL)=VALUE/FLEX(NVBL,NVBL)</td>	1003		FLEX(NVBL,KCOL)=VALUE/FLEX(NVBL,NVBL)
100550CONTINUE1006ELSE1007FLEX(1, 2)=FLEX(1, 2)/FLEX(1, 1)1008EDU 171009RETURN10109910111011011PORMAT(1HO, 1OX, 'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')1012STOP1013DEBUG SUBCHK1014END1015C1016C1017C1018C1020C1021C1022C1023C1024C1025C1026C1027C1028C1029C1024C1025C1026C1027C1028C1029C1029C1029C1021C1022C1023C1024C1025C1026C1027C1028C1029C1030C1031SUBROUTINE DELTA1032INCLUBE DAGNYS-COMPROC1033C1034SUBROUTINE DELTA1035IFFLAGI=01036IFFLAGI=01037IFLAGI=01038IFFLAGI=01039IFLAGI=11040IFLAG2=01039IFLAGI=11041DO 9 I=1,NRED1042I	1004	45	CONTINUE
1006ELSE1007FLEX(1,2)=FLEX(1,2)/FLEX(1,1)1008ERUUN1009RETURN10109910111011012STOP1013DEBUG SUBCHK1014EXD1015C1016C1017C1020C1021C1022C1032C1044EXD1055C1056C1077C1086C1097C1098C1099C1019C1020C1021C1022C1022C1023C1024C1025C1025C1026C1027C1028C1029C1030C1031SUBROUTINE DELTA1032INCLUDE DAGNYS.COMPROC1033C1034DOUBLE PRECISION XDUM(2*NN)1035IFLAGI=01036IFLAGI=01037IFLAGI=01038IF(TREAC.GT.1)THEN1039IFLAGI=11041DO 9 I=1,NED1042IF (DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)).GT.1.D-6)THEN1044END IF104591044END IF104591046END IF1047DO 9 I=1,	1005	50	CONTINUE
1007FLEX(1,2)=FLEX(1,2)/FLEX(1,1)1008END IF1009RETURN101090101110110111011011FORMAT(1H0,10X,'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')1012STOP1013DEBUG SUBCHK1014END1015C1016C1017C1018C1020C1021C1022C1023C AND MODE ANALYSIS IS REQUIRED, REDUNDANT1024C1025C1026C1027C1028C1024C1026C1027C1028C1029C1029C1021C1022C1023C1024C1025C1026C1027C1028C1029C1029C1030C1031SUBROUTINE DELTA1032INCLUBE DAGNS.COMPROC1033C1034DOUBLE PRECISION XDUM(2*N)1035IFLAG=0117IFLAG=0118IF(ATREAC.GT.1)THEN1035IFLAG2=11036IFLAG2=11037IFLAG2=11038IF(ATREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041DO 9 I=1,NRED <td>1006</td> <td></td> <td>ELSE</td>	1006		ELSE
1008END IF RETURN101099WRITE(IPRINT,101)1011101FORMAT(1H0,10X,'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')1011DEBUG SUBCHK1014END1015C1016C1017C1018C1020C1021C1022C1031DEBUG SUBCHK1041END1052C1053C1054C1052C1052C1053C1054C1055C1056C1057C1058C1059C1050C1051C1051C1052C1053C1054C1055C1056C1057C1058C1059C1050C1051C1052INCLUDE DAGNYS.COMPROC1054IFLAG2=01054IFLAG2=01055IFLAG2=11056IFLAG2=11057IFLAG2=11058IF(1)/XDUM(1)-1.1.GT.1.D-2)IFLAG2=01054IF(1)/XDUM(1)-1.1.GT.1.D-2)IFLAG2=01054IF(1)/XDUM(1)-1.1.GT.1.D-2)IFLAG2=01054IF(1)/XDUM(1)-1.NED1055C1051C1052IFLAG2.CG.1.NND.IFLAG2.EG.1)IFLAG=11053X(1)	1007		FLEX(1,2) = FLEX(1,2) / FLEX(1,1)
1000RETURN10109WRITE (IPRINT, 101)1011101FORMAT (1H0, 10X, 'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')1011DEBUG SUBCHK1013DEBUG SUBCHK1014END1015C1016C1017C1018C1020C1019C1021C1022C1033C1044END1054C1055C1056C1057C1058C1059C1050C1051C1052C1053C1054C1055C1056C1057C1058C1059C1050C1051C1051FEAGI-01051IFLAGI-01051IFLAGI-01051IFLAGI-01051IFLAGI-01051IFLAGI-01051IFLAGI-11051IFLAGI-11051IFLAGI-11052IFLAGI-11053IFLAGI-11054IFLAGI-11055IFLAGI-11056IFLAGI-11051IFLAGI-11051IFLAGI-11051IFLAGI-11051IFLAGI-11051IFLAGI-11051IFLAGI-11051IFLAGI-1	1008		END IF
1010 99 WRTTE (IPRINT, 101) 1011 101 FORMAT(1H0, 10X, 'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT') 1013 DEBUG SUBCHK 1014 END 1015 C 1016 C 1017 C 1018 C 1019 C 1020 C 1021 C 1022 C 103 MODE ANALYSIS IS REQUIRED, REDUNDANT 1042 C 1052 C 1053 C AND MODE ANALYSIS IS REQUIRED, REDUNDANT 1024 C FORCES ARE DETERMINED BY NEWTON-RAPHSON 1025 C PROCEDURE.RESIDUAL REDUNDANT FORCES ARE 1026 C CHECKED HERE TO DETERMINE IF CONVERCENCE 1027 C HAS OCCURED(FLAGE-1).IF NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1029 C SUBROUTINE DELTA 1031 SUBROUTINE DELTA 1032 INCLUDE DAGNYS.COMPROC 1033 C AND THAT THE COMPATIBILITY EQUATIONS ARE NEAR ZERO 1034 DOUBLE	1009		RETURN
1011 101 FORMAT(1H0,10X, 'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT') 1012 STOP 1013 DEBUG SUBCHK 1014 END 1015 C 1016 C 1017 C 1018 C 1017 C 1020 C D E L T A 1021 C 1022 C IF STRUCTURE IS STATICALLY INDETERMINATE 1023 C AND MODE ANALYSIS IS REQUIRED, REDUNDANT 1024 C FORCES ARE DETERMINED BY NEWTON-RAPHSON 1025 C PROCEDURE.RESIDUAL REDUNDANT FORCES ARE 1027 C HAS OCCURED(IFLAG=1).IF NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1029 C 1030 C **********************************	1010	99	WRITE(IPRINT, 101)
1012 STOP 1013 DEBUG SUBCHK 1014 END 1015 C 1017 C 1017 C 1020 C D E L T A 1021 C J F STRUCTURE IS STATICALLY INDETERMINATE 1022 C IF STRUCTURE IS STATICALLY INDETERMINATE 1023 C AND MODE ANALYSIS IS REQUIRED, REDUNDANT 1024 C FORCES ARE DETERMINED BY NEWTON-RAPHSON 1025 C PROCEDURE.RESIDUAL REDUNDANT FORCES ARE 1026 C CHECKED HERE TO DETERMINE IF CONVERGENCE 1027 C HAS OCCURED(IFLAG=1).IF NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1029 C 1031 SUBROUTINE DELTA 1032 C ***********************************	1011	101	FORMAT(1H0, 10X, 'SOLUTION NOT FEASIBLE.NEAR ZERO ON PIVOT')
1013 DEBUG SUBCHK 1014 END 1015 C 1016 C 1017 C 1018 C 1020 C D E L T A 1021 C 1022 C IF STRUCTURE IS STATICALLY INDETERMINATE 1023 C AND MODE ANALYSIS IS REQUIRED, REDUNDANT 1024 C FORCES ARE DETERMINED BY NEWTON-RAPHSON 1025 C PROCEDURE.RESIDUAL REDUNDANT FORCES ARE 1026 C CHECKED HERE TO DETERMINE IF CONVERGENCE 1027 C HAS OCCURED(IPLAG=1).IF NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1029 C 1030 C **********************************	1012		STOP
1014 END 1015 C 1017 C 1017 C 1018 C 1019 C 1020 C D E L T A 1021 C 1022 C IF STRUCTURE IS STATICALLY INDETERMINATE 1023 C AND MODE ANALYSIS IS REQUIRED, REDUNDANT 1024 C FORCES ARE DETERMINED BY NEWTON-RAPHSON 1025 C FROCEDURE. RESIDUAL REDUNDANT FORCES ARE 1026 C CHECKED HERE TO DETERMINE IF CONVERGENCE 1027 C HAS OCCURED (IFLAG=1). IF NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1029 C 1030 C **********************************	1013		DEBUG SUBCHK
1015C1016C1017C1018C1020C1021C1022C1023C1024C1025C1026C1027C1028C1029C1029C1024C1025C1025C1026C1027C1028C1029C1029C1030C1031SUBROUTINE DELTA1032INCLUDE DACNYS.COMPROC1033C1034DOUBLE PRECISION XDUM(2*NN)1035IFLAG1=01036IFLAG2=01037IFLAG2=01038IF(ITREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041DO 9 I=1,NED1042IF(DABS(X(I)).GT.1.D=6)THEN1044END IF104591044END IF104591044END IF104591044DO 9 I=1,NED1045IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG2=01046END IF1047DO 99 I=1,NED1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051C1052IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11053X(I)=X(I)+PLEX(I,NED+1) <t< td=""><td>1014</td><td></td><td>END</td></t<>	1014		END
1016C1017C1018C1019C1020C1021C1022C1023C1024C1024C1025C1026C1027C1028C1029C1029C1021C1024C1025C1025C1026C1027C1028C1028C1029C1029C1030C1031SUBROUTINE DELTA1032INCLUDE DAGNUS.COMPROC1033C1034DOUBLE PRECISION XDUM(2*NN)1035IFLAGI=01036IFLAG2=01037IFLAG2=01038IF(ITREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041DO 9 I=1,NRED1042IF (DABS(X(I)).GT.1.D-6)THEN1043IF (DABS(X(I)).GT.1.D-6)THEN1044END IF1045P1046END IF1047D0 99 I=1,NRED1048XDUM(1)=X(1)1049991050IF (IFLAGI.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051C1051C1052D 10 I=1,NRED1053X(1)=FEEXURBATION TO CURRENT VALUE OF REDUNDANT1054D1055C1056IF (IFLA	1015	С	
1017 C 1018 C 1019 C 1020 C D E L T A 1021 C 1022 C IF STRUCTURE IS STATICALLY INDETERMINATE 1023 C AND MODE ANALYSIS IS REQUIRED, REDUNDANT 1024 C FORCES ARE DETERMINED BY NEWTON-RAPHSON 1025 C PROCEDURE.RESIDUAL REDUNDANT FORCES ARE 1026 C CHECKED HERE TO DETERMINE IF CONVERCENCE 1027 C HAS OCCURED(IFLAG=1).IF NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1029 C 1030 C **********************************	1016	C	
1018C*********************************	1017	C	
1019C1020CD E L T A1021C1022C1023C1024C1025C1026C1027C1028C1029C1029C1020C1021C1022C1025C1026C1027C1028C1029C1029C1030C1031SUBROUTINE DELTA1032INCLUDE DAGNVS.COMPROC1033C1034DOUBLE PRECISION XDUM(2*NN)1035IFLAG1=01036IFLAG2=01037IFLAG2=01038IF(TTREAC.GT.1)THEN1040IFLAG2=11041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)).GT.1.D-6)THEN1044END IF104591046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)1049901051C1052ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1053X(I)=X(I)+FLEX(I,NRED+1)1054IO1055C1056IF (FIRLAG.EQ.0) GO TO 151057C1058C1059C1050C1051CONTINUE1052CONTINUE1053X(I)=X(I)+FLEX(I,NRE	1018	C	*****
Display="body>Display=transmine="body>Display	1019	C	
1021CDELTA1022CIF STRUCTURE IS STATICALLY INDETERMINATE1023CAND MODE ANALYSIS IS REQUIRED, REDUNDANT1024CFORCES ARE DETERMINED BY NEWTON-RAPHSON1025CPROCEDURE.RESIDUAL REDUNDANT FORCES ARE1026CCHECKED HERE TO DETERMINE IF CONVERGENCE1027CHAS OCCURED(IFLAG=1).IF NOT THEN1028CMORE EQUILIBRIUM ITERATIONS REQUIRED.1029C*********************************	1020	C	ר ד ד ת א
1012CIF STRUCTURE IS STATICALLY INDETERMINATE1023CAND MODE ANALYSIS IS REQUIRED, REDUNDANT1024CFORCES ARE DETERMINED BY NEWTON-RAPHSON1025CPROCEDURE.RESIDUAL REDUNDANT FORCES ARE1026CCHECKED HERE TO DETERMINE IF CONVERGENCE1027CHAS OCCURED(IFLAG=1).IF NOT THEN1028CMORE EQUILIBRIUM ITERATIONS REQUIRED.1029C*********************************	1020	C	
1023CAND MODE ANALYSIS IS REQUIRED, REDUNDANT1024CFORCES ARE DETERMINED BY NEWTON-RAPHSON1025CPROCEDURE.RESIDUAL REDUNDANT FORCES ARE1026CCHECKED HERE TO DETERMINE IF CONVERGENCE1027CHAS OCCURED(IFLAG=1).IF NOT THEN1028CMORE EQUILIBRIUM ITERATIONS REQUIRED.1029C1030C*********************************	1021	C	TE SUBICUIDE IS SUMMICALLY INDEVERMINAUE
1024CFORCES ARE DETERMINED BY NEWTON-RAPHSON1025CPROCEDURE.RESIDUAL REDUNDANT FORCES ARE1026CCHECKED HERE TO DETERMINE IF CONVERGENCE1027CHAS OCCURED(IFLAG=1).IF NOT THEN1028CMORE EQUILIBRIUM ITERATIONS REQUIRED.1029C1030C***********************************	1022	C	AND MODE ANALYSIS IS DECUIDED DEDUNDANT
1025CFORCEDURE RESIDUAL REDUNDANT FORCES ARE1026CCHECKED HERE TO DETERMINE IF CONVERGENCE1027CHAS OCCURED(IFLAG=1).IF NOT THEN1028CMORE EQUILIBRIUM ITERATIONS REQUIRED.1029C1030C*********************************	1023	C	FORCES ADE DETERMINED DV NEWTON_DADUCON
1025CFROEBORALTESTOOD EREPTOR DETERMINE IF CONVERGENCE1026CHAS OCCURED(IFLAG=1).IF NOT THEN1028CMORE EQUILIBRIUM ITERATIONS REQUIRED.1029C1030C*********************************	1025	C	DOCEDUDE DECEDUAL DEDUNDANT FORCES ADE
1020CCHECKED HERK IG DERMINE IF CONFRENCE1027CHAS OCCURED(IFLAG=1).IF NOT THEN1028CMORE EQUILIBRIUM ITERATIONS REQUIRED.1029C1030C1031SUBROUTINE DELTA1032INCLUDE DAGNVS.COMPROC1033C1034DOUBLE PRECISION XDUM(2*NN)1035IFLAG1=01036IFLAG2=01037IFLAG2=11040IFLAG2=11041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)).GT.1.D-6)THEN1044END IF104591044END IF1045OO 9 I=1,NRED1046IF(IFLAGL.EQ.1.AND.IFLAG2.EQ.1)IFLAG2=01047DO 99 I=1,NRED1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAGL.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051C1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054C1055C1056IF PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054C1055C1056IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1057C1058C1059C1059C1059C1059C1050C1050C1050C	1025	C	CHECKED HEDE TO DETERMINE LE CONVEDCENCE
1027 C HAS OCCURED(IFLAG=1).IF NOT THEN 1028 C MORE EQUILIBRIUM ITERATIONS REQUIRED. 1029 C 1030 C **********************************	1020		UNC OCCUPED (IFLIG 1) IF NOT THEN
1025CMORE EQUILIBRIUM TIERATIONS REQUIRED.1029C1030C1031SUBROUTINE DELTA1032INCLUDE DAGNVS.COMPROC1033C1034DOUBLE PRECISION XDUM(2*NN)1035IFLAGI=01036IFLAG2=01037IFLAG=11040IFLAG2=11041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)).GT.1.D-6)THEN1044END IF104591046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)1049991050IF(IFLAGI.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051C1052ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1053X(I)=X(I)+FLEX(I,NRED+1)1054IO1055C1056IF(IFLAG.EQ.0) GO TO 151057C1058C1059C1059C1059C1059C1059C1050C1051C1052IF(IFLAG.EQ.O) GO TO 151054C1055C1056C1057C1058C1059C1059C1059C1059C1050C1051C1052C1054C1055C1054 <td>1027</td> <td></td> <td>HAS OCCURED (IFLAG=I). IF NOT THEN</td>	1027		HAS OCCURED (IFLAG=I). IF NOT THEN
1029 C 1030 C ************************************	1028	C	MORE EQUILIBRIUM ITERATIONS REQUIRED.
1030 C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	1029	C	****
1031SUBROUTINE DELTA1032INCLUDE DAGNVS.COMPROC1033CAND THAT THE COMPATIBILITY EQUATIONS ARE NEAR ZERO1034DOUBLE PRECISION XDUM(2*NN)1035IFLAG1=01036IFLAG2=01037IFLAG=01038IF(ITREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=01044END IF104591046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1059C1059C1050CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO	1030	C	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
1032INCLUDE DAGNVS.COMPROC1033CAND THAT THE COMPATIBILITY EQUATIONS ARE NEAR ZERO1034DOUBLE PRECISION XDUM(2*NN)1035IFLAG1=01036IFLAG2=01037IFLAG=01038IF(ITREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=01044END IF104591046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1059C1059C1059C1059C1059C1050IF (IFLAG.EQ.O) GO TO 15	1031		SUBROUTINE DELTA
1033 C AND THAT THE COMPATIBILITY EQUATIONS ARE NEAR ZERO 1034 DOUBLE PRECISION XDUM(2*NN) 1035 IFLAG1=0 1036 IFLAG2=0 1037 IFLAG=0 1038 IF(ITREAC.GT.1)THEN 1039 IFLAG1=1 1040 IFLAG2=1 1041 DO 9 I=1,NRED 1042 IF(DABS(X(I)).GT.1.D-6)THEN 1043 IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0 1044 END IF 1045 9 CONTINUE 1046 END IF 1047 DO 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99 CONTINUE 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 C ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT 1052 DO 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10 CONTINUE 1055 C IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF(IFLAG.EQ.0) GO TO 15 1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO	1032		INCLUDE DAGNVS.COMPROC
1034DOUBLE PRECISION XDUM(2*NN)1035IFLAG1=01036IFLAG2=01037IFLAG=01038IF(ITREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=01044END IF104591044END IF104591044END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)1049991050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1056IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF (IFLAG.EQ.O) GO TO 151057C1058C1059C1059C1060C2ERO	1033	С	AND THAT THE COMPATIBILITY EQUATIONS ARE NEAR ZERO
1035IFLAG1=01036IFLAG2=01037IFLAG=01038IF(ITREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=01044END IF104591046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)1049991050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C156IF (IFLAG.EQ.O) GO TO 151057C1058C1059C1059C1050C1051C1052OG TO 151054C1055C1057C1058C1059C1059C1059C1050ZERO	1034		DOUBLE PRECISION XDUM(2*NN)
1036 IFLAG2=0 1037 IFLAG=0 1038 IF(ITREAC.GT.1)THEN 1039 IFLAG1=1 1040 IFLAG2=1 1041 DO 9 I=1,NRED 1042 IF(DABS(X(I)).GT.1.D-6)THEN 1043 IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0 1044 END IF 1045 9 CONTINUE 1045 9 CONTINUE 1046 END IF 1047 DO 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99 CONTINUE 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 C ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT 1052 DO 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10 CONTINUE 1055 C IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF(IFLAG.EQ.0) GO TO 15 1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO	1035		IFLAG1=0
1037 IFLAG=0 1038 IF(ITREAC.GT.1)THEN 1039 IFLAG1=1 1040 IFLAG2=1 1041 DO 9 I=1,NRED 1042 IF(DABS(X(I)).GT.1.D-6)THEN 1043 IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0 1044 END IF 1045 9 1046 END IF 1047 DO 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 C 1052 DO 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10 1055 C 1054 IF PERTURBATION TO CURRENT VALUE OF REDUNDANT 1055 C 1056 IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF(IFLAG.EQ.0) GO TO 15 1057 C 1058 C 1059 C 1059 C 1059 C 1059 C 1050 CHECK IF ANY REACTANT IS NEAR ZE	1036		IFLAG2=0
1038IF(ITREAC.GT.1)THEN1039IFLAG1=11040IFLAG2=11041D0 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=01044END IF104591044END IF104591046END IF1047D0 99 I=1,NRED1048XDUM(I)=X(I)1049991050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052D0 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C155C1656IF(IFLAG.EQ.0) GO TO 151057C1058C1059C1059C1050CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO	1037		IFLAG=0
1039 IFLAG1=1 1040 IFLAG2=1 1041 DO 9 I=1,NRED 1042 IF(DABS(X(I)).GT.1.D-6)THEN 1043 IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0 1044 END IF 1045 9 1046 END IF 1047 DO 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 C 1052 DO 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10 1055 C 1056 IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF (IFLAG.EQ.O) GO TO 15 1057 C 1058 C 1059 C 1050 CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO	1038		IF(ITREAC.GT.1)THEN
1040IFLAG2=1 1041 D0 9 I=1,NRED 1042 IF(DABS(X(I)).GT.1.D-6)THEN 1043 IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0 1044 END IF 1045 9 1046 END IF 1047 D0 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99CONTINUE 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT 1052 D0 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10CONTINUE 1055 CIF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF (IFLAG.EQ.0) GO TO 15 1057 CCONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 CC HECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 CZERO	1039		IFLAG1=1
1041DO 9 I=1,NRED1042IF(DABS(X(I)).GT.1.D-6)THEN1043IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=01044END IF104591046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)1049991049991050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051C1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1059C1050C2050C1054C1055C1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1059C1050C1050C1050C1050C1050C1051105710571058105910591050105010501050105010501050105010511051105210531054105510571057105810591050105010501050105010501050 <t< td=""><td>1040</td><td></td><td>IFLAG2=1</td></t<>	1040		IFLAG2=1
1042IF(DABS(X(I)).GT.1.D-6)THEN 1043 IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0 1044 END IF 1045 9 1046 END IF 1047 DO 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99CONTINUE 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT 1052 DO 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10CONTINUE 1055 CIF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF(IFLAG.EQ.0) GO TO 15 1057 CCONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 CCHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 CZERO	1041		DO 9 I=1, NRED
1043 IF (DABS $(X(I)/XDUM(I)-1.).GT.1.D-2)$ IFLAG2=0 1044 END IF 1045 9 CONTINUE 1046 END IF 1047 DO 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99 CONTINUE 1050 IF (IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 C ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT 1052 DO 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10 CONTINUE 1055 C IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF (IFLAG.EQ.0) GO TO 15 1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO	1042		IF(DABS(X(I)).GT.l.D-6)THEN
1044END IF10459CONTINUE1046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055CIF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1060CZERO	1043		IF(DABS(X(I)/XDUM(I)-1.).GT.1.D-2)IFLAG2=0
<pre>1045 9 CONTINUE 1046 END IF 1047 DO 99 I=1,NRED 1048 XDUM(I)=X(I) 1049 99 CONTINUE 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 C ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT 1052 DO 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10 CONTINUE 1055 C IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF(IFLAG.EQ.0) GO TO 15 1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO</pre>	1044		END IF
1046END IF1047DO 99 I=1,NRED1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAGI.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)105410CONTINUE1055CIF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1060CZERO	1045	9	CONTINUE
1047DO 99 I=1,NRED1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)105410CONTINUE1055CIF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF (IFLAG.EQ.0) GO TO 151057CCONSTRUCT FINAL BENDING MOMENT DIAGRAM1058C1059CCHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO1060CZERO	1046		END IF
1048XDUM(I)=X(I)104999CONTINUE1050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1656IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF(IFLAG.EQ.0) GO TO 151057C1058C1059C1060C2ERO	1047		DO 99 I=1.NRED
<pre>1049 99 CONTINUE 1050 IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=1 1051 C ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT 1052 D0 10 I=1,NRED 1053 X(I)=X(I)+FLEX(I,NRED+1) 1054 10 CONTINUE 1055 C IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF(IFLAG.EQ.0) GO TO 15 1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO</pre>	1048		XDUM(I) = X(I)
1050IF(IFLAG1.EQ.1.AND.IFLAG2.EQ.1)IFLAG=11051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1656IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1060C2ERO	1049	99	CONTINUE
1051CADD PERTURBATION TO CURRENT VALUE OF REDUNDANT1052DO 10 I=1,NRED1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1056IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1060CZERO	1050		IF(IFLAG1.EO.1.AND.IFLAG2.EO.1)IFLAG=1
DO 10 I=1,NRED DO 10 I=1,NRED X(I)=X(I)+FLEX(I,NRED+1) CONTINUE CONTINUE DO 10 I=1,NRED X(I)=X(I)+FLEX(I,NRED+1) CONTINUE DO 10 I=1,NRED I = 1,NRED DO 10 I=1,NRED I = 1,NRED I = 1,NRED	1051	С	ADD PERTURBATION TO CURRENT VALUE OF REDUNDANT
1053X(I)=X(I)+FLEX(I,NRED+1)1054101055C1055C1056IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF(IFLAG.EQ.O) GO TO 151057C1058C1059C1050C1060CZERO	1052	-	DO 10 I=1.NRED
105410CONTINUE1055CIF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE1056IF(IFLAG.EQ.O) GO TO 151057CCONSTRUCT FINAL BENDING MOMENT DIAGRAM1058C1059CCHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO1060CZERO	1053		X(T) = X(T) + FLEX(T, NRED+1)
<pre>1055 C IF PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE 1056 IF(IFLAG.EQ.O) GO TO 15 1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO</pre>	1054	10	CONTINUE
1056IF (IFLAG.EQ.0) GO TO 151057C1058C1059C1060CZERO	1055	C	TE PERTURBATION IS NOT SMALL ENOUGH THEN ITERATE
1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO	1056	0	IF (IFLAG. EO. 0) GO TO 15
1057 C CONSTRUCT FINAL BENDING MOMENT DIAGRAM 1058 C 1059 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO	1057	C	CONSTRUCT FINAL RENDING MOMENT DIAGRAM
1050 C CHECK IF ANY REACTANT IS NEAR ZERO . IF SO , SET EQUAL TO 1060 C ZERO	1059	C	CONSTRUCT LINNE DENDING NOMENI DIVARAM
1059 C CHECK IF ANI REACIANT IS NEAR ZERO . IF SO , SET EQUAL TO	1050	C	
	1060	C	ZERO

1061		IF(NRED.GT.1)THEN
1062		DUM1=1.E16
1063		DUM2=1.D-16
1064	С	FIND MINIMUM REACTANT
1065		DO 1 I=1,NRED
1066		<pre>IF(DABS(X(I)).LT.DUM1)DUM1=DABS(X(I))</pre>
1067		IF(DABS(X(I))-1.D-6.LE.DUM1)IMIN=I
1068	1	CONTINUE
1069	С	FIND MAXIMUM REACTANT
1070		DO 2 I=1, NRED
1071		IF(DABS(X(I)).GT.DUM2)DUM2=DABS(X(I))
1072		IF(DABS(X(I))+1.D-6.GT.DUM2)IMAX=I
1073	2	CONTINUE
1074	C	DETERMINE RATIO BETWEEN MAX AND MIN VALUES.
1075	Ċ	IF MIN/MAX LESS THAN 1.D-9 THEN SET X(MIN) EQUAL TO ZERO.
1076	•	RATIO=X(IMIN)/X(IMAX)
1077		IF(DABS(RATIO), LE, 1, D-9)X(IMIN)=0.
1078		END IF
1079		DO 12 IE=1.NE
1080		DO [13] IJ=1.3
1081		FORCET(IE, IJ) = 0.
1082		FORCET(IE, IJ) = FORCE(I, IE, IJ)
1083		DO 14 IK=1.NRED
1084		FORCET(IE, IJ)=FORCET(IE, IJ)+X(IK)*FORCE(IK+1, IE, IJ)
1085	14	CONTINUE
1086	13	CONTINUE
1087	12	CONTINUE
1088	15	CONTINUE
1089	15	PETUDN
1000		DEBUG SUBCHK
1090		FND
1091	C	
1092	C	
1095		* * * * * * * * * * * * * * * * * * * *
1094	C	
1095	C	
1096	C	VELOC
1097	C	UNITED ON THE TOTAL DENDING NOVENT
1098	C	HAVING OBTAINED THE TOTAL BENDING MOMENT
1099	C	AND AXIAL FORCE DIAGRAMS , VELOCITIES ARE
1100	С	DETERMINED USING THE PRINCIPLE OF VIRTUAL
1101	C	VELOCITIES
1102	C	
1103	C	***********************
1104		SUBROUTINE VELOC
1105		INCLUDE DAGNVS.COMPROC
1106		DOUBLE PRECISION UNITMA(NDF,NN)
1107		IJ=2
1108		DO 1 $I=1$, NDF
1109		VEL(I,2)=0.
1110	C	IF THE D.O.F. IS A BOUNDARY CONDITION THEN VELOCITY IS ZERO
1111		IF(IBC(I).EQ.1) GO TO 1
1112	С	DETERMINE WHETHER D.O.F. IS A ROTATION
1113		IK=0
1114		IF(IJ*NF.EQ.I)IK=1
1115		IF(IK.EQ.1)IJ=IJ+1
1116		DO 2 IE=1,NE
1117	С	FOR AN APPLIED UNIT MOMENT , THERE IS A
1118	С	DISCONTINUITY OF MOMENT AT POINT OF APPLICATION
1119		IF(I.EQ.3) GO TO 3

1120		IF(I.EQ.NDF) GO TO 3
1121		UNITMA(I, NBEAM(IE, 1)) = 0.
1122		IF(IK.EQ.1.AND.(IJ-1).EQ.NBEAM(IE,1))THEN
1123		UNITMA(I, NBEAM(IE, 1))=UNITM(I, NBEAM(IE, 1))-1.
1124		UNITM(I, NBEAM(IE, 1))=UNITMA(I, NBEAM(IE, 1))
1125		END IF
1126	3	CONTINUE
1127	•	SUMA=UNTTM(T, NBEAM(TE, 2)) - UNTTM(T, NBEAM(TE, 1))
1128		SUMB = (FORCET(IE, 2) - FORCET(IE, 1))/RMO
1129		SUMC=FORCET(IE.1)/ RMO +FORCET(IE.3)/ RNO
1130		SUMD= $(FORCET(IE, 3)/RNO-FORCET(IE, 1)/RMO)$
1131		SIG1=1.
1132		SIG2=1.
1133		SIG3=1.
1134		SIG4=1.
1135		IF((SUMB+SUMC), UT, 0) $SIG]=-1.$
1136		IF((-SUMB+SUMD), LT, 0) $SIG2=-1$.
1137		$IF(SUMC_LT, 0) SIG3=-1.$
1138		IF(SUMD, LT, 0) $SIG4=-1$.
1139		PRODI=SIGI*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)*(SUMB+SUMC)
1140		PROD 2=SIG1*((DABS(SUMB+SUMC))**EN)*(SUMB+SUMC)
1141		PROD 3 = SIG2*((DABS(-SUMB+SUMD))**EN)*(-SUMB+SUMD)
1142		#*(-SUMB+SUMD)
1143		PROD4=SIG2*((DABS(-SUMB+SUMD))**EN)*(-SUMB+SUMD)
1145		PROD5=SIG3*(DABS(SUMC)**EN)*SUMC*SUMC
1145		PROD6=SIG3* (DABS (SUMC) **EN) *SUMC
11/6		PROD 7=SIG4*(DABS(SUMD)**EN)*SUMD*SUMD
1147		PROP = SIG4*(DARS(SUMD) **FN)*SUMD
11/10		CONST = SIMA / CL(IF)
11/0		CONSTI = SOMA/CH(IE) $CONST2 = CL(IE) * CL(IE) / (EN2*SUMB*SUMB)$
1150		CONSTZ = CD(TE) + CD(TE) + (DNZ BOND BOND)
1151		CONSTJ = (SOMC) CH(TH) / (HAT BOAD BOAD) / (SUMC)
1152		CONST4 = CONST3 (SOMD) / (SOME)
1152		CONSTS = CD(TE)/(ENT (SOMB))
1155		$CONSTO = RNO^{2} EPSIO/(2. RMO)$
1155	C	VELOCITY AT D O F I
1156	C	$v_{ELOCITI AT D.O.F. T}$ $v_{ELOCITI AT D.O.F. T$ $v_{ELOCITI AT D.O.F. T$ $v_{ELOCITI AT D.O.F. T$
1150		$\#$ *CONGT2_(DPOD2_DPOD6) *CONGT3_(DPOD3_DPOD7) *CONGT2+
1150		# (PROD4 - PROD8) * CONST4 + UNITM(I, NBEAM(IE, 1)) *
1150		#((DBOD) - DBOD() + CONST + (DBOD) + (DBOD
1160		#((PROD2=PROD6)*CONSTS+(PROD4=PROD6)*CONSTS)
1160	n	#CONSITT ((FRODZ-FRODO) CONSIS-(FROD4-FRODO) CONSIST
1160	2	
1162	· 1	
1164		DEBUG SUBCHK
1165		
1102		עוני

1167	С	******************
1168	С	
1169	С	MODECH
1170	С	
1171	C	DETERMINES WHETHER CONVERGENCE ONTO
1172	C	A MODE SHAPE HAS OCCURED (INMSOL=1)
1173	C	
1174	Č	*****
1175	Č	SUBROUTINE MODECH
1176		INCLUDE DACHUS COMPROC
1177	C	INCHODE DAGN VS.COMPROC
1170	C	NO OF IMPRIMIENC TO DEMERMINE MODE
1170		TE MODE NOW ODWITTED AFTER FORMY IMERATIONS STORE
1100	C	TE MODE NOT OBTAINED AFTER FORT TIERATIONS STOP
1100	100	TE (TIMODE, EQ, TO) WRITE (TERINT, TOO)
1101	100	FORMAT(IN ,//, ZUX, MODE NOT FOUND AFTER IEN IIERATIONS :
1182		# STOP',/)
1183		IF (ITMODE.EQ.IU) STOP
1184		ITMODE=ITMODE+I
1185		A=0.
1186		B=0.
1187		DISIP(1) = DISIP(2)
1188	0	DISIP(2)=0.
1189	C	CHECK FOR CONVERGENCE
1190	C	CALCULATE CURRENT DISSIPATION RATE
1191		DO I I=1, NDF, 3
1192		1 K = 1 + 1
1193		11=1NT(FLOAT(1)/NF+0.7)
1194		DISIP(2) = DISIP(2) + VEL(1, 2) * RMASS(11) * PHI(1, 1)
1195		DISIP(2) = DISIP(2) + VEL(IK, 2) * RMASS(II) * PHI(IK, I)
1196	1	CONTINUE
1197	C	
1198	С	NORMALISE VELOCITIES FOR NEW MODE SHAPE
1199	C	
1200		DUM=0.
1201		DO 5 $I=1$, NDF, 3
1202		II = INT(FLOAT(I)/NF+0.7)
1203		<pre>DUM=DUM+VEL(I,2)*VEL(I,2)*RMASS(II)</pre>
1204		IK=I+1
1205		<pre>DUM=DUM+VEL(IK,2)*VEL(IK,2)*RMASS(II)</pre>
1206	5	CONTINUE
1207		RLAMDA=SQRT (DUM)
1208	С	
1209		DO 2 I=1,NDF
1210		PHI(I,2) = VEL(I,2) / RLAMDA
1211		PHI(I,1)=PHI(I,2)
1212	2	CONTINUE
1213	С	CHECK CHANGE IN DISSIPATION RATE
1214	С	RLAMDA=ANORM
1215		IDISIP=0
1216		IF(ABS(DISIP(2)/DISIP(1)-1.D0).LT.5.D-2)IDISIP=1
1217	С	
1218	С	OBTAIN AMPLITUDE OF VELOCITY BY PERFORMING MOMENTUM BALANCE
1219		IF(IDISIP.EQ.1.AND.T.LT.1.D-9)THEN
1220		DO 3 I=1,NDF
1221		II=INT(FLOAT(I)/NF+0.7)
1222		IJ = (INT(FLOAT(I)/NF+0.001))*NF
1223		IF(I.NE.IJ)THEN
1224		A=A+VEL(I,1)*RMASS(II)*PHI(I,1)
1225		B=B+PHI(I,1)*RMASS(II)*PHI(I,1)

1226		END IF
1227	3	CONTINUE
1228		AMP(1) = A/B
1229		AMP(2) = AMP(1)
1230		DO 4 T=1.NDF
1231		VEL(1,2) = PHI(1,1) * AMP(2)
1 2 3 2	Λ	CONTINUE
1222	- T	ENDIE
1222	0	END IF
1234	C	THE OF A NUR TREATE TO I AND THREAT TO A MUTH
1235		IF(T.GT.U.AND.IDISIP.EQ.I.AND.INMSOL.EQ.U)THEN
1236		DO 6 $I=1$, NDF
1237		11=1NT(FLOAT(I)/NF+0.7)
1238		IJ = (INT(FLOAT(I)/NF+0.001))*NF
1239		IF(I.NE.IJ)THEN
1240		A=A+VEL(I,1)*RMASS(II)*PHI(I,1)
1241		B=B+PHI(I,1)*RMASS(II)*PHI(I,1)
1242		END IF
1243	6	CONTINUE
1244		AMP(2) = A/B
1245		ENDIF
1246	С	
1247	C	
1248	Ŭ	RETURN
1249		DEBUG SUBCHK
1250		FND
1250	C	END
1251		
1252	C	* * * * * * * * * * * * * * * * * * * *
1253	C	* * * * * * * * * * * * * * * * * * * *
1254	C	
1255	C	INMODE
1,256	С	
1257	С	SETS UP TIME FUNCTION T AND AMPLITUDE
1258	С	OF MODE SOLUTION, ESTIMATES TOTAL TIME AND
1259	С	IN INSTANTANEOUS MODE SOLUTION TECHNIQUE
1260	С	CHECKS IF NEW MODE HAS BEEN FOUND.LATTER
1261	С	ONLY PERFORMED IF COMBINED MODE AND DIRECT
1262	С	ANALYSIS (ISYM=0) IS REQUESTED .
1263	Č	
1264	Č	*****
1265	C	SUBDOUTINE INMODE
1265		INCLUDE DACING COMPROC
1260	Ċ	INCLODE DAGNVS.COMPROC
1267	C	MODEG-0
1268		MODES=0
1269		A=0.D0
1270		B=0.D0
1271		ITMODE=0
1272		POWA=1.DO/EN
1273		POWB=(EN-1.DO)/EN
1274		POWC=1.DO/POWB
1275		POWD=1.DO/(EN-1.DO)
1276	С	
1277		<pre>RK=1.D0/((DABS(RLAMDA)**POWA)*(AMP(2)**POWB))</pre>
1278		TIME=DT
1279		FACT=POWB*RK*TIME
1280	С	
1281	C	CALCULATE EXPRESSION FOR T(T)
1282	C	TT(2) = (1, DO - FACT) ** POWC
1202	C	DIRIVATIVE OF $T(T)$
1284	C	I = (MCOUNT = C, 1) D = D = -RK * (TT (2) * * POWA)
1201		τ

1285	С	CONSIDER CASES :A) BEFORE MODE SOLN (INMSOL=0)
1286	Ĉ	SEPARATELY B) AFTER MODE SOLN. (INMSOL=1)
1287	•	IF (INMSOL, EO, 0) THEN
1288	С	MODE VELOCITIES
1289	•	$DO \ 1 \ I=1. NDF$
1290		VMODE(T, 2) = AMP(2) * PHT(T, 1) * TT(2)
1291	1	CONTINUE
1292	Ċ	CONTINUE
1293	C	ELSE
1294		
1205		DTTDT(2) - (DTTDT(2) - DK*(TTT(2) * DOTT)) / 2 DO
1 2 9 5	C	$DIIDI(2) = (DIIDI(2) - RR^{*}(II(2) * * POWR))/2.00$
1290	C	
1200		$\frac{11(2)-11(1)+(D11D1(1)+D11D1(2))-D1/2}{END}$
1298		
1299		$DU \ge 1=1, NDF$ TE(MCOUNT = 0, 1)U(T, 1)=0, DO
1300		IF(MCOUNT.EQ.I)U(I,I)=0.D0
1301		VEL(1,2) = VMODE(1,2)
1302		IF(MCOUNT.EQ.1)VMODE(1,2)=AMP(2)*PHI(1,1)*TT(2)
1303	C	
1304		IF(MCOUNT.GT.1)VMODE(I,2) = (VMODE(I,2) + AMP(2) + PHI(I,1)
1305		#*TT(2))/2.D0
1306	C	
1307	2	CONTINUE
1308	С	CHECK IF NEW MODE HAS BEEN REACHED
1309	С	
1310		DO 5 $I=1, NDF, 3$
1311		II=I+1
1312		A=A+(VMODE(I,2)-VEL(I,2))*(VMODE(I,2)-VEL(I,2))
1313		B=B+(VMODE(I,2)*VMODE(I,2))
1314		A=A+(VMODE(II,2)-VEL(II,2))*(VMODE(II,2)-VEL(II,2))
1315		B=B+(VMODE(II,2)*VMODE(II,2))
1316	5	CONTINUE
1317	С	
1318	С	IF NEW MODE REACHED SET NNORM = 1
1319	С	
1320		IF(DSQRT(A)/DSQRT(B).LT.0.05D0.AND.MCOUNT.GT.1)NNORM=1
1321	С	
1322	-	END IF
1323	С	
1324	C	CALCULATE MODE MOMENTS AND AXIAL FORCES
1 3 2 5	C	
1326	, Č	$DO_3 IE=1.NE$
1327		DO 4 J=1.3
1328		DUM=RMO
1320		TE(TEO 3)DIM=RNO
1220		$\mathbb{E} \cap \mathbb{E} \cap $
1331		$\frac{1}{100} = \frac{1}{100} = \frac{1}$
1333	Λ	
1 2 2 2	4	CONTINUE
1 2 2 4		TE AVIAI ECDOES ADE LADGE SET ELAG TO DECHEST NEW MATCHING
1005		ELCHOD
1335	C	FACTOR. In New Engroup the DEEN DREVIOUELY CALCULATED THEN SKID
1330	C	IF NEW FACTOR HAS BEEN PREVIOUSLI CALCULATED THEN SKIP.
133/		IF(MATCHA.NEI.UK.MATCHA.EQ.U)THEN $IF(DADG(FORMOD(IE.2)) GF(0.2)MATCHA.EQ.U) = 1$
1338		IF(DABS(FORMOD(IE,3)).GT.U.2)MATCHA=1
1339	•	END IF
1340	3	CONTINUE
1341		KETUKN
1342		DEBUG SUBCHK
1343		END

;

1344 С 1345 С ***** 1346 С 1347 MATCH С 1348 С MATCHING PROCEDURE (SLOPE ALONE) FOR 1349 1350 С MODE SOLUTION TECHNIQUE . THIS MATCHING 1351 С IS ONLY OPERATIVE IF COMBINED MODE AND С 1352 DIRECT ANALYSIS IS REQUESTED (ISYM=0) 1353 С С 1.354 1355 SUBROUTINE MATCH 1356 INCLUDE DAGNVS.COMPROC 1357 DMATCH=1. 1358 DUM=0. С 1359 LOCATE MAXIMUM BENDING MOMENT OR AXIAL FORCE IN STRUCTURE 1360 N1 = 11361 N2 = 21362 IF (MATCHA.EQ.1) THEN 1363 N1 = 31364 N2=31365 END IF 1366 С 1367 DO 1 IE=1,NE 1368 DO 2 I=N1, N2IF(DABS(FORMOD(IE,I)).GE.DUM)DUM=ABS(FORMOD(IE,I)) 1369 2 1370 CONTINUE 1 1371 CONTINUE 1372 С AMPD=AMP(1) **(1./EN) 1373 1374 RMMAX=DUM/AMPD 1375 IF(RMATCH.GT.1.01)THEN 1376 RMMAX=RMMAX*RMATCH 1377 DMATCH=RMATCH 1378 END IF 1379 С RKMAX=AMP(1)*((RMMAX)**EN) 1380 1381 RMATCH=(1.+(RKMAX)**(1./EN))/(RKMAX**(1./EN)) С 1382 С CALCULATE TOTAL TIME 1383 1384 С TF = (EN/(EN-1.))/(RK*RMATCH)1385 DT=TF/RINT 1386 1387 С 1388 С MATCH YIELD MOMENT AND AXIAL YIELD STRESS 1389 RMO=RMO*RMATCH/DMATCH RNO=RNO*RMATCH/DMATCH 1390 1391 IF (MATCHA.EQ.0) WRITE (IPRINT, 4) RMATCH FORMAT(1H ,///,20X, 'MATCHING FACTOR IS :',E11.6,///) 1392 4 1393 С IF (MATCHA.EQ.1) THEN 1394 1395 WRITE (IPRINT, 3) RMATCH FORMAT(1H ,///,20X, 'REVISED MATCHING FACTOR FOR MEMBRANE 3 1396 #ACTION IS : ',Ell.6,///) 1397 1398 MATCHA=-1 1399 END IF 1400RETURN 1401 DEBUG SUBCHK 1402 END

1403	С	
1404	С	
1405	С	******
1406	С	
1407	С	VMATCH
1408	С	
1409	С	CALCULATE MATCHING FACTOR(S) IN
1410	С	DIRECT SOLUTION PROCEDURE . MATCHING
1411	Ċ	PROCEDURE IS CHOSEN BY DATA INPUT
1412	Č	
1413	Č	* * * * * * * * * * * * * * * * * * * *
1414	Ŭ	SUBBOUTTNE WATCH
1415		INCLUDE DAGNUS COMPROC
1416		DOUBLE DECISION DECEMINOC
1417	C	DODDLE I RECISION RHOMI (NE, 27, DARICH, DIMACH
1/10	C	тматсн-О
1/10		
1419		
1420		
1421		
1422		ITMACH=ITMACH+I
1423		IF (ITMACH.EQ.I) THEN
1424		DMATCH=1.DO
1425		DPMACH=1.DU
1420	0	
1427	C	DEMERMINE DENDING NONENE DIACRAM
1420	C	DETERMINE BENDING MOMENT DIAGRAM
1429	C	DO 6 JE-1 NE
1430		DO O IE-I, NE
1431		RMOMT(IE,I) = 0.00
1432	C	RMOMI(1E, Z) = 0, D0
1433	6	CONTINUE
1434		DO I IE=I, NE
1435		DO 2 I=1, NINV
1436		RMOMT(IE,I) = RMOMT(IE,I) + FORCE(I,IE,I) * XX(I)
1437	0	RMOMT(1E, 2) = RMOMT(1E, 2) + FORCE(1, 1E, 2) * XX(1)
1438	2	CONTINUE
1439	1	CONTINUE
1440	C	
1441		IF(T.LT.I.D-9)THEN
1442	С	
1443	С	FIND MAXIMUM B.M.
1444	, C	
1445		DO 3 IE=1,NE
1446		DO 4 $I=1, 2$
1447		IF(DABS(RMOMT(IE,I)).GT.DUM)DUM=DABS(RMOMT(IE,I))
1448	4	CONTINUE
1449	3	CONTINUE
1450		IF (PAMTCH.LT.1) THEN
1451		PMATCH=(1.DO+(DUM/RMO))/((DUM/RMO))
1452		END IF
1453		RMATCH=(1.DO+DUM/RMO)/((DUM/RMO)**(1.DO/PMATCH))
1454		RMO=RMO*RMATCH/DMATCH
1455		RNO=RNO*RMATCH/DMATCH
1456		EN=EN*PMATCH/DPMACH
1457		EN1=EN+1.DO
1458		EN2=EN1+1.DO
1459		END IF
1460	С	
1461	С	RESET REACTIONS TO GET BETTER ESTIMATE OF REACTIONS AT

1462 1463	C	MATCHED VALUES IF(ITMACH.EQ.1)THEN
1464		DO 11 I=1, NINV
1465		XX(T) = XXDIM
1466	11	CONTINUE
1467		END IF
1468	C	
1460	C	
1409	0	$IF(I \cdot DI \cdot I \cdot D - 9)TF = (EN/(EN - I \cdot))/(RK^RMATCH^20 \cdot D0)$
1470	C	
14/1	-	WRITE (IPRINT, 5) ITMACH, RMATCH, PMATCH
14/2	5	FORMAT(1H ,//,20X, 'ITERATION NO. ',1X,12,5X, 'STRESS FACTOR
1473		#IS :',2X,E11.6,10X,'POWER FACTOR IS :',2X,E11.6,//)
1474		DT=TF/RINT
1475	С	
1476	С	HAS MATCHING FACTOR CONVERGED
1477	С	
1478		IF(DABS((RMATCH/DMATCH)-1,DO),LT,0,05D0,AND,
1479		#DABS((PMATCH/DPMACH)-1, DO), LT, 0, 05DO) IMATCH=1
1480	C	
1481	Ŭ	
1401		
1402	7	WRITE(IPRINI,/)
1483	/	FORMAT(IH ,///, 30X, 'MATCHING FACTOR HAS NOT CONVERGED AFTER
1484		#TEN ITERATIONS : STOP',///)
1485		STOP
1486		END IF
1487		IF(IMATCH.EQ.1)ITMACH=0
1488	С	
1489		RETURN
1490		DEBUG SUBCHK
1491		END
1492	C	
1493	C	
1494	C	*****
1494		
1495	C	
1496	C	COMDIF
1497	С	
1498	С	SET UP COMPATIBILITY EQUATIONS , ONE FOR
1499	С	EACH DEGREE OF FREEDOM OF THE RELEASED
1500	С	STRUCTURE AS FUNCTIONS OF NODAL FORCES.
1501	С	TAKE PARTIAL DIRIVATIVES OF EQUATIONS W.R.T.
1502	С	EACH NODAL FORCE IN TURN (MATRIX PARDIF).
1503	· C	· · ·
1504	C	*******
1505	•	SUBROUTINE COMDIE
1506		INCLUDE DACANG COMPROC
1505		DOUDLE DECISION VDUM(2*NN)
1507		DOUBLE PRECISION ADUM(2"NN)
1508		
1509	~	ITREAC=ITREAC+I
1510	C	
1511		DO 55 I=1,NINV
1512	С	DONT NEED TO TEST FOR CONVERGENCE IF IN IMPLICIT MODE
1513	С	
1514		IF(IRND.EQ.0)THEN
1515		XDUM(I) = XX(I)
1516		IF(ITREAC.GT.1)THEN
1517		XX(T) = XX(T) + FLEX(T, NINV+1)
1510		TE(DABS(XX(T)) GT = D-3)THEN
1510		TE(DABC(XX(T))/YDIM(T) = 1 DO) CT 1 D = 3)TELAC=0
1520		$\mathbf{T} \left(\mathbf{D} \mathbf{T} \mathbf{D} \right) \left(\mathbf{A} \mathbf{A} \left(\mathbf{T} \right) \right) \mathbf{A} \mathbf{D} \mathbf{D} \left(\mathbf{T} \right)^{-1} \cdot \mathbf{D} \mathbf{D} \right) \cdot \mathbf{O} \mathbf{T} \cdot \mathbf{T} \cdot \mathbf{D} \mathbf{D} \mathbf{D} \mathbf{T} \mathbf{D} \mathbf{A} \mathbf{O}^{-1} \mathbf{O}$
1 7 2 0		END IF

1521 END IF 1522 END IF 1523 FLEX(I,NINV+1)=0.1524 COMPAT(I)=0.55 1525 CONTINUE 1526 С 1527 IF(ITREAC.EQ.1)IFLAG=0 1528 С 1529 DO 6 I=1, NINV1530 DO 9 J=1,NINV PARDIF(I, J) = 0.1531 9 CONTINUE 1532 1533 6 CONTINUE 1534 С 1535 IFLAG=1 С 1536 SET UP NINV COMPATIBILITY EQUATIONS (NINV=NO. OF NODAL С 1537 С 1538 FORCES) 1539 С SUM OVER ALL ELEMENTS IE 1540 DO 1 IE=1, NEDO 2 I=1,NINV 1541 MOMENTS AND AXIAL FORCES ARE NORMALISED 1542 С AMOM(I) = (FORCE(I, IE, 2) - FORCE(I, IE, 1))/RMO1543 1544ANORM1(I)=FORCE(I,IE,1)/RMO+FORCE(I,IE,3)/RNO ANORM2(I)=FORCE(I, IE, 3)/RNO-FORCE(I, IE, 1)/RMO 1545 1546 2 CONTINUE SUMA=0. 1547 SUMB=0. 1548 1549 SUMC=0. 1550 DO 3 I=1,NINV 1551 SUMA=SUMA+XX(I)*ANORM1(I) SUMB=SUMB+XX(I)*ANORM2(I) 1552 1553 SUMC=SUMC+XX(I)*AMOM(I) 1554 CONTINUE 3 1555 С SUM1=SUMA+SUMC 1556 1557 SUM2=SUMB-SUMC IF(DABS(SUM1).LT.1.D-12)SUM1=0. 1558 IF(DABS(SUM2).LT.1.D-12)SUM2=0.1559 1560 С 1561 С SET UP SIGNUM FUNCTIONS FOR POWERED TERMS 1562 SIG=1.D0 SIGA=1.D0 1563 1564 SIGB=1.D0 1565 SIGC=1.D0 1566 SIGD=1.D0 IF((SUM1).LT.0) SIGA=-1.D0 1567 1568 IF((SUM2).LT.0) SIGB=-1.D0IF(SUMA.LT.O) SIGC=-1.DO 1569 IF(SUMB.LT.0) SIGD=-1.DO 1570 POWI=SIGA*(DABS(SUM1)**EN) 1571 POWJ=SIGC*(DABS(SUMA)**EN) 1572 1573 POWK=SIGB*(DABS(SUM2)**EN) POWL=SIGD*(DABS(SUMB)**EN) 1574 POWA=POWI*(SUM1) 1575 POWB=POWK*(SUM2) 1576 1577 POWC=POWA*(SUM1) 1578 POWD=POWB*(SUM2) POWE=POWJ*SUMA 1579

1580		POWF=POWL*SUMB
1581		POWG=POWE*SUMA
1582		POWH=POWF*SUMB
1583		PRODA=EN1*SUMC/CL(IE)
1584		PRODB=EN2*SUMC*SUMC/(CL(IE)*CL(IE))
1585		PRODC=EN1*SUMC*SUMC/(CL(IE)*CL(IE))
1586	С	SET UP COMPATIBILITY EQUATIONS, ONE FOR EACH COMPONENT OF
1587	С	VELOCITY
1588	-	DO 4 I=1.NINV
1589		COMPAT(T) = COMPAT(T) + (FORCE(T, TE, 3) * FPST(0*0, 5) *
1590		#((POWA-POWE)/PRODA-(POWE-POWE)/PRODA)
1591		#+RN0*EPSI0/(2 *RM0)*((AMOM/I)*PM0/CI(IF))*((POWC-POWG)/PRODB
1502		$\frac{1}{2} = \frac{1}{2} = \frac{1}$
1503		$\frac{1}{2} = \frac{1}{2} = \frac{1}$
1504		#=(rOWD - rOWD)/PRODE+SOME*(POWE - POWE)/PRODE) $#=(rOWD - rOWE)/PRODE)$
1594	0	# + FORCE(I, IE, I) " ((POWA-POWE) / PRODA+ (POWB-POWF) / PRODA))
1595	C	
1596	C	SET OP PARTIAL DIRIVATIVES
1597		DO 5 J=1, NINV
1598		PARTA=ANORM1(J)+AMOM(J)
1599		PARTB=ANORM2(J)-AMOM(J)
1600		IF(DABS(PARTA).LT.1.D-12)PARTA=0.
1601		IF(DABS(PARTB).LT.1.D-12)PARTB=0.
1602		PART1=((POWI*PARTA-POWJ*ANORM1(J))*EN1*PRODA
1603		#-EN1*AMOM(J)*(POWA-POWE)/CL(IE))/(PRODA*PRODA)
1604		PART2=((POWK*PARTB-POWL*ANORM2(J))*EN1*PRODA
1605		#-EN1*AMOM(J)*(POWB-POWF)/CL(IE))/(PRODA*PRODA)
1606		PART3=((POWA*PARTA-POWE*ANORM1(J))*EN2*EN2*PRODA*PRODA/
1607		#(EN1*EN1)-(POWC-POWG)*2.*EN2*PRODA*AMOM(J)/(EN1*CL(IE)))/
1608		#(PRODB*PRODB)
1609		PART4 = ((POWA * ANORM1 (J) + SUMA * EN1 * POWI * PARTA) - (POWE * ANORM1 (J))
1610		#+SUMA*EN1*POWJ*ANORM1(J)))*PRODA*PRODA/EN1
1611		# - ((POWA - POWE) * 2. * PRODA * AMOM(J) * SUMA) / CL(TE)) / (PRODC * PRODC)
1612		PART5 = ((POWB * PARTB - POWF * ANORM2(J)) * EN2*EN2*PRODA * PRODA /
1613		$\frac{1}{(FN)} = \frac{1}{(FN)} = 1$
1614		$#(ENI^{ENI}) = (FOWD = FOWH)^{2} \cdot EN2^{2} F KODA^{AMOM(O)} (ENI^{CD(IE)}) $
1615		$\frac{1}{2} \left(\frac{1}{2} - 1$
1015		PARIO = ((ANORM2(J) * POWBTSOMB*EN1* POWR*PARIB) = (POWF*ANORM2(J))
1010		$\frac{1}{2} + \frac{1}{2} + \frac{1}$
1617		$\#$ - ((POWB-POWF) ^ 2. ^ PRODA ^ AMOM(J) ^ SUMB)/CL(IE))/(PRODC ^ PRODC)
1618		$PARDIF(I,J) = PARDIF(I,J) + FORCE(I,IE,J) \times EPSIO(.5) \times (PARTI-PART2)$
1619		#+RNO*EPSIO/(2.*RMO)*((AMOM(I)*RMO/CL(IE))*(PART3
1620		#-PART4-PART5+PART6)+FORCE(I,IE,1)*(PART1+PART2))
1621	5	CONTINUE
1622	4	CONTINUE
1623	С	
1624	С	CHECK IF NODAL FORCES HAVE BEEN FOUND
1625	С	(ARE VIRTUAL VELOCITIES EQUAL TO REAL VELOCITIES ?)
1626	С	
1627		DO 11 I=1,NE
1628		$II = 2 \times I - 2$
1629		IJ=3*I
1630		DO 12 J=1,2
1631		IK=II+J
1632		IL=IJ+J
1633		T = (DABS(DUMVEL(TL)), GT = 1, D = 3) THEN
1634		TE(DABS(COMPAT(TK))/DIMVEL(TL)-1)DO) GT 1 D-3)TELAG=0
1625		LIGE TI (PUPP) (COULUI (TV) POULLIN(TN) - T. PO). GI. T. P-2) TL DUG-0
1626		
1627		TE (DADS (COMPAT(IX) - DOMVED(ID)).GI.I.D-4)IFDAG-O
1620	10	
1038	12	CONTINUE

1639	11	CONTINUE
1640		IF(NRED, NE, 4)THEN
1641		IF(DABS(COMPAT(NINV)), GT, 1, D-4)IFLAG=0
1642		
1643	C	
1644	C	
1044	C	
1645		IF (ITREAC.EQ.100) THEN
1646	• •	WRITE (IPRINT, 30)
1647	30	FORMAT(///, 3X, ' NO CONVERGENCE ONTO NODAL FORCES : CHANGE N
1648	-	#OR INITIAL NODAL FORCE ESTIMATE ')
1649		STOP
1650		END IF
1651		IF(IFLAG.EQ.1)ITREAC=0
1652	С	
1653	С	
1654	С	HAVING OBTAINED THE PARTIAL DIRIVATIVES OF COMPATIBILITY
1655	С	EQUATIONS, MUST INVERT TO OBTAIN PERTURBATIONS ON NODAL
1656	Ċ	FORCES . FIRST SET UP AUGMENTED MATRIX FOR INVERSION.
1657	Ĉ	
1658	Ŭ	DO 66 T=1 NINV
1659		$E = I N T (E I \cap A T (I) / 2 + 0.6)$
1660	0	
1661	C	TE (TE CE NE)IE-NE
1001		IF(IE.GI.NE)IE-NE
1662		11=1+NF+1E-1
1663		FLEX(1, NINV+1) = -COMPAT(1) + DUMVEL(11)
1664		DO // J=1, NINV
1665		FLEX(I, J) = PARDIF(I, J)
1666		FLEX(J,I) = PARDIF(I,J)
1667		AA(I,J) = PARDIF(I,J)
1668		AA(J,I) = PARDIF(I,J)
1669	С	
1670	77	CONTINUE
1671	66	CONTINUE
1672	С	
1673		RETURN
1674		DEBUG SUBCHK
1675		END
1676	C	
1677	C	
1670	C	*****
1078		
1679	C	
1680	C	ACCAXC
1681	C	
1682	С	CALCULATE ACCELERATIONS FROM THE NODAL FORCES
1683	C	USING THE EQUILIBRIUM EQUATIONS AT EACH D.O.F.
1684	С	
1685	С	***************************************
1686		SUBROUTINE ACCAXC
1687		INCLUDE DAGNVS.COMPROC
1688	С	
1689		DO 1 IE=1,NE
1690		IA=NBEAM(IE,1)
1691		IB=NBEAM(IE,2)
1692		IJ=3*IA-2
1693		IK=3*IB-2
1694		TIATJ
1695		$T_{2A=T_{1A+1}}$
1696		T I B = T K
1697		
1091		

1698		IIJ=IlB-IE-2
1699		IIK=I2B-IE-2
1700		II=INT(FLOAT(I1B)/3+0.7)
1701	С	
1702	С	CALCULATE ACCELERATIONS FROM NODAL FORCES
1703	С	
1704		ACC(T B,1)=0, D0
1705		ACC(12B,1) = 0.00
1706		ACC(TIB I) = -XX(TTT) / PMASS(TT)
1707		ACC(12B,1) = XX(110)/RMASS(11)
1709	C	ACC(12D,1) = -XX(11X)/(RIADD(11))
1700	1	CONTINUE
1710	Ċ	CONTINUE
1710	C	זאסוושיים
1712		NETURN DEDIC CUDCUK
1712		DEBUG SUBCIIK
1713	9	END
1/14	C	
1/15	C	
1716	С	
1717	С	***************************************
1718	С	
1719	С	IMPLIC
1720	С	
1721	С	ASSEMBLES MATRICES AND VECTORS REQUIRED
1722	С	TO SOLVE FOR NODAL FORCE RESIDUALS IN
1723	С	IMPLICIT FORWARD INTEGRATION PROCEDURE
1724	С	
1725	С	****************
1726		SUBROUTINE IMPLIC
1727		INCLUDE DAGNVS.COMPROC
1728	С	
1729	-	SIG=1.D0
1730		TF(TRND, GT, 1)SIG=-1, DO
1731		DO TF= NF
1732		TA = NBFAM(TF 1)
1733		IR=NBFAM(IE, 2)
1724		I D = N D D A H (I D , Z) $I I = 2 \times I \lambda = 2$
1725		$TV = 2 \times TR = 2$
1735		
1736		
1/3/		1 2B=1 K+1
1738		11J=11B-1E-2
1/39		11 K = 12 B - 1 E - 2
1740		11=1NT(FLOAT(11B)/3+0.7)
1741	C	
1742		FLEX(IIJ,NINV+I) = -(COMPAT(IIJ) - DUMVEL(IIB))
1743		#+(SIG*ACC(I1B,1)-XX(IIJ)/RMASS(II))*DT/2.D0
1744		<pre>FLEX(IIK,NINV+1)=-(COMPAT(IIK)-DUMVEL(I2B))</pre>
1745		#+(SIG*ACC(I2B,1)-XX(IIK)/RMASS(II))*DT/2.D0
1746	С	
1747	1	CONTINUE
1748	С	
1749	С	FOR INDETERMINATE STRUCTURES ADD ROTATIONAL CONTRIBUTION
1750	С	v.
1751		IF(NRED.NE.4)THEN
1752		<pre>FLEX(NINV,NINV+1)=-(COMPAT(NINV)-DUMVEL(NDF))</pre>
1753		#+(SIG*ACC(NDF,1)-XX(NINV)/RMASS(NN))*DT/2.D0
1754		END IF
1755	С	
1756		IOUT=0

1757		DO 2 I=1,NINV
1758		II = INT(FLOAT(I+2)/2+0.6)
1759		DO 3 J=1,NINV
1760		SIG=0.
1761		IF(I.EQ.J)SIG=1.
1762		IF(II.GT.NN)II=NN
1763		<pre>FLEX(I,J)=AA(I,J)+DT*SIG/(2.D0*RMASS(II))</pre>
1764	3	CONTINUE
1765	2	CONTINUE
1766		RETURN
1767		DEBUG SUBCHK
1768		END
1769	С	
1770	С	
1771	С	
1772	С	* * * * * * * * * * * * * * * * * * * *
1773	С	
1774		REVISE
1775	С	
1776	C	UPDATES OR REVISES ACCELERATIONS,
1777	С	VELOCITIES , DISPLACEMENTS AND NODAL
1778	C	FORCES BY RESIDUAL QUANTITIES.
1779	C	
1780	С	*******
1781		SUBROUTINE REVISE
1782		INCLUDE DAGNVS.COMPROC
1783		DOUBLE PRECISION DELTAV(NDF), IMAXM(NDF)
1784		ITREAC=0
1785	C	
1/86		DO 9 I=1, NINV
1787		ANORMI(I) = 0.00
1788		DO $10 J=1, NINV$
1789	1.0	ANORMI(1) = ANORMI(1) + AA(1, J) * FLEX(J, NINV+1)
1790	10	CONTINUE
1791	9	CONTINUE
1792		SIG=1.DU
1793		1F(1RND.EQ.1)SIG=-1.DO
1794		DO I IE=1, NE
1795		IB = NBEAM(IE, 2)
1796		$1K=3\times1B-2$
1700		
1700		IZBELNTI
1000		IIJ = IIB - IE - 2
1800		$\frac{11}{11} = \frac{2}{12} = \frac{2}{12}$
1001		$\frac{11-101(FLOAI(IID)/5+0.7)}{12(IICOM)(IID)/5+0.7)}$
1902		$\frac{11}{11} \frac{11}{10} 11$
1803		$\frac{1}{4} \operatorname{ELEX}(110) - (\operatorname{MASS}(11) - \operatorname{SIG} \operatorname{ACC}(110, 1) + \operatorname{ACC}(110) +$
1805		$\frac{1}{2} \frac{1}{2} \frac{1}$
1806		$\frac{1}{4} \operatorname{FLFX}(11K \operatorname{NINV}+1))$
1807		DELTAV(T B) = DELTAV(T B) * DT/(2, DO*RMASS(TT))
1808		DELTAV(I2B) = DELTAV(I2B) * DT/(2 DO * RMASS(II))
1809	1	CONTINUE
1810	-	DELTAV(NDF) = -(RMASS(NN) * SIG*ACC(NDF, 1) + XX(NINV) +
1811		#FLEX(NINV.NINV+1))
1812		DELTAV(NDF) = DELTAV(NDF) * DT/(2, DO*RMASS(NN))
1813	С	
1814	C	UPDATE ACCELERATIONS, VELOCITIES AND DISPLACEMENTS
1815	-	DO 4 I=1.NDF

1816		<pre>IF(IRND.EQ.1)DUMVEL(I)=VEL(I,1)</pre>
1817		ACC(I,2) = ACC(I,1)
1818		ACC(I, 1) = 2.DO*DELTAV(I)/DT+SIG*ACC(I, 1)
1819		VEL(I,2)=DUMVEL(I)+DELTAV(I)
1820		DUMVEL(I) = VEL(I, 2)
1821		U(I,2)=U(I,1)+(VEL(I,1)+VEL(I,2))*DT/2.D0
1822	4	CONTINUE
1823	С	
1824		DO 2 I=1,NINV
1825		IE=INT(FLOAT(I)/2+0.6)
1826		II=I+3+IE-1
1827		IF(II.GT.NDF)II=NDF
1828		XX(I) = XX(I) + FLEX(I, NINV+1)
1829	С	
1830	2	CONTINUE
1831	С	
1832	С	CHECK FOR CONVERGENCE
1833	С	
1834		IOUT=1
1835		DO 5 I=1,NINV
1836		IE=INT(FLOAT(I)/2+0.6)
1837		II=I+IE+2
1838		IF(II.GT.NDF)II=NDF
1839		IF(DABS(VEL(II,2)).GT.1.D-1)THEN
1840		IF(DABS(COMPAT(I)/VEL(II,2)-1.D0).GT.1.D-2)IOUT=0
1841		END IF
1842	5	CONTINUE
1843	С	
1844		IF(IRND.EQ.1)IOUT=0
1845	С	
1846	С	SET UP VELOCITY FOR MODE DETERMINATION
1847		IF(IOUT.EQ.1)THEN
1848		IF(IRND.LE.5)THEN
1849		DT=DT*1.5
1850		WRITE(IPRINT,123)
1851	123	FORMAT(///,15X,'STABLE SOLUTION : INCREASE TIME STEP BY
1852		#FACTOR OF 1.5 ',///)
1853		END IF
1854		DO 6 $I=1$, NDF
1855		VEL(I,1) = VEL(I,2)
1856	6	CONTINUE
1857		END IF
1858	С	
1859		DO 3 IE=1,NE
1860		I=NBEAM(IE,2)
1861		II=2*I+IE-1
1862		IJ=II+1
1863		COORDX(I) = XCOORD(I) + U(II, 2)
1864		COORDY(I) = YCOORD(I) + U(IJ, 2)
1865	3	CONTINUE
1866	C	
1867		KETUKN DEDUG GUDGUK
1868		DEBOG SUBCHK
1869		END
1810	C	

1872	С	***************************************
1873	С	
1874	С	СНЕСК
1875	С	
1876	С	CHECKS IF RESULTS OBTAINED FROM THE
1877	С	DIRECT METHOD OF ANALYSIS HAVE CONVERGED
1878	С	ONTO THE INSTANTANEOUS MODE SOLUTIONS.
1879	C	THIS CHECK IS ONLY PERFORMED IF THE
1880	č	COMBINED DIRECT AND MODE SOLUTION OPTION
1991	c	IC EMDIOVED
1992	C	15 EMPLOIED.
1002	C	*****
1003	C	
1005		SUBROUTINE CHECK
1992	9	INCLUDE DAGNVS.COMPROC
1886	C	FOR MODE STRUCTURES ONLY
1887		IF(ISYM.EQ.0)THEN
1888		A=0.D0
1889		B=0.D0
1890		DO 1 I=1,NDF,3
1891		II=I+1
1892		A=A+(VMODE(I,2)-VEL(I,2))*(VMODE(I,2)-VEL(I,2))
1893		B=B+VMODE(I,2)*VMODE(I,2)
1894		A=A+(VMODE(II,2)-VEL(II,2))*(VMODE(II,2)-VEL(II,2))
1895		B=B+VMODE(II,2)*VMODE(II,2)
1896	1	CONTINUE
1897		A=DSQRT(A)/DSQRT(B)
1898		IF(A.LT.0.05D0)INMSOL=1
1899		IF(INMSOL.EQ.1)THEN
1900		WRITE(IPRINT,100)
1901	100	FORMAT(1H ,///,20X, 'MODE SOLUTION REACHED : TREBLE TIME
1902		#STEP',///)
1903		DT=DT*3.DO
1904		END IF
1905		END IF
1906	С	SET UP VELOCITY ARRAYS FOR DIRECT ANALYSIS PROCEDURE
1907	С	RESET DISPLACEMENT ARRAY.
1908		DO 2 $I=1, NDF$
1909		U(I,1) = U(I,2)
1910	С	IF MODE FOUND STORE CURRENT DISPLACEMENTS
1911		IF(INMSOL.EO.1)THEN
1912		UMODE(T) = U(T, 2)
1913		VMODE(I, I) = VMODE(I, 2)
1914		VEL(I,1) = VMODE(I,2)
1915		VEL(1, 2) = VMODE(1, 2)
1916		U(I, 1) = 0, D0
1917		U(T, 2) = 0.D0
1918		END TF
1919		VEL(T, 1) = VEL(T, 2)
1920	2	CONTINUE
1921	Ĉ	CONTINUE
1922	C	STORE INITIAL VALUES OF T(T) AND DT(T)/DT
1923	C	BIONE INTITUE VIEWING OF I(I) AND DI(I)/DI
1924	0	TE (INMSOL, EQ. 1) THEN
1 9 2 5		DTT (1) = DTT (2)
1926		TT(1) = TT(2)
1007		
1000		
1020		DEBUG SUBCHK
1030		
1930		

1931	С	
1932	С	******
1933	С	
1934	C	STORE
1935	C	
1936	Ċ	REVISES AND STORES DISPLACEMENTS
1937	č	AND CHEPENER GEOMERDY OF CUPHICELINE
1020	C	INCRANENT GEOMETRY OF STRUCTURE IN
1930	C	NOTE THAT THE CONTRACT OF A CONTRACTACT OF A CONTRACT OF A CONTRACTACT OF A CONTRACTACT OF A CONTRACT OF A CONTRAC
1939	C	NOTE THAT THIS TECHNIQUE IS ONLY
1940	C	EMPLOYED WHEN COMBINED MODE AND
1941	C	DIRECT ANALYS IS REQUESTED (ISYM=0)
1942	C	
1943	C	******
1944		SUBROUTINE STORE
1945		INCLUDE DAGNVS.COMPROC
1946	С	
1947	С	UPDATES DISPLACEMENTS IN LARGE DISPLACEMENT
1948	С	MODE SOLUTION PROCEDURE
1949	С	
1950		DO 1 I=1,NDF
1951		U(I,2) = (VMODE(I,1) + VMODE(I,2)) * DT/2.D0
1952	С	
1953	1	CONTINUE
1954	С	
1955		DO 2 IE=1.NE
1956		T = NBEAM(TE, 2)
1957		TT = 2*T + TE - 1
1958		T.T=TT+1
1959		COOPN(I) - COOPN(I) + U(II 2) - U(II 1)
1960		COOPDY(I) = COOPDY(I) + U(II, 2) = U(II, 1)
1960		UMODE(II) = COORDY(I) + U(10, 2) = U(10, 1)
1901		UMODE(II) = COURDX(I) - XCOORD(I)
1962		UMODE(IJ) = COORDI(I) - ICOORD(I)
1963		U(11,1)=U(11,2)
1964	•	U(1J, 1) = U(1J, 2)
1965	2	CONTINUE
1966	C	
1967		RETURN
1968		DEBUG SUBCHK
1969		END
1970	С	
1971	С	
1972	C	*******************
1973	С	UPDATE
1974	С	
1975	С	UPDATES VELOCITY AND AMPLITUDE
1976	С	QUANTITIES IN INSTANTANEOUS MODE
1977	С	SOLUTION TECHNIQUE .NOTE THAT
1978	С	THIS TECHNIQUE IS ONLY USED WHEN
1979	С	COMBINED MODE AND DIRECT SOLUTION
1980	С	PROCEDURE IS EMPLOYED (ISYM=0)
1981	С	
1982	С	******
1983		SUBROUTINE UPDATE
1984		INCLUDE DAGNVS, COMPROC
1985	С	
1986	č	SET UP INITIAL VELOCITY ARRAY FOR NEW MODE DETERMINATION
1007	C	SHI OF INTITUD VEROCITI IMANIFION NEW HODE DITERNITION
1988	C	A=0.D0
1989		B=0.00

1990		DO 1 I=1,NDF
1991		VMODE(I,1)=VMODE(I,2)
1992	С	
1993	С	DETERMINE NEW AMPLITUDE
1994	C	
1995	•	TT = TNT(FLOAT(T)/NF+0.7)
1996		I = (I = (I = 0) + (I =
1007		TE/T NE TI) GUEN
1997		$\frac{1}{1} \left(1 \cdot NE \cdot IO \right) \left(I \cap EN \right)$
1998		A = A + VMODE(1, 1) * RMASS(11) * PHI(1, 1)
1999		$B=B+PHI(1,1) \cap KMASS(11) \cap PHI(1,1)$
2000	2	END IF
2001	C	
2002	1	CONTINUE
2003		AMP(2) = A/B
2004		TT(1) = 1.D0
2005		DTTDT(1) = DTTDT(2)
2006	С	
2007		RETURN
2008		DEBUG SUBCHK
2009		END
2010	С	
2011	C	* * * * * * * * * * * * * * * * * * * *
2012	Ċ	
2013	Č	PICTUR
2014	C	
2014	C	TE DICT FOULIS 'DIOT' THEN DIOTS
2015	C	ADE CREATED OF DEFORMED SHADE OF
2010	C	
2017		THE SIRUCIURE CORRESPONDING IO
2018	C	TIME STEP WHEN RESULTS ARE OUTPUT
2019	C	
2020	С	*********
2021		SUBROUTINE PICTUR
2022		INCLUDE DAGNVS.COMPROC
2023		DOUBLE PRECISION XMAX,YMAX
2024	С	
2025		CALL NEWPAG
2026		CALL PAGSIZ(20.5,29.)
2027		CALL PLOT $(2.0, 5.0, -3)$
2028	С	
2029	-	XMAX=-999.D0
2030		YMAX = -999.D0
2030		DO 22 J=1 JPLTS
2031		DO = 1 = 1 NN
2032		T_{T} T_{T
2033		TE(DABS(XCOPLI(1,0)).GE(XMAX)XMAX-DABS(XCOPLI(1,0))
2034	1	IF(DADS(ICOPLI(1,0)).GE.IMAA)IMAA-DADS(ICOPLI(1,0))
2035	1	CONTINUE
2036	22	CONTINUE RNAN RNAN (NNAN NNAN)
2037		RMAX = DMAX I (XMAX, YMAX)
2038		FX=18.0/RMAX
2039		FY=18.0/RMAX
2040	С	
2041		DO 21 J=1, IPLTS
2042		DO 2 I=1,NN
2043		XCOPLT(I,J)=XCOPLT(I,J)*FX
2044		YCOPLT(I,J)=YCOPLT(I,J)*FY
2045	2	CONTINUE
2046		CALL $PLOT(0., 0., 3)$
2047		DO $3 I=1, NN$
2048		CALL PLOT (XCOPLT(I,J), YCOPLT(I,J),2)

2049 CALL SYMBOL(XCOPLT(I,J), YCOPLT(I,J), 0.2, 11, 0., -1) 2050 3 CONTINUE 21 2051 CONTINUE 2052 RETURN 2053 DEBUG SUBCHK 2054 END 2055 С С ****** 2056 С 2057 С 2058 ΟυΤΡυΤ С 2059 С 2060 OUTPUTS RESULTS OF ANALYSIS AFTER 2061 С NDIV TIME STEPS AND WHEN STRUCTURE С 2062 IS AT OR NEAR REST 2063 С С 2064 С 2065 2066 SUBROUTINE OUTPUT 2067 INCLUDE DAGNVS.COMPROC 2068 С 2069 IF(T.LT.1.D-9)GO TO 23 2070 С С OUTPUT RESULTS OF MODE SOLUTION 2071 3 2072 FORMAT(1H , I3, 3X, E13.6, 6X, E13.6, 9X, E13.6, /) 9 2073 FORMAT(1H ,2X,12,8X,E13.6,4X,E13.6,4X,E13.6,/) 4 FORMAT(1H ,2X,12,8X,E13.6,4X,E13.6,/) 2074 2075 AMPERC=100.*AMP(2)/AMP(1)WRITE(IPRINT, 11)T, DT, AMPERC 2076 FORMAT(1H ,/,' TIME ',Ell.6,15X, 'TIME INCREMENT IS', 2077 11 #Ell.6,15X, 'VELOCITY AMPLITUDE (PERCENT) IS ',F7.3,/) 2078 2079 IF(INMSOL.EQ.1)THEN IF (DISPL.EQ. 'LARGE') THEN 2080 WRITE(IPRINT,60) 2081 60 2082 FORMAT(1H ,18X,' MODE SHAPE',//, #'NODE',6X,'X',17X,'Y',19X,'ROTATION',/) 2083 DO 40 I=1,NDF-2,3 2084 2085 II=INT(FLOAT(I)/NF+0.7) WRITE(IPRINT, 3)II, PHI(I, 1), PHI(I+1, 1), PHI(I+2, 1) 2086 2087 40 CONTINUE 2088 END IF WRITE(IPRINT, 12) 2089 2090 12 FORMAT(1H, 20X, 'MOMENTS AND AXIAL FORCES', //, #'ELEMENT',8X, 'MOMENT(A)', 2091 #8X, 'MOMENT(B)',8X, 'AXIAL FORCE',/) 2092 DO 13 IE=1,NE 2093 WRITE(IPRINT,9)IE,FORMOD(IE,1),FORMOD(IE,2),FORMOD(IE,3) 2094 2095 13 CONTINUE 2096 WRITE (IPRINT, 14) FORMAT(1H ,/,26X, 'VELOCITY',//, 2097 14# 'NODE', 6X, 'X', 17X, 'Y', 19X, 'ROTATION', /) 2098 2099 DO 15 I=1,NDF-2,3 II = INT(FLOAT(I)/NF+0.7)2100 WRITE(IPRINT, 3)II, VMODE(I, 2), VMODE(I+1, 2), VMODE(I+2, 2) 2101 CONTINUE 2102 15 WRITE (IPRINT, 16) 2103 FORMAT(1H ,/,21X, 'DISPLACEMENTS',//, 2104 16 #'NODE',11X,'X',17X,'Y',/) 2105 2106 DO 17 I=1,NDF-2,3 2107 II=INT(FLOAT(I)/NF+0.7)

```
2108
               WRITE(IPRINT, 4) II, UMODE(I), UMODE(I+1)
        17
2109
               CONTINUE
2110
        С
2111
               END IF
               IF (INMSOL.EQ.O.AND.ISYM.EQ.O) THEN
2112
2113
        С
2114
               WRITE (IPRINT, 18)
2115
        18
               FORMAT(1H ,/,33X, 'VELOCITIES',//, 'NODE',
2116
              #11X, 'X(DIRECT)', 10X, 'X(MODE)'
              #16X, 'Y(DIRECT)', 12X, 'Y(MODE)', /)
2117
2118
               DO 19 I=1,NDF-2,3
2119
               II=INT(FLOAT(I)/NF+0.7)
2120
               WRITE(IPRINT,20)II,VEL(I,1),VMODE(I,2),VEL(I+1,1),
2121
              \#VMODE(I+1,2)
2122
        20
               FORMAT(1H ,2X,12,7X,E13.6,6X,E13.6,9X,E13.6,9X,E13.6,/)
2123
        19
               CONTINUE
2124
        С
2125
               WRITE(IPRINT, 16)
2126
               DO 22 I=1,NDF-2,3
2127
               II = INT(FLOAT(I)/NF+0.7)
2128
               WRITE(IPRINT, 4) II, U(I, 1), U(I+1, 1)
         22
2129
               CONTINUE
               END IF
2130
        С
2131
2132
               IF(ISYM.EO.1)THEN
               WRITE (IPRINT, 24)
2133
         24
               FORMAT(1H ,/,26X, 'VELOCITY',//,' NODE',6X, 'X',17X, 'Y',/)
2134
2135
               DO 25 I=1.NDF-2.3
               II=INT(FLOAT(I)/NF+0.7)
2136
               WRITE(IPRINT,4)II,VEL(I,1),VEL(I+1,1)
2137
         25
               CONTINUE
2138
2139
               WRITE (IPRINT, 16)
2140
               DO 26 I=1,NDF-2,3
               II=INT(FLOAT(I)/NF+0.7)
2141
2142
               WRITE(IPRINT, 4) II, U(I, 1), U(I+1, 1)
         26
2143
               CONTINUE
2144
               END IF
2145
        С
        23
2146
               CONTINUE
2147
        С
               STORE CURRENT COORDINATES
2148
               IF (PICT.EO. 'PLOT') THEN
2149
               IPLTS=IPLTS+1
2150
               DO 21 I=1, NN
               XCOPLT(I,IPLTS)=COORDX(I)
2151
2152
               YCOPLT(I, IPLTS)=COORDY(I)
2153
         21
               CONTINUE
2154
               END IF
2155
        С
2156
               RETURN
2157
               DEBUG SUBCHK
2158
               END
2159
         С
        С
2160
```
****** 2162 С 2163 С 2164 С COMPROC С 2165 С 2166 COMMON BLOCK OF ALL ARRAYS AND С 2167 PARAMETERS USED IN DAGNVS. С 2168 ****** 2169 С 2170 COMPROC PROC 2171 PARAMETER NE=5 2172 PARAMETER NN=6 2173 PARAMETER NRED=4 2174 PARAMETER NF=3 2175 PARAMETER IREAD=8 2176 PARAMETER IPRINT=5 2177 PARAMETER NDF=NN*NF 2178 COMMON/BLK1/NBEAM(NE,2),COORDX(NN),COORDY(NN), 2179 #RMASS(NN), IBC(NDF), RELEAS(NRED), COMPAT(2*NN), 2180 #UNITN(NDF,NE),SSIN(NE),CCOS(NE),FORCE(2*NN,NE,3), 2181 #FLEX(2*NN,2*NN+1),PHI(NDF,2),DISIP(2),RLOAD(NDF), 2182 #XX(2*NN),X(NRED),FORCET(NE,3),AMOM(2*NN),ANORM1(2*NN), 2183 #VMODE(NDF,2),UMODE(NDF),FORMOD(NE,3),TF,RINT, 2184 #XCOPLT(NN, 20), YCOPLT(NN, 20), XCOORD(NN), YCOORD(NN), 2185 #U2(NDF), EPSI2(NE,2), RLO(NE), ACC(NDF,2), ISTAT(2*NN), 2186 #VINIT(NDF),TIME,PARDIF(2*NN,2*NN),UNITM(NDF,NN),CL(NE), 2187 #DTTDT(2),HH,BB,VEL(NDF,2),TT(2),ANORM2(2*NN),DT, #RLAMDA,U(NDF,2),AA(2*NN,2*NN),DUMVEL(NDF) 2188 2189 COMMON/BLK2/RM0, RN0, EPSI0, EN, EN1, EN2, STADET, TITLE, 2190 #DISPL, PICT, RK, YSTRS, RMATCH, PMATCH, AMP(2), XXDUM 2191 COMMON/BLK3/IFLAG, IDISIP, ITREAC, ITMODE, NNORM, IOUT, #ICOUNT, NDIVA, IPLTS, MATCHA, INMSOL, MODES, NINV, 2192 2193 #IRND, NDIV, INEWT, MCOUNT, ISYM, IMATCH 2194 COMMON/BLK4/PARTA, PARTB, SUMA, SUMB, SUMC, SUMD, POWA, #POWB, POWC, POWD, POWE, POWF, POWG, POWH, POWI, POWJ, POWK, 2195 #POWL, PRODA, PRODB, PRODC, PART1, PART2, PART3, PART4, PART5, 2196 2197 #PART6, PROD1, PROD2, PROD3, PROD4, PROD5, PROD6, PROD7, PROD8, 2198 #CONST1, CONST2, CONST3, CONST4, CONST5, CONST6, CONST7, #SIG1,SIG2,SIG3,SIG4,SIG,SIGA,SIGB,SIGC,SIGD,SUM1,SUM2 2199 2200 CHARACTER STADET*4 CHARACTER TITLE*80 2201 2202 CHARACTER DISPL*5 2203 CHARACTER PICT*4 2204 DOUBLE PRECISION COORDX, COORDY, PARDIF, RMASS, COMPAT, 2205 #UNITM,CL,UNITN,SSIN,CCOS,FORCE,FLEX,PHI,RLOAD,VEL, 2206 #TT,ANORM1,ANORM2,RM0,RN0,EPSI0,EN,EN1,EN2,VMODE, 2207 #T,X,AMOM,DTTDT,RK,UMODE,FORMOD,XX,TIME,ACC, 2203 #AMP, ANORM, FACT, EPSI2, RLO, AA, U2, DUMVEL, U2MODE, PARTA, #PARTB, SUMA, SUMB, SUMC, SUMD, POWA, POWB, POWC, POWD, POWE, 2209 #POWF, POWG, POWH, POWI, POWJ, POWK, POWL, PRODA, PRODB, PRODC, 2210 2211 #PART1, PART2, PART3, PART4, PART5, PART6, PROD1, PROD2, PROD3, #PROD4, PROD5, PROD6, PROD7, PROD8, CONST1, CONST2, CONST3, 2212 #CONST4, CONST5, CONST6, CONST7, SIG1, SIG2, SIG3, SIG4, SIG, 2213 #SIGA, SIGB, SIGC, SIGD, SUM1, SUM2, RMATCH, PMATCH, XXDUM 2214 2215 INTEGER RELEAS 2216 END

APPENDIX D

Published Work

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

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