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.

University of Cape Town

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 technique. 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.

signature removed

Signed by candidate

R. Trossbach

September 1984

to my parents

ACKNOWLEDGEMENTS

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

NOMENCLATURE

UPPER CASE CHARACTERS

F	pulse
I_0	initial impulse
K	kinetic energy
$\dot{K}$	time rate of change of kinetic energy
M	moments
M_0	yield moment
T	time function in mode analysis
X	global cartesian X - axis
Y	global cartesian Y - axis
V	local velocity

LOWER CASE CHARACTERS

h	rectangular section depth
ℓ	length of an element
n	power in constitutive relation
s	spatial variable (two-dimensional)
t	time variable
u,v	displacement components
$\dot{u},\dot{v}$	velocity components
$\ddot{u},\ddot{v}$	acceleration components

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
$\underset{\sim}{P}$	load vector
$\underset{\sim}{u}$, $\underset{\sim}{v}$	displacement vectors
$\dot{\underset{\sim}{u}}$, $\dot{\underset{\sim}{v}}$	velocity vectors
$\ddot{\underset{\sim}{u}}$, $\ddot{\underset{\sim}{v}}$	acceleration vectors
$\underset{\sim}{\phi}$	mode shape vector

SPECIAL SYMBOLS

$[\ \ ]$	a matrix
$\underset{\sim}{u}$	a vector u
$\dot{v}$	the differential of v with respect to time
$\|c\|$	the absolute value of c
d	differentiation with respect to
∂	partial differentiation with respect to

GREEK CHARACTERS

γ	specific mass
Δ	increment in
$\dot{\varepsilon}$	axial strain rate
$\dot{\varepsilon}_0$	strain rate material constant
θ	rotation
$\dot{\theta}$	rotation rate
$\dot{\kappa}$	curvature rate
$\dot{\kappa}_0$	curvature rate material constant
μ	stress matching factor
ν	power matching factor
σ_0	yield stress
ϕ	mode shape
λ	scalar with units [^{1}/time]

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

$$\begin{aligned} \dot{\varepsilon} &= \dot{\varepsilon}_0\left(\frac{\sigma}{\sigma_0} - 1\right)^n && \text{for } \sigma > \sigma_0 \\ \dot{\varepsilon} &= 0 && \text{for } 0 < \sigma < \sigma_0 \end{aligned} \qquad (1.1)$$

where $\dot{\varepsilon}$, σ are strain rate and stress respectively and $\dot{\varepsilon}_0$, σ_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 $\dot{\underset{\sim}{u}}^m(s,t)$ which is a separable function of space and time. Performing separation of variables, the mode velocity field can be written as

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

where $\underset{\sim}{\phi}(s)$ is the mode shape and a function of space only, and $T(t)$ is a scalar function of time. They showed that actual velocities $\dot{\underset{\sim}{u}}(s,t)$ in an impulsive loading case with initial velocities $\dot{\underset{\sim}{u}}(s,o) = \dot{\underset{\sim}{u}}^0(s)$ converge onto a mode solution; once $\underset{\sim}{\phi}(s)$ 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 (equn (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

$$\dot{\varepsilon} = \dot{\varepsilon}_0 \left(\frac{\sigma}{\mu\ \sigma_0}\right)^{\nu n} \tag{1.3}$$

where μ,ν 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.

$$\dot{\underset{\sim}{u}}(s,t) = \underset{\sim}{\phi}^t\ T^t \quad . \tag{1.4}$$

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.

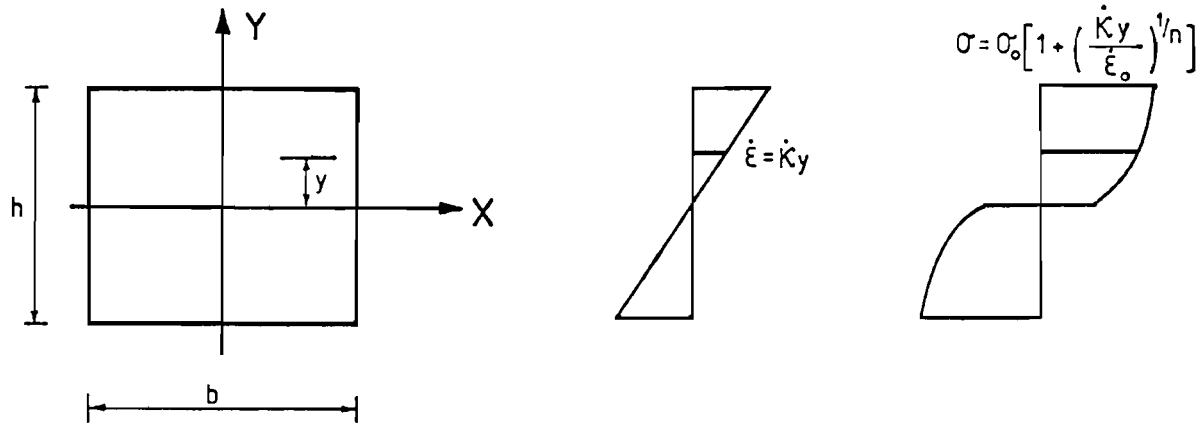
Griffin and Martin [20] used both the instantaneous mode technique and a direct method of analysis based on the Tamuzh principle to estimate the response of beams and frames. For both cases they used the homogeneous viscous law (equn (1.3)), suitably matched on the rigid-viscoplastic relation. For analyses 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 μ and ν, 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 equn (1.1) as

$$\dot{\kappa} = \dot{\kappa}_0 \left(\frac{M}{M_0} - 1\right)^n \qquad \text{for } M > M_0 \qquad (1.5)$$

$$\dot{\kappa} = 0 \qquad \text{for } 0 < M_0 < M \ ,$$

where $\dot{\kappa}_0$ 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 $\dot{\kappa}_0$ are given below.



(a) cross-section (b) strain rate (c) stress

Figure 1 Beam of Rectangular Cross-Section

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.

$$\dot{\varepsilon} = \dot{\kappa}\, y \qquad , \qquad (1.6)$$

where $\dot{\kappa}$ is the curvature rate. Rewriting equn (1.1) in terms of stress σ and substituting for strain rate $\dot{\varepsilon}$, we obtain

$$\sigma = \sigma_0 \left\{1 + \left(\frac{\dot{\kappa}\, y}{\dot{\varepsilon}_0}\right)^{1/n}\right\} \quad , \qquad (1.7)$$

which is illustrated in Fig 1.1(c).

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

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

We define the yield moment M_0 as the moment resulting from a yield stress σ_0 across the section and write

$$M_0 = 2\int_0^{h/2} b\, \sigma_0\, y\, dy = \sigma_0\, b\, h^2/4 \quad . \qquad (1.9)$$

In order to derive the curvature rate constant $\dot{\kappa}_0$, we write equn (1.8) as

$$M = 2\int_0^{h/2} b\, \sigma_0 \left\{1+\left(\frac{\dot{\kappa}\, y}{\dot{\varepsilon}_0}\right)^{1/n}\right\} y\, dy$$

$$= M_0 \left\{1+\left(\frac{2n}{2n+1}\right)\left(\frac{\dot{\kappa}\, h}{2\,\dot{\varepsilon}_0}\right)^{1/n}\right\} \quad . \qquad (1.10)$$

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

$$\dot{\kappa} = \frac{2\dot{\varepsilon}_0}{h}\left(\frac{2n+1}{2n}\right)^n \left(\frac{M}{M_0} - 1\right)^n \quad . \qquad (1.11)$$

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

$$\dot{\kappa}_0 = \frac{2\dot{\varepsilon}_0}{h}\left(\frac{2n+1}{2n}\right)^n \quad , \qquad (1.12)$$

where $\dot{\varepsilon}_0$ 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

$$F(t) = I_0 \delta(t) \quad . \tag{2.1}$$

Integrating over the initial instant of time, we obtain

$$\int_{0^-}^{0^+} F(t)\, dt = I_0 \int_{0^-}^{0^+} \delta(t)\, dt = I_0 \quad , \tag{2.2}$$

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 $\dot{\underset{\sim}{u}}(s,o)$ 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 equn (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

$$K = \tfrac{1}{2}\, \dot{\underset{\sim}{u}}^{T} [G] \dot{\underset{\sim}{u}} \quad . \tag{2.3}$$

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

$$\dot{\underset{\sim}{u}}^m (s,t) = \underset{\sim}{\phi}(s)\, T(t) \quad . \tag{2.4}$$

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

$$\ddot{u}^m = -\lambda\, \dot{\underset{\sim}{u}}^m \quad , \tag{2.5}$$

where

$$\lambda = -\frac{\dot{T}}{T} \quad . \tag{2.6}$$

The implementation of the technique requires that algorithms should exist for the determination of the mode shape $\underset{\sim}{\phi}(s)$ 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 $\dot{K}$.

The determination of the primary mode shape can be achieved by the principle that, if a structure is loaded by $\lambda[G]\dot{\underset{\sim}{u}}^m$, the resulting velocities are the mode velocities $\dot{\underset{\sim}{u}}^m$ (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 ϕ 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 ϕ.

In contrast to the rigid-viscoplastic case, the homogeneous viscous case thus permits the determination of the mode shape $\underset{\sim}{\phi}$ and λ to be performed in two independent calculations. The mode shape can be determined by loading the structure with $[G]\underset{\sim}{\phi}$ which result in velocities $\dot{\underset{\sim}{u}}$; 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 $\underset{\sim}{\phi}$ has been determined, the right hand side of equn (2.6) can be found, giving λ. It should be re-emphasized at this point that for the homogeneous case, λ 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

$$\dot{\underset{\sim}{u}}^{m}(s,t) = \underset{\sim}{\phi}(s,K(t)) \quad . \tag{2.7}$$

For this class of problems, the mode shape $\underset{\sim}{\phi}$ and λ 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

$$\dot{K} = \dot{\underset{\sim}{u}}^{T} [G] \ddot{\underset{\sim}{u}} \quad , \tag{2.8}$$

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

$$\dot{K} = -2 \lambda K \quad , \tag{2.9}$$

where

$$\lambda = \lambda(K) \quad . \tag{2.10}$$

In the following section we present an algorithm whereby λ and $\underset{\sim}{\phi}$ can be determined for a given value of K. We use the value of λ 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 λ and $\dot{\underset{\sim}{u}}$. The inner loop iterates on λ 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 $\lambda[G]\dot{\underset{\sim}{u}}$ applied to the structure results in a velocity field $\dot{\underset{\sim}{u}}$ which is identical to that used in the load. This assures that equn (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 λ and guess a set of velocities, denoted by λ^k and $\dot{\underset{\sim}{u}}^i$ respectively.

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

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

Step 3 : Determine the kinetic energy K^{i+1} corresponding to the velocities $\dot{\underset{\sim}{u}}^{i+1}$ 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 λ, replacing λ^k by λ^{k+1}, say, and repeat steps 1, 2 and 3 for λ^{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 $\dot{\underset{\sim}{u}}^i$ by the latest set of $\dot{\underset{\sim}{u}}^{i+1}$ determined from step 4.

Iterations are continued until a satisfactory convergence on the velocities has been attained assuring that equn (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 Δt. 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

$$\underset{\sim}{P} = \lambda^k \, [G] \, \dot{\underset{\sim}{u}}^i \quad , \tag{2.11}$$

where [G] is the diagonal mass matrix and the first estimates of λ^k and $\dot{\underset{\sim}{u}}^i$ 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:

$$\underset{\sim}{M} = [m] \, \underset{\sim}{P} \quad , \tag{2.12}$$

where $\underset{\sim}{P}$ is the load vector given by equn (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 ℓ_e, and M_a, M_b are the nodal moments, the bending moment distance s from node a is given by

$$M(s) = M_a \left(1 - \frac{s}{\ell_e}\right) + M_b \left(\frac{s}{\ell_e}\right) \quad . \tag{2.13}$$

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 $\underset{\sim}{P}$. 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 $\underset{\sim}{P}$. With the velocities $\dot{u}_j$ and curvature rates $\dot{\kappa}$ as the kinematic system, and a unit value of the j-th component of $\underset{\sim}{P}$ together with its associated m_j as the static system, the principle of virtual velocities gives the j-th component $\dot{u}_j$ of $\underset{\sim}{\dot{u}}$ as

$$\dot{u}_j = \sum_{\text{elements}} \int_{\ell_e} m_j \dot{\kappa} \, ds \quad , \tag{2.14}$$

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

In Step 4 we perform a standard bisection algorithm, the parameters of which are λ and kinetic energy K. Before the bisection can proceed, however, a pair of λ values must be available for which the sets of computed velocities $\underset{\sim 1}{\dot{u}}^{i+1}$ and $\underset{\sim 2}{\dot{u}}^{i+1}$ say, result in kinetic energy levels which are bigger and smaller than K* respectively. The bisection algorithm is started by changing λ^k , evaluating new loads resulting in new velocities with corresponding energy K. Velocities $\underset{\sim}{\dot{u}}^{i+1}$ are recomputed for every new λ^{k+1} 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 $\underset{\sim}{\dot{u}}^{i+1}$ and λ^{k+1} from Step 4, giving $\lambda^{k+1}[G]\underset{\sim}{\dot{u}}^{i+1}$. 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 $\dot{u}_a$ while the remainder of the beam is initially stationary. The initial velocity $\dot{u}_a$ is given by

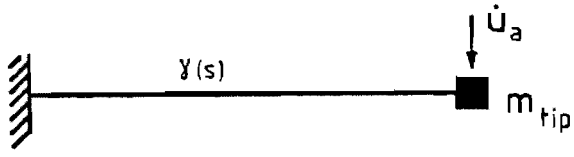
$$\dot{u}_a = \frac{I}{m_{tip}} \quad .$$

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 $\dot{u}_a$ is adjusted so that the initial momenta for the actual beam

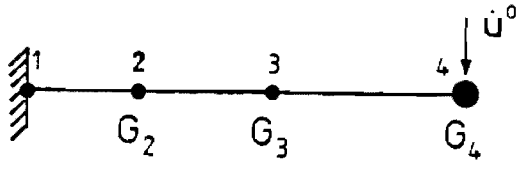
and the discretised beam are identical, i.e.

$$\dot{u}^0 = \frac{m_{tip}\,\dot{u}_a}{G_4} \quad .$$

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



(a) actual system



(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

$$\dot{\underset{\sim}{u}}^T [G] \dot{\underset{\sim}{u}}^0 = \dot{\underset{\sim}{u}}^T [G] \dot{\underset{\sim}{u}} \quad , \tag{2.15}$$

where $[G]$, $\dot{\underset{\sim}{u}}$ and $\dot{\underset{\sim}{u}}^0$ 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 $\dot{\underset{\sim}{u}}^0$.

Step 2 : Calculate velocities $\dot{\underset{\sim}{u}}_1$ from mode algorithm in Section 2.2.

Step 3 : Compute $\xi = \dfrac{\dot{\underset{\sim}{u}}_1^T [G] \dot{\underset{\sim}{u}}^0}{\dot{\underset{\sim}{u}}_1^T [G] \dot{\underset{\sim}{u}}_1}$ which represents the out-of-balance condition.

Step 4 : Multiply velocities $\dot{\underset{\sim}{u}}_i$ from step 2 by ξ to obtain scaled velocities $\dot{\underset{\sim}{u}}_{i+1}$.

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

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

This iteration is continued until ξ 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

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

then its derivative with respect to time has the form

$$\dot{K} = \frac{dK}{dt} = \dot{\underset{\sim}{u}}^{T} [G] \ddot{\underset{\sim}{u}} \quad , \tag{2.16}$$

where $\ddot{\underset{\sim}{u}}$ is the vector of nodal accelerations.

Replacing $\ddot{\underset{\sim}{u}}$ by $-\lambda\dot{\underset{\sim}{u}}$, we rewrite $\dot{K}$ as

$$\begin{aligned} \dot{K} &= \dot{\underset{\sim}{u}}^{T} [G] (-\lambda \dot{\underset{\sim}{u}}) \\ &= -2 \lambda K \quad . \end{aligned} \tag{2.17}$$

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

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

$$K^{t+1} = K^{t} + \dot{K}^{t} \Delta t \quad , \tag{2.18}$$

where the superscripts t+1 and t denote instants in time and Δt 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

$$K^{t+1} = K^{t} + \frac{\Delta t}{2} (\dot{K}^{t} + \dot{K}^{t+1}) \quad . \tag{2.19}$$

In applying this equation we assume that K^t and $\dot{K}$ are known. This is not sufficient information to compute K^{t+1}, since $\dot{K}^{t+1}$ 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

$$K_{i+1}^{t+1} = K^t + \frac{\Delta t}{2} (\dot{K}^t + \dot{K}_i^{t+1}) \quad . \tag{2.20}$$

The initial value of $\dot{K}_1^{t+1}$ is taken as $\dot{K}^t$, whereafter the velocities for K_1^{t+1}, are computed and $\dot{K}_1^{t+1}$ 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 evergy 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 Δt_f was done. Forward integration is stopped when one of the following conditions becomes true:

$$\begin{aligned} &(a) \quad K < \varepsilon_K \\ &(b) \quad K < \dot{K}\,\Delta t \quad . \end{aligned} \tag{2.21}$$

ε_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 Δt_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).

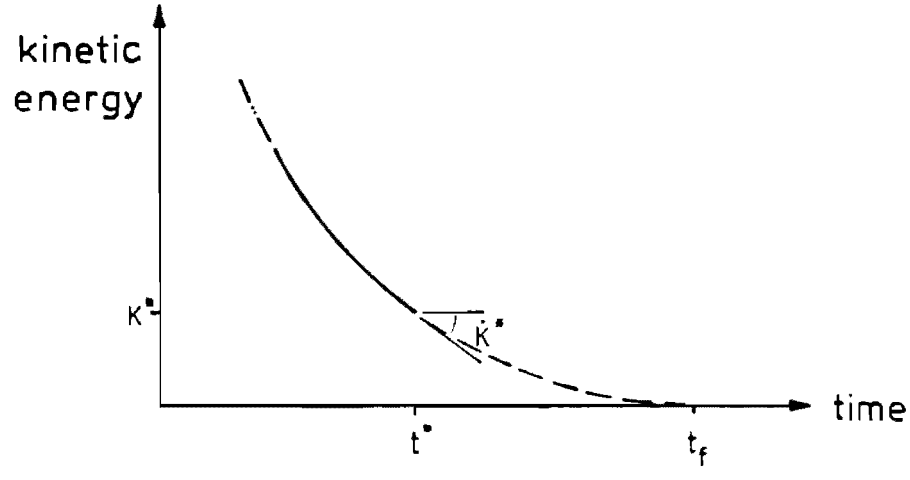


Figure 2.2 Extrapolation of Kinetic Energy Curve

The shape of the energy-time curve is nearly parabolic and thus we assume that for $t > t^*$

$$\begin{aligned} K &= -\tfrac{1}{2} a t^2 + bt + c \\ \dot{K} &= -at + b \end{aligned} \qquad (2.22)$$

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

$$\begin{aligned} K(t^*) &= K^* \\ \dot{K}(t^*) &= \dot{K}^* \\ K(t_f) &= 0 \\ \dot{K}(t_f) &= 0 \end{aligned} \qquad (2.23)$$

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

$$a = \frac{\dot{K}^*}{\Delta t_f} \quad , \quad b = \frac{\dot{K}^*}{\Delta t_f} t_f \quad , \quad c = -\tfrac{1}{2} \frac{\dot{K}^*}{\Delta t_f} t_f^2$$

and finally

$$\Delta t_f = -\frac{2K^*}{\dot{K}^*} \qquad (2.24)$$

Once the kinetic energy is low enough to satisfy either of conditions (2.21), equn (2.24) is used to evaluate Δt_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.

Chapter 3

EXTENSION TO GEOMETRICALLY NONLINEAR CASES

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.

$$\dot{\varepsilon} = 0 \quad , \tag{3.1a}$$

which implies that

$$\dot{\ell} = 0 \quad , \tag{3.1b}$$

where ℓ is the length of any element.

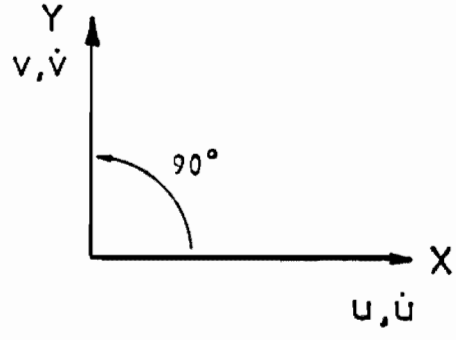


Figure 3.1 Cartesian Coordinate System

We define $\underset{\sim}{\dot{u}}$, $\underset{\sim}{\dot{v}}$ as the velocity components in the positive x and y directions of a Cartesian co-ordinate system; similarly the

displacement components $\underline{u}$ and $\underline{v}$ are defined. (see Fig 3.1).

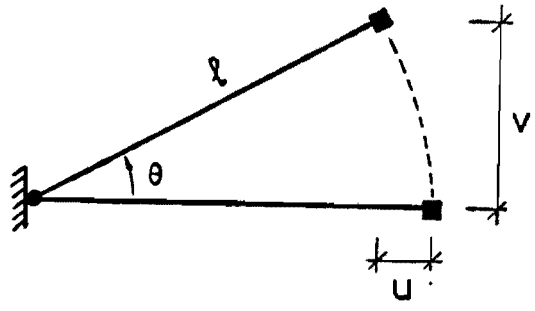


Figure 3.2 1 DOF System

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

$$\ell^2 = v^2 + (\ell-u)^2 \quad . \qquad (3.2)$$

Differentiating with respect to time, we obtain

$$2\ell\dot{\ell} = 2v\dot{v} + 2(\ell-u)\,(\dot{\ell}+\dot{u}) \quad ; \qquad (3.3)$$

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

$$\dot{u} = \dot{v}\,\frac{-v}{\ell-u} = -\dot{v}\,\tan\theta \quad .$$

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 $\underset{\sim}{\dot{u}}$ and $\underset{\sim}{\dot{v}}$ 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

$$\dot{u} = -V\sin\theta \tag{3.6a}$$

$$\dot{v} = V\cos\theta \quad . \tag{3.6b}$$

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

$$\begin{aligned} \ddot{u} &= -\dot{V}\sin\theta - V\dot{\theta}\cos\theta \quad , \\ \ddot{v} &= \dot{V}\cos\theta - V\dot{\theta}\sin\theta \quad , \end{aligned} \tag{3.7}$$

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

$$\dot{\theta} = \frac{V}{\ell} \quad . \tag{3.8}$$

Substituting the latter into equns (3.7) we get

$$\ddot{u} = -\dot{V}\sin\theta - \frac{V^2}{\ell}\cos\theta$$

$$\ddot{v} = \dot{V}\cos\theta - \frac{V^2}{\ell}\sin\theta \quad , \tag{3.9}$$

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

$$\dot{V} = -\lambda V \tag{3.10}$$

and substitute it into equns (3.9) to get

$$\ddot{u} = \lambda V\sin\theta - \frac{V^2}{\ell}\cos\theta$$

$$\ddot{v} = -\lambda V\cos\theta - \frac{V^2}{\ell}\sin\theta \quad . \tag{3.11}$$

Looking again at equns (3.7) it should be noted that the right hand sides contain derivatives of both V and θ. Whereas the term involving V can be seen as an ordinary "velocity rate" expression, the term containing $\dot{\theta}$ 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.

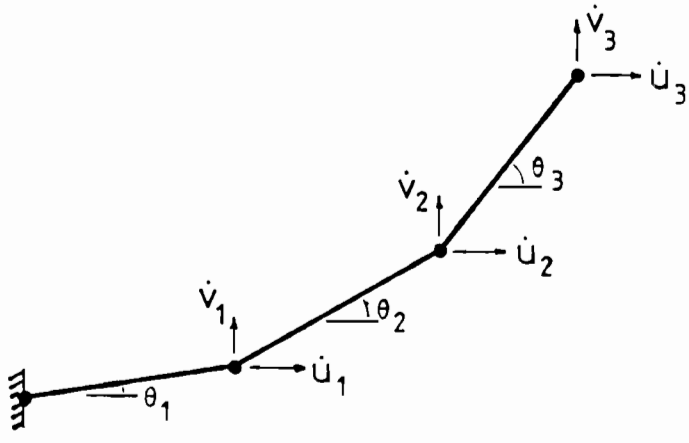


Figure 3.3 Multi DOF System

We redefine $\underset{\sim}{V}$ 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.

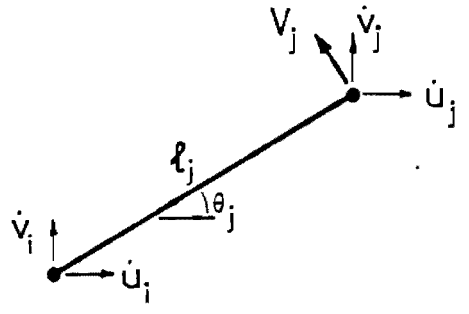


Figure 3.4 Element j

We can write the constraint equation as follows,

$$2\ell_j\dot{\ell}_j = (\dot{v}_j-\dot{v}_i)\cos\theta_j + (\dot{u}_j-\dot{u}_i)\sin\theta = 0 \qquad (3.12)$$

and

$$(\dot{u}_j-\dot{u}_i)^2 + (\dot{v}_j-\dot{v}_i)^2 = V_j^2 \qquad (3.13)$$

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

$$\dot{u}_j = \dot{u}_i - V_j\sin\theta_j \qquad (3.14a)$$

$$\dot{v}_j = \dot{v}_i + V_j\cos\theta_j \qquad (3.14b)$$

The rotation rate $\dot{\theta}_j$ can be expressed as

$$\dot{\theta}_j = \frac{\dot{v}_j - \dot{v}_i}{\ell_j \cos\theta_j} = \frac{V_j}{\ell_j} \tag{3.15}$$

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

$$\ddot{u}_j = \ddot{u}_i + \lambda V_j \sin\theta_j - \frac{V_j^2}{\ell_j} \cos\theta_j \tag{3.16a}$$

$$\ddot{v}_j = \ddot{v}_i - \lambda V_j \cos\theta_j - \frac{V_j^2}{\ell_j} \sin\theta \quad . \tag{3.16b}$$

Equations (3.14) and (3.16) are the relevant velocity and acceleration equations used in the formulation of geometric nonlinearity. Once the local velocities $\underset{\sim}{V}$ 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

$$\hat{V}_j = (\dot{u}_j^2 + \dot{v}_j^2)^{1/2} \quad . \tag{3.17}$$

The kinetic energy is then given by

$$K = \tfrac{1}{2} \underset{\sim}{\hat{V}}^T [G] \underset{\sim}{\hat{V}} \quad . \tag{3.18}$$

The load vector $\underset{\sim}{P}$ is divided up into a horizontal and vertical component respectively, giving

$$\begin{aligned} \underset{\sim x}{P} &= \lambda [G] \underset{\sim}{\dot{u}} \\ \underset{\sim y}{P} &= \lambda [G] \underset{\sim}{\dot{v}} \end{aligned} \quad , \tag{3.19}$$

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

$$\underset{\sim}{M} = [m_x] \underset{\sim y}{P} + [m_y] \underset{\sim x}{P} \quad , \tag{3.20}$$

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 $\underset{\sim}{X}$, whose components X_j act transversely to elements j as shown in Fig 3.5.

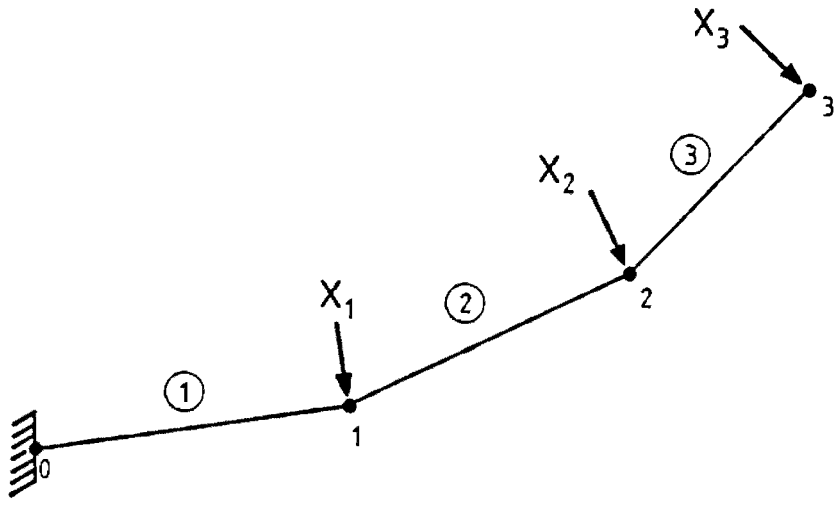


Figure 3.5 **Transverse loads $\underset{\sim}{X}$**

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 $\underset{\sim}{X}$. 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

$$V_j' = \sum_{\text{elements}} \int_{\ell_e} m_{t_j} \dot{\kappa} \, ds \quad , \tag{3.21}$$

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

$$V_j' = V_j + \sum_{i=1}^{j-1} V_i \cos(\theta_j - \theta_i) \quad , \tag{3.22}$$

where V_j is the local velocity at node j. Once the complete vector $\underset{\sim}{V}'$ 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

$$V_1 = V_1' \quad , \tag{3.23}$$

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 $\underset{\sim}{\hat{V}}$, we write

$$\underset{\sim}{\hat{V}}^T[G]\ \underset{\sim}{\hat{V}}^0 = \underset{\sim}{\hat{V}}^T[G]\ \underset{\sim}{\hat{V}} \tag{3.24}$$

where $\underset{\sim}{\hat{V}}^T$ is the transpose of the vector of global velocities and $\underset{\sim}{\hat{V}}^0$ is the set of the global initial velocities.

In the next section we proceed with the adaptation of the forward intergration 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

$$K_{i+1}^{t+1} = K^t + \frac{\Delta t}{2}(\dot{K}^t + \dot{K}_i^{t+1}) \quad . \tag{3.25}$$

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

$$\underset{\sim}{u}^{t+1} = \underset{\sim}{u}^t + \frac{\Delta t}{2}(\dot{\underset{\sim}{u}}^t + \dot{\underset{\sim}{u}}^{t+1}) \tag{3.26a}$$

$$\underset{\sim}{v}^{t+1} = \underset{\sim}{v}^t + \frac{\Delta t}{2}(\dot{\underset{\sim}{v}}^t + \dot{\underset{\sim}{v}}^{t+1}) \tag{3.26b}$$

In applying equns (3.26) we assume that $\dot{\underset{\sim}{u}}^t$, $\dot{\underset{\sim}{v}}^t$ and $\underset{\sim}{u}^t$, $\underset{\sim}{v}^t$ are known.This is not sufficient information to compute $\dot{\underset{\sim}{u}}^{t+1}$, $\dot{\underset{\sim}{v}}^{t+1}$ from equns (3.26), however, and an iterative scheme must be used. We put

$$\underset{\sim i+1}{u}^{t+1} = \underset{\sim}{u}^t + \frac{\Delta t}{2}(\dot{\underset{\sim}{u}}^t + \dot{\underset{\sim i}{u}}^{t+1}) \tag{3.27a}$$

$$\underset{\sim i+1}{v}^{t+1} = \underset{\sim}{v}^t + \frac{\Delta t}{2}(\dot{\underset{\sim}{v}}^t + \dot{\underset{\sim i}{v}}^{t+1}) \quad . \tag{3.27b}$$

where subscript i indicated the i-th iteration.

Equns (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 $\dot{K}^{t}$, $\dot{\underset{\sim}{u}}^{t}$ and $\dot{\underset{\sim}{v}}^{t}$ as the initial values for $\dot{K}_{i}^{t+1}$, $\dot{\underset{\sim}{u}}_{i}^{t+1}$ and $\dot{\underset{\sim}{v}}_{i}^{t+1}$; thereafter we compute the mode velocities for K_{1}^{t+1} and $\underset{\sim}{u}_{1}^{t+1}$, $\underset{\sim}{v}_{1}^{t+1}$. Subsequently $\dot{K}_{1}^{t+1}$ 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 $\underset{\sim}{u}_{2}^{t+1}$ and $\underset{\sim}{v}_{2}^{t+1}$. This process is repeated until satisfactory convergence in the values of K^{t+1} and $\underset{\sim}{u}^{t+1}$, $\underset{\sim}{v}^{t+1}$ 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_{i+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 $-\lambda\dot{u}$, $\dot{K}$ could be written as

$$\dot{K} = -2 \lambda K \quad . \tag{3.28}$$

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 λ and the velocities, but also contain a term involving rotation rate $\dot{\theta}$ (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

$$K = \tfrac{1}{2} (\underset{\sim}{\dot{u}}^T [G] \underset{\sim}{\dot{u}} + \underset{\sim}{\dot{v}} [G] \underset{\sim}{\dot{v}}) \quad , \tag{3.29}$$

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

$$\dot{K} = \underset{\sim}{\dot{u}}^T [G] \underset{\sim}{\ddot{u}} + \underset{\sim}{\dot{v}}^T [G] \underset{\sim}{\ddot{v}} \quad . \tag{3.30}$$

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

$$\dot{K} = -2\lambda K + \sum_{j=2}^{n} G_j \left\{ \sum_{i=1}^{j-1} ((\dot{u}_i \dot{v}_j - \dot{v}_i \dot{u}_j)(\frac{V_i}{\ell_i} - \frac{V_{i+1}}{\ell_{i+1}})) \right\} , \tag{3.31}$$

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.

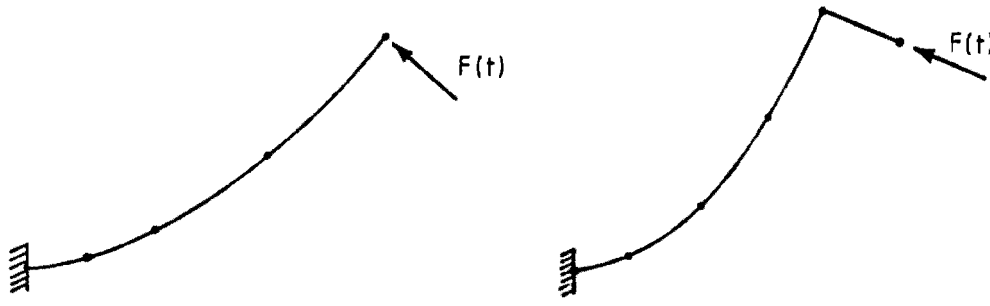
Let us now introduce the concept of pulse forces of the form

$$\underset{\sim}{F} = \underset{\sim}{F}(t) \quad , \tag{3.32}$$

where $\underset{\sim}{F}(t)$ 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

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.



(a) transversely (b) tangentially

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.

(i) Tangentially applied pulses:

$$P_x = \lambda[G]\ddot{\underset{\sim}{u}} - F(t)\cos\theta$$
$$P_y = \lambda[G]\ddot{\underset{\sim}{v}} - F(t)\sin\theta \qquad (3.33)$$

(ii) Transversely applied pulses:

$$P_x = \lambda[G]\underset{\sim}{\dot{u}} - F(t)\sin\theta$$
$$P_y = \lambda[G]\underset{\sim}{\dot{v}} + F(t)\cos\theta \qquad (3.34)$$

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 $\dot{\varepsilon}_0$, σ_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 matrixes $[m_x]$, $[m_y]$ and $[m_t]$ 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 λ 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 λ 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 λ to evaluate a new set of loads. Otherwise the program calls up subroutine BISEC in which a bisection algorithm is performed to obtain λ for the initial kinetic energy K_0 .

Once the above λ has been found, a check is done on whether momentum balance has been achieved by calculating ξ (equn (3.24)) ; if not, the velocities as calculated in BISEC are scaled by multiplying them with ξ 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 λ 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.

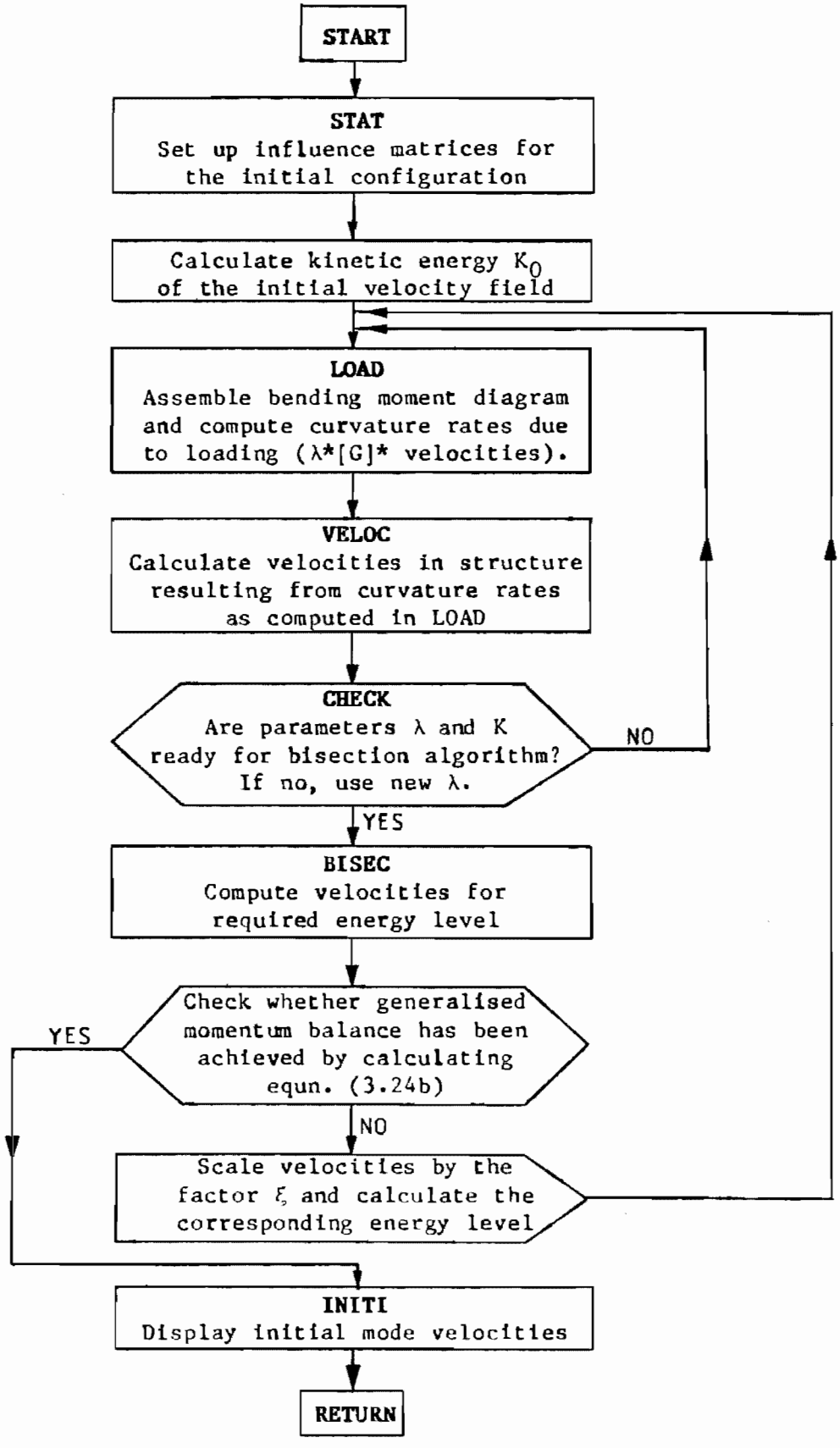
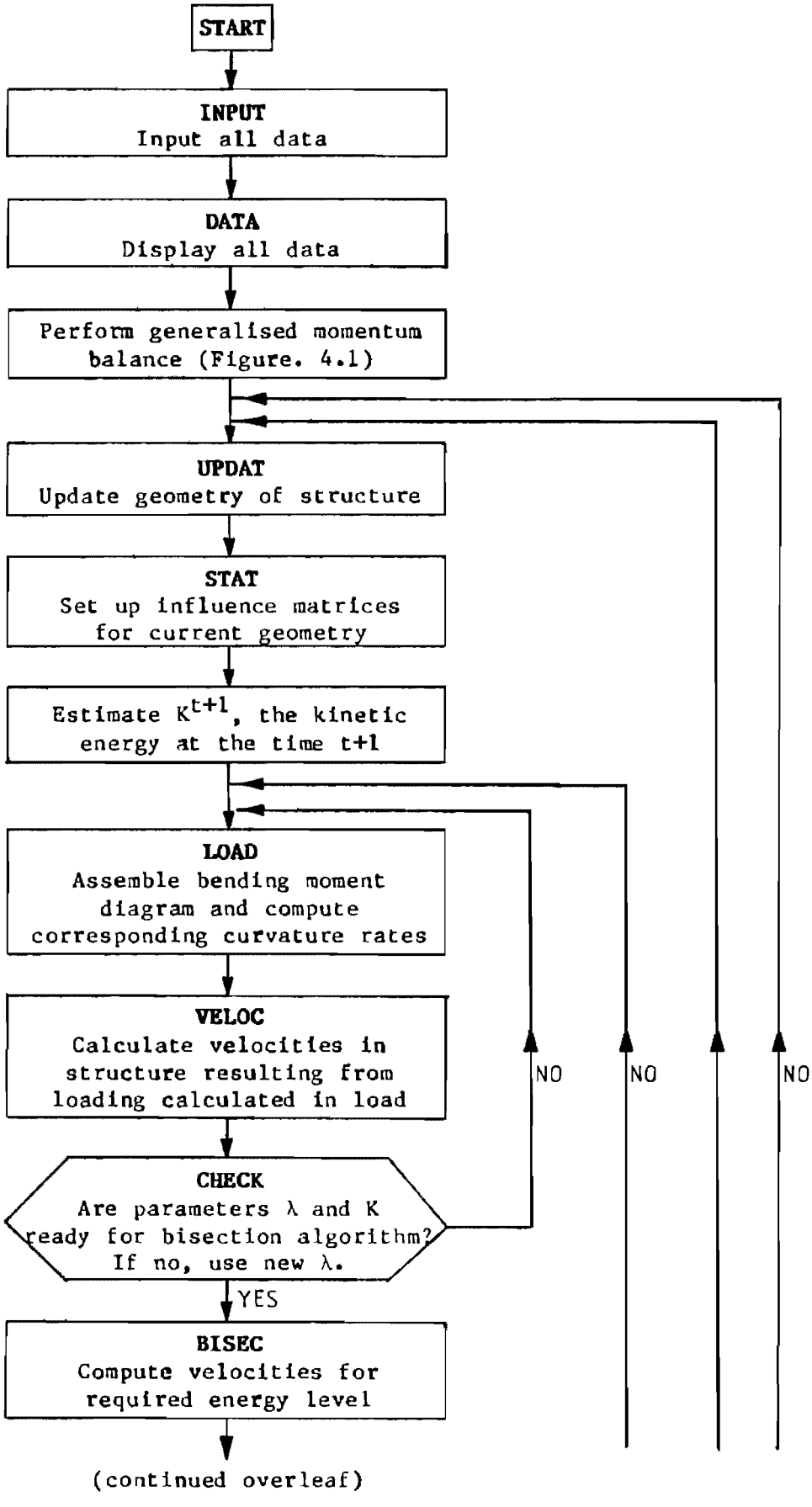


Figure 4.1 **Generalised Momentum Balance**


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 λ and K ready for bisection algorithm? If no, use new λ.
NO
NO
NO
YES
BISEC
Compute velocities for required energy level
(continued overleaf)

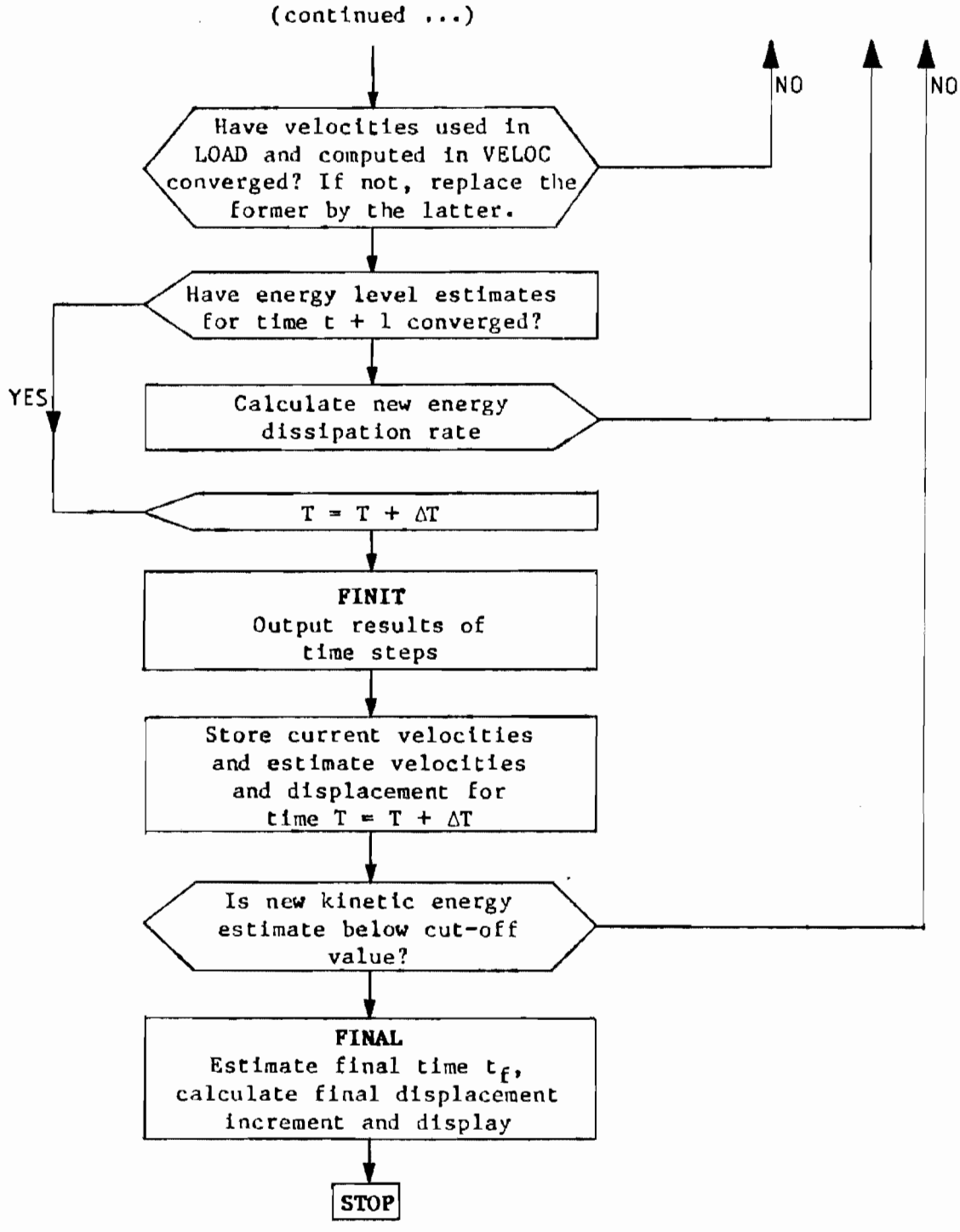


Figure 4.2 **Instantaneous Mode Algorithm**

CHAPTER 5

ILLUSTRATIVE EXAMPLES

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 algorithm 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 11.715m/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 0.054s, 50° and 0.239m respectively; including geometric effects, the calculations gave t_f as 0.054s, the tip rotation as 41,7° and a tip displacement of 0.247m. 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 0.047s, 37.1° and 0.194m , while for Fig 5.11, values of 0.063s, 65° and 0.314m 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 (equn (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

$$I\ddot{\theta} = - M_p \quad , \tag{5.1}$$

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

$$I = m\ell^2 \quad , \tag{5.2}$$

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

$$t_f = \frac{v_0\, m\ell}{M_p} \tag{5.3}$$

and

$$\theta(t_f) = \frac{v_0^2\, m}{2M_p} \quad . \tag{5.4}$$

(ii) L-shaped cantilever: I is given by

$$I = m\ell^2 + m(\ell^2 + h^2) = m(2\ell^2 + h^2) \tag{5.5}$$

and hence the final time and rotation can be derived:

$$t_f = \frac{v_0\, m(2\ell^2 + h^2)}{\ell\, M_p} \quad , \tag{5.6}$$

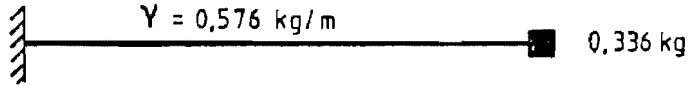
$$\theta\,(t_f) = \frac{v_0^2\, m(2\ell^2 + h^2)}{2\, M_p\, \ell^2} \quad , \tag{5.7}$$

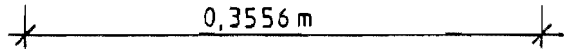
Substituting the relevant data from Figs 5.15(a) and (b) into equns (5.3), (5.4) and equns (5.6), (5.7) respectively, values are obtained which are given in Table 1 and compared to the analytical results: excellent agreement between the theoretical rigid-plastic and approximate analytical results (using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

	theory		analysis (n=100)	
	t_f	$\theta(t_f)$	t_f	$\theta(t_f)$
case (i)	0.01867s	1.706 rads	0.01868s	1.703 rads
case (ii)	0.0455s	2.022 rads	0.0457s	2.0455 rads

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.





section depth h = 4.5mm
section breadth b = 16.3mm
mass density $\rho = 7850 kg/m^3$
$\sigma_0 = 200MPa$, $\dot{\varepsilon}_0 = 40/s$, n = 5

Figure 5.1 Description of E4-Beam

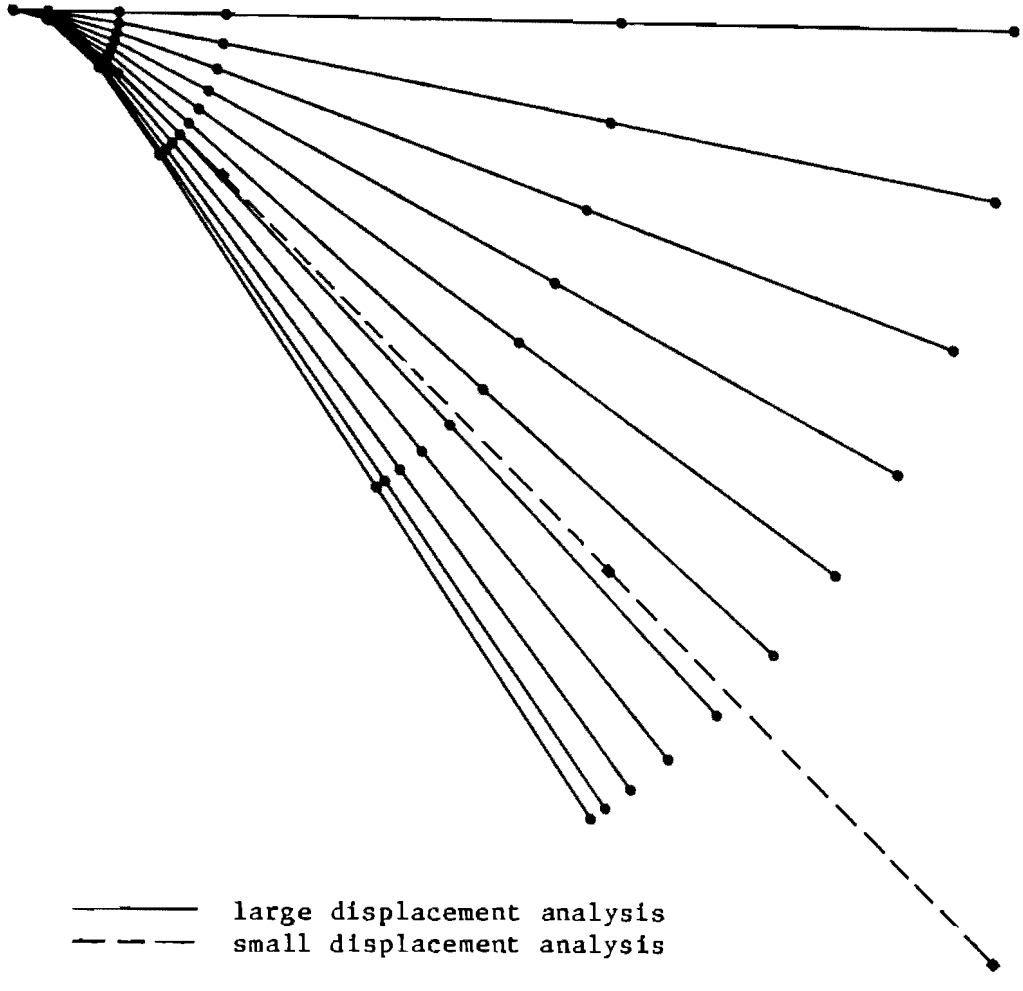


Figure 5.2 Displaced shape of E4-beam at successive time intervals for large displacement analysis and final displaced shape for small displacement analysis.

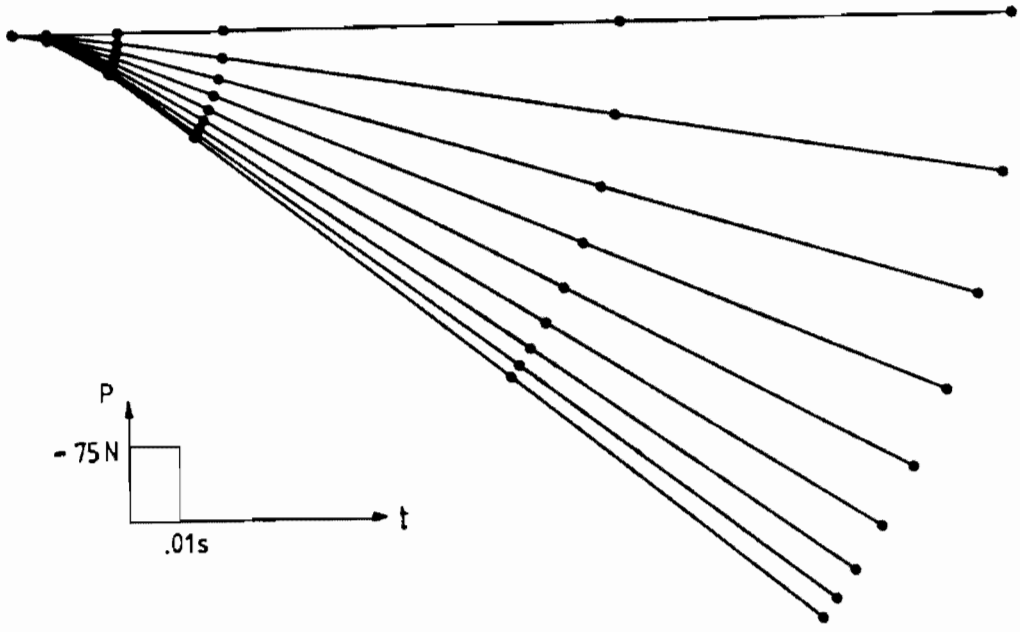


Figure 5.3 Displaced shape at successive time intervals of E4-beam subjected to a combination of an initial impulse and a negative follower pulse at tip.

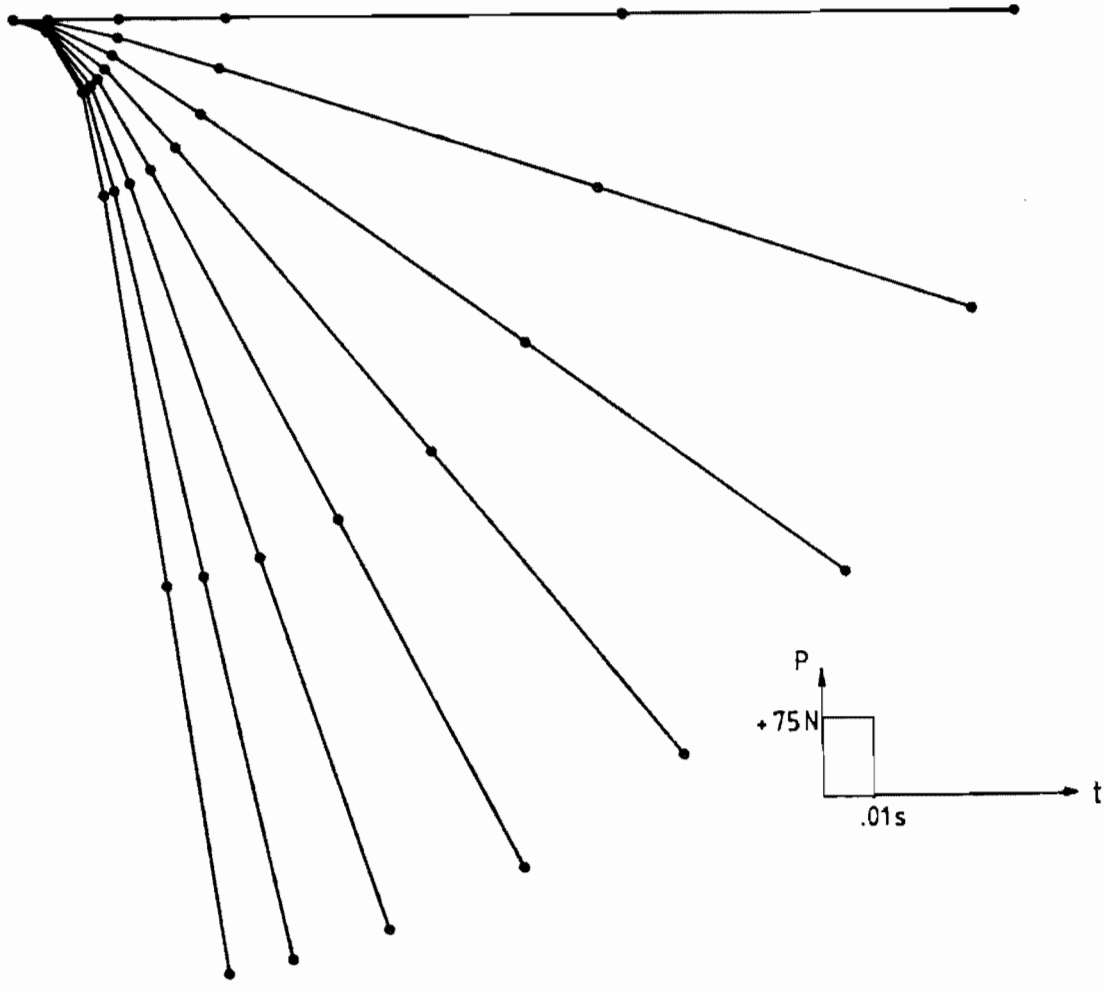


Figure 5.4 Displaced shape at successive time intervals of E4-beam subjected to a combination of an initial impulse and a positive follower pulse at tip.

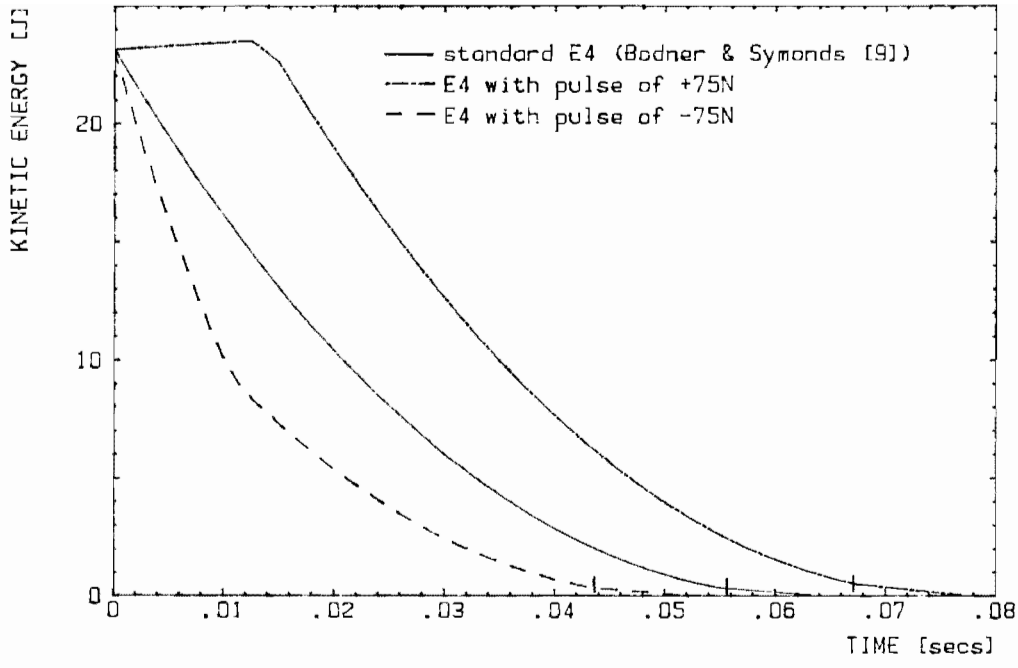


Figure 5.5 Plots of kinetic energy vs time of E4-beam examples

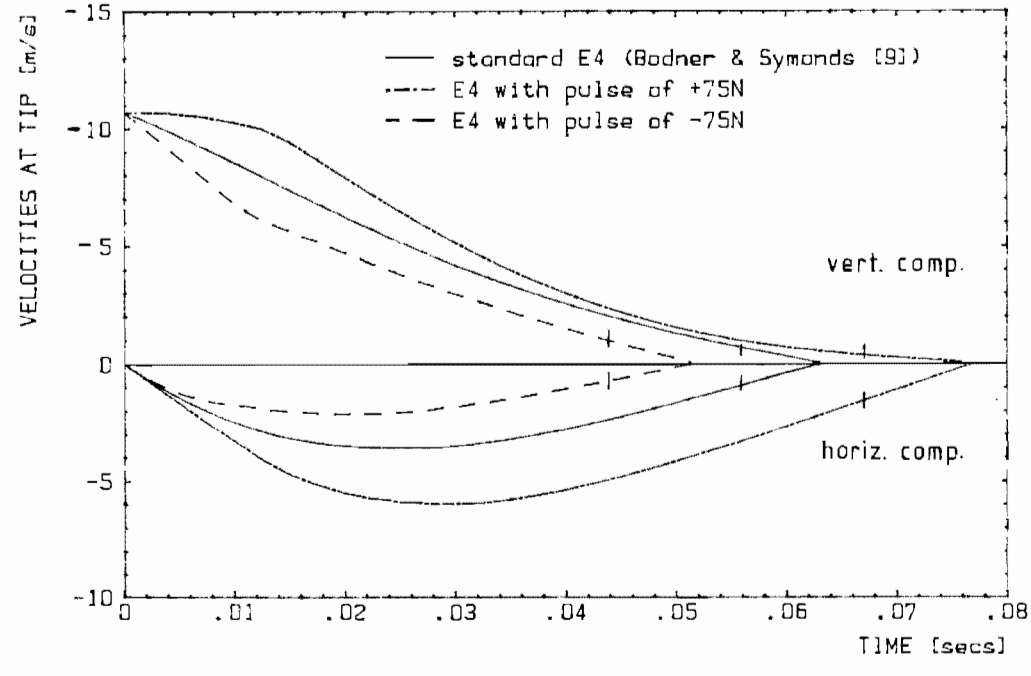


Figure 5.6 Plots of velocity components vs time of E4-beam examples

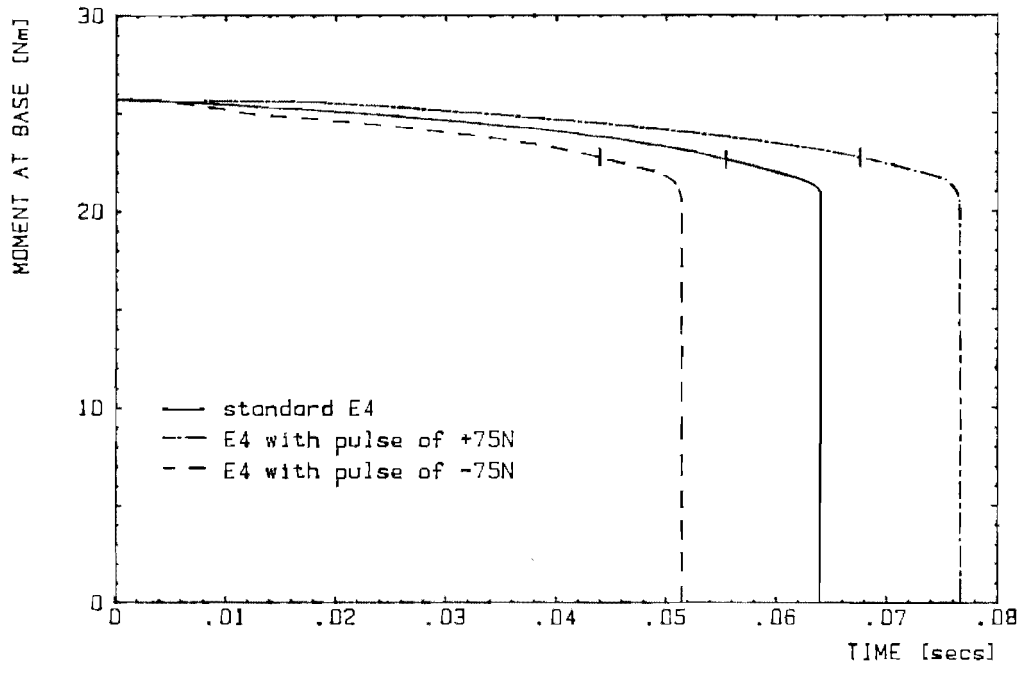


Figure 5.7 Plots of support bending moment vs time of E4-beam example

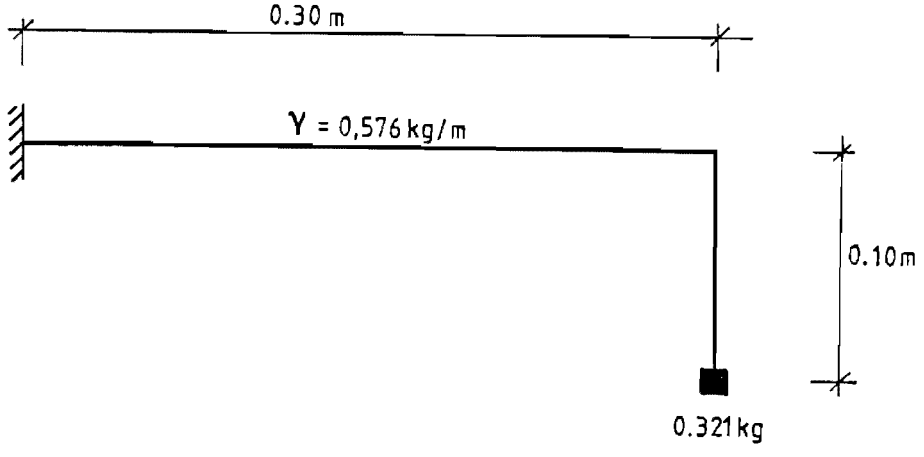


Figure 5.8 Description of L-shaped cantilever. The material properties are identical to those of the E4-beam.

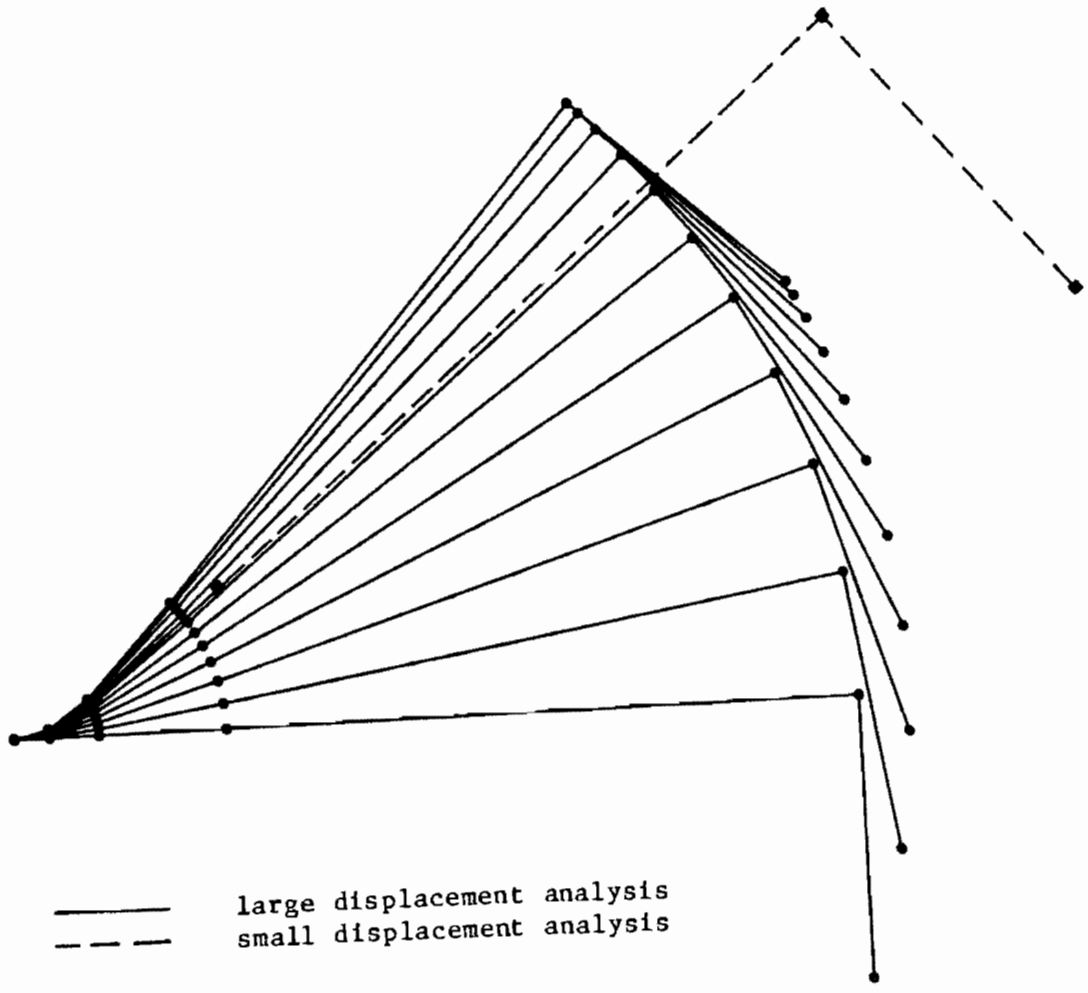


Figure 5.9 Displaced shape of L-shaped cantilever beam subjected to an initial impulse only.

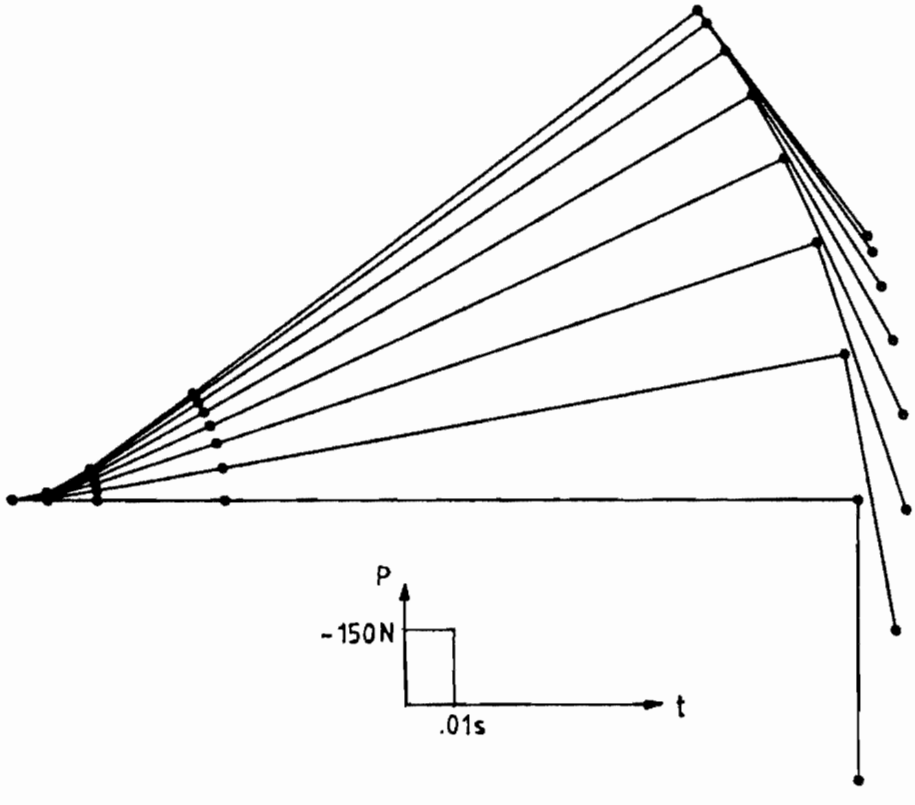


Figure 5.10 Displaced shape of cantilever subjected to a combination of initial impulse and a negative follower pulse at tip.

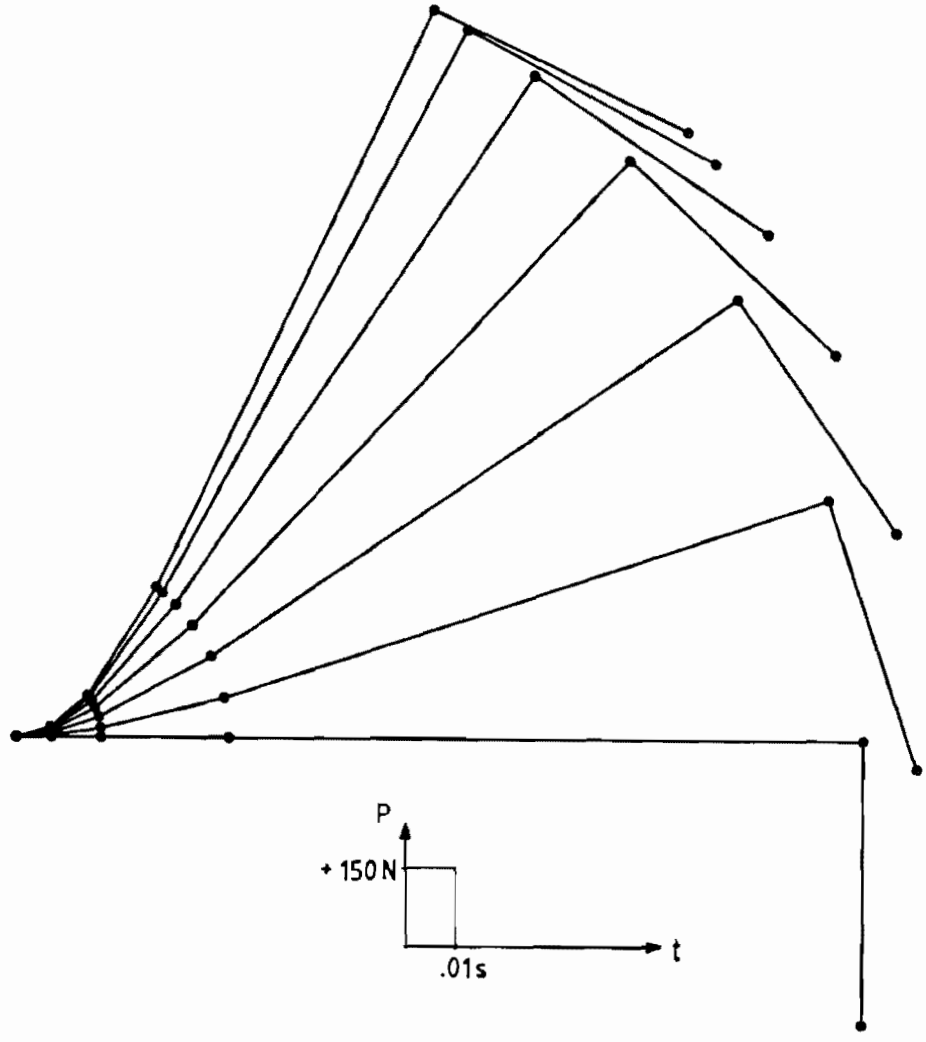


Figure 5.11 Displaced shape of cantilever subjected to a combination of initial impulse and a positive follower pulse at tip.

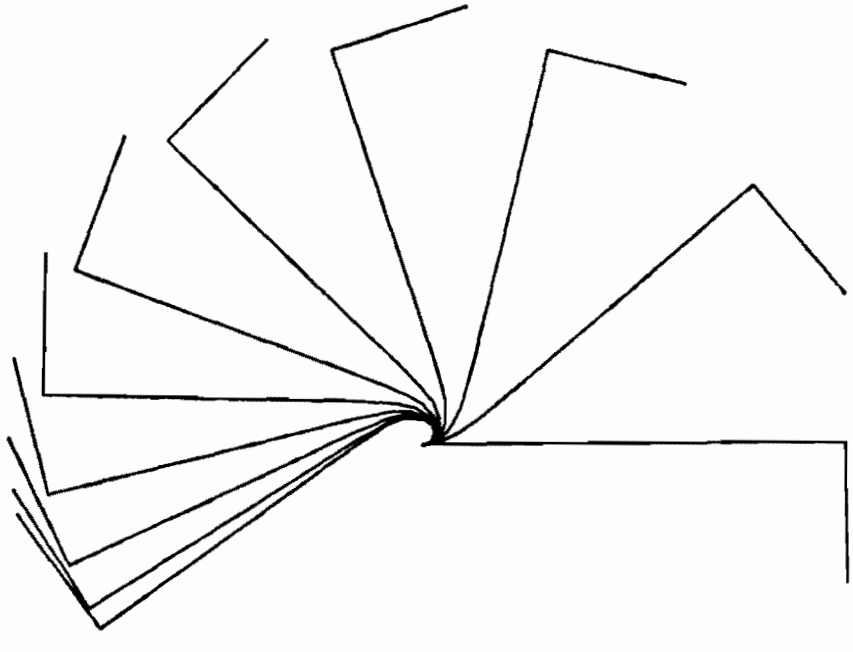


Figure 5.12 Cantilever subjected to an initial impulse and a positive follower pulse applied at the tip.

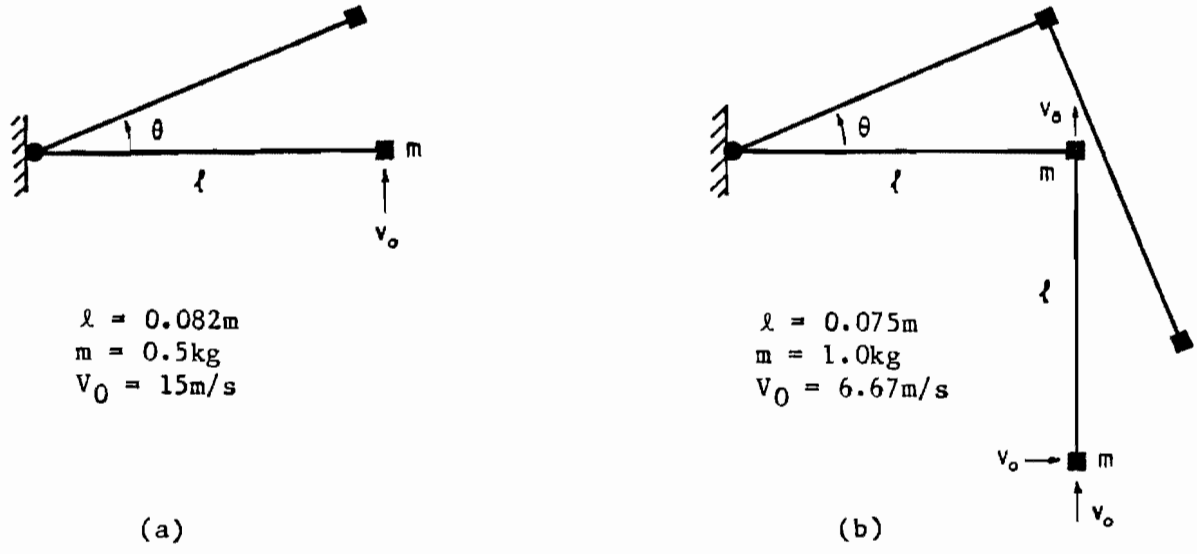


Figure 5.13 Description of two rigid-plastic cantilever beam examples with material constants identical to the E4-beam.

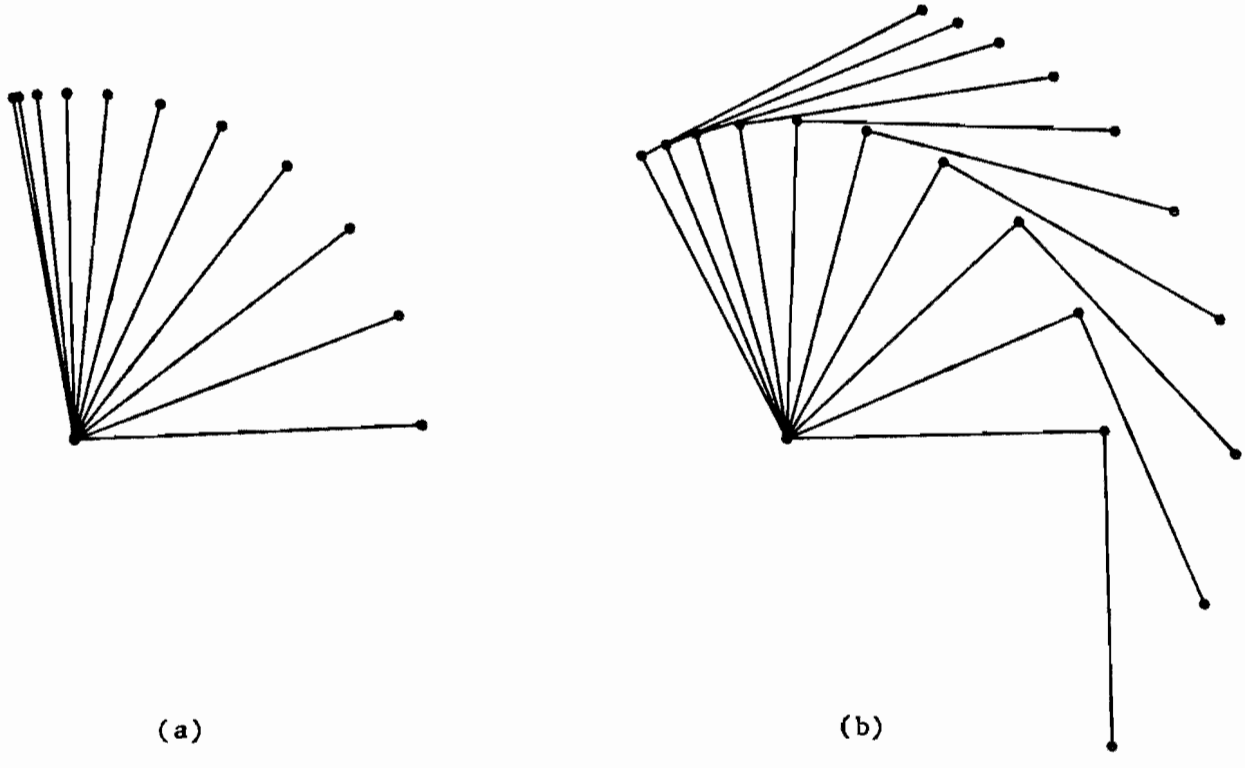


Figure 5.14 Displaced shapes at successive time intervals of the rigid-plastic cantilever beam example.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

16. P.D. Griffin and J.B. Martin, "Geometrically Nonlinear Mode Approximations for Impulsively Loaded Homogeneous Beams and Frames", Int. J. Mech. Sci., 25, No 1, 15-26, 1983.

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

18. J.B. Martin, "The Determination of Mode Shapes for Dynamically Loaded Rigid-Plastic Structures", Meccanica, 16, 42-45, 1981.

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

20. P.D. Griffin, "The Analysis of Rigid-Viscoplastic Plane Structures Subjected to Large Impulsive Loading", Ph.D. thesis University of Cape Town, 1982.

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

HH	BB	YSTRS	EPSIO	EN

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;

ε_K = TOL * K^0

(typically TOL = 10^{-2})

TOL1 - tolerance for kinetic energy convergence during forward integration

$$\left| K_{i-1}^{t+1} - K_{i}^{t+1} \right| < TOL1 * K^{0}$$

(typically TOL1 = 10^{-6})

TOL2 - tolerance for velocity convergence

$$\varepsilon_v = TOL2 * \dot{u}_{max}^{0}$$

(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.

PLOAD

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 [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.

TIME FUNCTION

Then the following card is required.

IPTS							
T_1	F_1	T_2	F_2	·	·	T_{IPTS}	F_{IPTS}

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

T_i F_i - 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).

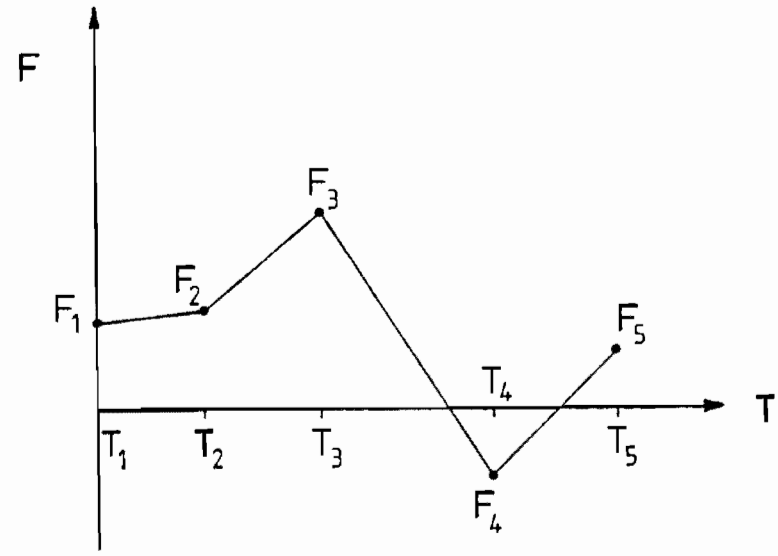


Figure A-1 A TYPICAL LOAD - TIME FUNCTION

Note that the pulse load at any instant is given by the load value PLOAD multiplied by the value F_i 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 [m/s] and [m] respectively. Node bending moments are given in units [Nm]. In addition, the magnitude of the current pulse is given in [N], together with the current value of λ and the kinetic energy (units [1/s] and [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

```
1 C     ************************************************************
2 C
3 C     DRIVER PROGRAM FOR SOLVING DYNAMICS OF RIGID-VISCOPLASTIC
4 C     CANTILEVERS
5 C
6 C     ************************************************************
7       IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
8
9       INCLUDE VISCO.PROCA
10       CALL INPUT
11       CALL DATA
12
13
14       DO 111 I=2,NN
15       VELO(I,1)=VEL(I,1)
16       VELO(I,2)=VEL(I,2)
17  111  CONTINUE
18
19
20 C     TIME LOOP  ........
21       DO 12 ITIME=1,200
22
23 C     KINETIC ENERGY LOOP  ........
24       DO 11 INERGY=1,444
25
26       CALL STAT
27
28       ITER=0
29
30       IF(IFLAG.EQ.1) GOTO 22
31
32 C     IMPLICIT FORWARD INTEGRATION OF KIN. ENERGY
33
34       ENER(1)=EOLD+0.5*DT*(DEOLD+DENER(1))
35 C     PRINT*,'NEW APPROX. ENER(1) =',ENER(1)
36 C     PRINT*,'  EOLD,DEOLD,DENER(1)',EOLD,DEOLD,DENER(1)
37
38
39 C     CHECK WHETHER KIN. EN. IS BELOW LIMIT
40       IF(ENER(1).LT.TOL.OR.ENER(1).LT.0.01) GOTO 800
41       KIN=DEOLD*DT
42       KIN=DABS(KIN)
43       IF(ENER(1).LT.KIN) GOTO 800
44
45
46       LOOP=0
47   22  CONTINUE
48
49
50 C     MODE VELOCITY LOOP  ..........
51
52       DO 1 IVELO=1,400
53       LOOP=LOOP+1
54
55       IF(LOOP.NE.1) THEN
56       DO 2 I=2,NN
57       VEL(I,1)=FXCOM(I,1)
58       VEL(I,2)=FXCOM(I,2)
59       AVEL(I)=AFX(I)
60   2   CONTINUE
61       END IF
62
```

```
C     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)
 25   CONTINUE
      ELSE
      GOTO 777
      END IF

999   CONTINUE

777   CALL BISEC

C     GENERALISED MOMENTUM BALANCE FOR INITIAL CONDITIONS
      IF(IFLAG.EQ.1) THEN
C     DO 220 I=2,NN
C     PRINT*,VEL(I,1),VEL(I,2),FXCOM(I,1),FXCOM(I,2)
C220  CONTINUE
      DO 221 J=1,2
      PSID=0.
      PSIN=0.
 221  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
 222  CONTINUE
      PSI=PSIN/PSID
C     PRINT*,'    PSI =',PSI
      DO 333 I=2,NN
      FXCOM(I,1)=PSI*FXCOM(I,1)
      FXCOM(I,2)=PSI*FXCOM(I,2)
      AFX(I)=DSQRT(FXCOM(I,1)**2+FXCOM(I,2)**2)
 333  CONTINUE
      ENER(1)=0.
      DO 444 I=2,NN
      ENER(1)=ENER(1)+0.5*GMASS(I,I)*(AFX(I))**2
 444  CONTINUE
C     PRINT*,'  ENER(1)  ENER(2)',ENER(1),ENER(2)
      CDIFF=ENER(1)-ENER(2)
      CDIFF=DABS(A)
C     PRINT*,'  CDIFF =',CDIFF
      IF(CDIFF.LT.TOL1) THEN
       GOTO 888
      ELSE
       GOTO 85
      END IF
      END IF
```

```
125
126 C      CHECK WHETHER VELOCITIES HAVE CONVERGED
127
128        IX=0
129        DO 555 I=2,NN
130        DO 555 J=1,2
131        DIFF=VEL(I,J)-FXCOM(I,J)
132        DIFF=DABS(DIFF)
133        IF(DIFF.GT.TOL2) IX=1
134 555    CONTINUE
135
136        IF(IX.EQ.0) THEN
137         GOTO 666
138        END IF
139
140    85  CONTINUE
141
142
143        DGAM=DABS(DGAM)
144
145
146   1    CONTINUE
147
148   888  CALL INITI
149
150   666  CONTINUE
151
152        DGAM=DABS(DGAM)
153
154        DO 3 I=2,NN
155        VEL(I,1)=FXCOM(I,1)
156        VEL(I,2)=FXCOM(I,2)
157    3   CONTINUE
158
159 C      CHECK WHETHER KINETIC ENERGY HAS CONVERGED
160
161        EE=E-ENER(1)
162        EE=DABS(EE)
163 C       PRINT*,' EE =',EE
164        E=ENER(1)
165        IF(EE.LT.TOL1) THEN
166         ITER=1
167         DEOLD=DENER(1)
168         GOTO 700
169        END IF
170
171 C      EVALUATE TIME RATE OF CHANGE OF ENERGY
172
173
174        DENER(1)=0.
175        DENER(1)=-2.*GAMM*ENER(2)
176 C      PRINT*,'  NORMAL DENER(1)',DENER(1)
177
178        IF(GEOM.EQ.'SMALL') GOTO 11
179
180 C      EVALUATE ADDITIONAL TERM FOR ENERGY RATE
181
182        DO 68 I=1,NN
183        CENT(I)=0.
184    68  CONTINUE
185
186        DO 69 I=3,NN
```

```
      DO 70 II=2,I-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)))
   70 CONTINUE
      CENT(I)=CENT(I)*GMASS(I,I)
   69 CONTINUE

      DO 72 I=2,NN
      DENER(1)=DENER(1)+CENT(I)
   72 CONTINUE

C     PRINT*,'  UPDATED DENER(1)',DENER(1)

   11 CONTINUE

  700 T = T + DT

      CALL FINIT

       DENER(1)=DEOLD
       EOLD=ENER(2)
       ICOUNT=0
C     NEW VELOCITY ESTIMATES
       DO 33 I=2,NN
       VEL(I,1)=FXCOM(I,1)-GAMM*FXCOM(I,1)*DT
       VEL(I,2)=FXCOM(I,2)-GAMM*FXCOM(I,2)*DT
   33  CONTINUE


      DO 55 I=2,NN
      VELO(I,1)=FXCOM(I,1)
      VELO(I,2)=FXCOM(I,2)
   55 CONTINUE

   12 CONTINUE

  800 CALL FINAL

      STOP
      END



      SUBROUTINE INPUT
C     ****************************************************
C
C     SUBROUTINE FOR DATA INPUT
C
C     ****************************************************
      IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
      INCLUDE VISCO.PROCA
C
  100 FORMAT( )
  101 FORMAT(A5)
  102 FORMAT(A72)
  103 FORMAT(A10)
  104 FORMAT(A4)

      READ(IREAD,102)TITLE


      DO 1 IJKL=1,8
```

```
249
250         READ(IREAD,104) DUMMY
251
252         IF(DUMMY.EQ.'CONT') THEN
253          READ(IREAD,100) NE,NN
254          READ(IREAD,101) GEOM
255          GOTO 1
256         END IF
257
258         IF(DUMMY.EQ.'SECT') THEN
259          READ(IREAD,100) HH,BB,YSTRS,EPSIO,EN
260          GOTO 1
261         END IF
262
263         IF(DUMMY.EQ.'SOLU') THEN
264          READ(IREAD,100) IFREQ,TZERO,TS,TOL,TOL1,TOL2,TOL3
265          GOTO 1
266         END IF
267
268         IF(DUMMY.EQ.'NODA') THEN
269          READ(IREAD,100)(J,COORDX(I),COORDY(I),I=1,NN)
270          GOTO 1
271         END IF
272
273         IF(DUMMY.EQ.'LUMP') THEN
274          READ(IREAD,100)(J,GMASS(I,I),I=1,NN)
275          GOTO 1
276         END IF
277
278         IF(DUMMY.EQ.'INIT') THEN
279          READ(IREAD,100)(J,VEL(I,1),VEL(I,2),I=1,NN)
280          GOTO 1
281         END IF
282
283         IF(DUMMY.EQ.'PULS') THEN
284          READ(IREAD,103) PULFOR
285          IF(PULFOR.EQ.'NONE') THEN
286           DO 10 I=1,NN
287           PULS(I)=0.
288    10     CONTINUE
289          GOTO 14
290          ELSE
291           READ(IREAD,100) P(NN,1)
292          END IF
293          GOTO 1
294         END IF
295
296         IF(DUMMY.EQ.'TIME') THEN
297          READ(IREAD,100) IPTS
298          DO 12 J=1,IPTS
299          READ(IREAD,100) I,FACT(1,I),FACF(1,I)
300    12    CONTINUE
301          FACT(1,IPTS+1)=50.
302          FACF(1,IPTS+1)= 0.
303          GOTO 1
304         END IF
305
306         IF(DUMMY.EQ.'END ') THEN
307          GOTO 14
308         END IF
309
310     1   CONTINUE
```

```
311
312    14  CONTINUE
313  C     CALCULATE ALL CONSTANTS  .....
314
315        T=TZERO
316
317        DO 2 I=1,NN
318        XGEO(I)=COORDX(I)
319        YGEO(I)=COORDY(I)
320    2   CONTINUE
321
322        V0=0.
323        DO 8 I=1,NN
324        IF(DABS(VEL(I,1)).GT.V0) V0=DABS(VEL(I,1))
325        IF(DABS(VEL(I,2)).GT.V0) V0=DABS(VEL(I,2))
326    8   CONTINUE
327
328        PIE=3.14159265358979324
329
330        AA=HH*BB
331        RMO=AA*HH*YSTRS/4*1000000
332        RKO=2.*EPSIO/((((2.*EN)/(2.*EN+1))**EN)*HH)
333
334        DO 3 I=1,NN
335        AVEL(I)=DSQRT(VEL(I,1)**2+VEL(I,2)**2)
336        AVELO(I)=AVEL(I)
337    3   CONTINUE
338
339        ENER(1)=0.
340        DO 4 I=2,NN
341        ENER(1)=ENER(1)+0.5*GMASS(I,I)*(AVEL(I))**2
342    4   CONTINUE
343        ENER(3)=ENER(1)
344
345        GAMM=1.5
346        DGAM=1.
347
348        TOL=TOL*ENER(3)
349        TOL1=TOL1*ENER(3)
350        TOL2=TOL2*V0
351        TOL2=DABS(TOL2)
352        TOL3=TOL3*ENER(3)
353
354        IF(V0.LT.1.D-3) THEN
355        TOL =.200000000
356        TOL1=.000020000
357        TOL2=.030000000
358        TOL3=.000020000
359        END IF
360
361        IEXIT=0
362        ITP=0
363        IPLTS=-1
364        IFLAG=1
365        ITER=1
366
367        SIMP=DFLOAT(ISIMP)
368
369        RETURN
370        DEBUG SUBCHK
371        END
372
```

```
373
374
375       SUBROUTINE DATA
376 C     ******************************************************
377 C
378 C     SUBROUTINE FOR DATA DISPLAY
379 C
380 C     ******************************************************
381       IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
382       INCLUDE VISCO.PROCA
383
384       WRITE(IPRINT,123)
385  123  FORMAT(1H1,/////////
386      &,/,20X,'*********************************************************'
387      &,/,20X,'*                                                       *'
388      &,/,20X,'*  VV      V  II    SSSSS     CCCCC     OOOOO          *'
389      &,/,20X,'*  VV      V  II   SSS   S   CC    C   OO    O         *'
390      &,/,20X,'*   VV   V   II     SSS     CC        OO       O      *'
391      &,/,20X,'*    VVVV    II    S    SS   CC    C   OO    O         *'
392      &,/,20X,'*     VV     II    SSSSS     CCCCC     OOOOO          *'
393      &,/,20X,'*                                                       *'
394      &,/,20X,'*                                                       *'
395      &,/,20X,'*                                                       *'
396      &,/,20X,'*    A PROGRAM FOR SOLVING DYNAMICS OF RIGID-           *'
397      &,/,20X,'*     VISCOPLASTIC CANTILEVER BEAMS .........           *'
398      &,/,20X,'*                                                       *'
399      &,/,20X,'*         VERSION @ 29/8/84                             *'
400      &,/,20X,'*                                                       *'
401      &,/,20X,'*********************************************************',/)
402 C
403 C     DISPLAYS ALL INPUT DATA
404 C
405       WRITE(IPRINT,1)TITLE
406 1     FORMAT(1H1,5X,80('*'),/10X,A80,/,6X,80('*'),/)
407       WRITE(IPRINT,7) GEOM
408  7    FORMAT(1H ,//,10X,'RIGID-VISCOPLASTIC ANALYSIS WITH ',
409      &A5,' DISPLACEMENT ASSUMPTIONS .',/,10X,
410      &'THE INITIAL DISPLACEMENTS ARE ASSUMED TO BE ZERO .')
411       WRITE(IPRINT,13)EN,YSTRS,EPSIO,RMO,RKO,HH,BB,NE,NN
412 13    FORMAT(1H ,///,10X,'MATERIAL ASSUMPTIONS',/,10X,20('-'),//,5X,
413      &'RIGID-VISCOPLASTIC WITH POWER N =',D12.7,/,5X,
414      &'STATIC YIELD STRESS             =',D12.7,/,5X,
415      &'INITIAL STRAIN RATE             =',D12.7,/,5X,
416      &'YIELD MOMENT                    =',D12.7,/,5X,
417      &'CURVATURE RATE CONSTANT         =',D12.7,//,5X,
418      &'SECTION HEIGHT   =',D12.7,/,5X,
419      &'SECTION WIDTH    =',D12.7,//,5X,
420      &'NUMBER OF ELEMENTS  :',I5,/,5X,
421      &'NUMBER OF NODES     :',I5,/)
422       WRITE(IPRINT,3)
423 3     FORMAT(1H ,/,10X,'COORDINATES OF NODES',/,10X,20('-'),//,4X,
424      &'NODE',10X,'X',12X,'Y',/)
425       DO 110 I=1,NN
426       WRITE(IPRINT,4)I,COORDX(I),COORDY(I)
427 4     FORMAT(1H ,3X,I3,5X,2(2X,D11.5))
428 110   CONTINUE
429       WRITE(IPRINT,5)
430 5     FORMAT(1H ,//,10X,'LUMPED MASSES',/,10X,13('-'),//,4X,'NODE',/)
431       DO 6 I=2,NN
432       WRITE(IPRINT,15)I,GMASS(I,I)
433 6     CONTINUE
434       WRITE(IPRINT,14)
```

```
435 14    FORMAT(1H ,///,10X,'INITIAL VELOCITIES',/,10X,18('-'),//,4X,
436      &'NODE',11X,'X',18X,'Y',/)
437       DO 112 I=1,NN
438       WRITE(IPRINT,15)I,VEL(I,1),VEL(I,2)
439 112   CONTINUE
440 15    FORMAT(1H ,3X,I3,2(3X,D16.5))
441       IF(PULFOR.EQ.'NONE') THEN
442       WRITE(IPRINT,16)
443  16   FORMAT(1H ,10X,'NO PULSE APPLIED TO STRUCTURE',///)
444       ELSE
445       WRITE(IPRINT,17)
446  17   FORMAT(1H ,//10X,'TIME FUNCTION VALUES',/,10X,20('-'),/)
447       DO 115 I=1,IPTS
448       WRITE(IPRINT,4) I,FACT(1,I),FACF(1,I)
449 115   CONTINUE
450       WRITE(IPRINT,20) PULFOR,P(NN,1)
451  20   FORMAT(1H ,4X,A10,' PULSE AT TIP =',D11.5,/)
452       END IF
453       WRITE(IPRINT,18) ENER(1),T
454  18   FORMAT(1H ,///,5X,'INITIAL KINETIC ENERGY =',D16.6,/,5X,
455      &'STARTING TIME          =',D16.6,//)
456
457
458       IF(PULFOR.EQ.'NONE') IPTS=5
459
460       RETURN
461       DEBUG SUBCHK
462       END
463
464
465
466       SUBROUTINE ABVEL
467 C     ************************************************************
468 C
469 C     SUBROUTINE FOR CALCULATING VELOCITY COMPONENTS AND
470 C     ABSOLUTE VELOCITIES FROM LOCAL VELOCITIES   ......
471 C
472 C     ************************************************************
473       IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
474
475       INCLUDE VISCO.PROCA
476
477       DO 1 J=1,2
478       DO 1 I=1,NN
479       FXCOM(I,J)=0.
480   1   CONTINUE
481
482       DO 2 I=2,NN
483       FXCOM(I,1)=FXCOM(I-1,1)-FX(I)*DSIN(THETA(I-1))
484       GSTOR(2,I,1)=FXCOM(I,1)
485       FXCOM(I,2)=FXCOM(I-1,2)+FX(I)*DCOS(THETA(I-1))
486       GSTOR(2,I,2)=FXCOM(I,2)
487   2   CONTINUE
488
489       DO 3 I=1,NN
490       AFX(I)=0.
491   3   CONTINUE
492
493       DO 4 I=2,NN
494       AFX(I)=DSQRT(FXCOM(I,1)**2+FXCOM(I,2)**2)
495   4   CONTINUE
496
```

```
      RETURN
      DEBUG SUBCHK
      END



      SUBROUTINE UPDAT
C     ************************************************************
C
C     SUBROUTINE FOR UPDATING OF GEOMETRY  .........
C
C     ************************************************************
      IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
      INCLUDE VISCO.PROCA


      IF(ITER.EQ.0) THEN
       DO 6 I=2,NN
       XINC(I)=0.5*(VELO(I,1)+FXCOM(I,1))*DT
       YINC(I)=0.5*(VELO(I,2)+FXCOM(I,2))*DT
    6  CONTINUE

       DO 2 I=1,NN
       XGEO(I)=XCOORD(I)+XINC(I)
       YGEO(I)=YCOORD(I)+YINC(I)
    2  CONTINUE
      END IF

      IF(ITER.EQ.0.OR.IFLAG.EQ.1) THEN
       DO 3 I=1,NE
       IF((XGEO(I+1)-XGEO(I)).EQ.0.D0) THEN
        IF((YGEO(I+1)-YGEO(I)).GT.0.D0) THETA(I)=PIE/2.
        IF((YGEO(I+1)-YGEO(I)).LT.0.D0) THETA(I)=-PIE/2.
        GOTO 3
       END IF
       THETA(I)=DATAN((YGEO(I+1)-YGEO(I))/(XGEO(I+1)-XGEO(I)))
       IF(XGEO(I+1).LT.XGEO(I).AND.YGEO(I+1).GT.YGEO(I))
     & THETA(I)=THETA(I)+PIE
       IF(XGEO(I+1).LT.XGEO(I).AND.YGEO(I+1).LT.YGEO(I))
     & THETA(I)=THETA(I)+PIE
    3  CONTINUE
      END IF


      IF(ITER.EQ.1) THEN
       DO 4 I=1,NN
       XCOORD(I)=XGEO(I)
       YCOORD(I)=YGEO(I)
    4  CONTINUE
      END IF

   11 CONTINUE

      RETURN
      DEBUG SUBCHK
      END



      SUBROUTINE STAT
C     ************************************************************
C
```

```
C     SUBROUTINE FOR SETTING UP INFLUENCE MATRICES  ......
C
C     ************************************************************
      IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
      INCLUDE VISCO.PROCA

      IF(IFLAG.EQ.0.AND.GEOM.EQ.'SMALL') GOTO 50

      CALL UPDAT

C     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)
 10   CONTINUE
      END IF


C     ASSEMBLING ALL INFLUENCE MATRICES  .....

      IF(NE.EQ.1) GOTO 222

      DO 5 I=1,NN
      DO 5 J=1,NE
      VM(I,J)=0.
 5    CONTINUE

      DO 6 I=1,NE
      VM(I,I)=CL(I)
 6    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))
 7    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./SIMP)*(DFLOAT(I)-1.-
     &SIMP*(DFLOAT(K)-1.))*CL(K)*DCOS(THETA(J)-THETA(K))
 8    CONTINUE

      DO 3 I=1,NN
      DO 4 J=1,NE
      UNITMX(I,J)=0.
      UNITMY(I,J)=0.
4     CONTINUE
3     CONTINUE

      UNITMX(1,1)=CL(1)*DCOS(THETA(1))
      UNITMY(1,1)=CL(1)*DSIN(THETA(1))*(-1.)

      DO 1 J=2,NE
      UNITMX(1,J)=UNITMX(1,J-1)+CL(J)*DCOS(THETA(J))
      UNITMY(1,J)=UNITMY(1,J-1)-CL(J)*DSIN(THETA(J))
 1    CONTINUE
      DO 2 I=2,NE
      DO 2 J=I,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)
```

```
  2   CONTINUE


      DO 24 I=1,ISIMP*NE+1
      DO 25 J=1,NE
      XINFX(I,J)=0.
      XINFY(I,J)=0.
25    CONTINUE
24    CONTINUE

      DO 27 K=1,NE
      DO 27 I=(K-1)*ISIMP+1,ISIMP*K
      DO 27 J=K,NE
      XINFX(I,J)=UNITMX(K,J)-(1./SIMP)*(DFLOAT(I)-1.-
     &SIMP*(DFLOAT(K)-1.))*CL(K)*DCOS(THETA(K))
      XINFY(I,J)=UNITMY(K,J)+(1./SIMP)*(DFLOAT(I)-1.-
     &SIMP*(DFLOAT(K)-1.))*CL(K)*DSIN(THETA(K))
 27   CONTINUE

222   CONTINUE
      IF(NE.EQ.1) THEN
       XVM(1,1)=CL(1)
       XINFX(1,1)=CL(1)*DCOS(THETA(1))
       XINFY(1,1)=CL(1)*DSIN(THETA(1))*(-1.)
       DO 122 I=2,11
       XVM(I,1)=XVM(I-1,1)-(1./SIMP)*XVM(1,1)
       XINFX(I,1)=XINFX(I-1,1)-(1./SIMP)*XINFX(1,1)
       XINFY(I,1)=XINFY(I-1,1)-(1./SIMP)*XINFY(1,1)
 122   CONTINUE
      END IF


 50   CONTINUE

      RETURN
      DEBUG SUBCHK
      END



      SUBROUTINE PULSE
C     ****************************************************************
C
C     SUBROUTINE EVALUATING FOLLOWER PULSE MAGNITUDE FROM
C     PULSE-TIME HISTORY .........
C
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.D0) THEN
      PULS(I)=FACF(ITFUN,IT)*P(I,ITFUN)
      GOTO 5
      END IF

```

```
683         IF(TDEL.GT.0.D0) GOTO 4
684    3    CONTINUE
685    4    PULS(I)=(FACF(ITFUN,IT)-FACF(ITFUN,IT-1))
686         PULS(I)=PULS(I)*(T-FACT(ITFUN,IT-1))/(FACT(ITFUN,IT)-FACT(ITFUN,
687        &IT-1))
688         PULS(I)=PULS(I)+FACF(ITFUN,IT-1)
689
690         PULS(I)=PULS(I)*P(I,ITFUN)
691
692    2    CONTINUE
693    1    CONTINUE
694
695
696    5    CONTINUE
697
698         RETURN
699         DEBUG SUBCHK
700         END
701
702
703
704         SUBROUTINE LOAD
705  C      ************************************************************
706  C
707  C      SUBROUTINE EVALUATING FORCES DUE TO MASS * ACCELERATION
708  C      AND POSSIBLY AXIAL OR TRANSVERSE PULSE .......
709  C
710  C      ************************************************************
711         IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
712         INCLUDE VISCO.PROCA
713
714
715  C      FORCE=LAMBDA*MASS*VELOCITY ( + PULSE )
716
717
718         DO 20 I=2,NN
719         FORC(I,1)=GAMM*GMASS(I,I)*VEL(I,1)
720         FORC(I,2)=GAMM*GMASS(I,I)*VEL(I,2)
721         IF(PULFOR.EQ.'NONE') GOTO 20
722         CALL PULSE
723         IF(PULFOR.EQ.'TANGENTIAL') THEN
724          FORC(I,1)=FORC(I,1)-PULS(I)*DCOS(THETA(I-1))
725          FORC(I,2)=FORC(I,2)-PULS(I)*DSIN(THETA(I-1))
726         END IF
727         IF(PULFOR.EQ.'TRANSVERSE') THEN
728          FORC(I,1)=FORC(I,1)-PULS(I)*DSIN(THETA(I-1))
729          FORC(I,2)=FORC(I,2)+PULS(I)*DCOS(THETA(I-1))
730         END IF
731  20     CONTINUE
732         DO 33 J=1,NE
733         DO 32 I=1,ISIMP+1
734         BMOM(I,J)=0.
735         CURV(I,J)=0.
736  32     CONTINUE
737  33     CONTINUE
738
739  C      BENDING MOMENT = (INFLUENCE MATRIX) * (LOAD VECTOR)
740
741         DO 35 I=1,ISIMP+1
742         DO 36 K=1,NE
743         DO 37 J=1,NE
744         BMOM(I,K)=BMOM(I,K)+FORC(J+1,1)*XINFY(ISIMP*(K-1)+I,J)
```

```
      BMOM(I,K)=BMOM(I,K)+FORC(J+1,2)*XINFX(ISIMP*(K-1)+I,J)
37    CONTINUE
36    CONTINUE
35    CONTINUE

C     CALCULATE THE CURVATURE RATES ..........

      DO 39 J=1,NE
      DO 40 I=1,ISIMP+1
      JIK=1
      POP=DABS(BMOM(I,J))/RMO-1.DO
      IF(DABS(BMOM(I,J)).LE.RMO) POP=0.
      IF(BMOM(I,J).LE.-RMO) JIK=-1
      CURV(I,J)=DFLOAT(JIK)*RKO*POP**EN
40    CONTINUE
39    CONTINUE


      RETURN
      DEBUG SUBCHK
      END



      SUBROUTINE VELOC
C     ************************************************************
C
C     SUBROUTINE FOR EVALUATING VELOCITIES BY THE PRINCIPLE
C     OF VIRTUAL VELOCITIES  .........
C     (SIMPSON'S RULE USED FOR INTEGRATION )
C
C     ************************************************************
      IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
      INCLUDE VISCO.PROCA


      DO 42 I=1,NN
      FXE(I)=0.
      FX(I)=0.
42    CONTINUE


      DO 44 K=1,NE
      DO 45 I=1,ISIMP+1
      DO 46 J=1,NE
      IF(I.EQ.1) GOTO 47
      IF(I.EQ.ISIMP+1) GOTO 47
      RI=DFLOAT(I)/2.+0.2
      IR=IDINT(RI)
      RR=DFLOAT(IR)+0.2
      IF(RI.NE.RR) THEN
      FXE(J+1)=FXE(J+1)+2.*XVM(ISIMP*(K-1)+I,J)*CURV(I,K)
      ELSE
      FXE(J+1)=FXE(J+1)+4.*XVM(ISIMP*(K-1)+I,J)*CURV(I,K)
      END IF
      GOTO 46
47    FXE(J+1)=FXE(J+1)+1.*XVM(ISIMP*(K-1)+I,J)*CURV(I,K)
46    CONTINUE
45    CONTINUE
      DO 166 IJ=1,NE
      FXE(IJ+1)=FXE(IJ+1)*(1./SIMP)*CL(K)/3.
      FX(IJ+1)=FX(IJ+1)+FXE(IJ+1)
```

```
      FXE(IJ+1)=0.
166   CONTINUE
44    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))
 9    CONTINUE
      DO 10 I=2,NN
      FX(I)=FXE(I)
 10   CONTINUE

      CALL ABVEL

      RETURN
      DEBUG SUBCHK
      END



      SUBROUTINE CHECK
C     ************************************************************
C
C     SUBROUTINE TO CHECK WHETHER LAMDA MUST BE INCREMENTED
C     UP OR DOWN , OR WHETHER BISECTION CAN COMMENCE  ....
C
C     ************************************************************
      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
 60   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


      RETURN
      DEBUG SUBCHK
      END



      SUBROUTINE BISEC
C     ************************************************************
C
```

```
C     SUBROUTINE PERFORMING A BISECTION ALGORITHM ON THE
C     DESIRED KINETIC ENERGY LEVEL .........
C
C     ************************************************************
      IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)

      INCLUDE VISCO.PROCA


C     PRINT*,'F(1)=',F(1)
      A=F(1)-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)
      DO 8 I=2,NN
      FXCOM(I,1)=GSTOR(1,I,1)
      FXCOM(I,2)=GSTOR(1,I,2)
      AFX(I)=DSQRT(FXCOM(I,1)**2+FXCOM(I,2)**2)
 8    CONTINUE
      GOTO 790
      END IF

C     PRINT*,'F(2)=',F(2)
      B=F(2)-ENER(1)
      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

      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*,I,VEL(I,1),VEL(I,2),FXCOM(I,1),FXCOM(I,2)
 10   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

       GAM(3)=0.5*(GAM(1)+GAM(2))
```

```
      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     ************************************************************
C
C      SUBROUTINE DISPLAYING INITIAL MODE VELOCITIES   .....
C
C     ************************************************************
      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',/)

      DO 16 I=2,NN
      IF(ENER(3).LT.1.D-3) THEN
      WRITE(IPRINT,15) I,VEL(I,1),VEL(I,2)
      ELSE
      WRITE(IPRINT,15)I,FXCOM(I,1),FXCOM(I,2)
15    FORMAT(1H ,4X,I3,5X,D11.5,4X,D11.5,/)
      END IF
16    CONTINUE
      PRINT*,'            LAMDA =',GAMM
      PRINT*,'   KINETIC ENERGY =',ENER(2),ENER(1)

      DENER(2)=-2.*GAMM*ENER(2)
      DEOLD=DENER(2)
      DENER(1)=DEOLD
      EOLD=ENER(1)
      E=ENER(3)
      IF(NE.EQ.1) E=0.
      FTIME=ENER(2)/(-1.*DENER(2))
      IF(FTIME.LE.0.OR.FTIME.GT.0.05) THEN
      FTIME=0.05
      PRINT*,'WARNING: FTIME ESTIMATE SET TO 0.05 '
      END IF
      PRINT*,'    '
      PRINT*,'    FTIME =',FTIME
      PRINT*,'  PULS(NN) =',PULS(NN)
      DT=FTIME/TS

      CALL LOAD

      DO 18 I=1,NE
22    FORMAT(1H ,4X,I3,5X,D11.5,/)
      WRITE(IPRINT,22)I,BMOM(1,I)
18    CONTINUE


      DO 17 I=2,NN
      VELO(I,1)=FXCOM(I,1)
      VELO(I,2)=FXCOM(I,2)
17    CONTINUE

      IFLAG=0
```

```
1055        ITER=1
1056
1057        RETURN
1058        DEBUG SUBCHK
1059        END
1060
1061
1062
1063        SUBROUTINE FINIT
1064  C     ************************************************************
1065  C
1066  C     SUBROUTINE DISPLAYING MODE VELOCITIES AND
1067  C     DISPLACEMENTS FOR TIME STEP T     .......
1068  C
1069  C     ************************************************************
1070        IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
1071        INCLUDE VISCO.PROCA
1072
1073        CALL PDATA
1074        DO 10 I=2,NN
1075        DISP(I,1)=DISP(I,1)+(VELO(I,1)+FXCOM(I,1))*DT/2.
1076        DISP(I,2)=DISP(I,2)+(VELO(I,2)+FXCOM(I,2))*DT/2.
1077   10   CONTINUE
1078
1079        IF(ENER(3).LT.1.D-3.AND.T.EQ.DT) THEN
1080        DO 20 J=1,2
1081        DO 20 I=2,NN
1082        DISP(I,J)=0.
1083    20  CONTINUE
1084        END IF
1085
1086        IF(IR.EQ.IPLTS) THEN
1087
1088        WRITE(IPRINT,14) T
1089  14    FORMAT(1H1,2X,' RESULTS FOR TIME T =',D16.6,/////,20X,
1090       &'VELOCITIES',24X,'DISPLACEMENTS',//,5X,'NODE',9X,'X',
1091       &14X,'Y',17X,'X',14X,'Y',/)
1092
1093        DO 16 I=2,NN
1094        WRITE(IPRINT,15)I,FXCOM(I,1),FXCOM(I,2),DISP(I,1),DISP(I,2)
1095  15    FORMAT(1H ,4X,I3,1X,2(4X,D11.5),3X,2(4X,D11.5),/)
1096  16    CONTINUE
1097        PRINT*,'          LAMDA =',GAMM
1098        PRINT*,'KINETIC ENERGY =',ENER(2)
1099        PRINT*,'  PULS(NN) =',PULS(NN)
1100
1101        T=T-DT
1102        CALL LOAD
1103        T=T+DT
1104
1105        DO 18 I=1,NE
1106   22   FORMAT(1H ,4X,I3,5X,D11.5,/)
1107        WRITE(IPRINT,22) I,BMOM(1,I)
1108   18   CONTINUE
1109
1110        END IF
1111
1112        RETURN
1113        DEBUG SUBCHK
1114        END
1115
1116
```

```
1117
1118        SUBROUTINE PDATA
1119 C      ************************************************************
1120 C
1121 C      DATA OF SELECTED TIME STEPS STORED FOR PLOTTING
1122 C      PURPOSES  .........
1123 C
1124 C      ************************************************************
1125        IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
1126
1127        INCLUDE VISCO.PROCA
1128
1129        IPLTS=IPLTS+1
1130
1131        IF(IPLTS.EQ.0) THEN
1132        DO 1 I=1,NN
1133        X(I,1)=COORDX(I)
1134        Y(I,1)=COORDY(I)
1135    1   CONTINUE
1136 C      CALL DPRINT
1137        ITP=1
1138        GOTO 111
1139        END IF
1140
1141        IF(IEXIT.EQ.1) THEN
1142        DO 2 I=1,NN
1143        X(I,ITP)=XGEO(I)
1144        Y(I,ITP)=YGEO(I)
1145    2   CONTINUE
1146        GOTO 222
1147        END IF
1148
1149        RI=DFLOAT(IPLTS)/DFLOAT(IFREQ)+0.001
1150        IR=IDINT(RI)
1151        IR=IR*IFREQ
1152        IF(IR.EQ.IPLTS) THEN
1153        DO 3 I=1,NN
1154        X(I,ITP)=XGEO(I)
1155        Y(I,ITP)=YGEO(I)
1156    3   CONTINUE
1157        GOTO 111
1158        ELSE
1159        GOTO 222
1160        END IF
1161
1162  111   ITP=ITP+1
1163
1164  222   CONTINUE
1165
1166
1167        RETURN
1168        DEBUG SUBCHK
1169        END
1170
1171
1172
1173        SUBROUTINE FINAL
1174 C      ************************************************************
1175 C
1176 C      SUBROUTINE DISPLAYING ESTIMATED FINAL TIME AND
1177 C      FINAL DISPLACEMENTS   ........
1178 C
```

```
1179 C     ************************************************************
1180       IMPLICIT DOUBLE PRECISION*8(A-H,O-Z)
1181       INCLUDE VISCO.PROCA
1182
1183       IEXIT=1
1184
1185 C     EXT=1./GAMM
1186       EXT=2*ENER(2)/(-DENER(1))
1187       FTIME=T+EXT
1188       DO 20 I=2,NN
1189       DISP(I,1)=DISP(I,1)+0.5*EXT*FXCOM(I,1)
1190       XGEO(I)=COORDX(I)+DISP(I,1)
1191       DISP(I,2)=DISP(I,2)+0.5*EXT*FXCOM(I,2)
1192       YGEO(I)=COORDY(I)+DISP(I,2)
1193  20   CONTINUE
1194       CALL PDATA
1195
1196       WRITE(IPRINT,14) FTIME
1197 14    FORMAT(1H1,2X,' ESTIMATED FINAL TIME =',D16.6,
1198      &/////,18X,'TOTAL DISPLACEMENTS',//,5X,'NODE',
1199      &16X,'X',18X,'Y',/)
1200
1201       DO 16 I=2,NN
1202       WRITE(IPRINT,15)I,DISP(I,1),DISP(I,2)
1203 15    FORMAT(1H ,3X,I3,5X,2(3X,F16.8),/)
1204 16    CONTINUE
1205       PRINT*,'  EXTENSION IN TIME :',EXT
1206       PRINT*,'                    '
1207       PRINT*,'  FINAL TIME      :',FTIME
1208
1209
1210 C     IF DATA IS TO BE WRITTEN TO A FILE
1211 C 100 FORMAT(1H ,2(5X,F10.5))
1212 C     PRINT*,'  X & Y COORDS FOR PLOTTING'
1213 C     DO 4 J=1,ITP
1214 C     DO 5 I=1,NN
1215 C     PRINT*,X(I,J),Y(I,J)
1216 C     WRITE(IDATA,100) X(I,J),Y(I,J)
1217 C 5   CONTINUE
1218 C 4   CONTINUE
1219 C     PRINT*,'  ITP =',ITP
1220
1221       RETURN
1222       DEBUG SUBCHK
1223       END
1224
1225
1226 C     ************************************************
1227 C
1228 C     COMMON BLOCKS   ......................
1229 C
1230 C     ************************************************
1231 PROCA PROC
1232       PARAMETER IREAD=10
1233       PARAMETER IPRINT=6
1234       PARAMETER IDATA=20
1235       PARAMETER ISIMP=10
1236       PARAMETER NEL=14
1237       PARAMETER NNO=NEL+1
1238       PARAMETER EPS1=1.E-15
1239       PARAMETER EPS2=0.000001
1240       PARAMETER NTFUN=1
```

```
1241        PARAMETER NPLOD=1
1242        PARAMETER NPTS =10
1243        COMMON /BLK1/ HH,BB,YSTRS,EPSIO,EN,AA,RMO,E,EE,NE,NN,INC,
1244       &RKO,POP,RI,IR,RR,IX,TITLE,IFLAG,ITER,DIFF,CDIFF,LOOP,A,B,C,
1245       &COORDX(NNO),COORDY(NNO),VEL(NNO,2),CENT(NNO),ILAMDA,
1246       &FORC(NNO,2),CL(NEL),UNITMX(NNO,NEL),UNITMY(NNO,NEL),
1247       &XINFX(ISIMP*NEL+1,NEL),XINFY(ISIMP*NEL+1,NEL),
1248       &BMOM(ISIMP+1,NEL),CURV(ISIMP+1,NEL),FX(NNO),FXE(NNO),
1249       &GAMM,GAM(3),DGAM,F(3),ENER(3),
1250       &FTIME,T,DT,EOLD,DEOLD,DENER(2),TZERO,VELO(NNO,2),
1251       &DISP(NNO,2),EXT,PSI,PSID,PSIN,TOL,TS,KIN,VO,
1252       &TOL1,TOL2,TOL3,GSTOR(3,NNO,2),GMASS(NNO,NNO),
1253       &VM(NNO,NEL),XVM(ISIMP*NEL+1,NEL),FXCOM(NNO,2),AFX(NNO),
1254       &AVEL(NNO),AVELO(NNO),SIMP,GEOM,DUMMY,XINC(NNO),
1255       &YINC(NNO),XGEO(NNO),YGEO(NNO),THETA(NEL),XCOORD(NNO),
1256       &YCOORD(NNO),X(NNO+2,99),Y(NNO+2,99),ITP,IEXIT,IPLTS,IFREQ,
1257       &PIE,TDEL,P(NNO,NTFUN),FACT(NTFUN,NPTS),FACF(NTFUN,NPTS),
1258       &PULS(NNO),PULFOR,IPTS
1259
1260        CHARACTER TITLE*72
1261        CHARACTER PULFOR*10
1262        CHARACTER DUMMY*4
1263        CHARACTER GEOM*5
1264
1265  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

An introduction to the elastic theory of plates and shells. Generalised stresses, generalised strains, elastic constitutive relations, coordinate systems. Analytical solutions of simple problems.

(iv) CE 522 (a) : **INTRODUCTION TO FINITE ELEMENT METHOD**

3 credits

Generalised displacement method of analysis for framed structures. Elastic energy theorems. Basic procedures of the finite element method illustrated for frame structures.

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

3 credits

Plane stress and plane strain elements, plate bending elements, shell elements, three-dimensional elements. Programming of the finite element method. Techniques for equation solving.

(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

The partial differential equations of mathematical physics and methods of solution. Separation of variables, integral transforms, Green's functions. Application in physics and engineering.