LARGE DEFORMATION OF RIGID–VISCOPLASTIC CANTILEVERS
SUBJECTED TO IMPULSIVE LOADING

by

R. Trossbach

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science in Engineering

September 1984
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

The problem of a ductile metal cantilever structure (not necessarily initially straight) subjected to dynamic loads leading to deformations of the order of the dimensions of the structure is considered. The material is treated as rigid-viscoplastic; in this idealisation elastic effects are ignored, and the dependence of the yield stress on the rate of strain is taken into account.

The problem is first analysed as one of impulsive loading, using the concepts of the mode approximation techniques. A new algorithm for the determination of mode shapes is presented for small displacement assumptions and then extended to incorporate geometric effects. An algorithm is given for the time integration of the motion in which the geometry of the structure is updated. Applications of the method are described for impulsive loading, and extended to a type of pipe-whip problem where the loading is a combination of an impulse and a pulse which acts in the direction of the tangent at the tip of the cantilever structure at each instant. Illustrative examples are presented which show that the algorithms can be used to give very good predictions of the displaced shape of the structures under consideration.
DECLARATION

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

R. Trossbach
September 1984
to my parents
I would like to express my gratitude to the following:

My supervisor, Professor J.B. Martin, for his encouragement and patience.

The Council of Scientific and Industrial Research for their financial assistance.

My postgraduate colleagues, particularly Mr Colin Mercer, for some fruitful discussions on general problems.

Miss Birgit Rethemeyer for her enthusiasm and the many hours spent on the word processor.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE PAGE</td>
<td>(i)</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>(ii)</td>
</tr>
<tr>
<td>DECLARATION</td>
<td>(iii)</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>(iv)</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>(v)</td>
</tr>
<tr>
<td>CONTENTS</td>
<td>(vi)</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>(viii)</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. THE MODE APPROXIMATION TECHNIQUE</td>
<td>9</td>
</tr>
<tr>
<td>2.1 The Basis of the Mode Approximation Technique</td>
<td>11</td>
</tr>
<tr>
<td>2.2 An Algorithm for the Determination of Mode Velocities</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1 Implementation of the Mode Algorithm</td>
<td>15</td>
</tr>
<tr>
<td>2.3 Generalised Momentum Balance</td>
<td>18</td>
</tr>
<tr>
<td>2.4 An Implicit Forward Integration Scheme</td>
<td>21</td>
</tr>
<tr>
<td>2.5 Estimating Time after which Structure will be at Rest</td>
<td>23</td>
</tr>
<tr>
<td>3. EXTENSION TO GEOMETRICALLY NONLINEAR CASES</td>
<td>27</td>
</tr>
<tr>
<td>3.1 Formulation of Velocity Components</td>
<td>28</td>
</tr>
<tr>
<td>3.2 The Approximation Technique and Generalised Momentum Balance</td>
<td>33</td>
</tr>
<tr>
<td>3.3 The Implicit Forward Integration Scheme</td>
<td>36</td>
</tr>
<tr>
<td>3.3.1 Modification of rate of dissipation of energy</td>
<td>38</td>
</tr>
<tr>
<td>3.4 Pulse Forces</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>COMPUTER IMPLEMENTATION OF THE MODE SOLUTION TECHNIQUE</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>5.</td>
<td>ILLUSTRATIVE EXAMPLES</td>
</tr>
<tr>
<td>6.</td>
<td>CONCLUSIONS</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
</tr>
</tbody>
</table>

**APPENDIX A**  'VISCO' User Manual  A.1 - A.12
**APPENDIX B**  Program Listing  B.1 - B.21
**APPENDIX C**  Course Work  C.1 - C.2
## NOMENCLATURE

### UPPER CASE CHARACTERS

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>pulse</td>
</tr>
<tr>
<td>$I_0$</td>
<td>initial impulse</td>
</tr>
<tr>
<td>K</td>
<td>kinetic energy</td>
</tr>
<tr>
<td>$\dot{K}$</td>
<td>time rate of change of kinetic energy</td>
</tr>
<tr>
<td>M</td>
<td>moments</td>
</tr>
<tr>
<td>$M_0$</td>
<td>yield moment</td>
</tr>
<tr>
<td>T</td>
<td>time function in mode analysis</td>
</tr>
<tr>
<td>X</td>
<td>global cartesian X-axis</td>
</tr>
<tr>
<td>Y</td>
<td>global cartesian Y-axis</td>
</tr>
<tr>
<td>V</td>
<td>local velocity</td>
</tr>
</tbody>
</table>

### LOWER CASE CHARACTERS

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>rectangular section depth</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>length of an element</td>
</tr>
<tr>
<td>n</td>
<td>power in constitutive relation</td>
</tr>
<tr>
<td>s</td>
<td>spatial variable (two-dimensional)</td>
</tr>
<tr>
<td>t</td>
<td>time variable</td>
</tr>
<tr>
<td>u,v</td>
<td>displacement components</td>
</tr>
<tr>
<td>$\dot{u},\dot{v}$</td>
<td>velocity components</td>
</tr>
<tr>
<td>$\ddot{u},\ddot{v}$</td>
<td>acceleration components</td>
</tr>
</tbody>
</table>
SUPERSCRIPTS

i, j, k  the i-th, j-th, k-th iteration
m        modal quantities
o        initial value
t        time
T        the transpose of a matrix

SUBSCRIPTS

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

MATRICES AND VECTORS

[G]      lumped mass matrix
[m]      influence matrix of nodal moments
p        load vector
u, v     displacement vectors
\dot{u}, \dot{v}  velocity vectors
\ddot{u}, \ddot{v} acceleration vectors
\phi    mode shape vector
### SPECIAL SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ]</td>
<td>a matrix</td>
</tr>
<tr>
<td>( u )</td>
<td>a vector ( u )</td>
</tr>
<tr>
<td>( \dot{v} )</td>
<td>the differential of ( v ) with respect to time</td>
</tr>
<tr>
<td>(</td>
<td>c</td>
</tr>
<tr>
<td>( d )</td>
<td>differentiation with respect to</td>
</tr>
<tr>
<td>( \partial )</td>
<td>partial differentiation with respect to</td>
</tr>
</tbody>
</table>

### GREEK CHARACTERS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \gamma )</td>
<td>specific mass</td>
</tr>
<tr>
<td>( \Delta )</td>
<td>increment in</td>
</tr>
<tr>
<td>( \dot{\varepsilon} )</td>
<td>axial strain rate</td>
</tr>
<tr>
<td>( \dot{\varepsilon}_0 )</td>
<td>strain rate material constant</td>
</tr>
<tr>
<td>( \theta )</td>
<td>rotation</td>
</tr>
<tr>
<td>( \dot{\theta} )</td>
<td>rotation rate</td>
</tr>
<tr>
<td>( \dot{k} )</td>
<td>curvature rate</td>
</tr>
<tr>
<td>( \dot{k}_0 )</td>
<td>curvature rate material constant</td>
</tr>
<tr>
<td>( \mu )</td>
<td>stress matching factor</td>
</tr>
<tr>
<td>( \nu )</td>
<td>power matching factor</td>
</tr>
<tr>
<td>( \sigma_0 )</td>
<td>yield stress</td>
</tr>
<tr>
<td>( \phi )</td>
<td>mode shape</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>scalar with units ([1/\text{time}])</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Experimental studies of impulsively loaded steel and aluminium structures have shown considerable evidence of the effects of strain rate sensitivity. In recent years increasing attention has been paid to the analysis of dynamic problems with this phenomenon included. The problem is a complex one due to materially and geometrically nonlinear behavior.

Some early analytical attempts made use of elastic-plastic constitutive laws with standard elastic mode techniques. Permanent plastic deformations were included by introducing plastic hinges. These techniques, however, were unable to incorporate large permanent deformations and were therefore limited to small impulses.

For the purpose of approximate methods, the incorporation of both elastic and plastic effects in the constitutive relation proved to be too complex and could generally be dealt with only by large finite element packages. It was recognized, however, that when a structure is subjected to large impulsive loading, the energy dissipated in plastic work far exceeds the ability of the structure to store energy elastically. Under these circumstances elastic effects could be ignored (e.g. Lee and Symonds [1], Parkes [2] and Symonds [3], [4]). For simplicity the geometric effects were assumed small. These assumptions were incorporated in what became known as the simple rigid-plastic theory. This theory gave some good insights into the problem but proved
useful only in limited applications.

The importance of including rate sensitivity in the plastic model was highlighted by, among others, Manjoine [5] and Parkes [6]. Parkes proposed a crude rate sensitive model in which the static yield stress in the rigid-plastic theory was adjusted simply by a constant factor appropriate to the average strain rate in the structure. Although improved solutions were obtained, factoring of the static yield moment did not correctly predict the pattern of plastic deformation in the structure, giving an overestimate in the case of a cantilever struck transversely at its tip.

The direct inclusion of strain rate behaviour into the constitutive relation improved analytical results to a great extent (eg. Ting and Symonds [7], Ting [8], Bodner and Symonds [9] and Bodner [10]). This rigid-viscoplastic model ignores strain hardening and was based on empirical stress - strain rate relations suggested by Manjoine [5] for mild steel, and Parkes [6] for aluminium alloys. In uniaxial form, the rigid-viscoplastic law can be written as

\[
\begin{align*}
\dot{\varepsilon} &= \dot{\varepsilon}_0 \left( \frac{\sigma}{\sigma_0} - 1 \right)^n \quad \text{for } \sigma > \sigma_0 \\
\dot{\varepsilon} &= 0 \quad \text{for } 0 < \sigma < \sigma_0
\end{align*}
\]

(1.1)

where \(\dot{\varepsilon}, \sigma\) are strain rate and stress respectively and \(\dot{\varepsilon}_0, \sigma_0 \) and \(n\) are constants. The index \(n\) is usually large, and in Manjoine's test for mild steel was about 5. Good correlation with experiments was obtained only in cases where the gross response was required. Despite the assumptions of small deflections, no elastic phase and no strain hardening, the analysis remained rather complicated and not easily generalised.
Martin and Symonds [11] developed a much simpler scheme for rigid-plastic structures. Their argument was that permanent deformations of rigid-plastic structures subjected to high intensity dynamic loading could be estimated by means of mode approximations. Assuming small displacements, they claimed that the actual velocity field could be approximated by a mode velocity field $\hat{u}^m(s,t)$ which is a separable function of space and time. Performing separation of variables, the mode velocity field can be written as

$$\hat{u}^m(s,t) = \varphi(s) T(t) \tag{1.2}$$

where $\varphi(s)$ is the mode shape and a function of space only, and $T(t)$ is a scalar function of time. They showed that actual velocities $\hat{u}(s,t)$ in an impulsive loading case with initial velocities $\hat{u}(s,0) = \hat{u}^0(s)$ converge onto a mode solution; once $\varphi(s)$ is known, the initial amplitude $T(0)$ was chosen so as to minimise a functional of the initial difference of the mode solution and the actual solution. Symonds [12] and later Bodner [10] extended the mode approximation technique to include rate sensitivity. This enabled them to successfully predict the deformations of a cantilever struck transversely at its tip.

Lee and Martin [13] attacked the problem of including rate sensitivity more formally by making direct use of the rigid-viscoplastic constitutive relation (eqn (1.1)). This equation is essentially inhomogeneous and natural mode solutions strictly do not exist, as they are not dynamically admissible. Assuming small displacements, they proposed a scheme whereby for each kinetic energy level there exists an instantaneous velocity field which satisfies the variational principle for the mode shape. The dynamic inadmissibility of the instantaneous
mode solution can be traced to the changes in shape which occur as the kinetic energy changes; mode solutions of the form of equn (1.2) do not exist and the local equilibrium conditions are not satisfied. If, however, the departure from a fixed mode shape is small, it can be expected that from an overall point of view the violations of equilibrium are small, and not likely to affect the gross behavior to any large extent. Good agreement was obtained with previous experimental and analytical results for the tip-loaded cantilever (Ting [8], Bodner and Symonds [9]). The technique, however, was not set out in a way which could be easily generalised to more complex problems.

Symonds [14] proposed a scheme whereby a rate sensitive constitutive law permitted separation of variables in order to obtain an exact mode solution. This law, called the homogeneous viscous relation, is of the form

$$\dot{\epsilon} = \dot{\epsilon}_0 \left( \frac{\sigma}{\mu \sigma_0} \right)^\nu \quad (1.3)$$

where $\mu, \nu$ are factors chosen so as to appropriately match equn (1.3) to equn (1.1). Results agreed reasonably well with results of Lee and Martin [13], but problems arose in the matching procedure.

This mode solution technique holds rigorously throughout the timespan of deformation when assuming small displacements. In many cases, however, this is not a realistic assumption as severe impulses often result in large deformations. Symonds and Chon [15] extended the technique to large displacement assumptions by means of instantaneous mode shapes, i.e.

$$\ddot{u}(s,t) = \dot{q}^T T^T \quad (1.4)$$
Here the mode shape is not only a function of space but also of time, as it is recomputed for every time step on the basis of the current geometry. The mode amplitude at the beginning of each time interval is computed in the same way as the initial amplitude for the small displacement case. The method is not exact, but for suitably chosen time steps and matching factors in the constitutive law gave good results in some structures.

Griffin and Martin [20] used both the instantaneous mode technique and a direct method of analysis based on the Tamuzh principle to estimate the response of beams and frames. For both cases they used the homogeneous viscous law (eqn (1.3)), suitably matched on the rigid-viscoplastic relation. For analyses where localised deformations are significant but where the dominant deformation pattern was modal, they combined the direct method with the mode technique: once the localised deformations were quantified by the direct method, the instantaneous mode technique was used to find the subsequent deformations. Good agreement with experimental results was obtained, provided the homogeneous viscous relation was suitably matched to the rigid-viscoplastic constitutive law. The problem was to make an unambiguous choice of the matching factors $\mu$ and $\nu$, and in some cases the matching resulted in numerical instability.

In this thesis we are concerned with a mode approximation technique whereby numerical solutions can be found for impulsively loaded structures whose material law is rigid-viscoplastic. Specifically we shall be concerned with statically determinate cantilever beam structures which lie in one plane. For the cases that will be dealt with, flexural stresses are predominant and hence, ignoring axial and shear strain rates, we rewrite eqn (1.1) as
\[ \dot{k} = \dot{k}_0 \left( \frac{M}{M_0} - 1 \right)^n \quad \text{for } M > M_0 \]  
\[ \dot{k} = 0 \quad \text{for } 0 < M_0 < M , \]

where \( \dot{k}_0 \) and \( M_0 \) are material constants with units of curvature rate and moment respectively. The beams under consideration in this thesis have a rectangular cross-section and the derivations of \( M_0 \) and \( \dot{k}_0 \) are given below.

Consider a beam of width \( b \) and height \( h \), as given in Fig 1.1(a). Assuming that bending occurs about the horizontal axis only, the strain rate is a linear function of the vertical distance from the centroid of the beam (see Fig 1.1(b)); i.e.

\[ \dot{\varepsilon} = \dot{k} y \]

Figure 1  Beam of Rectangular Cross-Section

\[ \sigma = C\epsilon \left[ 1 + \left( \frac{\dot{k} y}{\dot{\varepsilon}_0} \right)^n \right] \]

where \( \dot{k} \) is the curvature rate. Rewriting eqn (1.1) in terms of stress \( \sigma \) and substituting for strain rate \( \dot{\varepsilon} \), we obtain
\[ \sigma = \sigma_0 \left[ 1 + \left( \frac{\dot{\varepsilon}}{\xi_0} \right)^{1/n} \right], \quad (1.7) \]

which is illustrated in Fig 1.1(c).

Using elementary bending theory, the moment for the section is given by

\[ M = 2 \int_0^{h/2} b \sigma y \, dy \quad . \quad (1.8) \]

We define the yield moment \( M_0 \) as the moment resulting from a yield stress \( \sigma_0 \) across the section and write

\[ M_0 = 2 \int_0^{h/2} b \sigma y \, dy = \sigma_0 b \frac{h^2}{4} \quad . \quad (1.9) \]

In order to derive the curvature rate constant \( \kappa_0 \), we write equn (1.8) as

\[ M = 2 \int_0^{h/2} b \sigma_0 \left\{ 1 + \left( \frac{\dot{\varepsilon}}{\xi_0} \right)^{1/n} \right\} y \, dy \]

\[ = M_0 \left\{ 1 + \left( \frac{2n}{2n+1} \right) \left( \frac{\dot{\varepsilon}}{\xi_0} \right)^{1/n} \right\} \quad . \quad (1.10) \]

Reordering equn (1.10) in terms of \( \dot{\varepsilon} \), we get

\[ \dot{\varepsilon} = \frac{2 \varepsilon_0}{h} \left( \frac{2n+1}{2n} \right)^n \left( \frac{M}{M_0} - 1 \right) \quad . \quad (1.11) \]

Comparing equn (1.11) with equn (1.5), we see that

\[ \dot{\varepsilon}_0 = \frac{2 \varepsilon_0}{h} \left( \frac{2n+1}{2n} \right)^n \quad , \quad (1.12) \]

where \( \varepsilon_0 \) and \( n \) are the material constants obtained experimentally by Manjoine [5].
In Chapter 2 of this thesis, impulsive loading is described in more detail and a discretisation procedure is presented. Thereafter mode approximation techniques as they were used by Griffin [16] and Lee [13] are described in more detail, followed by a presentation of the mode approximation technique as used in this thesis. Small displacement assumptions are made and then an implicit forward integration scheme is given.

In Chapter 3, the proposed mode approximation scheme is described specifically for geometrically nonlinear cases, the relevant changes in the techniques are discussed and extensions given. Following this, the method is further extended to include pulse forces which can be used in addition to or in place of the impulsive loading.

In Chapter 4, the implementation of the analytical techniques to the computer is discussed and flow charts of the various techniques are presented.

The results of the analyses of various cantilever beam structures using the mode approximation technique are given in Chapter 5 as an illustration of the concepts put forward in this thesis. Approximate rigid-plastic results are also given and compared to analytical results.

In Appendix A, a user manual is presented for a program which was specifically written for the implementation of the proposed mode approximation technique, followed in Appendix B by a listing of this program. In Appendix C a short description is given of the coursework that was done in partial fulfilment of the degree.
CHAPTER 2

THE MODE APPROXIMATION TECHNIQUE

In this thesis we are dealing with metal beams which are subjected to large impulsive loading or a combination of impulses and pulses. The main concern is with impulses which are idealised as short duration pulses to take the form

\[ F(t) = I_0 \delta(t) \quad \text{(2.1)} \]

Integrating over the initial instant of time, we obtain

\[ \int_0^{0^+} F(t) \, dt = I_0 \int_0^{0^+} \delta(t) \, dt = I_0 \quad \text{(2.2)} \]

where \( I_0 \) is the impulse applied at time \( t=0 \) and \( \delta(t) \) is a Dirac Delta function. The impulse \( I_0 \) imparts an initial velocity to the structure. It is assumed for the impulsive loading case that no further external loads are applied to the structure after time \( t=0 \); the initial conditions are thus fully defined by some initial velocity field \( \dot{u}(s,0) \) and zero external loads.

The beams under consideration are statically determinate cantilever beam structures with uniform rectangular cross-section. In order to perform numerical analyses, the beams are discretised by nodes which are defined along the centre line of the structure. Adjacent nodes are connected by straight massless elements which are assumed to be able to transmit bending moment, axial and shear force from one node to
another. Shear and axial strain rates are ignored and thus the only generalised stress associated with deformation is the bending moment $M$ which varies linearly across the elements. The curvature rate $\dot{\kappa}$ is related to $M$ through the constitutive relation given by eqn (1.5). The velocities of the nodes are the kinematic variables and it will be assumed that the velocity of the constrained (support) node is zero. Rotation rates are not used as kinematic variables.

Instead of using the conventional finite element method, a force method type of approach is employed together with the principle of virtual velocities. Massless elements imply that the bending moment should vary linearly between the nodes. If, however, the usual cubic interpolation function for transverse velocities is used together with the rigid-viscoplastic relation, the bending moment will be nonlinear between nodes. Alternatively, if linear variation of moments is assumed, the interpolation function for transverse velocities cannot be explicitly computed. An important consideration in the solution procedures to be presented in this thesis is that the interpolation function for the velocity field across an element will not be explicitly defined.

Mass is lumped at the nodes in the conventional way (Newmark [17]). A diagonal mass matrix is defined in such a way that the kinetic energy at any instant is given by

$$ K = \frac{1}{2} \ddot{\mathbf{u}}^T \mathbf{G} \ddot{\mathbf{u}} $$

The mass term corresponding to the constrained node can be arbitrarily defined.

An implicit assumption in any mode approximation technique is that final deformations are predominantly of the modal type and that any
localised, non-modal response which occurs contributes negligibly to the overall behaviour of the structure. In this thesis we limit ourselves to problems where the response is predominantly modal such as the tip-loaded cantilever case, and we make direct use of the rigid-viscoplastic constitutive relation given by equn (1.5).

In the following section the basis of the mode approximation technique is described.

2.1. The Basis of the Mode Approximation Technique

For geometrically linear problems and for a homogeneous viscous material (including a rigid, perfectly plastic idealisation) the mode approximation technique is based on the existence of solutions of the form

\[ \ddot{u}^m (s,t) = \phi(s) T(t) \]  

(2.4)

This implies that the acceleration field has the same spatial distribution as the velocity field, since one can write, from equn (2.4),

\[ \ddot{u}^m = -\lambda \dot{u}^m \]  

(2.5)

where

\[ \lambda = -\frac{T}{T} \]  

(2.6)
The implementation of the technique requires that algorithms should exist for the determination of the mode shape \( \phi(s) \) and the time function \( T(t) \).

The primary mode shape, which we shall be concerned with in this study, is governed by a minimum principle (see for example Martin [19], Griffin and Martin [16]), and is that velocity among a class of fields which all have the same kinetic energy \( K \) and which maximises the rate of change of kinetic energy \( K \).

The determination of the primary mode shape can be achieved by the principle that, if a structure is loaded by \( \lambda [G] \dot{u}^m \), the resulting velocities are the mode velocities \( \dot{u}^m \) (see Martin [18]). Griffin and Martin [16] made use of this technique in the determination of the mode shape for homogeneous viscous material. For the latter case, the shape function \( \phi \) is independent of the kinetic energy \( K \) and the time function \( T(t) \) respectively, and an arbitrary level of the initial kinetic energy is used for the determination of \( \phi \).

In contrast to the rigid-viscoplastic case, the homogeneous viscous case thus permits the determination of the mode shape \( \phi \) and \( \lambda \) to be performed in two independent calculations. The mode shape can be determined by loading the structure with \( [G] \dot{\phi} \) which result in velocities \( \dot{\gamma} \); these velocities are scaled by some parameter to give a new trial mode shape which in turn is used for a new loading until convergence has occurred on the mode shape. Once the mode shape \( \phi \) has been determined, the right hand side of equn (2.6) can be found, giving \( \lambda \). It should be re-emphasized at this point that for the homogeneous case, \( \lambda \) is not included in the calculation of the mode shape.

In the rigid-viscoplastic case, however, the mode velocities are dependent on the instantaneous kinetic energy and can be written as
\[ \ddot{u}^m(s,t) = \phi(s,K(t)) \]  

For this class of problems, the mode shape \( \phi \) and \( \lambda \) cannot be found independently as both are functions of the kinetic energy. The time rate of change of the kinetic energy is of the form

\[ \dot{K} = u^T [G] u \]  

and substituting for the accelerations from equn (2.5), equn (2.8) can be rewritten as

\[ \dot{K} = -2 \lambda K \]  

where

\[ \lambda = \lambda(K) \]  

In the following section we present an algorithm whereby \( \lambda \) and \( \phi \) can be determined for a given value of \( K \). We use the value of \( \lambda \) so obtained in equn (2.9) to integrate forward in time.

2.2. An Algorithm for the Determination of Mode Velocities

In the algorithm given below we present a new technique in which for a given kinetic energy level velocities can be found which satisfy the variational principle for the instantaneous mode. The determination of the instantaneous mode velocities is achieved by enforcing equn (2.5) in an iterative procedure.
The algorithm consists of a pair of nested loops which in turn are used to evaluate $\lambda$ and $\dot{\mathbf{u}}$. The inner loop iterates on $\lambda$ to assure that the given kinetic energy level is retained while the outer loop iterates on the velocities. Iteration is continued until a state is reached in which a load of $\lambda[G]\dot{\mathbf{u}}$ applied to the structure results in a velocity field $\dot{\mathbf{u}}$ which is identical to that used in the load. This assures that eqn (2.5) has been satisfied.

The steps in this algorithm in principle apply to both geometrically linear and nonlinear cases and are given below.

The kinetic energy for which instantaneous mode velocities are to be found is $K^*$.  

Step 0: Select a trial value for $\lambda$ and guess a set of velocities, denoted by $\lambda^k$ and $\dot{\mathbf{u}}^k$ respectively.

Step 1: Apply loads $\lambda^k[G]\dot{\mathbf{u}}^k$ to the structure and determine resulting bending moments with corresponding curvature rates.

Step 2: Compute velocities $\dot{\mathbf{u}}^{i+1}$ resulting from above loading.

Step 3: Determine the kinetic energy $K^{i+1}$ corresponding to the velocities $\dot{\mathbf{u}}^{i+1}$ and check $K^{i+1}$ against $K^*$.  

Step 4: If $K^{i+1}$ is not close to $K^*$, return to step 1 with a new estimate of $\lambda$, replacing $\lambda^k$ by $\lambda^{k+1}$, say, and repeat steps 1, 2 and 3 for $\lambda^{k+1}$. Perform a
series of iterations on this loop until $K^{i+1}$ is close to $K^*$.  

Step 5: Return to step 1, replacing $\mathbf{u}_2^i$ by the latest set of $\mathbf{u}_2^{i+1}$ determined from step 4.

Iterations are continued until a satisfactory convergence on the velocities has been attained assuring that eqn (2.5) is satisfied for the energy level $K^*$. Numerically, convergence is rapid, requiring only one or two iterations of the outer loop. The algorithm has been successfully applied to the tip-loaded cantilever case, comparing very favourably with experimental results.

The following subsection provides detail information of the solution procedures for the above algorithm.

2.2.1. Implementation of the Mode Algorithm

In this subsection details are given of the mode algorithm as they are applicable to small displacement assumptions. The extension to geometrically nonlinear problems is given in Chapter 3.

In finding rigid-viscoplastic solutions, the total response time is divided up into suitable time intervals $\Delta t$. Suppose that for a certain time step $t-\Delta t$ the mode velocities and corresponding kinetic energy are known. A forward integration technique (described in Section 2.4) is then used to predict the kinetic energy $K^*$, say, at time $t$ for which the mode velocities have to be computed.
In Step 1 of the algorithm loads are applied to the structure in the form of

\[ p = \lambda^k [G] \dot{u} \]

(2.11)

where \([G]\) is the diagonal mass matrix and the first estimates of \(\lambda^k\) and \(\dot{u}^1\) are those of the previous time step. In order to determine the resulting bending moments, we set up an influence matrix \([m]\) whose columns are sets of nodal moments due to unit loads. We can write the nodal moments as follows:

\[ M = [m] \sim p \]

(2.12)

where \(p\) is the load vector given by eqn (2.11). The bending moments are distributed linearly across elements and can thus be easily determined. If \(a, b\) are adjacent nodes separated by distance \(\lambda_e\), and \(M_a, M_b\) are the nodal moments, the bending moment distance \(s\) from node \(a\) is given by

\[ M(s) = M_a (1 - \frac{s}{\lambda_e}) + M_b \left(\frac{s}{\lambda_e}\right) \]

(2.13)

Using these relations, we can define the bending moment \(m_j\) along each element resulting from a unit value of the \(j\)-th component \(p_j\) of the load vector \(p\). Using the constitutive equn (1.5) we can determine the curvature rate \(\dot{\kappa}\) at each point on the structure resulting from a bending moment \(M(s)\).

In Step 2 the principle of virtual velocities is used to determine
the velocities resulting from the loading $\mathcal{P}$. With the velocities $u_j$ and curvature rates $\dot{k}$ as the kinematic system, and a unit value of the $j$-th component of $\mathcal{P}$ together with its associated $m_j$ as the static system, the principle of virtual velocities gives the $j$-th component $\dot{u}_j$ of $\dot{u}$ as

$$\dot{u}_j = \sum_{\text{elements}} \int_{l_e} m_j k \, ds,$$  \hspace{1cm} (2.14)

where $l_e$ is the length of an element. This principle is applied repeatedly to obtain the velocities for all the unconstrained nodes, giving $\dot{u}_{i+1}$, say. In Step 3 the kinetic energy corresponding to $u_{i+1}$ is evaluated.

In Step 4 we perform a standard bisection algorithm, the parameters of which are $\lambda$ and kinetic energy $K$. Before the bisection can proceed, however, a pair of $\lambda$ values must be available for which the sets of computed velocities $\dot{u}_{i+1}^1$ and $\dot{u}_{i+1}^2$ say, result in kinetic energy levels which are bigger and smaller than $K^*$ respectively. The bisection algorithm is started by changing $\lambda^k$, evaluating new loads resulting in new velocities with corresponding energy $K$. Velocities $\dot{u}_{i+1}^k$ are recomputed for every new $\lambda^k$ until resulting kinetic energy is close to $K^*$.

Once Step 4 has been completed, we return to Step 1 of the algorithm and again we evaluate new loads: to compute the loads we use the latest values of $u_{i+1}^k$ and $\lambda^{k+1}$ from Step 4, giving $\lambda^{k+1}[G]u_{i+1}^k$. The algorithm is repeated until the velocities have converged onto a set for which equn (2.5) holds true and for which the corresponding energy is $K^*$. This concludes the determination of instantaneous mode velocities for a specific kinetic energy level.
Note that the algorithm entails solving a static problem and that the dynamic aspect of the solution procedure lies with a forward integration technique which will be described in Section 2.4.

In the next section the initial conditions for the mode algorithm are outlined and an algorithm given for a generalised momentum balance.

2.3 Generalised Momentum Balance

Before proceeding with the rigid-viscoplastic analysis, two independent momentum balances have to be performed. The first one is done to obtain an exact initial velocity field for the discretised structure and the second momentum balance is done to obtain an equivalent mode velocity field.

The first generalised momentum balance is demonstrated for the case of the tip-loaded cantilever. Fig 2(a) shows the actual beam with tip mass $m_{tip}$ and specific mass $\gamma(s)$. Suppose an impulse $I$ is applied to the tip at time $t = 0$. This results in an initial transverse tip velocity $u_a$ while the remainder of the beam is initially stationary. The initial velocity $u_a$ is given by

$$u_a = \frac{I}{m_{tip}}.$$

Fig 2(b) shows a three element lump mass model. Note that part of the third element is lumped at node 4 with the tip mass. The initial velocity $u_a$ is adjusted so that the initial momenta for the actual beam
and the discretised beam are identical, i.e.

\[ \ddot{u}^0 = \frac{m_{\text{tip}} \dot{u}_a}{G_4} \]

This adjustment is done manually and the velocity \( \dot{u}^0 \) is used as input for the computer program.

\[ \delta(s) \]

(a) actual system

\[ m_{\text{tip}} \]

(b) three element model

Figure 2.1 Cantilever beam example

The second and main generalised momentum balance is part of the instantaneous mode technique. An inherent assumption in the mode approximation technique, however, is that the velocities at any instant satisfy or at least are forced to satisfy equn (2.5) so that the variational principle for the mode holds true; the latter also applies to the initial velocities.
We refer to Fig 2.1(b) as an example. Initially only node 4 has a velocity whereas the other two nodes are stationary. This velocity field is clearly non-modal. In the initial phase of the response, however, there are very rapid changes in the velocities until they converge onto mode velocities. This transient phase is of very short duration compared to the total response time, and we assume that this transient phase can be ignored. We use the instantaneous mode technique and incorporate a generalised momentum balance into it to get an approximation of the "initial" mode velocities. We seek a balance in the form of

\[ \dot{\mathbf{u}}^T (\mathbf{G}) \dot{\mathbf{u}}^0 = \dot{\mathbf{u}}^T (\mathbf{G}) \dot{\mathbf{u}} \]  

(2.15)

where \( [G] \), \( \dot{\mathbf{u}} \) and \( \dot{\mathbf{u}}^0 \) are the diagonal mass matrix and the sets of the mode velocities and the initial velocities respectively.

This generalised momentum balance is the first major step in the rigid-viscoplastic solution and provides the mode velocities for time \( t = 0 \). The algorithm is presented below.

Step 1: Evaluate initial kinetic energy \( K^0 \) corresponding to the set of initial velocities \( \dot{\mathbf{u}}^0 \).

Step 2: Calculate velocities \( \dot{\mathbf{u}}_A \) from mode algorithm in Section 2.2.

Step 3: Compute \( \xi = \frac{\dot{\mathbf{u}}^T [G] \dot{\mathbf{u}}^0}{\dot{\mathbf{u}}^T [G] \dot{\mathbf{u}}_A} \) which represents the out-of-balance condition.
Step 4: Multiply velocities $\dot{u}_i$ from step 2 by $\xi$ to obtain scaled velocities $\dot{u}_{i+1}'$.

Step 5: Compute kinetic energy $K_{i+1}$ corresponding to the scaled velocities $\dot{u}_{i+1}'$.

Step 6: Return to step 2 and evaluate mode velocities for kinetic energy $K_{i+1}$.

This iteration is continued until $\xi$ has converged to unity implying that the generalised momentum balance has been achieved. Convergence is rapid and in the case of the tip-loaded cantilever takes three to four iterations.

In the next section a scheme is presented by means of which the solution can be carried forward in time.

2.4 An Implicit Forward Integration Scheme

In order to march the solution forward in time, a suitable integration scheme has to be implemented. In rigid-viscoplastic solutions the instantaneous mode shape is dependent on the level of the kinetic energy and thus we use the independent parameter in the form of the kinetic energy to achieve this aim. For this purpose, however, the time rate of change of the kinetic energy needs to be evaluated. If the kinetic energy is given by

$$K = \frac{1}{2} \dot{u}_T [G] \dot{u}_T,$$
then its derivative with respect to time has the form

\[ \dot{K} = \frac{dK}{dt} = \dot{\ddot{u}} [G] \ddot{u} \quad , \]  

(2.16)

where \( \ddot{u} \) is the vector of nodal accelerations.

Replacing \( \ddot{u} \) by \(-\lambda \dot{u}\), we rewrite \( K \) as

\[ \dot{K} = \dot{\ddot{u}} [G] (-\lambda \dot{u}) \]

\[ = -2 \lambda K \quad . \]  

(2.17)

From equn (2.17) it can be seen that the rate of change of the energy is a function of that energy and of \( \lambda \), which implies that \( K \) can be evaluated once \( K \) and its corresponding mode velocities together with \( \lambda \) is known.

In order to do the forward integration, we could use the Euler method

\[ K_{t+1} = K_{t} + K \Delta t \quad , \]  

(2.18)

where the superscripts \( t+1 \) and \( t \) denote instants in time and \( \Delta t \) denotes a time increment. The procedure would be numerically stable but since the shape of the energy curve is essentially parabolic, errors would propagate in time resulting in an underestimate of \( K_{t+1} \).

In order to improve estimates of \( K_{t+1} \), we use the improved Euler method which is a predictor-corrector type scheme with average rate of change given by

\[ K_{t+1} = K_{t} + \frac{\Delta t}{2} (\dot{K}_{t} + \dot{K}_{t+1}) \quad . \]  

(2.19)
In applying this equation we assume that $K^t$ and $K$ are known. This is not sufficient information to compute $K^{t+1}$, since $\mathbf{K}^{t+1}$ is a function of $K^{t+1}$. Hence an iterative scheme must be used and if subscript $i$ indicated the $i$-th iteration, we put

$$K_i^{t+1} = K^t + \frac{\Delta t}{2} (\mathbf{K}_i^t + \mathbf{K}_i^{t+1}) .$$  \hspace{1cm} (2.20)

The initial value of $\mathbf{K}_i^{t+1}$ is taken as $K^t$, whereafter the velocities for $K_i^{t+1}$, are computed and $K_i^{t+1}$ can be evaluated. Equn (2.20) is used again for obtaining a new kinetic energy estimate and the process is repeated until satisfactory convergence in $K^{t+1}$ has occurred. Numerical trials have indicated that convergence is fairly rapid under normal circumstances, requiring five to six iterations to obtain satisfactory convergence.

In the following section a method is derived which enables us to find the final time of the response.

2.5 Estimating time after which structure will be at rest

As the rigid-viscoplastic solution is integrated forward in time, the kinetic energy will drop parabolically and as the mode velocities decrease the kinetic energy approaches zero with zero slope. This last section in the response makes it very difficult for the forward integration procedure as the kinetic energy is easily overshot to a negative value.

In order to overcome the problem, two conditions were set up to cut off the analysis and from which an estimate of the remaining time
increment $\Delta t_f$ was done. Forward integration is stopped when one of the following conditions becomes true:

\[
\begin{align*}
(a) \quad & K < \varepsilon_K \\
(b) \quad & K < K_{\text{fit}} \\
\end{align*}
\]  

$\varepsilon_K$ is a set tolerance and is usually taken as $10^{-2} K^0$. Condition (b) is set up to prevent the kinetic energy estimate from overshooting the zero level. Once one of the two conditions is met, the analysis is halted and an extrapolation on the kinetic energy is done to establish the time increment $\Delta t_f$.

Suppose an energy level $K^*$ satisfies one of the two conditions above and corresponds to an instant in time $t^*$ and we wish to extrapolate (see Fig 2.2).

![Figure 2.2 Extrapolation of Kinetic Energy Curve](image-url)
The shape of the energy-time curve is nearly parabolic and thus we assume that for \( t > t^* \)

\[
K = -\frac{1}{2}at^2 + bt + c
\]

\[
K = -at + b
\]  \hspace{1cm} \text{(2.22)}

We have four unknowns \( a, b, c \) and \( t_f \) and the essential four conditions are

\[
K(t^*) = K^*
\]
\[
K(t_f^*) = K^*
\]
\[
K(t_f) = 0
\]
\[
K(t_f) = 0
\]  \hspace{1cm} \text{(2.23)}

Substituting the conditions (2.23) into eqns (2.22), we obtain the following results:

\[
a = \frac{K^*}{\Delta t_f}, \quad b = \frac{K^*}{\Delta t_f} t_f, \quad c = -\frac{1}{2} \frac{K^*}{\Delta t_f} t_f^2
\]

and finally

\[
\Delta t_f = -\frac{2K^*}{K^*}
\]  \hspace{1cm} \text{(2.24)}

Once the kinetic energy is low enough to satisfy either of conditions (2.21), equn (2.24) is used to evaluate \( \Delta t_f \) and hence \( t_f \) can be found, giving approximately the time after which the structure has come to a rest.

For small displacement assumptions, displacement increments do not need to be updated after every time step, as all parameters are computed.
according to the original configuration. The calculation of displacements was done by integrating the velocity - time curves and using either Simpson's or the trapezoidal rule.

In the next chapter a formulation for the geometrically nonlinear case is presented together with all the necessary adjustments that have to be done on the algorithms.
In the last chapter a rigid-viscoplastic solution method was described using small displacement assumptions. Stresses and strain rates were evaluated according to the initial configuration of the structure and the resulting velocity vector at any point on the structure was assumed to keep its direction throughout the response. For impulsive loading cases, however, where the deformation of the structure is of the same order of magnitude as its largest dimension, small displacement assumptions are inappropriate; nevertheless they give a good insight into the problem and provide a good base for more complex formulations.

A fundamental concept in geometrically nonlinear formulations is that the structural geometry is continuously updated during the response and that all parameters are evaluated according to the present or instantaneous configuration. Another important point is that as the geometry changes the velocity vector at any point on the structure changes direction, necessitating appropriate extensions to the geometrically linear formulation.

In the following section a formulation is given for geometrically nonlinear problems and the subsequent adjustments that have to be made to the algorithms.
3.1 Formulation of Velocity Components

An important assumption in our present formulation of velocity components for the geometrically nonlinear case is that axial deformations for the examples under consideration are negligible. Hence we put a constraint on the axial strain rate, i.e.

\[ \dot{\varepsilon} = 0 \quad , \quad (3.1a) \]

which implies that

\[ \dot{l} = 0 \quad , \quad (3.1b) \]

where \( l \) is the length of any element.

Figure 3.1 Cartesian Coordinate System

We define \( \dot{u}, \dot{v} \) as the velocity components in the positive \( x \) and \( y \) directions of a Cartesian co-ordinate system; similarly the
displacement components $u$ and $v$ are defined. (see Fig 3.1).

![Diagram](image)

**Figure 3.2 1 DOF System**

Imagine a 1-DOF system where the mass is lumped at the end of a bar of length $l$ and which is rotated around the opposite end of the bar, as shown in Fig. (3.2). By using Pythagoras, we can write

$$l^2 = v^2 + (l-u)^2 \tag{3.2}$$

Differentiating with respect to time, we obtain

$$\dot{l}l = 2v\dot{v} + 2(l-u)\dot{(l-u)} \tag{3.3}$$

in order to satisfy the assumption made earlier that axial strain rates are neglected, we solve equn (3.3) subject to the constraint (3.1b) to obtain

$$\dot{u} = \dot{v} \cdot \frac{-v}{l-u} = -\dot{v} \tan \theta$$
Multiplying out by \( \cos \theta \), we derive an equation which assures that the length of the bar remains constant:

\[
\dot{u} \cos \theta + \dot{v} \sin \theta = 0 \quad . \tag{3.4}
\]

Now let us define an absolute velocity \( V \) which is transverse to the bar. In this case the absolute velocity is simply related to components \( \dot{u} \) and \( \dot{v} \) as

\[
\dot{u}^2 + \dot{v}^2 = V^2 \quad . \tag{3.5}
\]

Using the last two equations simultaneously we solve for \( \dot{u} \) and \( \dot{v} \) to obtain

\[
\dot{u} = -V \sin \theta \quad . \tag{3.6a}
\]
\[
\dot{v} = V \cos \theta \quad . \tag{3.6b}
\]

Differentiating eqns (3.6) with respect to time, we have the acceleration components as

\[
\ddot{u} = -V \sin \theta - \dot{\theta} V \cos \theta \quad , \tag{3.7}
\]
\[
\ddot{v} = V \cos \theta - \dot{\theta} V \sin \theta \quad ,
\]

where \( \dot{\theta} \) is the rotation rate and is defined as

\[
\dot{\theta} = \frac{V}{\lambda} \quad . \tag{3.8}
\]
Substituting the latter into eqns (3.7) we get

\[ \ddot{u} = -\dot{V} \sin \theta - \frac{V^2}{\lambda} \cos \theta \]
\[ \ddot{v} = \dot{V} \cos \theta - \frac{V^2}{\lambda} \sin \theta \quad \text{(3.9)} \]

where \( \dot{V} \) is the absolute acceleration and due to the earlier definition of \( \lambda \), we can write

\[ \dot{V} = -\lambda V \quad \text{(3.10)} \]

and substitute it into eqns (3.9) to get

\[ \ddot{u} = \lambda V \sin \theta - \frac{V^2}{\lambda} \cos \theta \]
\[ \ddot{v} = -\lambda V \cos \theta - \frac{V^2}{\lambda} \sin \theta \quad \text{(3.11)} \]

Looking again at eqns (3.7) it should be noted that the right hand sides contain derivatives of both \( V \) and \( \theta \). Whereas the term involving \( V \) can be seen as an ordinary "velocity rate" expression, the term containing \( \theta \) can be viewed as the acceleration due to change in geometry.

We now proceed to derive velocity acceleration equations for a multi-degree-of-freedom system as shown in Fig 3.3.
We redefine \( V \) as the set of local velocities such that \( V_j \) is the velocity of a node \( j \) if the frame of reference is moving with node \( i \). This is illustrated in Fig 3.4. The element \( j \) with nodes \( i \) and \( j \) is taken as one element of Fig 3.3, and due to the axial constraint, \( V_j \) is acting transverse to the element.

![Figure 3.4 Element j](image)

We can write the constraint equation as follows,

\[
2l_j \ddot{x}_j = (\ddot{v}_j - \ddot{v}_1) \cos \theta_j + (\ddot{u}_j - \ddot{u}_1) \sin \theta = 0 \quad (3.12)
\]

and

\[
(\ddot{u}_j - \ddot{u}_1)^2 + (\ddot{v}_j - \ddot{v}_1)^2 = V_j^2 \quad (3.13)
\]

Solving eqns (3.12) and (3.13) simultaneously, we obtain

\[
\ddot{u}_j = \ddot{u}_1 - V_j \sin \theta_j \quad (3.14a)
\]

\[
\ddot{v}_j = \ddot{v}_1 + V_j \cos \theta_j \quad (3.14b)
\]
The rotation rate $\dot{\theta}_j$ can be expressed as

$$\dot{\theta}_j = \frac{\dot{V}_j - \dot{V}_i}{\dot{X}_j \cos \theta_j} = \frac{V_j}{\dot{X}_j}$$

(3.15)

and consequently differentiating eqns (3.14) with respect to time and substituting eqns (3.15) and (3.10), we write

$$\ddot{u}_j = \ddot{u}_i + \lambda V_j \sin \theta_j - \frac{V_j^2}{\dot{X}_j} \cos \theta_j$$

(3.16a)

$$\ddot{v}_j = \ddot{v}_i - \lambda V_j \cos \theta_j - \frac{V_j^2}{\dot{X}_j} \sin \theta$$

(3.16b)

Equations (3.14) and (3.16) are the relevant velocity and acceleration equations used in the formulation of geometric nonlinearity. Once the local velocities $V_j$ have been found, the velocity and acceleration components can be evaluated. The method of finding the local velocities and the effect of this new formulation on the algorithms is described in the next section.

3.2 The Approximation Technique and Generalised Momentum Balance

Although the steps for the instantaneous mode technique as given in Section 2.2 apply in principle also to the geometrically nonlinear case, the numerical implementation is different as velocities are now defined in terms of $X$ and $Y$ components.
We define the global or absolute velocity of a node \( j \) as

\[
\hat{v}_j = (\dot{u}_j^2 + \dot{v}_j^2)^{1/2}
\]  \hspace{1cm} (3.17)

The kinetic energy is then given by

\[
K = \frac{1}{2} \hat{v}^T \dot{v}
\]  \hspace{1cm} (3.18)

The load vector \( \mathbf{P} \) is divided up into a horizontal and vertical component respectively, giving

\[
\begin{align*}
P_x &= \lambda \mathbf{G} \hat{u} \\
P_y &= \lambda \mathbf{G} \hat{v}
\end{align*}
\]  \hspace{1cm} (3.19)

and consequently the influence matrix \( [m] \) as defined in Chapter 2 also has to be extended. We set up an influence matrix \( [m_x] \) whose columns are sets of nodal moments due to unit vertical loads \( P_y \), and similarly a matrix \( [m_y] \) whose columns are sets of nodal moments due to unit horizontal loads \( P_x \). Subsequently the nodal moments can be written as

\[
\mathbf{m} = [m_x] P_y + [m_y] P_x
\]  \hspace{1cm} (3.20)

and the moments across the elements can still be found by equn (2.13).

In the determination of velocities by the principle of virtual velocities a different approach is used. We define a set of dummy loads \( \mathbf{X} \), whose components \( X_j \) act transversely to elements \( j \) as shown in Fig 3.5.
Subsequently we define a new influence matrix \([m_t]\) whose \(j\)-th column, say, is a set of nodal moments resulting from a unit value of the component \(X_j\) of \(\vec{X}\). As an example we take the influence matrix \([m_t]\) for the structure in Fig 3.5. The last column would be made up of a set of nodal moments due to a unit value of \(X_3\).

The principle of virtual velocity can now be written as

\[
V_j' = \sum_{\text{elements}} \int_{\lambda_e} m_{t,j} \dot{\lambda} \, ds \quad ,
\]

(3.21)

where \(m_{t,j}\) is the \(j\)-th column of the influence matrix \([m_t]\). Due to the definition of the matrix \([m_t]\), the computed velocities \(V_j\) are then given by

\[
V_j = V_j + \sum_{i=1}^{j-1} V_i \cos(\theta_j - \theta_i) \quad ,
\]

(3.22)
where \( V_j \) is the local velocity at node \( j \). Once the complete vector \( V' \) has been computed, start at node 2 to evaluate the local velocities using equn (3.22) and solving for \( V_j \). For node 1 we have

\[
V_1 = V'_1
\]

(3.23)

and thus the reason for starting at node 2 becomes obvious. The evaluation of the velocity components \( \dot{u} \) and \( \dot{v} \) can now proceed using equns (3.14) and starting at node 1.

The generalised momentum balance in the form of equn (2.11) is extended to both horizontal and vertical velocity components and using the global velocities \( \vec{V} \), we write

\[
\vec{V}^T[G] \vec{V} = \vec{V}^T[G] \vec{V}
\]

(3.24)

where \( \vec{V}^T \) is the transpose of the vector of global velocities and \( \vec{V}^0 \) is the set of the global initial velocities.

In the next section we proceed with the adaptation of the forward integration scheme and introduce a procedure which essentially makes the formulation geometrically nonlinear.

3.3. The Implicit Forward Integration Scheme

In the geometrically linear case no update of the geometry was done during the response and moments and curvature rates were evaluated according to the initial configuration of the structure. In order to step forward in time, an implicit integration was performed whereby the
kinetic energy at the next time step was found by an iterative scheme.

In the geometrically nonlinear case we use this same forward integration scheme for the kinetic energy, given by

\[ k_{i+1} = k_i + \frac{\Delta t}{2} (\dot{k}_i + \dot{k}_{i+1}) \]  

(3.25)

In addition we wish to update the geometry of the structure in order to account for geometric nonlinearity. We use the improved Euler method in the form

\[ u_{i+1}^t = u_i^t + \frac{\Delta t}{2} (\ddot{u}_i^t + \ddot{u}_{i+1}^t) \]  

(3.26a)

\[ v_{i+1}^t = v_i^t + \frac{\Delta t}{2} (\ddot{v}_i^t + \ddot{v}_{i+1}^t) \]  

(3.26b)

In applying eqns (3.26) we assume that \( \ddot{u}_i^t, \ddot{v}_i^t \) and \( u_i^t, v_i^t \) are known. This is not sufficient information to compute \( \ddot{u}_{i+1}^t, \ddot{v}_{i+1}^t \) from eqns (3.26), however, and an iterative scheme must be used. We put

\[ u_{i+1}^t = u_i^t + \frac{\Delta t}{2} (\ddot{u}_i^t + \ddot{u}_{i+1}^t) \]  

(3.27a)

\[ v_{i+1}^t = v_i^t + \frac{\Delta t}{2} (\ddot{v}_i^t + \ddot{v}_{i+1}^t) \]  

(3.27b)

where subscript \( i \) indicated the \( i \)-th iteration.
Eqns (3.25) and (3.27) are now applied alternatively in the following way. Suppose we know all parameters at time $t$ and wish to get a first estimate of the displacements and kinetic energy at time $t+1$. We take $K^t$, $\dot{u}^t$ and $\dot{v}^t$ as the initial values for $K^{t+1}$, $\dot{u}^{t+1}$ and $\dot{v}^{t+1}$; thereafter we compute the mode velocities for $K_{1}^{t+1}$ and $u_{1}^{t+1}$, $v_{1}^{t+1}$. Subsequently $K_{1}^{t+1}$ can be evaluated and substituted into equn (3.25) to obtain $K_{2}^{t+1}$. Similarly a better estimate for the displacements are found by evaluating $u_{2}^{t+1}$ and $v_{2}^{t+1}$. This process is repeated until satisfactory convergence in the values of $K^{t+1}$ and $u_{1}^{t+1}$, $v_{1}^{t+1}$ is obtained.

This new procedure thus entails integrating the kinetic energy forward in time, updating the geometry and performing the instantaneous mode technique for the kinetic energy $K_{1}^{t+1}$ and the new configuration. Convergence is fairly rapid, requiring five to eight iterations.

3.3.1 Modification of rate of dissipation of energy

In the case where displacements are assumed small, we derived the time rate of change of the kinetic energy as a function of velocities and accelerations; since for small displacement assumptions the accelerations are given by $-\lambda \ddot{u}$, $K$ could be written as

$$K = -2 \lambda \dot{K} \quad .$$  \hspace{1cm} (3.28)

In Section 3.1, however, we derived expressions for the acceleration components for the geometrically nonlinear case, given by
equns (3.16); these accelerations are not only a function of $\lambda$ and the velocities, but also contain a term involving rotation rate $\theta$ (see equns (3.7)). Hence we also expect the rate of dissipation of energy to be a function of rotation rates.

Rewriting equn (3.18) in terms of velocity components $\dot{u}$ and $\dot{v}$, we get

$$K = \frac{1}{2} (\dot{u}^T [G] \dot{u} + \dot{v}^T [G] \dot{v}),$$

(3.29)

and taking the first derivative of equn (3.29) with respect to time, we obtain


(3.30)

After substituting equns (3.16) and (3.14) into equn (3.30), we can derive an expression for $K$ given by

$$K = -2\lambda K + \sum_{j=2}^{n} G_j \left\{ \sum_{i=1}^{j-1} (\dot{u}_i \dot{v}_j - \dot{u}_j \dot{v}_i) \left( \frac{V_i}{x_i} - \frac{V_{i+1}}{x_{i+1}} \right) \right\},$$

(3.31)

where $G_j$ is the $j$-th term in the diagonal mass matrix $[G]$.

It should be noted that the first summation sign in equn (3.31) starts with $j = 2$, i.e. the second node, since there is no contribution to the correction term from the first node.
3.4 Pulse Forces

The instantaneous mode approximation technique as outlined in the previous sections was based on the assumption that an impulsive load was applied to the structure at time $t = 0$, resulting in an initial velocity field. It was also assumed that after time $t = 0$ no external loads are applied to the structure.

Let us now introduce the concept of pulse forces of the form

$$F_e = F(t),$$

(3.32)

where $F(t)$ is of limited time duration. Pulse loads are associated with shocks due to impact of collision, or air or fluid induced pressure waves due to explosive detonation. A typical curve is shown in Fig 3.6.

![Figure 3.6 Typical Pulse Load](image)
For simplicity, we have limited attention to pulse loads which are applied at nodes either tangentially (i.e. typical "pipe whip" problem), or transversely. Figs 3.7(a) and 3.7(b) are examples of the former and latter application. In the geometrically nonlinear case the load directions follow the deformation of the structure.

\[ P_x = \lambda[G] \ddot{y} - F(t) \cos \theta \]
\[ P_y = \lambda[G] \ddot{x} - F(t) \sin \theta \]  

Figure 3.7  Pulses applied at cantilever tip

These pulse loads can be easily incorporated by adding their values to the load vectors. We distinguish between the tangentially and transversely applied pulses.

(1) Tangentially applied pulses:
(ii) Transversely applied pulses:

\[
P_x = \lambda[G]\ddot{u} - F(t) \sin \theta
\]
\[
P_y = \lambda[G]\ddot{v} + F(t) \cos \theta
\]  \hspace{1cm} (3.34)

Pulse loads do not have a direct effect on the calculation of the kinetic energy; the energy dissipation rate, however, is affected by the pulse loads and depending on whether the pulse is acting in the same or the opposite direction of the motion, the energy dissipation rate is increased or decreased.
CHAPTER 4

COMPUTER IMPLEMENTATION OF THE MODE SOLUTION TECHNIQUE

A computer program VISCO has been developed to implement the solution procedures given in Chapter 2 and Chapter 3, and has been used successfully in the analysis of a variety of cantilever beam structures.

The data input will be discussed in detail in the user manual given in Appendix A and comprises material constants $\varepsilon_0$, $\sigma_0$ and $n$, the coordinates of the discretised structure, node masses, the initial velocity field and control parameters such as time step size and output requirements. A listing of the program is given in Appendix B.

VISCO is a program written in FORTRAN language. It 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), it is displayed in order that it may be verified (subroutine DATA). Thereafter the influence matrices $[m_x]$, $[m_y]$ and $[m_z]$ as described in Section 3.2 are assembled. The numerical formulation of these matrixes is a straightforward static problem which is easily automated. This procedure is performed in subroutine STAT.

Once the influence matrices subject to the initial geometry have been found, the determination of the initial instantaneous mode shape may commence. A macro flow chart of this procedure is shown in Fig 4.1.
The kinetic energy $K_0$ of the input velocities is evaluated and thereafter a suitable $\lambda$ and these initial velocities used to compute load vectors which are in turn multiplied by the influence matrices to obtain bending moments on the structure (subroutine LOAD). Curvature rates corresponding to the bending moments are computed using equn (1.5).

The velocity field corresponding to the bending moment diagram is calculated using the principle of virtual velocities (equn (3.21)), which is evaluated over each element and the contribution from each element summed over the structure to give local velocities plus components of local velocities of other nodes (equn (3.22)). The true local node velocities are computed by subtracting the components and subsequently the velocity components in the global X and Y directions and the corresponding kinetic energy are evaluated. All this is done in subroutines VELOC and ABVEL.

Subsequently the subroutine CHECK is called up to check whether the last two sets of parameters $\lambda$ and $K$ are appropriate for a bisection routine to begin. If the answer is no, the program returns to LOAD with a new value for $\lambda$ to evaluate a new set of loads. Otherwise the program calls up subroutine BISEC in which a bisection algorithm is performed to obtain $\lambda$ for the initial kinetic energy $K_0$.

Once the above $\lambda$ has been found, a check is done on whether momentum balance has been achieved by calculating $\xi$ (equn (3.24)); if not, the velocities as calculated in BISEC are scaled by multiplying them with $\xi$ and thereafter a new kinetic energy is computed. The program returns to subroutine LOAD with the new scaled velocities to obtain new loads and subsequently in BISEC a $\lambda$ and new velocities are evaluated for the required kinetic energy level.
Once the generalised momentum balance has been achieved, the initial mode velocities are displayed in INITI and time integration can commence. This is shown in the flow chart in Fig 4.2.

The kinetic energy of the first time step is estimated and subroutines LOAD, VELOC and CHECK are used as above. Subroutine BISEC evaluates velocities corresponding to the energy estimate and subsequently a check is done on whether velocities used in LOAD and those computed from VELOC have converged; if not, the former are replaced by the latter and the program returns to LOAD to establish a new load vector of a slightly different shape. If the answer is yes, a check is done on the convergence of the kinetic energy estimate. If convergence has not been attained yet, a new energy dissipation rate is computed and a new kinetic energy estimate $K_{t+1}$ is done. The geometry is updated (subroutine UPDAT) using displacement increments computed from the latest set of velocities and new influence matrices are set up in STAT based on the current geometry. Subsequently mode velocities are computed for this kinetic energy level based on the current geometry. Once the kinetic energy $K_{t+1}$ has converged and the corresponding mode velocities have been found, the results are output in FINIT. The current mode velocities are stored and the instant in time is updated. A kinetic energy estimate for the new time step is done and if above a set cut-off value, mode velocities are evaluated for the new time step. If the kinetic energy level is below the cut-off value, subroutine FINAL estimates the final time $t_f$ and calculates the final displacement increments and outputs final results.
START

STAT
Set up influence matrices for the initial configuration

Calculate kinetic energy $K_0$ of the initial velocity field

LOAD
Assemble bending moment diagram and compute curvature rates due to loading ($\lambda[G]$ velocities).

VELOC
Calculate velocities in structure resulting from curvature rates as computed in LOAD

CHECK
Are parameters $\lambda$ and $K$ ready for bisection algorithm? If no, use new $\lambda$.

YES

BISEC
Compute velocities for required energy level

Check whether generalised momentum balance has been achieved by calculating eqn. (3.24b)

YES

INITI
Display initial mode velocities

RETURN

Figure 4.1 Generalised Momentum Balance
START

INPUT
Input all data

DATA
Display all data

Perform generalised momentum balance (Figure 4.1)

UPDAT
Update geometry of structure

STAT
Set up influence matrices for current geometry

Estimate $k^{t+1}$, the kinetic energy at the time $t+1$

LOAD
Assemble bending moment diagram and compute corresponding curvature rates

VELOC
Calculate velocities in structure resulting from loading calculated in load

CHECK
Are parameters $\lambda$ and $K$ ready for bisection algorithm? If no, use new $\lambda$.

YES

BISEC
Compute velocities for required energy level

(continued overleaf)
Have velocities used in LOAD and computed in VELOC converged? If not, replace the former by the latter.

Have energy level estimates for time $t + 1$ converged?

Calculate new energy dissipation rate

$T = T + \Delta T$

FINIT
Output results of time steps

Store current velocities and estimate velocities and displacement for time $T = T + \Delta T$

Is new kinetic energy estimate below cut-off value?

FINAL
Estimate final time $t_f$, calculate final displacement increment and display

STOP

Figure 4.2 Instantaneous Mode Algorithm
The program developed in this thesis, VISCO, has been successfully used for rigid-viscoplastic analyses of statically determinate cantilever beams. Initially the program was limited to perform geometrically linear analyses and the proposed mode approximation technique was tested with the constitutive relation. Subsequently geometric effects were incorporated in the program and a number of successful analyses were performed. As the algorithms presented in Chapters 2 and 3 are mode type approximations, examples were chosen for analyses which deform in a predominantly modal fashion.

Two shapes of cantilever structures with impulse or a combination of pulse and impulse loading were analysed and the results presented. The cases are:

(a) a straight cantilever loaded transversely at its tip

(b) an L-shaped cantilever loaded at its tip.

The problem of a cantilever struck transversely at its tip has been treated extensively, both experimentally and theoretically. A particular beam, E4, from the tests by Bodner and Symonds [9] was analysed using the proposed mode algorithms for both small and large displacement assumptions.
In their results for the experiment, Bodner and Symonds [9] gave no transverse deflection but a tip rotation of 52° and an estimated total time of deformation of 0.052s. Using a small displacement rigid-plastic analysis with strain rate sensitivity, they calculated the final time $t_f$ to be 0.064s. From a rigid-viscoplastic material model with small assumptions, Ting [8] estimated the final time to be 0.065s with a tip rotation of 59°. Lee and Martin [13] used an instantaneous mode technique together with the rigid-viscoplastic relation and estimated the tip rotation and $t_f$ to be 45.3° and 0.064s respectively.

Symonds [14], also neglecting geometric effects and using a matched viscous constitutive relation, calculated $t_f$ as 0.066s, the tip rotation as 52.4° and the transverse tip displacement as 0.348m. Griffin and Martin [16] obtained deflections of 0.347m and 0.348m with small displacement assumptions, using their mode solution technique with the homogeneous viscous material model and based their matching on the value of $t_f$ given by Symonds[14]. Their instantaneous mode technique, a large displacement analysis, gave a tip deflection of 0.302m and the total time of deformation as 0.065s. Using a direct method of analysis, with the inclusion of geometric effects, Griffin [20] obtained a transverse tip displacement of 0.330m with a final time of deformation of 0.065s.

Two methods of analysis are presented here. The first analysis of the beam E4 is based on small displacement assumptions as described in Chapter 2; the tip rotation and transverse deflection were calculated as 44.3° and 0.331m respectively, and $t_f$ was estimated to be 0.063s. The second analysis assumes large displacements and, based on the theory in Chapter 3, gave a tip rotation of 55.8° and the transverse tip deflection as 0.282m. The final time of deformation was also estimated as 0.063.
Comparing the geometrically nonlinear analyses of Griffin and Martin [16] with the present one, it can be seen that both the final time estimate and the transverse tip deflection are less in the latter case; this is to be expected since the rigid-viscoplastic relation represents a slightly stiffer material. The physical description of the beam E4 is given in Fig 5.1, while plots of the displaced shape at successive time intervals for both the small and large displacement analyses are illustrated in Fig 5.2.

Examples of a cantilever subjected to a combination of impulse and pulse loading are given in Figs 5.3 and 5.4. The dimensions and material properties of both these cases are identical to those of the standard E4 beam, and the cantilevers are subjected to an initial impulse similar to that of the E4 beam. In addition, transverse follower pulses are applied at the tip in the direction of and opposite to the motion of the tip, respectively, and the histories of which are also illustrated in Figs 5.3 and 5.4. For the negative pulse case (Fig 5.3), the total time of deformation was estimated as 0.052s, the tip rotation as 38.5° and the transverse tip deflection as 0.212m; the positive pulse case gave 0.077s, 81.6° and 0.341m respectively (see Fig 5.4). In Fig 5.5, a plot of kinetic energy versus time is given for the standard E4 problem together with the two cases which are subjected to additional pulse loading. Figs 5.6 and 5.7 show plots of tip velocity components and support moment versus time respectively.

In order to test the proposed algorithms on a slightly more complicated structure, an L-shaped cantilever was chosen as illustrated in Fig 5.8. The objective was to establish whether the algorithms could also be used in the analysis of cantilever beams which are not necessarily straight. As for the straight cantilever case, three
loading cases were analysed: in the first case, the cantilever was loaded impulsively at its tip in the axial direction, resulting in an initial vertical velocity field of 11.715 m/s along the short arm; no other loading was applied after time $t = 0$. In the other two cases, the cantilever was subjected to a combination of impulse and pulse loading. A pulse load of 150 N was applied at the tip of the cantilever in the direction of or opposite to the elbow.

Fig 5.9 shows plots of the displaced shape at successive time intervals of the impulsive loading case for both small and large displacement assumptions. For the small displacement case, the total time of deformation, the tip rotation and transverse tip displacement were estimated as 0.054 s, $50^\circ$ and 0.239 m respectively; including geometric effects, the calculations gave $t_f$ as 0.054 s, the tip rotation as $41.7^\circ$ and a tip displacement of 0.247 m. In Figs 5.10 and 5.11 plots of the displaced shapes and pulse histories are given for the two cases where a combined loading of impulse and pulse was used. The latter two analyses were performed using large displacement assumptions. For Fig 5.10, the results were 0.047 s, $37.1^\circ$ and 0.194 m, while for Fig 5.11, values of 0.063 s, $65^\circ$ and 0.314 m were calculated.

To demonstrate the capability of the program of handling geometric nonlinearities, an L-shaped cantilever, modelled by twelve elements, was subjected to severe dynamic loading: an impulse resulting in initial velocities together with a linearly decaying pulse load with its maximum at the initial time. Plots of the deformed shape at successive time intervals are given in Fig 5.12.

Noting that the rigid-viscoplastic constitutive relation (eqn (1.5)) approaches the rigid-perfectly plastic law as the index $n$ is raised to infinity, two separate analyses were performed as
approximations to theoretical rigid-plastic solutions. A value of 100 for the index n proved sufficiently high to give excellent agreement with theoretical results. The examples together with their physical description are illustrated in Figs 5.13(a) and 5.13(b). The theoretical rigid-plastic solutions for the two cases are governed by

\[ I \theta = - M_p \]  \hspace{2cm} (5.1)

where \( I \) is the moment of inertia as calculated with respect to the support, and \( M_p \) is the plastic moment at the support. In both cases, the cross-section was taken as that of the beam E4 (as tested by Bodner and Symonds [9]), and the other physical characteristics were so chosen that the moment at the support was twice the yield moment \( M_0 \). The theoretical rigid-plastic solutions for the two cases were calculated as follows:

(i) straight cantilever: the moment of inertia is given by

\[ I = m \ell^2 \]  \hspace{2cm} (5.2)

where \( m \) is the lumped mass. Solving for final time of deformation \( t_f \) and final rotation, we obtain

\[ t_f = \frac{v_0 m \ell}{M_p} \]  \hspace{2cm} (5.3)

and

\[ \theta(t_f) = \frac{v_0^2 m}{2 M_p} \]  \hspace{2cm} (5.4)
(ii) L-shaped cantilever: \( l \) is given by

\[
l = m\dot{\lambda}^2 + m(\dot{\lambda}^2 + h^2) = m(2\dot{\lambda}^2 + h^2)
\]

and hence the final time and rotation can be derived:

\[
t_f = \frac{v_0 m(2\dot{\lambda}^2 + h^2)}{\dot{\lambda} M_p},
\]

\[
\theta(t_f) = \frac{v_0^2 m(2\dot{\lambda}^2 + h^2)}{2 M_p \dot{\lambda}^2},
\]

Substituting the relevant data from Figs 5.15(a) and (b) into eqns (5.3), (5.4) and eqns (5.6), (5.7) respectively, values are obtained which are given in Table 1 and compared to the analytical results: excellent agreement between the theoretical rigid-plastic and approximate analytical results (using an index \( n \) of 100) gives further evidence of the applicability of the proposed algorithms.

<table>
<thead>
<tr>
<th></th>
<th>theory</th>
<th>analysis (n=100)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( t_f )</td>
<td>( \theta(t_f) )</td>
</tr>
<tr>
<td>case (i)</td>
<td>0.01867s</td>
<td>1.706 rads</td>
</tr>
<tr>
<td>case (ii)</td>
<td>0.0455s</td>
<td>2.022 rads</td>
</tr>
</tbody>
</table>

Table 1  Comparison of theoretical rigid-plastic with approximate analytical results.
All analyses were performed on the SPERRY 1100 mainframe computer. A typical C.P.U. time, i.e. for the standard E4 beam problem, using five elements and forty time steps, is 7 minutes and 10 seconds. Although the analyses using the proposed algorithms are rather costly in terms of computing time, they have the main advantage that no matching has to be performed on the constitutive relation as is the case for homogeneous viscous materials.
mass density \( \rho = 7850\text{kg/m}^3 \)

\[ \sigma_0 = 200\text{MPa}, \quad \dot{\varepsilon}_0 = 40/\text{s}, \quad n = 5 \]

**Figure 5.1** Description of E4-Beam

**Figure 5.2** Displaced shape of E4-beam at successive time intervals for large displacement analysis and final displaced shape for small displacement analysis.
Figure 5.3 Displaced shape at successive time intervals of E4-beam subjected to a combination of an initial impulse and a negative follower pulse at tip.

Figure 5.4 Displaced shape at successive time intervals of E4-beam subjected to a combination of an initial impulse and a positive follower pulse at tip.
Figure 5.5  Plots of kinetic energy vs time of E4-beam examples

Figure 5.6  Plots of velocity components vs time of E4-beam examples
Figure 5.7 Plots of support bending moment vs time of E4-beam example

Figure 5.8 Description of L-shaped cantilever. The material properties are identical to those of the E4-beam.
large displacement analysis
small displacement analysis

Figure 5.9 Displaced shape of L-shaped cantilever beam subjected to an initial impulse only.

Figure 5.10 Displaced shape of cantilever subjected to a combination of initial impulse and a negative follower pulse at tip.
Figure 5.11  Displaced shape of cantilever subjected to a combination of initial impulse and a positive follower pulse at tip.

Figure 5.12  Cantilever subjected to an initial impulse and a positive follower pulse applied at the tip.
Description of two rigid-plastic cantilever beam examples with material constants identical to the E4-beam.

Displaced shapes at successive time intervals of the rigid-plastic cantilever beam example.
CHAPTER 6

CONCLUSIONS

The instantaneous mode algorithm as presented in this thesis for rigid-viscoplastic materials proves to be a worthy tool in the analysis of ductile metal cantilevers subjected to large impulses and pulses. The numerical implementation of the algorithm enables the determination of instantaneous mode shapes corresponding to arbitrary energy levels. Geometric effects are included in the formulation and deformations of the order of the dimensions of the structures can be traced successfully.

Very good agreement is obtained with experimental results of the tip-loaded cantilever case. The algorithms have also been successfully applied in the analysis of cantilevers which are not straight initially; the loading cases considered include the combination of an impulse and a follower pulse. Approximate rigid-plastic solutions, obtained by using a value of 100 for the index n in the rigid-viscoplastic constitutive relation, compare excellently with theoretical rigid-plastic results.

The program is capable of handling pipe-whip problems where the structure is initially stationary and the loading is that of a follower pulse. Attempts were done to analyse examples given in the literature; the information, however, was found to be insufficient to properly test the program.
REFERENCES


APPENDIX A

'VISCO' User Manual

Introduction

VISCO is a finite element program for the large displacement analysis of rigid-viscoplastic cantilever beams, subjected to large impulsive (and pulse) loading. The theory and computer implementation of this program are outlined in Chapters 2, 3 and 4 of this thesis.

The program was written in FORTRAN 77 and was run on a SPERRY 1100 mainframe computer. All data is to be 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 and N stand for integer numbers and all others for real numbers. The program is written in double precision.

In Section A-1 the data input will be described, and in Section A-2 some guidelines for the efficient use of the program will be suggested. Section A-3 contains a typical program input.
Section A-1

DATA INPUT

The input data for VISCO is divided into 9 sections which are described below in the order in which they must appear.

Except for the title and end of data, all sections begin with a header card which contains the name of that section or sub-section followed by zero or more numerical data cards. These header cards must be input, unless stated otherwise. These serve not only as terminators of groups of data but, more importantly, as comments in the data input. These comments together with the fixed input order facilitate easier data checking.

At least the first four letters of each header and subsection headers cards must appear in the first four columns.

All numerical data is to be input in free format. The only disadvantage is that values not required must be input as zero.

1.1. Title

The VISCO data check begins with a single line title card. This title is printed at the start of the printout.

```
title
```

The title may occupy columns 1 through 72 inclusive.
1.2. Control Data

The control data section begins with the card

```
CONTROL DATA
```

Following the header card, the program requires that the control parameters are input in the form

```
NEL  NNO
GEOM
```

NEL  number of elements in the structural model (NEL < 20)
NNO  number of nodes (NNO < 21)
GEOM is the control parameter for geometric assumptions and one of the following two words has to be input:

LARGE    for geometrically nonlinear analysis
SMALL    for geometrically linear analysis

1.3. Section and Material Properties

```
SECTION AND MATERIAL PROPERTIES
```

The following numerical data are required for the section and material properties
<table>
<thead>
<tr>
<th>HH</th>
<th>BB</th>
<th>YSTRS</th>
<th>EPSIO</th>
<th>EN</th>
</tr>
</thead>
</table>

HH - section depth in meters  
BB - section width in meters  
YSTRS - yield stress of section in MPa  
EPSIO - strain rate constant (40 for mild steel)  
EN - power in constitutive relation (5 for mild steel)

1.4. Solution Details

**SOLUTION DETAILS**

IFREQ TZERO TSTEP TOL TOL1 TOL2 TOL3

IFREQ - frequency, in number of time steps, of printout of results.  
The final output, when the structure is at rest, is always printed.

TZERO - starting time for the analysis

TSTEP - number of steps into which the total time of deformation $t_f$ is to be subdivided. A crude estimate of $t_f$ is automatically calculated by the program.

TOL - determines the cut-off for the analysis;  
$$\varepsilon_K = TOL \times K^0$$  
(typically TOL = $10^{-2}$)
TOL1 - tolerance for kinetic energy convergence during forward integration

\[ \left| \frac{K_{t+1}^{1-1} - K_{t+1}^1}{K_{t+1}^1} \right| < TOL1 \times K^0 \]

(typically \( TOL1 = 10^{-6} \))

TOL2 - tolerance for velocity convergence

\[ \epsilon_v = TOL2 \times \dot{u}_{max}^0 \]

(typically \( 10^{-3} \))

TOL3 - tolerance for kinetic energy convergence in bisection algorithm; use the same as for TOL1.

1.5. Nodal Coordinates

The header card for this section is

NODAL COORDINATES

The program then requires NNO cards, as follows, giving the node number and the x and y coordinates for each node.

| node | X | Y |

The above set of NNO cards must be input in ascending order starting with node 1, where node 1 is the constrained node.
1.6. Initial Velocities

For the impulsive loading case a set of initial velocities is required. These are input beginning with the header:

```
INITIAL VELOCITIES
```

followed by

```
node  XVEL  YVEL
```

where XVEL, YVEL are the initial velocities in the x and y directions respectively.

The above set must consist of NNO cards and must be input in ascending order, starting with node 1. The velocities for node 1 can be defined arbitrarily but will be taken as zero automatically.

1.7. Lumped Masses

The input begins with

```
LUMPED MASSES
```

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 program requires NNO cards as follows, giving the node number and its mass.
node GMASS

Again the above set of cards must be input in ascending order, starting with node 1. The mass for node 1 can be arbitrarily defined.

1.8. Pulse Loads

Even if zero pulse loads are applied to the structure, the following header must be input,

```
PULSE LOADS
```

If no pulses are applied, omit section (1.8.1)

1.8.1 Nonzero Pulse Loads

The program distinguishes between two types of "follower" pulse forces, applied at the last node, and must be input as follows

```
type
```

type - TANGENTIAL the pulse force is acting in the direction of the last element.

- TRANSVERSE the pulse force is acting transversely to the last element.
Subsequently, a card is required giving the magnitude of the pulse load.

\[\text{PLOAD}\]

- a positive value denotes a pulse force following the movement of the last node and vice versa. It must be input in units of \([\text{N}]\).

1.8.1.1 Time Function

A load-time function must be input if pulses are applied to the structure, beginning with the header.

\[\text{TIME FUNCTION}\]

Then the following card is required.

\[\text{IPTS} \quad T_1 \quad F_1 \quad T_2 \quad F_2 \quad \cdots \quad T_{\text{IPTS}} \quad F_{\text{IPTS}}\]

- the number of function points to be input (3 < IPTS < 20).

- the time coordinate and its corresponding function value of a function point. The function is assumed to be linear between two successive function points (see Fig A-1).
Figure A-1  A TYPICAL LOAD - TIME FUNCTION

Note that the pulse load at any instant is given by the load value PLOAD multiplied by the value $F_1$ of the load - time function at that instant.

1.8.2 Zero Pulse Load

If no pulse is applied to the structure, the following card must be input:

```
NONE
```

1.9. End of Data

The very last card to be input must be:

```
END OF DATA
```
Section A-2

Efficient and meaningful results can only be expected once the user has gained a certain level of experience with the program. An unwise choice of certain parameters could result in an unrealistic solution or the nonconvergence of the algorithms with no solution. Some guidelines are given here for a reasonable choice of certain parameters.

2.1. Time Step Size

The program calculates a crude estimate of the final time once the initial kinetic energy and its time rate of change have been evaluated. This final time estimate divided by the specified number of time steps determines the time step size. Typically a value of 20-40 can be used for the number of time steps TSTEP, depending on the complexity of the program.

2.2. Tolerances

Suggested values for the tolerance magnitudes are given in Section 1.4. The user can make his own choice, however, depending on accuracy requirements.
2.3. **Output**

At the beginning of the program output a reflection is given of the input data. The output of the selected time steps consists of nodal velocity and displacement components, given in \([\text{m/s}]\) and \([\text{m}]\) respectively. Node bending moments are given in units \([\text{Nm}]\). In addition, the magnitude of the current pulse is given in \([\text{N}]\), together with the current value of \(\lambda\) and the kinetic energy (units \([1/\text{s}]\) and \([\text{J}]\) respectively).
Section A-3

Sample Program Input

1 LARGE DISPLACEMENT ANALYSIS OF E4-BEAM WITH PULSE (75N)
2 CONTROL DATA
3 5 6
4 LARGE
5 SECTION AND MATERIAL PROPERTIES
6 .004496 .016307 200. 40. 5.
7 SOLUTION DETAILS
8 4 .0 20. 0.01 1.E-6 1.E-3 1.E-6
9 NODAL COORDINATES
10 1 0.00000 0.00000
11 2 0.01250 0.00000
12 3 0.03750 0.00000
13 4 0.07500 0.00000
14 5 0.21530 0.00000
15 6 0.35560 0.00000
16 LUMPED MASSES
17 1 0.00000
18 2 0.01080
19 3 0.01800
20 4 0.05119
21 5 0.08078
22 6 0.37640
23 INITIAL VELOCITIES
24 1 0.00000 0.00000
25 2 0.00000 0.00000
26 3 0.00000 0.00000
27 4 0.00000 0.00000
28 5 0.00000 0.00000
29 6 0.00000 11.51541
30 PULSE LOADS
31 TRANSVERSE TYPE
32 75.
33 TIME FUNCTION
34 3
35 1 .0000 1.
36 2 .0100 1.
37 3 .0101 0.
38 END OF DATA
APPENDIX B

VISCO program listing
B.1

********************************************************************************************
DRIVER PROGRAM FOR SOLVING DYNAMICS OF RIGID-VISCOPLASTIC CANTILEVERS
******************************************************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)

INCLUDE VISCO.PROCA
CALL INPUT
CALL DATA

DO 11 I=2,NN
VELO(I,1)=VEL(I,1)
VELO(I,2)=VEL(I,2)
11 CONTINUE

C TIME LOOP .........
DO 12 ITIME=1,200
C KINETIC ENERGY LOOP .........
DO 11 INERGY=1,444
CALL STAT
ITER=0
IF(IFLAG.EQ.1) GOTO 22

C IMPLICIT FORWARD INTEGRATION OF KIN. ENERGY
ENER(1)=EOLD+0.5*DT*(DEOLD+DENER(1))
PRINT*,"NEW APPROX. ENER(1) =",ENER(1)
PRINT*,"EOLD, DEOLD, DENER(1) =",EOLD,DEOLD,DENER(1)

C CHECK WHETHER KIN. EN. IS BELOW LIMIT
IF(ENER(1).LT.TOL.OR.ENER(1).LT.0.01) GOTO 800
KIN=DEOLD*DT
KIN=DABS(KIN)
IF(ENER(1).LT.KIN) GOTO 800

LOOP=0
22 CONTINUE

C MODE VELOCITY LOOP ............
DO 1 IVELO=1,400
LOOP=LOOP+1
IF(LOOP.NE.1) THEN
DO 2 I=2,NN
VEL(I,1)=FXCOM(I,1)
VEL(I,2)=FXCOM(I,2)
AVEL(I)=AFX(I)
2 CONTINUE
END IF
B.2  

PREPARING LAMDA FOR BISECTION ALGORITHM

DO 999 ILAMDA=1,555

CALL LOAD
CALL VELOC
CALL CHECK

IF(ENER(1).LT.1.D-5.AND.ENER(3).LT.1.D-5) GOTO 888

IF(INC.EQ.1) THEN
  GAM(1)=GAMM
  F(1)=ENER(2)
  GAM(2)=GAM(1)+DGAM
  GAMM=GAM(2)
  DO 25 I=2,NN
  GSTOR(1,I,1)=GSTOR(2,I,1)
  GSTOR(1,I,2)=GSTOR(2,I,2)
  CONTINUE
ELSE
  GOTO 777
END IF

999 CONTINUE

C GENERALISED MOMENTUM BALANCE FOR INITIAL CONDITIONS

IF(IFLAG.EQ.1) THEN
  DO 220 I=2,NN
  PRINT*,VEL(I,1),VEL(I,2),FXCOM(I,1),FXCOM(I,2)
  CONTINUE
  DO 221 J=1,2
  PSID=0.
  PSIN=0.
  CONTINUE
  DO 222 J=1,2
  DO 222 I=2,NN
  PSIN=PSIN+GMASS(I,I)*VELO(I,J)*FXCOM(I,J)
  PSID=PSID+GMASS(I,I)*(FXCOM(I,J))**2
  CONTINUE
  PSI=PSIN/PSID
  PRINT*," PSI =",PSI
  DO 333 I=2,NN
  FXCOM(I,1)=PSI*FXCOM(I,1)
  FXCOM(I,2)=PSI*FXCOM(I,2)
  CONTINUE
  AFX(I)=DSQRT(FXCOM(I,1)**2+FXCOM(I,2)**2)
  CONTINUE
  ENER(1)=0.
  DO 444 I=2,NN
  ENER(1)=ENER(1)+0.5*GMASS(I,I)*(AFX(I))**2
  CONTINUE
  PRINT*," ENER(1) ENER(2)",ENER(1),ENER(2)
  CDIFF=ENER(1)-ENER(2)
  CDIFF=DABS(A)
  PRINT*," CDIFF =",CDIFF
  IF(CDIFF.LT.TOL) THEN
    GOTO 888
  ELSE
    GOTO 85
  END IF
B.3

CHECK WHETHER VELOCITIES HAVE CONVERGED

IX=0
DO 555 I=2,NN
DO 555 J=1,2
DIFF=VEL(I,J)-FXCOM(I,J)
DIFF=DABS(DIFF)
IF(DIFF.GT.TOL2) IX=1
CONTINUE

IF(IX.EQ.0) THEN
  GOTO 666
END IF

CONTINUE

DGAM=DABS(DGAM)
CONTINUE

CALL INITI

CONTINUE

DGAM=DABS(DGAM)
DO 3 I=2,NN
VEL(I,1)=FXCOM(I,1)
VEL(I,2)=FXCOM(I,2)
CONTINUE

CHECK WHETHER KINETIC ENERGY HAS CONVERGED

EE=E-ENER(1)
EE=DABS(EE)
PRINT*, ' EE=', EE
E=ENER(1)
IF(EE.LT.TOL1) THEN
  ITER=1
  DEOLD=DENER(1)
GOTO 700
END IF

EVALUATE TIME RATE OF CHANGE OF ENERGY

DENER(1)=0.
DENER(1)=-2.*GAMM*ENER(2)
PRINT*, ' NORMAL DENER(1)', DENER(1)

IF(GEOM.EQ.'SMALL') GOTO 11

EVALUATE ADDITIONAL TERM FOR ENERGY RATE

DO 68 I=1,NN
CENT(I)=0.
CONTINUE

DO 69 I=3,NN
B.4

DO 70 II=2,1-1
    CENT(I)=CENT(I)+(FXCOM(II,1)*FXCOM(I,2)-FXCOM(II,2)*
    &FXCOM(I,1))*(FX(II)/CL(II-1)-(FX(II+1)/CL(II))
190 70 CONTINUE
191 CENT(I)=CENT(I)*GMASS(I,I)
192 69 CONTINUE
193
194 DO 72 I=2,NN
195 DENER(1)=DENER(1)+CENT(I)
196 72 CONTINUE
197
198 C PRINT*, ' UPDATED DENER(1)', DENER(1)
199 11 CONTINUE
200
202 T = T + DT
203
204 CALL FINIT
205
206 DENER(1)=DEOLD
207 EOLD=ENER(2)
208 ICOUNT=0
209 C NEW VELOCITY ESTIMATES
210 DO 33 I=2,NN
211 VEL(I,1)=FXCOM(I,1)-GAMM*FXCOM(I,1)*DT
212 VEL(I,2)=FXCOM(I,2)-GAMM*FXCOM(I,2)*DT
213 33 CONTINUE
214
216 DO 55 I=2,NN
217 VELO(I,1)=FXCOM(I,1)
218 VELO(I,2)=FXCOM(I,2)
219 55 CONTINUE
220
221 12 CONTINUE
222
223 CALL FINAL
224
225 STOP
226 END
227
229 SUBROUTINE INPUT
230 C
231 C*****************************************************************************
232 C
233 C SUBROUTINE FOR DATA INPUT
234 C
235 C*****************************************************************************
236 IMPLICIT DOUBLE PRECISION*(A-H,O-Z)
237 INCLUDE VISCO,PROCA
238 C
239 100 FORMAT( )
240 101 FORMAT(A5)
241 102 FORMAT(A72)
242 103 FORMAT(A10)
243 104 FORMAT(A4)
244
245 READ(IREAD,102)TITLE
246
247 DO 1 IJKL=1,8
READ(IREAD,104) DUMMY

IF(DUMMY.EQ.'CONT') THEN
   READ(IREAD,100) NE,NN
   READ(IREAD,101) GEOM
   GOTO 1
END IF

IF(DUMMY.EQ.'SECT') THEN
   READ(IREAD,100) HH,BB,YSTRS,EPSIO,EN
   GOTO 1
END IF

IF(DUMMY.EQ.'SOLU') THEN
   READ(IREAD,100) IFREQ,TZERO,TS,TOL,TOL1,TOL2,TOL3
   GOTO 1
END IF

IF(DUMMY.EQ.'NODA') THEN
   READ(IREAD,100)(J,COORDX(I),COORDY(I),I=I,NN)
   GOTO 1
END IF

IF(DUMMY.EQ.'LUMP') THEN
   READ(IREAD,100)(J,GMASS(I,I),I=I,NN)
   GOTO 1
END IF

IF(DUMMY.EQ.'INIT') THEN
   READ(IREAD,100)(J,VEL(I,I),VEL(I,2),I=I,NN)
   GOTO 1
END IF

IF(DUMMY.EQ.'PULS') THEN
   READ(IREAD,103) PULFOR
   IF(PULFOR.EQ.'NONE') THEN
      DO 10 I=I,NN
         PULS(I)=0.
      10 CONTINUE
      GOTO 14
   ELSE
      READ(IREAD,100) P(NN,I)
   END IF
   GOTO 1
END IF

IF(DUMMY.EQ.'TIME') THEN
   READ(IREAD,100) IPTS
   DO 12 J=I,IPTS
      READ(IREAD,100) I,FACT(I,I),FACF(l,I)
   12 CONTINUE
   FACT(I,IPTS+1)=50.
   FACF(I,IPTS+1)= 0.
   GOTO 1
END IF

IF(DUMMY.EQ.'END ') THEN
   GOTO 14
END IF
GOTO 1

1 CONTINUE
CONTINUE
C CALCULATE ALL CONSTANTS .....  
T=TZERO
DO 2 I=1,NN
XGEO(I)=COORDX(I)
YGEO(I)=COORDY(I)
2 CONTINUE
VO=0.
DO 8 I=1,NN
IF(DABS(VEL(I,1)).GT.VO) VO=DABS(VEL(I,1))
IF(DABS(VEL(I,2)).GT.VO) VO=DABS(VEL(I,2))
8 CONTINUE
PIE=3.14159265358979324
AA=HH*BB
RMO=AA*HH*YSTRS/4*1000000
RKO=2.*EPSIO/(((2.*EN)/(2.*EN+1))**EN)*HH)
DO 3 I=1,NN
AVEL(I)=DSQRT(VEL(I,1)**2+VEL(I,2)**2)
AVELO(I)=AVEL(I)
3 CONTINUE
ENER(1)=0.
DO 4 I=2,NN
ENER(I)=ENER(I)+0.5*GMASS(I,I)*(AVEL(I)**2
4 CONTINUE
ENER(3)=ENER(1)
GAMM=1.5
DGAM=1.
TOL=TOL*ENER(3)
TOL1=TOL1*ENER(3)
TOL2=TOL2*VO
TOL2=DABS(TOL2)
TOL3=TOL3*ENER(3)
IF(VO.LT.1.D-3) THEN
TOL = .200000000
TOL1= .000020000
TOL2= .030000000
TOL3= .000020000
END IF
IEXIT=0
ITP=0
IPLTS=-1
IFLAG=1
ITER=1
SIMP=DFLOAT(ISIMP)
RETURN
DEBUG SUBCHK
END
SUBROUTINE DATA
*****************************************************************************
SUBROUTINE FOR DATA DISPLAY
*****************************************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA
WRITE(IPRINT,123)
FORMAT(1H1,///////)
&//,2OX,'*****************************************************************************'
&//,2OX,' A PROGRAM FOR SOLVING DYNAMICS OF RIGID-
&//,2OX,' VISCOPLASTIC CANTILEVER BEAMS ...........
&//,2OX,' VERSION @ 29/8/84
&//,2OX,'*****************************************************************************'
DISPLAYS ALL INPUT DATA
WRITE(IPRINT,1)TITLE
FORMAT(1H1,SX,80('*'),/10X,A80,I,6X,80('*'},/) 
WRITE(IPRINT,7) GEOM 
FORMAT(1H ,III,lOX,'RIGID-VISCOPLASTIC ANALYSIS WITH ',
&'AS,' DISPLACEMENT ASSUMPTIONS '.'/10X,
&'THE INITIAL DISPLACEMENTS ARE ASSUMED TO BE ZERO .')
WRITE(IPRINT,13)EN,YSTRS,EPSIO,RMO,RKO,HH,BB,NE,NN
FORMAT(1H ,1,10X,'COORDINATES OF NODES'
&'NODE' ,lOX,'X',12X,'Y' 
DO 110 I=1,NN
WRITE(IPRINT,4)I,COORDX(I),COORDY(I)
CONTINUE
WRITE (IPRINT, 14)
SUBROUTINE ABVEL
C **********************************************************
C SUBROUTINE FOR CALCULATING VELOCITY COMPONENTS AND
C ABSOLUTE VELOCITIES FROM LOCAL VELOCITIES .......
C **********************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA

DO 1 J=1,2
DO 1 I=1,NN
FXCOM(I,J)=0.
1 CONTINUE

DO 2 I=2,NN
FXCOM(I,1)=FXCOM(I-1,1)-FX(I)*DSIN(THETA(I-1))
GSTOR(2,I,1)=FXCOM(I,1)
FXCOM(I,2)=FXCOM(I-1,2)+FX(I)*DCOS(THETA(I-1))
GSTOR(2,I,2)=FXCOM(I,2)
2 CONTINUE

DO 3 I=1,NN
AFX(I)=0.
3 CONTINUE

DO 4 I=2,NN
AFX(I)=DSQRT(FXCOM(I,1)**2+FXCOM(I,2)**2)
4 CONTINUE
8.9
497 RETURN
498 DEBUG SUBCHK
499 END
500
501
502
503 SUBROUTINE UPDAT
504 C
505 C
506 C
507 C
508 C
509 IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
510 INCLUDE VISO.PROCA
511
512 IF(ITER.EQ.0) THEN
513 DO 6 I=2,NN
514 XINC(I)=0.5*(VELO(I,1)+FXCOM(I,1))*DT
515 YINC(I)=0.5*(VELO(I,2)+FXCOM(I,2))*DT
516 6 CONTINUE
517
518 DO 2 I=1,NN
519 XGEO(I)=XCOORD(I)+XINC(I)
520 YGEO(I)=YCOORD(I)+YINC(I)
521 2 CONTINUE
522 END IF
523
524 IF(ITER.EQ.0.OR.IFLAG.EQ.1) THEN
525 DO 3 I=1,NE
526 IF((XGEO(I+1)-XGEO(I)).EQ.0.DO) THEN
527 IF((YGEO(I+1)-YGEO(I)).GT.0.DO) THETA(I)=PIE/2.
528 IF((YGEO(I+1)-YGEO(I)).LT.0.DO) THETA(I)=-PIE/2.
529 3 CONTINUE
530 END IF
531 THETA(I)=DATAN((YGEO(I+1)-YGEO(I))/(XGEO(I+1)-XGEO(I)))
532 IF(XGEO(I+1).LT.XGEO(I).AND.YGEO(I+1).GT.YGEO(I))
533 & THETA(I)=THETA(I)+PIE
534 IF(XGEO(I+1).LT.XGEO(I).AND.YGEO(I+1).LT.YGEO(I))
535 & THETA(I)=THETA(I)+PIE
536 3 CONTINUE
537 END IF
538
539
540 IF(ITER.EQ.1) THEN
541 DO 4 I=1,NN
542 XCOORD(I)=XGEO(I)
543 YCOORD(I)=YGEO(I)
544 4 CONTINUE
545 END IF
546
547 11 CONTINUE
548 RETURN
549 DEBUG SUBCHK
550 END
551
552
553
554
555
556 SUBROUTINE STAT
557 C
558 C
SUBROUTINE FOR SETTING UP INFLUENCE MATRICES

**********************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA
IF(IFLAG.EQ.0.AND.GEOM.EQ.'SMALL') GOTO 50
CALL UPDAT
CALCULATE ELEMENT LENGTHS ....
IF(IFLAG.EQ.1) THEN
DO 10 I=1,NE
CL(I)=DSQRT((YGEO(I+1)-YGEO(I))**2+(XGEO(I+1)-XGEO(I))**2)
CONTINUE
END IF
ASSEMBLING ALL INFLUENCE MATRICES ....
IF(NE.EQ.1) GOTO 222
DO 5 I=1,NN
DO 5 J=1,NE
VM(I,J)=0.
CONTINUE
VM(I,I)=CL(I)
CONTINUE
DO 7 J=2,NN-1
DO 7 I=J,1,-1
VM(I,J)=VM(I+1,J)+CL(I)*DCOS(THETA(J)-THETA(I))
CONTINUE
DO 8 K=1,NE
DO 8 I=(K-1)*ISIMP+1,ISIMP*K
DO 8 J=K,NE
XVM(I,J)=VM(K,J)-(1./ISIMP)*(DFLOAT(I)-1.-
&ISIMP*(DFLOAT(K)-1.))*CL(K)*DCOS(THETA(J)-THETA(K))
CONTINUE
UNITMX(I,J)=CL(I)*DCOS(THETA(I))
UNITMY(I,J)=CL(I)*DSIN(THETA(I))*(-1.)
DO 10 J=2,NE
UNITMX(I,J)=UNITMX(I,J-1)+CL(J)*DCOS(THETA(J))
UNITMY(I,J)=UNITMY(I,J-1)-CL(J)*DSIN(THETA(J))
CONTINUE
DO 17 I=2,NE
UNITMX(I,J)=UNITMX(I-1,J)-UNITMX(I-1,I-1)
UNITMY(I,J)=UNITMY(I-1,J)-UNITMY(I-1,I-1)
SUBROUTINE PULSE

C ***********************************************************
C SUBROUTINE EVALUATING FOLLOWER PULSE MAGNITUDE FROM
C PULSE-TIME HISTORY ............
C ***********************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA

DO 1 I=NN,NN
   DO 2 ITFUN=1,NTFUN
      DO 3 IT=1,NPTS
         TDEL=FACT(ITFUN,IT)-T
         IF(TDEL.EQ.0.DO) THEN
            PULS(I)=PACF(ITFUN,IT)*P(I,ITFUN)
            GOTO 5
         END IF
      3 CONTINUE
   2 CONTINUE
1 CONTINUE
RETURN
DEBUG SUBCHK
END
IF(TDEL.GT.0.DO) GOTO 4
   CONTINUE
   PULS(I)=(FACF(ITFUN,IT)-FACF(ITFUN,IT-1))
   PULS(I)=PULS(I)*(T-FACT(ITFUN,IT-1))/(FACT(ITFUN,IT)-FACT(ITFUN, IT-1))
   PULS(I)=PULS(I)+FACF(ITFUN,IT-1)
   PULS(I)=PULS(I)*P(I,ITFUN)
2   CONTINUE
1   CONTINUE
5   CONTINUE
RETURN
DEBUG SUBCHK
END

SUBROUTINE LOAD
C **********************************************************
C SUBROUTINE EVALUATING FORCES DUE TO MASS * ACCELERATION
C AND POSSIBLY AXIAL OR TRANSVERSE PULSE ..........
C **********************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA
C FORCE=LAMBDA*MASS*VELOCITY ( + PULSE )

DO 20 I=2,NN
   FORC(I,1)=GAMM*GMASS(I,I)*VEL(I,1)
   FORC(I,2)=GAMM*GMASS(I,I)*VEL(I,2)
   IF(PULFOR.EQ.'NONE') GOTO 20
   CALL PULSE
   IF(PULFOR.EQ.'TANGENTIAL') THEN
      FORC(I,1)=FORC(I,1)-PULS(I)*DCOS(THETA(I-1))
      FORC(I,2)=FORC(I,2)-PULS(I)*DSIN(THETA(I-1))
   END IF
   IF(PULFOR.EQ.'TRANSVERSE') THEN
      FORC(I,1)=FORC(I,1)-PULS(I)*DSIN(THETA(I-1))
      FORC(I,2)=FORC(I,2)+PULS(I)*DCOS(THETA(I-1))
   END IF
20 CONTINUE

DO 33 J=1,NE
   DO 32 I=1,ISIMP+1
      BMOM(I,J)=0.
      CURV(I,J)=0.
32 CONTINUE
33 CONTINUE

BENDING MOMENT = (INFLUENCE MATRIX) * (LOAD VECTOR)

DO 35 I=1,ISIMP+1
   DO 34 K=1,NE
      BMOM(I,K)=BMOM(I,K)+FORC(J+1,1)*XINFY(ISIMP*(K-1)+I,J)
34 CONTINUE
35 CONTINUE
B.13

745 BMOM(I,K)=BMOM(I,K)+FORC(J+1,2)*XINFX(ISIMP*(K-1)+I,J)
746 37 CONTINUE
747 36 CONTINUE
748 35 CONTINUE
749
750 C CALCULATE THE CURVATURE RATES ............
751
752 DO 39 J=1,NE
753 DO 40 I=1,ISIMP+1
754 JIK=1
755 POP=DABS(BMOM(I,J))/RMO-1.DO
756 IF(DABS(BMOM(I,J)).LE.RMO) POP=0.
757 IF(BMOM(I,J).LE.-RMO) JIK=-1
758 CURV(I,J)=DFLOAT(JIK)*RKO*POP**EN
759 40 CONTINUE
760 39 CONTINUE
761
762 RETURN
763 DEBUG SUBCHK
764 END
765

766 SUBROUTINE VELOC
767 C **********************************************************
768 C SUBROUTINE FOR EVALUATING VELOCITIES BY THE PRINCIPLE
769 C OF VIRTUAL VELOCITIES ...........
770 C (SIMPSON'S RULE USED FOR INTEGRATION )
771 C **********************************************************
772 IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
773 INCLUDE VISCO.PROCA
774
775 DO 42 I=1,NN
776 FXE(I)=0.
777 FX(I)=0.
778 42 CONTINUE
779
780 DO 44 K=1,NE
781 IF(I.EQ.1) GOTO 47
782 IF(I.EQ.ISIMP+1) GOTO 47
783 RI=DFLOAT(I)/2.+0.2
784 IR=IDINT(RI)
785 RR=DFLOAT(IR)+0.2
786 IF(RI.NE.RR) THEN
787 FXE(J+1)=FXE(J+1)+2.*XVM(ISIMP*(K-1)+I,J)*CURV(I,K)
788 ELSE
789 FXE(J+1)=FXE(J+1)+4.*XVM(ISIMP*(K-1)+I,J)*CURV(I,K)
790 END IF
791 GOTO 46
792 44 CONTINUE
793 46 CONTINUE
794 47 CONTINUE
795 46 CONTINUE
796 45 CONTINUE
797 47 CONTINUE
798 46 CONTINUE
799 45 CONTINUE
800 46 CONTINUE
801 47 CONTINUE
802 46 CONTINUE
803 45 CONTINUE
804 46 CONTINUE
805 FXE(IJ+1)=FXE(IJ+1)*(1./SIMP)*CL(K)/3.
806 FX(IJ+1)=FX(IJ+1)+FXE(IJ+1)
FXE(I+1) = 0.

CONTINUE

FXE(2) = FX(2)

DO 9 I = 3, NN
FXE(I) = FX(I)

DO 9 J = 2, I - 1
FXE(I) = FXE(I) - FXE(J) * DCOS(THETA(I-1) - THETA(J-1))

CONTINUE

CONTINUE

CALL ABVEL

RETURN

DEBUG SUBCHK

END

SUBROUTINE CHECK

********************************************************************************

SUBROUTINE TO CHECK WHETHER LAMDA MUST BE INCREMENTED
UP OR DOWN, OR WHETHER BISECTION CAN COMMENCE.....

********************************************************************************

IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)

INCLUDE VISCO.PROCA

ENER(2) = 0.

DO 60 I = 2, NN
ENER(2) = ENER(2) + 0.5 * GMASS(I, I) * (AFX(I))**2
CONTINUE

F(2) = ENER(2)

IF(ILAMDA.EQ.1) THEN
   INC = 1
   IF(ENER(2).GT.ENER(1)) DGAM = -DGAM
ELSE
   INC = 0
   IF(DGAM.GT.0.) THEN
      IF(ENER(2).LT.ENER(1)) INC = 1
   ELSE
      IF(ENER(2).GT.ENER(1)) INC = 1
   END IF
END IF

END IF

RETURN

DEBUG SUBCHK

END

SUBROUTINE BISEC

********************************************************************************
SUBROUTINE PERFORMING A BISECTION ALGORITHM ON THE DESIRED KINETIC ENERGY LEVEL

** implicit double precision*8(a-h,o-z) **

INCLUDE VISCO.PROCA

PRINT*,'F(1)='F(1)
A=F(1)-ENER(1)
IF(DABS(A).LE.TOL3) THEN
  ENER(2)=F(1)
  PRINT*,' GAM(1)='GAM(1)
  PRINT*,' ENER(2)='F(1)
  GAMM=GAM(1)
  DO 8 I=2,NN
    FXCOM(I,1)=GSTOR(I,1,1)
    FXCOM(I,2)=GSTOR(I,1,2)
    AFX(I)=DSQRT(FXCOM(I,1)**2+FXCOM(I,2)**2)
  CONTINUE
  GOTO 790
END IF

PRINT*,'F(2)='F(2)
B=F(2)-ENER(1)
IF(DABS(B).LE.TOL3) THEN
  ENER(2)=F(2)
  PRINT*,' GAM(2)='GAM(2)
  PRINT*,' ENER(2)='F(2)
  GAMM=GAM(2)
  GOTO 790
END IF

A=F(1)-ENER(1)
B=F(2)-ENER(1)
IF(A*B.GT.0.) THEN
  PRINT*,'ERROR EXIT -> F1*F2> 0 '
  PRINT*,' F(1)='F(1)
  PRINT*,' F(2)='F(2)
  PRINT*,' ENER(1)='ENER(1)
  PRINT*,' GAM(1), GAM(2)='GAM(1),GAM(2)
  DO 10 I=1,NN
    PRINT*,' VEL(I,1), VEL(I,2), FXCOM(I,1), FXCOM(I,2)
  CONTINUE
  STOP
END IF

DO 789 IJ=1,100
  IF(DABS(A).GT.TOL3.AND.DABS(B).GT.TOL3) THEN
    IF(DABS(GAM(1)-GAM(2)).LT.EPS1) THEN
      PRINT*,'TOO STEEP !'
      PRINT*,' GAM(1), GAM(2)='GAM(1),GAM(2)
      STOP
      END IF
  END IF
  GAM(3)=0.5*(GAM(1)+GAM(2))
B.16

GAMM = GAM(3)
CALL LOAD
CALL VELOC
DO 7 I=2,NN
GSTOR(3,I,1) = FXCOM(I,1)
GSTOR(3,I,2) = FXCOM(I,2)
7 CONTINUE
F(3) = 0.
DO 3 I=2,NN
F(3) = F(3) + 0.5*GMASS(I,I)*(AFX(I)**2
3 CONTINUE
C PRINT*, 'F(3) = ', F(3)
C = F(3) - ENER(1)

IF(A*C.LT.0.) THEN
GAM(2) = GAM(3)
GAMM = GAM(2)
F(2) = F(3)
DO 4 I=2,NN
GSTOR(2,I,1) = GSTOR(3,I,1)
GSTOR(2,I,2) = GSTOR(3,I,2)
4 CONTINUE
C PRINT*, 'F(2) = ', F(2)
ELSE
GAM(1) = GAM(3)
GAMM = GAM(1)
F(1) = F(3)
DO 5 I=2,NN
GSTOR(1,I,1) = GSTOR(3,I,1)
GSTOR(1,I,2) = GSTOR(3,I,2)
5 CONTINUE
C PRINT*, 'F(1) = ', F(1)
END IF
END IF
A = F(1) - ENER(1)
B = F(2) - ENER(1)

IF(DABS(A).LE.TOL3) THEN
ENER(2) = F(1)
C PRINT*, 'GAM(1) = ', GAM(1)
C PRINT*, 'ENER(2) = ', F(1)
GAMM = GAM(1)
GOTO 790
ELSE IF(DABS(B).LE.TOL3) THEN
ENER(2) = F(2)
C PRINT*, 'GAM(2) = ', GAM(2)
C PRINT*, 'ENER(2) = ', F(2)
GAMM = GAM(2)
GOTO 790
END IF
789 CONTINUE
790 CONTINUE
RETURN
DEBUG SUBCHK
END

SUBROUTINE INITI
**********************************************************
C SUBROUTINE DISPLAYING INITIAL MODE VELOCITIES .......
**********************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA
CALL PDATA
WRITE(IPRINT,14) T
14 FORMAT(1H1,2X,' RESULTS FOR TIME T =',D16.6,///,
&5X,'ZERO DISPLACEMENTS AT TIME T=0 ',',
&//,20X,'VELOCITIES',//,5X,'NODE',9X,'X',14X,'Y',/
1013
1014 DO 16 I=2,NN
1015 IF(ENER(3).LT.l.D-3) THEN
1016 WRITE(IPRINT,15) I,VEL(I,1),VEL(I,2)
1017 ELSE
1018 WRITE(IPRINT,15)I,FXCOM(I,1),FXCOM(I,2)
15 FORMAT(lH ,4X,I3,5X,Dll.5,4X,Dll.5,/) END IF
1020 16 CONTINUE
1021 PRINT* , , LAMDA =' ,GAMM
1022 PRINT* ' KINETIC ENERGY =' ,ENER(2),ENER(1)
1023 DENER(2)=-2.*GAMM*ENER(2)
1024 DEOLD=DENER(2)
1025 DENER(1)=DEOLD
1026 EOLD=ENER(1)
1027 E=ENER(3)
1028 IF(NE.EQ.1) E=O.
1029 FTIME=ENER(2)/(-1.*DENER(2))
1030 IF(FTIME.LE.0. OR.FTIME.GT.0.05) THEN
1031 FTIME=0.05
1032 PRINT*,'WARNING: FTIME ESTIMATE SET TO 0.05 '
1033 END IF
1034 PRINT*,' ' END IF
1035 PRINT*,' ' END IF
1036 PRINT*,' ' END IF
1037 PRINT*,' ' END IF
1038 PRINT*,' ' END IF
1039 DT=FTIME/TS
1040 CALL LOAD
1041 CALL LOAD
1042 DO 18 I=1,NE
1043 22 FORMAT(1H1,4X,I3,5X,Dll.5,/) WRITE(IPRINT,22)I,BMOM(I,1)
1045 18 CONTINUE
1046
1047 DO 17 I=2,NN
1048 VELO(I,1)=FXCOM(I,1)
1049 VELO(I,2)=FXCOM(I,2)
1050 17 CONTINUE
1051 IFLAG=O
**SUBROUTINE FINIT**

```
C******************************************************************************
C SUBROUTINE DISPLAYING MODE VELOCITIES AND DISPLACEMENTS FOR TIME STEP T .......
C******************************************************************************
C IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA

CALL PDATA
DO 10 I=2,NN
DISP(I,1)=DISP(I,1)+(VELO(I,1)+FXCOM(I,1))*DT/2.
DISP(I,2)=DISP(I,2)+(VELO(I,2)+FXCOM(I,2))*DT/2.
CONTINUE

IF(ENER(3).LT.1.D-3.AND.T.EQ.DT) THEN
 DO 20 J=1,2
 DO 20 I=2,NN
 DISP(I,J)=0.
 CONTINUE
 END IF

IF(IR.EQ.IPLTS) THEN
 WRITE(IPRINT,14) T
 FORMAT(1H1,2X,' RESULTS FOR TIME T      ',D16.6,IIIII,20X,
 'VELOCITIES',24X,'DISPLACEMENTS',I5X,'NODE',9X,'X',14X,'Y',17X,'X',14X,'Y',/)
 DO 16 I=2,NN
 WRITE(IPRINT,15)I,FXCOM(I,1),FXCOM(I,2),DISP(I,1),DISP(1,2)
 FORMAT(1H    ,4X,13,1X,2(4X,D11.5),3X,2(4X,D11.5),/)
 CONTINUE
 PRINT*,'    LAMDA = ',GAMM
 PRINT*,'KINETIC ENERGY = ',ENER(2)
 PRINT*,' PULS(NN) = ',PULS(NN)

T=T-DT
CALL LOAD
T=T+DT
DO 18 I=1,NE
 FORMAT(1H ,4X,I3,5X,D11.5,/)
 WRITE(IPRINT,22) I,BMOM(I,I)
 CONTINUE

END IF
RETURN
DEBUG SUBCHK
END
```
SUBROUTINE PDATA

***********************************************************************
DATA OF SELECTED TIME STEPS STORED FOR PLOTTING PURPOSES...........
***********************************************************************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA

IPLTS=IPLTS+1

IF(IPLTS.EQ.0) THEN
  DO 1 I=1,NN
  X(I,1)=COORDX(I)
  Y(I,1)=COORDY(I)
  CONTINUE
  CALL DPRINT
  ITP=1
  GOTO 111
END IF

IF(IEXIT.EQ.1) THEN
  DO 2 I=1,NN
  X(I,ITP)=XGEO(I)
  Y(I,ITP)=YGEO(I)
  CONTINUE
  GOTO 222
END IF

RI=DFLOAT(IPLTS)/DFLOAT(IFREQ)+O.001
IR=IDINT(RI)
IR=IR*IFREQ

IF(IR.EQ.IPLTS) THEN
  DO 3 I=1,NN
  X(I,ITP)=XGEO(I)
  Y(I,ITP)=YGEO(I)
  CONTINUE
  GOTO 111
END IF

ELSE
  GOTO 222
END IF

ITP=ITP+1
CONTINUE

RETURN
DEBUG SUBCHK

END

***********************************************************************
SUBROUTINE DISPLAYING ESTIMATED FINAL TIME AND FINAL DISPLACEMENTS
***********************************************************************
SUBROUTINE FINAL

***********************************************************************

RETURN
DEBUG SUBCHK
END

SUBROUTINE FINAL
************
IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
INCLUDE VISCO.PROCA

IEIXT=1

EXT=1./GAMM
EXT=2*ENER(2)/(-ENER(1))
FTIME=T+EXT
DO 20 I=2,NN
DISP(I,1)=DISP(I,1)+0.5*EXT*FXCOM(I,1)
XGEO(I)=COORDX(I)+DISP(I,1)
DISP(I,2)=DISP(I,2)+0.5*EXT*FXCOM(I,2)
YGEO(I)=COORDY(I)+DISP(I,2)
CONTINUE
CALL PDATA
WRITE(IPRINT,14) FTIME
DO 16 I=2,NN
WRITE(IPRINT,15)I,DISP(I,1),DISP(I,2)
CONTINUE
PRINT*, ' EXTENSION IN TIME :' ,EXT
PRINT*, ' FINAL TIME :' ,FTIME
IF DATA IS TO BE WRITTEN TO A FILE
PRINT*, ' X & Y COORDS FOR PLOTTING'
DO 4 J=1,ITP
DO 5 I=1,NN
PRINT*,X(I,J),Y(I,J)
WRITE(IDATA,100) X(I,J),Y(I,J)
CONTINUE
CONTINUE
PRINT*, ' ITP = ',ITP
RETURN

COMMENTS

*****************************************************
PROCA
PARAMETER IREAD=10
PARAMETER IPRINT=6
PARAMETER ISIMP=10
PARAMETER NEL=14
PARAMETER NNO=NEL+1
PARAMETER EPSL=1.E-15
PARAMETER EPS2=0.000001
PARAMETER NTFUN=1

**********************************************************
PARAMETER NPLOD=1
PARAMETER NPTS =10
COMMON /BLK1/ HH,BB,YSTRS,EPSIO,EN,AA,RMO,E,EE,NE,NN,INC,
& RKO,POP,RI,IR,RR,IX,TITLE,IFLAG,ITER,DIFF,CDIFF,LOOP,A,B,C,
& COORDX(NNO),COORDY(NNO),VEL(NNO,2),CENT(NNO),ILAMDA,
& FORC(NNO,2),CL(NEL),UNITMX(NNO,NEL),UNITMY(NNO,NEL),
& XINFX(ISIMP*NEL+1,NEL),XINFY(ISIMP*NEL+1,NEL),
& BMOM(ISIMP+1,NEL),CURV(ISIMP+1,NEL),FX(NNO),FXE(NNO),
& GAMM,GAM(3),DGAM,F(3),ENER(3),
& DTIME,T,DT,EOLD,DEOLD,DENER(2),TZERO,VELO(NNO,2),
& DISP(NNO,2),EXT,PSI,PSID,PSIN,TOL,TS,KS,KIN,VO,
& TOL1,TOL2,TOL3,GSTOR(3,NNO,2),GMASS(NNO,NNO),
& VM(NNO,NEL),XVM(ISIMP*NEL+1,NEL),FXCOM(NNO,NNO),AFX(NNO),
& AVEL(NNO),AVELO(NNO),SIMP,GEOM,DUMMY,XINC(NNO),
& YINC(NNO),XGEO(NNO),YGEO(NNO),THETA(NEL),XCOORD(NNO),
& YCOORD(NNO),X(NNO+2,99),Y(NNO+2,99),ITP,EEXIT,IPLTS,IFREQ,
& PIE,TDEL,P(NNO,NTFUN),FACT(NTFUN,NPTS),FACF(NTFUN,NPTS),
& PULS(NNO),PULFOR,IPTS,

CHARACTER TITLE*72
CHARACTER PULFOR*10
CHARACTER DUMMY*4
CHARACTER GEOM*5

END
APPENDIX C

Course Work

In compliance with the requirements for the Master's degree, approved course work with a value of twenty-one credits was done in addition to the thesis. The courses are briefly described below:

(i) CE 551 (a): FRAME ANALYSIS
    2 credits
    The application of the force method of analysis to framed structures of straight and curved members. The stability of equilibrium of framed structures.

(ii) CE 551 (b): INTRODUCTION TO THE THEORY OF ELASTICITY
    2 credits
    Stress, strain, equilibrium, strain displacement relations. Elastic constants. Solutions of simple boundary value problems in plane stress and plane strain.

(iii) CE 551 (c): PLATES AND SHELLS
    2 credits
(iv) CE 522 (a): **INTRODUCTION TO FINITE ELEMENT METHOD**

3 credits


(v) CE 522 (b): **FINITE ELEMENT ANALYSIS**

3 credits


(vi) AM 343: **NUMERICAL ANALYSIS**

4 credits

Theory and practice of numerical methods including approximate solution of nonlinear equations, interpolation, numerical integration and differentiation, numerical solution of ordinary differential equations.

(vii) AM 401: **MATHEMATICAL METHODS**

6 credits