A PROCEDURE FOR PREDICTING THE STRESSES IN GAS TURBINE BLADES USING FINITE ELEMENT ANALYSIS

by

Christopher Geoffrey Burton

Thesis submitted through the Faculty of Engineering
to the University of London for the degree
of Doctor of Philosphy

Mechanical Engineering Department
Imperial College
London

April 1983

24 0442985 3

FOR
REFERENCE ONLY

SUMMARY

The objective of the work described in this thesis was to devise a technique to predict the stresses in modern cooled aircraft gas turbine blades, as the first part of a design life assessment procedure.

The finite element method was chosen as the analysis technique as it is capable of modelling the non-linear material behaviour which occurs in a blade. The code can model elastic, plastic and creep behaviour with temperature dependent material properties. Because of the geometric complexity of a modern blade, it has been found that typically 15000 degrees-of-freedom are required in an analysis. Thus a primary concern in the research has been the development of fast, reliable and accurate solution techniques.

Starting from the governing equations for temperature dependent elastic, plastic and creep behaviour an expression for the tangent constitutive matrix for the behaviour of the blade material has been derived. A suitable timestepping solution scheme has been devised. A novel method for controlling the size of the timesteps used in the analysis based on changes in the tangent constitutive matrix has been derived. A number of quasi- and modified Newton schemes have been incorporated into the code in order to solve the structural equilibrium equations. A sub-incrementation scheme is used to calculate the stresses and inelastic strains from the predicted displacement increments.

The predictions of the code have been validated by comparing them with a number of analytic solutions, and the optimum solution method has been identified. An analysis of a representative blade design has been undertaken, and the stresses after creep relaxation has occurred in the component have been predicted. A good correlation was obtained between the location where the largest creep strain accumulation was predicted and an observed in-service failure site, which increases confidence in the predictions of the code.

ACKNOWLEDGMENTS

The work described in this thesis was performed at the Royal Aircraft Establishment, Pyestock under the research in public research laboratories scheme. I wish to thank the Director, Royal Aircraft Establishment and the Head of Propulsion Department for permission and facilities to undertake this project. I am grateful for the advice and encouragement of my two Supervisors, Dr G.A. Webster at Imperial College and Dr G.F. Harrison at Pyestock. The help and advice of Dr T.J.W. Ward of Imperial College is also gratefully acknowledged. I am also grateful to my colleagues Mr W.C. Clifford for writing the front-width reduction program and Mr S.R. Cockrem for programming the alternative numerical solution for the thermal plate problem. I would also like to thank the members of the Library, Typing and Reprographic Services at the Royal Aircraft Establishment for their assistance.

NOMENCLATURE

$\underline{a}$	flow vector, *ie* derivative of yield criterion with respect to stress
a_i, b_i	parameters in Crisfield acceleration scheme
A	slope of uniaxial stress/plastic strain curve
[B]	displacement/strain matrix
C_1, C_2	dimensionless constants in elasto-plastic creep stress/strain relationship
C_3	dimensionless constant in elasto-creep stress/strain relationship
[D]	elastic constitutive (stress/strain) matrix
[Dep], [Depc]	elasto-plastic and elasto-plastic-creep constitutive matrices
E	Youngs modulus
$\underline{f}$	structural force vector
F	plastic yield criterion
[F]	flexibility matrix (inverse of [D])
$G(\eta_i)$	function to be minimised in line search procedure
[I]	identity matrix
[J]	Jacobian matrix
K_1	constant in elasto-plastic creep stress/strain relationship (units of stress)
[K]	structural stiffness matrix
[K*]	structural stiffness matrix used in modified Newton-Raphson iteration
$[K_i]$	structural stiffness matrix modified by BFGS update
$[K_T]$	tangent stiffness matrix
N_i	element shape function
Q	plastic potential
$\underline{R}$	vector of reactions
R_1, R_2	control values in Crisfield acceleration scheme
S_s, S_r	multiplying factors for Cormeau timestep limit in displacement iteration and strain to stress calculation

t	time
t^*	creep pseudo-time for strain hardening procedure
T	temperature
u_x, u_y, u_z	displacements in x,y,z directions
$\underline{u}$	structural displacement vector
$\Delta\underline{u}_i$	displacement increment on current iteration
$\Delta\underline{\tilde{u}}_i$	intermediate displacement increment calculated during BFGS procedure
$\Delta\underline{u}_i^*$	accelerated displacement increment
$\Delta\underline{u}_t$	total displacement increment over current load/time step
$\underline{v}$, $\underline{w}$	vectors in BFGS update procedure
x, y, z	global coordinates
α	mean coefficient of linear thermal expansion
$\underline{\delta}_i$	displacement increment after line search procedure
$\Delta\underline{\delta}_{el}$	virtual displacement in virtual work statement
$\underline{\varepsilon}$	total strain
$\underline{\varepsilon}_c$	creep strain
$\bar{\varepsilon}_c$	effective creep strain
$\bar{\varepsilon}_c^*$	creep strain hardening parameter
$\underline{\varepsilon}_e$	elastic strain
$\underline{\varepsilon}_p$	plastic strain
$\underline{\varepsilon}_T$	thermal strain
η_i	scalar multiplier for linesearch procedure
θ	explicit/implicit factor in single step θ timestepping method
θ_s, θ_r	explicit/implicit factors in displacement iteration and strain to stress calculation
θ_1, θ_2, θ_3, θ_4	constants in Swansea θ' creep constitutive equation
$\underline{\kappa}$	creep or plastic hardening parameters
$d\lambda$	plastic multiplier

ν	Poisson's ratio
ξ, η, ζ	element local coordinates
π	potential energy
$\underline{\sigma}$	stress
$\bar{\sigma}$	effective (Von Mises) stress
σ_y	yield stress
χ_{ec}, χ_{epc}	creep stiffness parameters for elasto-creep and elasto-plastic creep analysis
$\bar{\chi}_{ec}$, $\bar{\chi}_{epc}$	control values of creep stiffness parameters
$\underline{\Psi}$	residual force vector
$\underline{\tilde{\Psi}}$	modified residual force vector calculated during BFGS procedure

LIST OF CONTENTS

Page

SUMMARY

ACKNOWLEDGMENTS

NOMENCLATURE

1 INTRODUCTION 5

2 REVIEW OF PREVIOUS WORK 9

3 THE DATA GENERATION PROGRAMS 13

3.1 Introduction 13
3.2 Generation of the three-dimensional finite element mesh 13
3.3 Interpolation of temperature data onto the GANDALF mesh 16
3.4 Generation of nodal constraints 18
3.5 Suggestions for further work 19

4 THE DATA CHECKING PROGRAM GANDALF1N 20

5 FINITE ELEMENT STRESS ANALYSIS 23

5.1 Elastic analysis 23
5.2 Elasto-thermal analysis 30
5.3 Plastic analysis 32
5.4 Creep analysis 41

6 TURBINE BLADE ANALYSIS 50

6.1 Introduction 50
6.2 Derivation of the elasto-plastic thermal creep model 50
6.3 The subincrementation solution procedure 58

7 THE SOLUTION OF FINITE ELEMENT EQUATIONS 60

7.1 Linear equation systems 60
7.2 Nonlinear equation systems 63

7.2.1 The tangent stiffness method 63
7.2.2 Modified Newton-Raphson methods 65
7.2.3 Other nonlinear solution schemes 68

8 THE STRESS ANALYSIS PROGRAM GANDALF3P 76

8.1 The overall structure of the code 76
8.2 The calculation of the allowable timestep size 79
8.3 The calculation of the centrifugal load 84
8.4 The strain to stress calculation 85
8.5 The nonlinear subincrementation procedure 88
8.6 The restart capability 90

LIST OF CONTENTS (concluded)

		Page
9	APPLICATION OF GANDALF3P TO TEST PROBLEMS	92
9.1	Introduction to the test problems	92
9.2	Creep test cases	93
9.2.1	Introduction	93
9.2.2	The stress relaxation problem	94
9.2.3	The constant stress problem	98
9.2.4	Tests of the creep timestep control algorithm	100
9.2.5	Investigation of the optimum number of sub-increments for creep problems	104
9.2.6	Application of quasi-Newton methods to creep problems	107
9.3	Elasto-plastic test cases	111
9.3.1	Analysis of a hollow sphere under internal pressure loading	111
9.3.2	Analysis of a thermally loaded infinite plate	117
10	ANALYSIS OF THE RB211 "RACETRACK" BLADE	120
11	SUGGESTIONS FOR FURTHER WORK	126
12	CONCLUDING REMARKS	131
Appendix A	The data input files	139
Appendix B	The interface to the graphics package	144
Appendix C	Validation of the GANDALF creep model	149
Appendix D	Implementation of the frontal solution procedure on the VAX 11/780/FPS-164 attached processor computer system	155
References		165
Tables 1-20		
Illustrations		Figures 1-35

Chapter 1

INTRODUCTION

The development of the aircraft gas turbine over the last 25 years has been dominated by the need to obtain increased specific thrust and fuel economy. Considerable improvements can be obtained by increasing the maximum cycle temperature as shown in Fig 1. The efficiencies of the turbine stages can be increased by increasing the blade speed[1-5]. However, both increasing the peak cycle temperature and the blade speed increase the stresses within the engine, particularly in the hot section. This has led to a steady demand for materials of greater high temperature strength. Turbine blade alloys consist of a nickel matrix precipitation hardened with a γ' (Ni_3Al or Ni_3Ti) phase. The improved high temperature strength of more modern alloys has been gained at the expense of formability, and therefore forged blades have been replaced by cast blades of alloys such as Mar-M 002 and Mar-M 246. More recently directionally solidified and single crystal blades have been introduced. Despite these advances the pace of aero-engine development is still limited by the ability of metallurgists to devise materials of sufficient high temperature strength. However this limitation has to some extent been overcome by the introduction of turbine blade cooling. Air is bled from the high pressure end of the compressor and passed through passages within the blades, before exhausting onto the blade surface to provide a blanket of cool air. A useful summary of the advantage of blade cooling is given by Barnes[6], who states that improved blade materials have allowed turbine entry temperatures to increase by 10°C per year, whereas cooled blades have allowed an increase of 20°C per year to be obtained.

The use of cooling air however degrades the aerodynamic and thermal efficiency of the engine. It has been estimated[7] that each 1% of compressor mainstream flow used for cooling causes a 0.75% increase in specific fuel consumption. At 1984 fuel prices this translates into an additional fuel cost of about threequarters of a million dollars over the engine lifespan for a three engined civil airliner. It is thus essential to optimise the degree of cooling obtained from a given amount of cooling air. This requirement has led to increasingly complex cooling designs, so that modern blades now feature "multi-pass" systems, where the cooling air is passed up and down the blade several times, and arrays of pedestals, pin fins and other devices to promote cooling air turbulence[7,8].

Despite these improvements in cooling system technology, the cost of maintenance of the engine hot section remains high. Turbine blade life is limited by creep and thermal fatigue. It is estimated[9] that airline hot section maintenance costs of $400 million per year account for 60% of total engine maintenance costs. It is clear therefore that airline operating costs could be significantly reduced by increasing the life of engine hot section parts. It is not realistic to imagine that improved materials alone will overcome this problem. It is thus essential to make optimum use of available materials. The major barriers to extending blade life are a lack of understanding of the influence of cooling design on life, and perhaps as importantly, the fact that there can be significant differences between the duty cycle for which an engine is designed, and that which it actually experiences in service[9-13].

In order to overcome these deficiencies it is necessary to improve the methods by which turbine blade life is assessed at the design stage. The importance of achieving an acceptable life at the design stage should not be underestimated. Life problems are insidious and may not appear until there are hundreds of engines in the field, because even an inadequate blade life may correspond to several years of service usage[11,14]. Current designs are validated by the use of simple stress and temperature correlations, often based on past experience[10-12], and by extensive "cut and try" development engine testing[15,16]. Development engine testing is expensive and it is impractical to test a blade for the length of time it would last in service. In addition, it is not practicable to examine the effect of variations in service usage on blade life.

The objective of the work described in this Thesis was to develop an improved numerical method of predicting turbine blade stresses as the first stage of a design life assessment procedure. Given the cooling geometry of the blade and an engine duty cycle the prediction of turbine blade life can in principle be achieved as follows. Firstly, by predicting the internal and external heat transfer coefficients to the blade, the blade metal temperatures can be calculated for the various engine operating conditions. The metal temperatures are necessary both because of the important contribution of the thermal stresses to the total stress field, and also because of the dependence on temperature of the mechanical properties of blade alloys. From the predicted temperatures, and knowledge of the centrifugal and gas loads, the stresses and strains experienced by the blade during a flight can be determined. Using this information the damage caused to the blade can be quantified, and hence a life obtained as a number of flights. Such a scheme is suggested by Jenkins[13], who states

that "analysis of high pressure turbine blades ... pushes at the boundary of heat transfer, stress analysis and high temperature materials behaviour technology".

The work described in this Thesis has not addressed the problem of the prediction of blade metal temperatures. The uncertainties associated with this procedure have been analysed extensively by Stepka[17], who concluded that the major source of error is inaccuracies in the prediction of external heat transfer coefficients. A similar conclusion was reached by Haubert *et al*[18] and Hsia[19]. Current heat transfer research at NASA is reviewed by Rudy and Graham[20].

This Thesis describes the development of an analysis system to predict the stresses in gas turbine blades. The package is known as GANDALF (General Algorithm for Numerical Determination of Aerofoil Life and Failure). A nonlinear analysis is required, in order to model the creep and plasticity of the blade material. The finite element method has been chosen as the analysis technique as it is able to model the complex geometry of modern cooled turbine blades and can accommodate a wide range of material behaviours. Particular emphasis has been placed on the development of efficient techniques for solving the nonlinear equations arising from the analysis.

It was clear at an early stage that a fully three-dimensional analysis would be required, because the stresses due to thermal gradients act primarily in a chordwise direction, whilst those due to centrifugal loads act in a spanwise direction. An exact solution can be obtained for three-dimensional elastic stresses in a beam of any cross section with a temperature distribution which is linear along the length of the beam but arbitrary across the cross section[21]. However, this analysis is not appropriate as the maximum temperatures in turbine blades occur near mid-span. The use of a generalized plane strain analysis was not considered appropriate, as this would have been unable to allow for the significant twist in the aerofoils of modern blades of high aerodynamic loading. Lewis and Beckwith[22] noted significant differences between the results from three-dimensional and beam analyses. They concluded that "since a beam type analysis cannot accurately predict local response, the use of three-dimensional finite element techniques is felt to be required for an acceptable confidence level in turbine aerofoil stress predictions". Freche and Acurio[23] and Jenkins[13] both advocate three-dimensional analysis. The requirement for a three-dimensional analysis means that a large number of degrees of freedom are needed, and it has been found that some 15000 unknowns are required to analyse a typical turbine blade.

In addition to the development of the stress prediction code itself it has proved necessary to develop a suite of programs to generate and check the data required for the analysis. The code inevitably generates a considerable quantity of data and the use of graphical techniques to assess the results is essential. Thus the development of a link between the stress analysis and an interactive graphics package has been necessary.

The calculations were performed on a Digital Equipment Co. Vax 11/780 with a Floating Point Systems Inc. FPS-164 attached processor. The solution techniques employed have been optimised for this hardware configuration.

In addition to research into solution techniques, assessments of the accuracy with which the behaviour of blade materials under engine conditions is modelled in the code have been made using comparisons with experimental data.

Chapter 2

REVIEW OF PREVIOUS WORK

It is convenient to categorize previous work according to the kinematic assumptions made in the analysis. It is clear that, as would be expected with the increase in the power of computers, there has been a steady trend towards more complex analyses.

The simplest assumption is to regard the blade as a series of radial elements, each at a different temperature. The cross section of the blade is assumed to remain plane and therefore the total radial strain will vary linearly across it. Knowing the temperature at any point on the cross section and neglecting inplane stresses, the radial stresses can be immediately written in terms of the unknown radial strain. These stresses must be balanced by the centrifugal and gas bending loads on the blade in order for it to remain in equilibrium. If the loads are known therefore, it is possible to determine the stresses. If the creep properties of the blade material are available then the creep rate of each element can be determined, and using a suitable timestepping scheme the stress relaxation in the blade can be calculated. Chaboche[24], in an early paper extended the idea to consider a number of cross sections, and solved the coupled problem which arises when the restraining effect of moments due to the action of the centrifugal forces on the deflected blade are included in the analysis. Kaufman[25], Halls and Baker[14] and Policella and Culie[26] all analysed a single cross section.

A more realistic approach is to couple the radial element method to a plane strain analysis of the blade cross section, in order to determine the inplane stresses. This technique was used by Hall and Olsen[27].

A more advanced approach was adopted by Peschel and Schreieck[28] who evaluated the bending and twisting of a blade due to the thermal loading. This information enabled the radial strain distribution over the blade cross section to be calculated, and this was then used as input to a generalized plane strain finite element analysis. The radial and inplane stresses could therefore be determined.

Chaboche and Culie[29] corrected the results of a radial element analysis by using data from a three-dimensional elastic analysis. They considered that the hypothesis that plane sections remained plane gave unrealistically high constraints in the leading- and trailing-edge regions of the blade, thus resulting in an overestimation of the thermal stresses.

Kaufman[30] used the commercial finite element package NASTRAN[31] to analyse a test specimen which simulated a hollow aerofoil cross section. Two-dimensional plate elements were used. The model was unable to include the effects of temperature gradients through the aerofoil wall, which in the blade are the most significant. Only an elastic analysis was performed.

Wolf[32] used an inhouse finite element package to analyse a pair of nozzle guide vanes. Elastic-ideally plastic material behaviour was considered. Due to the constraint imposed at both ends of the vane aerofoil, the predominant strain in this configuration is in the radial direction. Thus little difference was observed between beam theory predictions and the results of the three-dimensional analysis. Compressive plastic flow occurred during the takeoff transient and the author considered that failure was due to low cycle fatigue caused by excessive strain excursions. However, it was not possible to quantify damage due to creep as it was not included in the analysis.

Three-dimensional elastic analyses of blades are reported by Shung and Xiao[33] and Maya, Katsumata and Itoh[34]. It is very doubtful whether the finite element mesh used by Shung and Xiao[33] was fine enough to model the stress distributions accurately. Maya, Katasumata and Itoh[34] used the NASTRAN package[31]. However, they neglected to include the effect of the variation of Young's modulus with temperature in the analysis. This approximation was found by Allen[35] to produce significant errors in both generalized plane strain and three-dimensional turbine blade analyses.

Kaufman[36,37] developed a simplified analysis procedure known as ANSYMP to calculate the stress-strain history at the critical location of a thermomechanically cycled structure. The fundamental assumption in the method was that inelastic behaviour was confined to a localized region. Thus the total strain history of the critical location could be determined from a three-dimensional elastic analysis. Simple one-dimensional models of creep and plasticity were then used to determine the stress-strain history of the critical location. Although this method is clearly far more economical than a full three-dimensional inelastic analysis, its validity when applied to the blade problem is very questionable. In general the critical location on a blade is not known *a priori*, inelastic effects may not be confined to a small region, and significant stress redistribution occurs.

Kaufman[38] compared the results of three-dimensional elastic and elastic-plastic analyses of a 'shell' blade using the commercial finite element package MARC[39]. Not surprisingly, good agreement was obtained between the elastic and elastic-plastic analyses at low strains. Agreement between the three-dimensional analyses and the predictions of a radial element analysis was poor.

Kaufman and Gaugler[40] analysed a 'shell' blade using the finite element package ELAS 55 [41,42]. Elastic, plastic and creep behaviour was considered. Blade lives were predicted for a number of different flight cycles using the strain range partitioning technique[43] and Robinson's creep life fraction rule[44]. In a thin-walled hollow aerofoil, as analysed in this study, the centre of the blade does not exert significant constraint on the leading and trailing edges. Thus as found in the analysis inelastic strain reversal is seldom induced even during severe thermal transients. Thus the damage due to inelastic strain hysteresis was insignificant compared to the creep damage. The same conclusion was reached from an anlysis of a similar blade by the same authors[45,46] using the MARC package.

McKnight, Laflen and Spamer[47], and McKnight *et al*[48] analysed the tip region of a blade with a history of tip cracking. The objective of the work was to determine whether a three-dimensional inelastic analysis could predict an observed inservice failure. A cyclic elasto-plastic creep analysis was performed using the commercial package ANSYS[49]. The maximum strain range was predicted in the region of the blade tip where the actual cracking occurred. The life predicted by strain range partitioning agreed with the observed life to within a factor of 2.

Manderscheid and Kaufman[50] analysed a fuel turbopump blade of the space shuttle main engine. The blades were manufactured from the nickel superalloy Mar-M 246, which was directionally solidified. A steady-state elastic-plastic analysis of the blade at the rated power level was performed using the MARC program. Although elastic anisotropy was included, the material was assumed to yield isotropically. It was found that the predicted steady-state stresses which were entirely in the elastic range were too low to cause a low cycle fatigue failure, thus underlining the importance of transient thermal stresses in determining blade life.

In addition to the turbine blade stress analyses described above, applications of finite element analysis to other structural problems in gas turbine engines have been reported.

Moreno[51] and Moreno *et al*[52] analysed a combustion chamber liner using the MARC program. Plasticity and creep were included in the analysis. The predicted mechanical strain and temperature cycle was applied to a test specimen. It was noted that the stress/strain cycles observed experimentally were not reproduced by the analysis. It was tentatively concluded that this might be due to the fact that the interaction between creep and plasticity seen in blade materials at high temperature was not included in the analysis.

Andress[53] and Sevcik and Stoner[54] used two-dimensional elastic analyses to examine the stresses in turbine blade Yttrium/Zirconium thermal barrier coatings. As these coatings are essentially ceramic materials, which will fail under low strain, the use of an elastic analysis is probably appropriate. However, it is doubtful whether the mechanical properties of the coating were known with sufficient accuracy for the analysis to be meaningful. The agreement between predicted and observed coating failure sites was poor. The use of the bulk ductility of the coating as a failure criterion, as in this study, is probably inappropriate as coating microcracking can occur without impairing its effectiveness.

In addition to failure caused by low cycle fatigue due to stresses at or near the yield point of the blade material, blade vibration can also cause failure by high cycle fatigue. Although the study of this type of failure does not form part of the work described in this Thesis, the use of finite element analysis to determine the resonant frequencies and mode shapes of turbine blades is well established (see, for example Peterson *et al*[55]).

The main conclusion which can be drawn from this review is that nearly all previous workers have used commercially available packages. Although commercial codes allow a wide range of analyses to be performed, the available solution techniques are often limited. For example the MARC code uses only the standard tangent stiffness method for plasticity behaviour, and an explicit timestepping scheme for creep. The development of a new code specifically optimised for the blade analysis problem does not appear to have been previously reported in the literature.

Chapter 3

THE DATA GENERATION PROGRAMS

3.1 Introduction

It has not been part of the work described in this Thesis to develop techniques for the prediction of blade metal temperatures. The decision was made therefore to obtain information from the Rolls-Royce plc TACITUS* suite. This suite of programs, which is described in Ref 8, enables a blade's aerodynamic profile and cooling system to be designed and the metal temperature distribution to be calculated. Data on the geometry of the blade, together with information on the blade metal temperatures, mainstream and coolant gas pressures, and rotational speed for various conditions around a flight cycle (which will be referred to as loadcases) were supplied from the TACITUS system on magnetic tape. It was thus necessary to develop a suite of programs to process the data into a form suitable for the stress analysis program. In essence the task involves the generation of a three-dimensional finite element mesh from the TACITUS geometry data, and then to interpolate onto this the temperature and pressure load data.

The structure of the suite of data generation programs is shown in Fig 2. As can be seen from Fig 2 it was decided that the various data file names should consist of five letters, followed by a three-figure number, *eg* TACITnnn.DAT. The five letters determine the type of data contained in the file, and the number nnn is chosen by the user to distinguish between data for different blades.

The first task is to read the TACITUS input tape onto the VAX 11/780 using the program FRODTAPE2. The data is output into a file TACITnnn.DAT. The program GTPCS splits the data in this file into four basic types. These are geometry, temperature, pressure and cycle data (such as rotational speed), which are output into files TGEOMnnn.DAT, TTEMPnnn.DAT, TPRESnnn.DAT and TCYCLnnn.DAT respectively. The four data sets are then used to generate the data required for the stress analysis.

3.2 Generation of the three-dimensional finite element mesh

In any finite element analysis the description of the mesh requires the largest proportion of the input data. A GANDALF turbine blade mesh consists of some 750 20-node brick elements, thus requiring some 30000 numbers to describe the nodal coordinates and mesh topology. In order to

---

* TACITUS Turbine Aerodynamics and Cooling Interactive Terminal Users Suite.

generate a mesh of this complexity the use of interactive graphics is essential. A number of commercial packages are available to perform this task, and it was decided that one of these FEMGEN[56], should be used.

The TACITUS data consists of the geometries of a number of cross sections up the blade. The program TGMDC converts this into a format suitable for FEMGEN. Essentially this involves describing the external and cooling geometry of the blade to FEMGEN as a set of points, lines and arcs. This information is written to file FGEOMnnn.DAT. A file PGEOMnnn.DAT containing the external geometry is also output for use by the temperature interpolation program.

The first task when running FEMGEN is to read the geometry from the file FGEOMnnn.DAT to create the FEMGEN database FGBASnnn.800. The mesh is then created interactively using FEMGEN commands and stored in the database. Although FEMGEN has considerable flexibility, experience has shown that the most efficient technique is as follows. Initially a grid of four-sided surfaces is created on each section, with each grid having an identical topology. When surfaces have been created on all of the sections they are joined up using a FEMGEN command to produce three-dimensional shapes known as bodies thus giving a three-dimensional geometric model of the blade. The automatic mesh generation routine in FEMGEN is then used to create 20-node "brick" elements on this geometry. A typical blade mesh is shown in Fig 3.

Once the mesh has been completed, it is then checked using the program MSHTB2. The program carries out a number of simple checks, such as ensuring that no nodes are coincident, and that no node appears twice in any element. The major task however is to ensure that the shapes of the elements are acceptable, and this test is described below.

It is the normal practice for a finite element to be defined in terms of a coordinate system (ξ,η,ζ) local to the element[57]. Thus the 20-node brick element used in GANDALF is a cube in local coordinates centred on the origin with node 1 situated at point (-1,-1,-1) and node 2 at (0,-1,-1), etc. To model the geometry of the structure being analysed the element is distorted from its original cubic shape by specifying nodal coordinates in global x,y,z coordinates. The global coordinates of any point within the element are obtained by interpolation from the global coordinates of the nodes using low order polynomial functions of the local coordinate system. It is necessary to ensure that each point within the finite element in local coordinates corresponds to a unique point in the global x,y,z space. It can be shown that a sufficient

condition to satisfy this requirement is that the determinant of the Jacobian matrix [J] where

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}$$

should have the same sign throughout the element[58]. For simple two-dimensional elements it is possible to establish rules of thumb which ensure that this criterion is not violated[59]. However, for more complex elements this approach is impracticable. The technique adopted has been to evaluate the determinant at the element nodes and the integration points used to calculate the element stiffness matrix. It is not known whether this is a mathematically sufficient condition to ensure that the determinant has the same sign throughout the element, and indeed this would be very difficult to determine. However, experience has shown that the method reliably detects misshapen elements.

During the calculation of the element stiffness matrix it is necessary to integrate over the volume of the element. The volume integral is transformed from the global x,y,z coordinates to the local coordinates to simplify the integration limits. It can be shown that the volume of the elementary parallelopiped whose sides have length $d\xi$, $d\eta$ and $d\zeta$ in the ξ,η,ζ directions respectively is given by

$$dvol = \det[J]d\xi d\eta d\zeta \quad .$$

As the element of volume must always be positive it is necessary to ensure that $\det[J]$ is always positive, and this can be achieved by numbering the element nodes in the correct order with respect to the global coordinate system. This is done automatically by the MSHTB program.

The GANDALF code uses the so called frontal solution technique[60,61] in reducing the structural stiffness matrix. This differs from the older band solution technique in that the entire structural stiffness matrix is never assembled. Instead, element stiffness matrices are assembled in the usual way, but only until it becomes possible to reduce one or more degrees of freedom. This occurs when no other elements, except those

which have already been assembled, contribute to the behaviour of a node. At this point the equations corresponding to the nodal degrees of freedom are reduced. The process continues in this way, intermixing element assembly and reduction of degrees of freedom until all the degrees of freedom have been reduced. This method is claimed to be up to four times faster than the band technique[61].

Irrespective of the solution technique employed, the time taken to reduce the structural stiffness matrix depends on some arbitrary numbering system. For the frontal solution the number of terms in each pivotal row is controlled by the element numbering. The distance between the first and last terms in the row (the so called frontwidth) determines the solution time. For a large three-dimensional analysis it is almost impossible to arrive at an efficient numbering of the elements whilst generating the mesh. A program to generate an efficient element numbering scheme automatically is thus essential, and in the GANDALF suite this function is performed by the program FWRED. The nodal renumbering scheme due to Gibbs, Poole and Stockmeyer[62] has been applied to the element renumbering problem. The algorithm due to Razzaque[63] is also used. A full description of the FWRED program is given in Ref 64. The renumbered mesh is output to a file GMESHnnn.DAT.

A graph of how the number of terms in each pivotal row will vary during the solution process for the original and revised numberings can be plotted using the program FWPLOT. An example is shown in Fig 4.

3.3 Interpolation of temperature data onto the GANDALF mesh

The GANDALF finite element analysis requires the temperature of each node of the mesh to be specified. Thus it is necessary to interpolate the TACITUS temperature information onto the GANDALF three-dimensional mesh. This is performed by the program TEMPINT. The temperatures are calculated in TACITUS using a series of two-dimensional finite element meshes of the blade cross sections. The 8-noded quadrilateral element is used. Spanwise conduction is ignored and thus by specifying the internal and external heat transfer coefficients, the temperature can be obtained using the standard finite element discretisation of Fourier's Law[57]. A nonlinear analysis is required as the thermal conductivity is weakly dependent on temperature.

The TACITUS data are contained in the file TTEMPnnn.DAT. These are the geometry and topology of the finite element meshes and the calculated temperatures. The file PGEOMnnn.DAT produced by the program TGMDC, which contains the geometry of the external profile of each cross section is

also required by the temperature interpolation program. The GANDALF mesh is contained in the file FMESHnnn.DAT.

The difficulty in carrying out the interpolation is that there is no fixed relationship between the GANDALF and TACITUS meshes. This is overcome by performing the interpolation in two stages. The first stage is to interpolate the temperatures from each TACITUS section onto a grid of squares which just overlaps the edges of the section. This grid will be referred to as a mask and there is one mask for each section. The second stage is to calculate the temperature of each GANDALF node from the temperature of the points on the masks.

The program proceeds as follows. The first task is to read in the coordinates of the nodes in the GANDALF mesh and sort them into groups according to the values of their radial coordinates, so that all the nodes in each group lie between an adjacent pair of TACITUS sections. Each pair of TACITUS sections and the nodes which lie between them are then considered in turn.

The next task is to establish the masks for each section. Each mask is a grid of squares which just overlaps the section. The coordinates of the blade profile are used to establish which intersections on the mask are inside the blade and which are outside. It is then necessary to calculate the temperature of all the intersections which lie within the blade. The temperature of any point within each TACITUS element will be a function of the nodal temperature of the element and the value of the element local coordinates ξ and η at the point of interest. It is thus necessary to calculate the local coordinates corresponding to the global coordinates of the intersection. As the interpolation function for the 8-noded quadrilateral contains quadratic terms in both ξ and η this is mathematically awkward. A compromise solution is adopted whereby the 8-noded quadrilateral element is split into four quarter elements, and a linear variation in properties is assumed in each. This means that the values of the local coordinates for the quarter elements can be simply obtained as the roots of two quadratic equations. It would be extremely inefficient to search all the quarter elements to locate the one which contained the intersection, so the quarter elements are sorted into groups as they are created according to the position of their centroids. The sorting and searching procedure has been adapted from that given by Griffiths[65] for a hidden surface algorithm.

Once the temperatures for all the intersections on each mask have been calculated the temperatures of the GANDALF nodes can be determined. The procedure for each node is as follows. The squares on each mask can be considered as the tops and bottoms of a set of cuboidal boxes which lie between the two sections. The box which contains each node is determined and the temperatures of the corners of the box are then used to calculate the nodal temperature, with each value weighted inversely according to the distance from the corner to the GANDALF node. Logic is provided to use an adjacent mask square automatically if the selected one contains no information.

The accuracy of this procedure is controlled by the fineness of the mask grid, as information can be lost if the grid is too coarse relative to the TACITUS mesh. Numerical experiments have shown that for typical TACITUS meshes acceptable results are produced for a mask grid of 125 squares in each direction.

3.4 Generation of nodal constraints

In a finite element analysis degrees of freedom are constrained to simulate earthing of the structure, planes of symmetry, etc. It is necessary to constrain sufficient of the degrees of freedom to prevent rigid body motion of the structure.

In the GANDALF suite the nodal constraints are input using the program CONSPROG. The program runs interactively and the user simply types in the list of nodes to be constrained together with the displacement of each degree of freedom.

Although this method is extremely simple there are two problems with it. Firstly the nodes in the model which are to be constrained have to be identified. It is essential to look at a graphical representation of the model, and for a large mesh identifying all the nodes can be quite difficult. Secondly with large meshes there could be a large number of nodes to be constrained, possibly around 300. These all have to be typed in by the user, thus increasing the possibility of errors.

It is suggested that in the future the generation of the nodal constraints is incorporated into the mesh generation package FEMGEN. This will enable the user to define areas of the mesh to be constrained such as the bottom surface of the blade with very few commands. FEMGEN will then automatically add the constraint data to the nodes as they are generated by the meshing routine.

3.5 Suggestions for further work

As can be seen by reference to Fig 2 some further work is needed to complete the entire GANDALF program suite. This is of a relatively minor nature but for completeness it is given here.

The largest item will be to write the pressure interpolation program PRESSINT. This program will take mainstream and coolant gas pressures and interpolate them onto the mesh to generate a set of nodal forces. The program would also add in the loads due to the change in momentum of the gas stream as it passes over the blade.

The stress analysis programs are able to analyse a series of conditions around a flight cycle. It will be necessary however to extend the data input programs to accommodate this feature. It is planned to develop a program TCYCDC to enable the user to specify what flight cycle is to be analysed. It is envisaged that a data file will be created containing information on various possible engine conditions. When carrying out an analysis the user will select conditions from this file and thus build up the required flight cycle for the stress analysis program to analyse.

At the present material property data are set up manually in a file in a format suitable for the analysis program. The format of this file (MATERnnn.DAT) is given in Appendix A. The analysis code assumes that Young's modulus and the coefficient of thermal expansion of the material vary quadratically with temperature and that Poisson's ratio is constant. Plastic properties are supplied to the code as sets of stress/plastic strain data points at the required temperatures. The code uses linear interpolation to obtain yield points at intermediate temperatures and plastic strains. Creep data is supplied through the GANDALF3P CRPLAW subroutine.

Although the present system is adequate it is possible that in the future, as a wider range of materials are analysed, the creation of a materials database might be advantageous. This would be managed by two programs. Program MATUPD will be used to update the database and the program MATEXT will extract the required material properties and produce a file MATERnnn.DAT for use by the analysis program.

Chapter 4

THE DATA CHECKING PROGRAM GANDALF1N

The program GANDALF1N is shown in Fig 5. The program is run interactively and its main functions are to sort the data produced by the data generation programs described in Chapter 3 into two output data files and, in response to user commands, to set the values of various control parameters to be used by the stress analysis. The input files to GANDALF1N are as follows:

(1) GMESHnnn.DAT Containing mesh topology and nodal coordinates.

(2) MATERnnn.DAT Containing information on the blade material.

(3) MCONSnnn.DAT Containing the constraints to be applied to the mesh.

(4) MTEMPnnn.DAT Containing the nodal temperatures for each point in the flight cycle.

(5) MPRESnnn.DAT Containing the nodal forces for each flight condition due for instance to pressure loading.

(6) GCYCLnnn.DAT Containing information on the rotational speed of the blade and the duration of each flight condition.

The data structures in these files are given in Appendix A. The first three files contain data which are unchanged throughout the analysis. The last three files however contain data which changes from flight condition to flight condition. Each flight condition will be referred to as a loadcase. A flight cycle is made up of a number of loadcases, which can conveniently be divided into two types, transient loadcases where the external conditions vary linearly with time, and steady state or cruise loadcases where the conditions remain constant (see Fig 6). Each loadcase is specified by its duration, whether or not it is a transient, and for transient loadcases the conditions at the end of the loadcase. The advantages of this scheme are that it is simple and flexible, and keeps the amount of data required to a minimum.

GANDALF1N produces two output data files. The first of these, GDATAnnn.DAT contains the structural topology and nodal coordinates. The second GLOADnnn.DAT contains all the data describing how the structure is loaded on each loadcase. The data structures in these files are given in Appendix A.

In addition to the two output data files, two further files are produced. The first file, GCONTnnn.DAT contains control information such as what types of nonlinearities are to be included in the analysis and the solution technique to be employed. The data structure in this file is given in Appendix A. The second file RUNnnn.COM contains the VAX/VMS operating system commands needed to run the analysis program GANDALF3P.

As is the normal procedure for nonlinear analysis GANDALF3P uses an incremental/iterative solution procedure, with a fraction of the loads on the structure being applied on each increment. A restart facility has been incorporated into the code. This enables an analysis to be restarted from any selected load increment using previously stored data. The incorporation of the restart capability has had a number of advantages. In order to investigate the performance of new solution algorithms or material models it is essential to carry out a number of analyses of the same structure. It is wasteful to obtain a complete solution if the model or algorithm only affects the later stages of the analysis. This can be avoided using the restart capability. In some cases it may be desirable to continue an analysis from a previous end condition, for example to investigate the effect of a further period of creep on the structure. This facility is also available using the restart capability. It is possible that for certain combinations of loading and solution procedure, convergence of the solution may not be obtained. Using the restart capability it would be possible to reanalyse the load increment using a different solution technique in order to obtain convergence. Finally, because the results of each load increment are stored as it is completed, the restart capability automatically provides a measure of protection against machine failure.

The reason that three separate input files (GDATA, GLOAD and GCONT) are used for the stress analysis program is in order to accommodate the restart capability efficiently. For an initial (*ie* non-restart) run all three files are required. However when a restart run is being prepared the data in the GDATA file remains unchanged and so the old version of this file is used. It is also possible to use the old version of the GLOAD file providing that the restart analysis is not to include any additional loadcases. The GCONT file which contains the control information is always generated irrespective of whether or not a restart analysis is being prepared. Thus by using three files duplication of the data is avoided, thus saving disc space and CPU time.

The data generation programs described in Chapter 3 perform extensive checking of the data and thus only the simplest checks need to be carried out by GANDALF1N, essentially to ensure that the various files contain the correct amount of data. If any of the checks in GANDALF1N fail the program is halted and no output files are created.

In order to assess the results of a three-dimensional finite element analysis the use of interactive graphics is essential. In the GANDALF suite this is provided by the commercial package FEMVIEW[66]. The interface between the stress analysis program and FEMVIEW is described in detail in Appendix B. A file containing the mesh coordinates and topology is produced by GANDALF1N, and this information is used to create a new database ready for access by FEMVIEW. If a restart analysis is being prepared then either a new database can be created, or the results can be added to the one used for the previous part of the analysis.

Chapter 5

FINITE ELEMENT STRESS ANALYSIS

5.1 Elastic analysis

Most physical problems are too complex to allow exact analytic solutions to be determined. One method of overcoming this difficulty is to model the continuum problem as a finite number of discrete components. This approach of discretizing a continuous problem has led to the development of a number of mathematical procedures, such as the finite difference method, various weighted residual procedures, and the Ritz method[67-71]. The finite element method however was developed primarily from the work of engineers. In the field of structural mechanics McHenry[72], Hrenikoff[73] and Newmark[74] showed that reasonable solutions to a continuum problem could be obtained by approximating small regions by an arrangement of elastic bars. Argyris[75] and Turner *et al*[76] showed that it was possible to approximate the continuum directly by simplifying the behaviour of each small region. The term 'finite element' was first used by Clough[77].

In the Ritz method a solution of the form

$$\Phi = \sum_{i=1}^{n} a_i f_i \qquad (5\text{-}1)$$

is assumed, where the f_i are linearly independent 'trial functions' which give the allowed variation of the solution across the domain, and the a_i are scalar multipliers to be determined in the solution. The Ritz method then requires that the a_i are calculated by finding the extreme value of a scalar functional π. Thus

$$\frac{\partial \pi}{\partial a_i} = 0 \qquad (5\text{-}2)$$

thus generating n equations for the unknown multipliers a_i.

The appropriate functional for structural analysis is the potential energy. Thus

$$\pi = U - W \qquad (5\text{-}3)$$

where U is the strain energy of the system, and W the potential energy of the applied loads.

In a finite element analysis the structure is divided into sub-regions, the elements, and the response of the whole structure is modelled by summing the behaviour of all the elements within it. If in a structural mechanics analysis the variation of displacement within each element is assumed to be a function of the displacements of discrete points within each element, known as nodes, then the procedure can be shown to be identical to the Ritz method, with the nodal displacements equivalent to the unknown multipliers a_i, and the element interpolation functions equivalent to the trial functions[78,79]. This insight enables the allowed forms of the interpolation functions, and the rate of convergence of the solution as the number of elements is increased to be determined[80].

A general purpose finite element package will have a large number of element types capable of analysing a wide range of geometries. Because the GANDALF package is designed specifically for the analysis of turbine blades only one element type has been included. This leads to considerable simplification of the code. The element used in GANDALF is a cube whose faces can be distorted to fit the geometry of the blade. The geometry of the element is specified by the coordinates of the element nodes.

The GANDALF element has 20 nodes, one in each corner and one at the middle of each edge (Fig 7). The coordinates of any point within the element are defined from the nodal values using the interpolation functions. In fact the value of any property such as temperature, nodal displacement, etc at a point within the element is found by such interpolation. If the same interpolation is used to define the coordinates of points within the element, and the values of the unknown displacements, the element is termed isoparametric. This approach was first used by Taig[81] who developed a linear quadrilateral element. Irons[82,83] generalized the idea for other elements.

The interpolation functions are low order polynomials known as shape functions, of a coordinate system (ξ,η,ζ) local to the element. In the GANDALF element the well established 'Serendipity' shape functions are used. These were first introduced for two-dimensional analysis by Ergatoudis, Irons and Zienkiewicz[84] and extended to three dimensions by Marples[85] and Zienkiewicz *et al*[86].

There will be one polynomial N_i associated with each node such that

$$N_i(\xi_i,\eta_i,\zeta_i) = 1, \qquad N_i(\xi_j,\eta_j,\zeta_j) = 0 \quad i \neq j \ , \tag{5-4}$$

where (ξ_i,η_i,ζ_i) and (ξ_j,η_j,ζ_j) are the local coordinates of the ith and jth nodes respectively. The value of a property p at the point (ξ,η,ζ) can be defined as

$$p(\xi,\eta,\zeta) = \sum_{j=1}^{n} N_j(\xi,\eta,\zeta)p_j \qquad (5\text{-}5)$$

where n is the number of nodes in the element and p_j is the value of the property at the jth node, and as required

$$p\left(\xi_i,\eta_i,\zeta_i\right) = p_i \ . \qquad (5\text{-}6)$$

If the vectors of the n nodal displacements relative to the global x, y and z axes are $\underline{u}_x$, $\underline{u}_y$ and $\underline{u}_z$ respectively, then the x displacement for example at any point (ξ,η,ζ) is defined as

$$u_x(\xi,\eta,\zeta) = \underline{N} \cdot \underline{u}_x \ , \qquad (5\text{-}7)$$

where $\underline{N} = [N_1,N_2,\ldots,N_n]$ are the element shape functions and the N_i are evaluated at the point (ξ,η,ζ). Thus differentiating

$$\frac{\partial u_x}{\partial \xi} = \frac{\partial \underline{N}}{\partial \xi} \cdot \underline{u}_x, \qquad \frac{\partial u_x}{\partial \eta} = \frac{\partial \underline{N}}{\partial \eta} \cdot \underline{u}_x, \qquad \frac{\partial u_x}{\partial \zeta} = \frac{\partial \underline{N}}{\partial \zeta} \cdot \underline{u}_x \qquad (5\text{-}8)$$

with similar expressions for the derivatives of u_y and u_z.

If the nodal global coordinates are given by $\underline{x}$, $\underline{y}$ and $\underline{z}$ then the coordinates (x,y,z) of any point (ξ,η,ζ) are given by equation (5-5)

$$x = \underline{N} \cdot \underline{x}, \qquad y = \underline{N} \cdot \underline{y}, \qquad z = \underline{N} \cdot \underline{z} \ . \qquad (5\text{-}9)$$

Thus

$$\frac{\partial x}{\partial \xi} = \frac{\partial \underline{N}}{\partial \xi} \cdot \underline{x}, \qquad \frac{\partial x}{\partial \eta} = \frac{\partial \underline{N}}{\partial \eta} \cdot \underline{x}, \qquad \frac{\partial x}{\partial \zeta} = \frac{\partial \underline{N}}{\partial \zeta} \cdot \underline{x} \ , \qquad (5\text{-}10)$$

with similar expressions for the y and z coordinates.

By the usual rules of partial differentiation

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \xi} \ . \qquad (5\text{-}11)$$

Thus by performing the same differentiation with respect to the other local coordinates

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = [J] \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \qquad (5\text{-}12)$$

where [J] is known as the Jacobian matrix. The terms in this matrix can be calculated from equation (5-10) and thus by inverting [J] the derivatives of the shape function with respect to the global coordinates (x,y,z) can be written

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = [J]^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} . \qquad (5\text{-}13)$$

For small strains, the strain at a point can be written (see for example Love[87] or Timoshenko and Goodier[88]),

$$\underline{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_z}{\partial z} \\ \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \\ \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \\ \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \end{bmatrix} . \qquad (5\text{-}14)$$

The global derivatives of the displacement u_x can be written

$$\frac{\partial u_x}{\partial x} = \frac{\partial \underline{N}}{\partial x} \cdot \underline{u}_x, \qquad \frac{\partial u_x}{\partial y} = \frac{\partial \underline{N}}{\partial y} \cdot \underline{u}_x, \qquad \frac{\partial u_x}{\partial z} = \frac{\partial \underline{N}}{\partial z} \cdot \underline{u}_x \qquad (5\text{-}15)$$

with similar expressions for the derivatives of u_y and u_z. The global derivatives of the shape functions are calculated using equation (5-13).

Thus we can write

$$\underline{\varepsilon} = \begin{bmatrix} \frac{\partial \underline{N}}{\partial x} & 0 & 0 \\ 0 & \frac{\partial \underline{N}}{\partial y} & 0 \\ 0 & 0 & \frac{\partial \underline{N}}{\partial z} \\ \frac{\partial \underline{N}}{\partial y} & \frac{\partial \underline{N}}{\partial x} & 0 \\ 0 & \frac{\partial \underline{N}}{\partial z} & \frac{\partial \underline{N}}{\partial y} \\ \frac{\partial \underline{N}}{\partial z} & 0 & \frac{\partial \underline{N}}{\partial x} \end{bmatrix} \begin{bmatrix} \\ \underline{u}_x \\ \\ \underline{u}_y \\ \\ \underline{u}_z \end{bmatrix} = [B]\underline{u}_{el} . \qquad (5\text{-}16)$$

The stresses will be related to the strain by the generalized Hooke's Law

$$\underline{\sigma} = [D]\underline{\varepsilon} \tag{5-17}$$

where $[D]$ is the standard matrix of elastic constants. For an isotropic material $[D]$ is simply a function of Young's Modulus and Poisson's ratio.

It is now necessary to relate the applied forces to the stresses within the structure. This relation is obtained by using the principle of virtual work which states that if a structure is in stable equilibrium and is then given a small arbitrary displacement the increase in internal strain energy ΔS equals ΔW the external work done in causing the displacement. In finite element analysis external loads are applied by resolving them into an equivalent set of forces acting at the nodes. Thus the external work done on an element by an arbitrary displacement vector $\Delta\underline{\delta}_{el}$ is

$$\Delta W_{el} = \Delta\underline{\delta}_{el}^{T}\underline{f}_{el} \tag{5-18}$$

where $\underline{f}_{el}$ is the vector of nodal forces acting on the element. This vector cannot in general be determined as it includes the forces imposed on the element by its neighbours.

The increase in internal strain energy of the element is given by the product of the increment of strain caused by the arbitrary displacement and the current stress. The strain increment from equation (5-16) is $[B]\Delta\underline{\delta}$ and thus using equation (5-17) the increase in internal strain energy assuming small displacements is given by

$$\Delta S_{el} = \int_{ve} \left([B]\Delta\underline{\delta}_{el}\right)^{T}[D][B]dv\ \underline{u}_{el} \tag{5-19}$$

where the integral is taking over the volume of the element.

The increase in internal strain energy equals the external work done and thus combining equations (5-18) and (5-19) and eliminating the arbitrary displacement $\Delta\underline{\delta}_{el}$

$$\int_{ve} [B]^{T}[D][B]dv\ \underline{u}_{el} = \underline{f}_{el} \cdot \tag{5-20}$$

The integral can only be evaluated analytically for the simplest elements and generally numerical integration is required. This approximates the integral by a weighted sum of the values of the integrand at specified sampling points. The Gaussian quadrature technique is used as this allows both the sampling points and the weights associated with them to be chosen so that in one dimension an 'n' point Gauss rule will exactly integrate a polynomial of degree '2n - 1'[89].

The required order of integration is, in principle, that required to evaluate the integral exactly. For the 20-node brick, if it is not distorted from its cubical shape, this means that 3 × 3 × 3 integration is required. However as the finite element method can be regarded as a Ritz analysis, it can be shown that the approximation to the system potential energy calculated using the finite element method will always be an overestimate. Thus, the predicted stiffness matrices are too stiff. This deficiency can be overcome to some extent by the use of so called 'reduced' integration where the integrand is evaluated at fewer points[80]. For ideally plastic materials obeying Von Mises' yield criterion plastic straining must occur at constant volume[90]. Nagtegaal *et al*[91] showed that the allowable deformations of most finite element meshes are incapable of satisfying this requirement. This leads to overestimation of collapse loads because the structure is overconstrained. Zienkiewicz[57] indicates how this problem can be overcome by the use of reduced integration. For these reasons, and because it requires less CPU time reduced integration is used in the GANDALF program. For the 20-node brick, this means 2 × 2 × 2 Gaussian quadrature.

As can be seen by reference to equation (5-16), the matrix [B] contains a significant proportion of zeroes. It is important, therefore, to organise the matrix multiplication in equation (5-20) to avoid multiplying by zero terms, as has been noted by many authors[92-94].

Equation (5-20) describes the equilibrium of a single element in terms of its nodal deflections. However some, if not all of the nodes will be shared by adjacent elements. Thus the corresponding nodal deflections will also appear in the equations generated by other elements. If all the equations produced by all the elements in the structure are added together a set of equations of the form

$$[K]\underline{u} = \underline{f} \qquad (5\text{-}21)$$

will be generated. This describes the nodal deflections $\underline{u}$ of the structure in terms of the structural force vector $\underline{f}$ and the stiffness

matrix [K]. In the assembly process the inter-element forces cancel out and $\underline{f}$ thus contains only the applied loads.

Equation (5-21) can be obtained more directly be considering the minimization of the potential energy of the structure. The potential energy π is given by

$$\pi = \tfrac{1}{2}\underline{u}^T[K]\underline{u} - \underline{u}^T\underline{f} \quad . \tag{5-22}$$

Hence

$$\delta\pi = [K]\underline{u} - \underline{f} = \underline{0} \tag{5-23}$$

as required.

The unknown vector of nodal displacements can be obtained using any technique for the solution of a set of linear equations, Gaussian elimination being the preferred algorithm. The implementation of this technique in the GANDALF program is discussed in section 7.1.

Once all the nodal displacements have been determined the stress at any point within each element can be calculated using equations (5-16) and (5-17). In practice the stresses are always evaluated at the Gauss quadrature points because these locations produce the most accurate results[95]. If results at nodes are required they are extrapolated from the Gauss point values.

5.2 Elasto-thermal analysis

In order to extend the analysis of section 5.1 to elasto-thermal problems it is necessary to include the effects of thermal strains and temperature dependent elastic constants.

By considering equations (5-16), (5-17), (5-20) and (5-21) it is easy to see that equation (5-21) is equivalent to

$$\int [B]^T\underline{\sigma} = \underline{f} \quad , \tag{5-24}$$

where the integral is now taken over the volume of the structure.

By considering equation (5-24) applied to two different equilibrium states it is readily seen that

$$\int [B]^T\Delta\underline{\sigma} = \Delta\underline{f} \quad . \tag{5-25}$$

At any temperature the stresses will be related to the elastic strains $\underline{\varepsilon}_e$ by the generalized Hooke's law

$$\underline{\sigma} = [D]\underline{\varepsilon}_e \quad . \tag{5-26}$$

Thus if we consider a change in state from condition 1 to condition 2 then

$$\begin{aligned} \Delta\underline{\sigma} \equiv \underline{\sigma}_2 - \underline{\sigma}_1 &= \left[D_2\right]\underline{\varepsilon}_{e_2} - \left[D_1\right]\underline{\varepsilon}_{e_1} \\ &= \left[D_2\right]\Delta\underline{\varepsilon}_e + [\Delta D]\underline{\varepsilon}_{e_1} \quad , \end{aligned} \tag{5-27}$$

where $[\Delta D] \equiv \left[D_2\right] - \left[D_1\right]$ and $\Delta\underline{\varepsilon}_e \equiv \underline{\varepsilon}_{e_2} - \underline{\varepsilon}_{e_1}$.

Substituting equation (5-27) in equation (5-25)

$$\int [B]^T \left[D_2\right] \Delta\underline{\varepsilon}_e = \Delta\underline{f} - \int [B]^T [\Delta D] \underline{\varepsilon}_{e_1} \quad . \tag{5-28}$$

In general the material will experience a thermal strain if the temperature at a point is different from some ambient temperature θ_a , which is usually taken as 20°C. Thus for a material displaying isotropic thermal expansion the thermal strains are given by

$$\underline{\varepsilon}_T = \left[\alpha T_d, \alpha T_d, \alpha T_d, 0, 0, 0\right]^T \quad , \tag{5-29}$$

where α is the mean coefficient of linear thermal expansion and $T_d \equiv T - T_a$ is the difference between the ambient temperature and the temperature of the point in the structure. In general the structure will be subject to a temperature distribution and $\underline{\varepsilon}_T$ will vary from point to point.

The total strain at any point is given by the sum of the elastic and thermal strains. Thus

$$\underline{\varepsilon} = \underline{\varepsilon}_e + \underline{\varepsilon}_T \tag{5-30}$$

and hence

$$\Delta\underline{\varepsilon} = \Delta\underline{\varepsilon}_e + \Delta\underline{\varepsilon}_T \quad . \tag{5-31}$$

Substituting (5-31) into (5-28) yields

$$\int [B]^T \left[D_2\right] \Delta\underline{\varepsilon} = \Delta\underline{f} - \int [B]^T [\Delta D] \underline{\varepsilon}_{e_1} + \int [B]^T \left[D_2\right] \Delta\underline{\varepsilon}_T \qquad (5\text{-}32)$$

and utilizing equation (5-16)

$$\int [B]^T \left[D_2\right] [B] \Delta\underline{u} = \Delta\underline{f} - \int [B]^T [\Delta D] \underline{\varepsilon}_{e_1} + \int [B]^T \left[D_2\right] \Delta\underline{\varepsilon}_T \qquad (5\text{-}33)$$

or

$$\int [B]^T \left[D_2\right] [B] \Delta\underline{u} = \Delta\underline{f} + \Delta\underline{f}_e + \Delta\underline{f}_T \ . \qquad (5\text{-}34)$$

Equation (5-34) thus gives the unknown displacement increment $\Delta\underline{u}$ in terms of the external load $\Delta\underline{f}$ and two so-called 'pseudo loads' $\Delta\underline{f}_e$ and $\Delta\underline{f}_T$. $\Delta\underline{f}_e$ is dependent on the current elastic strain in the structure and the change in elastic constants over the loading step. $\Delta\underline{f}_T$ is dependent on the increment of thermal strains in the structure and the value of the elastic constants at the new temperature distribution in the structure. It should be noted that equation (5-34) is valid irrespective of the size of load steps employed. Once $\Delta\underline{u}$ has been determined, the new total stress is calculated using equations (5-16), (5-27) and (5-31).

5.3 Plastic analysis

When portions of a structure enter the plastic regime the relation between stress and strain becomes nonlinear. In order to perform a finite element analysis it is necessary to derive a stress-strain law which reflects the current state of stress in the structure. Plasticity theory is reviewed in Refs 90 and 96 to 99.

The first recorded investigation of metal plasticity is due to Tresca[100], who in 1864 published an account of punching and extrusion processes, which led him to conclude that a metal yielded plastically when the maximum shear stress attained a critical value. Saint-Venant[101] proposed that during plastic flow the principal axes of the total strain increment coincide with the principal axes of stress. Levy[102] obtained the general relation in three dimensions. Levy's equations were suggested independently by Von Mises[103], who also proposed his well known yield criterion. Prandtl[104] and Reuss[105] extended the Levy-Mises equation to include elastic strains. The Prandtl-Reuss equations relate increments of plastic strain to the current stress. Hencky[106] related the total plastic strain to the current stress. It is easy to see that this

approach is incorrect. If the material is unloaded after a certain plastic deformation and then reloaded to yield at a different stress state, Hencky's theory would predict that the plastic strain should also have changed, which is absurd as no plastic straining can take place below the yield stress.

In addition to a criterion for yield and a relationship between stress and plastic straining it is also necessary to establish a hardening rule to model how the yield criterion is modified by plastic straining. Two well established criteria are work hardening and strain hardening. In the work hardening hypothesis the degree of hardening is related to the plastic work defined as

$$Wp = \int \sigma_{ij} \left(d\varepsilon_{ij}\right)_p \qquad (5\text{-}35)$$

where the integral is taken along the loading path. In strain hardening the degree of hardening is related to an effective plastic strain defined incrementally as

$$d\bar{\varepsilon}_p = \sqrt{\frac{2}{3}} \left\{ \left(d\varepsilon_{ij}\right)_p \left(d\varepsilon_{ij}\right)_p \right\}^{\frac{1}{2}} . \qquad (5\text{-}36)$$

Then the degree of hardening is assumed to be related to $\bar{\varepsilon}_p$. Bland[107] showed that these hypotheses were equivalent, *ie* $\bar{\varepsilon}_p$ is a function of Wp provided that the stress function describing the yield surface is linear or quadratic in the principal components of stress. Thus the above two criteria are equivalent for both the Von Mises and Tresca yield surfaces except at singular points of the Tresca surface. It should be noted however that if the Tresca yield surface is used the expression relating the stress and total strain increments obtained using the work hardening criterion is not identical to that obtained using the strain hardening theory, even at points away from the singularity. However the Von Mises surface produces identical expressions whichever of the criteria is used.

Because the work and strain hardening theories relate the degree of hardening to a single scalar variable, they both assume that the yield surface expands uniformly about its original position, so called isotropic hardening. This is at variance with the observed behaviour of many metals which exhibit a Bauschinger effect, whereby for example the magnitude of the yield stress in compression after tensile yielding is lower than that of the virgin material. To overcome this deficiency

Prager[108] proposed the so called kinematic hardening model whereby the yield surface was translated in stress space by hardening. Ziegler[109] modified Prager's model by allowing translation to occur parallel to the vector between the current stress state and the updated position of the centre of the yield surface. More complex hardening theories have also been proposed[110-112]. It should be noted that on the basis of thermodynamic arguments, the initial and all subsequent yield surfaces must be convex, as shown by Drucker[113]. A comparison of the performance of various hardening theories has been given by Hunsaker, Vaughan and Stricklin[114].

In order to develop plasticity theory in a form suitable for finite element analysis the first requirement is an equation to describe whether the material is in the plastic regime. This is given in general by

$$F(\underline{\sigma},\underline{\kappa},T) = 0 \quad . \tag{5-37}$$

This equation was first given by Prager[115] and describes all possible plastic states as a function of the applied stress, a (theoretically infinite) number of hardening parameters $\underline{\kappa}$ which account for the prior history of the material, and the temperature T. If the material is assumed to be isotropic, the yielding can only depend on the magnitude of the three stress invariants. Experimental observations[116] indicate, that to a good approximation plastic yielding of metals is unaffected by hydrostatic stress. Thus the yield criterion must only be a function of the deviatoric stress invariants J_2' and J_3' as J_1' is identically zero. In the GANDALF code the stress function used in the yield criterion is that of Von Mises. This has the considerable advantage of simplicity, and was shown to be more accurate than the Tresca criterion by Lode[117] and Taylor and Quinney[118]. Winstone[119] using tension-torsion tests on thin walled tubes of the nickel superalloy Mar-M 002 found that the initial yield surfaces lay between the Von Mises and Tresca criteria. As is well known the Von Mises criterion can be written

$$\left(3J_2'\right)^{\frac{1}{2}} = 0 \tag{5-38}$$

thus satisfying the requirements given above.

The work hardening rule is used in the code. It is considered that this will be adequate to model initial yield of turbine blade materials. However, if reverse yielding is encountered a more complex kinematic or combined hardening model may be required. Winstone[119] noted that considerable distortion to the original yield surface can occur after

yielding with the surface developing corners. The use of the work hardening rule implies that (*cf* equation (5-35)) only a single parameter κ is used and hence that only isotropic hardening occurs. Thus

$$d\kappa = dWp = \underline{\sigma}^T d\underline{\varepsilon}_p \quad . \tag{5-39}$$

It is assumed that the plastic strain increment is proportional to the derivative with respect to stress of a 'plastic potential' Q. Thus

$$d\underline{\varepsilon}_p = d\lambda \frac{\partial Q}{\partial \underline{\sigma}} \quad , \tag{5-40}$$

where $d\lambda$ is a constant of proportionality to be determined. This equation is known as the flow rule. It is normal to assume that $Q \equiv F$. This is termed the associated theory of plasticity[120]. It can be shown that if F is the Von Mises yield criterion then with associated plasticity equation (5-40) reduces directly to the Prandtl-Reuss relations.

When a state of plastic flow exists equation (5-37) must remain true, *ie* $F(\underline{\sigma},\kappa,T) = 0$. Thus

$$dF = \left(\frac{\partial F}{\partial \underline{\sigma}}\right)^T d\underline{\sigma} + \frac{\partial F}{\partial \kappa} d\kappa + \frac{\partial F}{\partial T} dT = 0 \quad . \tag{5-41}$$

If we define $\underline{a} \equiv \partial F/\partial \underline{\sigma}$ and $-Ad\lambda \equiv (\partial F/\partial \kappa)d\kappa$ equation (5-41) becomes

$$dF = \underline{a}^T d\underline{\sigma} - Ad\lambda + \frac{\partial F}{\partial T} dT = 0 \quad . \tag{5-42}$$

Hence

$$\underline{a}^T d\underline{\sigma} = Ad\lambda - \frac{\partial F}{\partial T} dT \quad . \tag{5-43}$$

The flow rule (equation (5-40)) can be rewritten as

$$d\underline{\varepsilon}_p = d\lambda \underline{a} \quad . \tag{5-44}$$

Considering a change in state from condition 1 to condition 2 the change in stress is

$$d\underline{\sigma} = \underline{\sigma}_2 - \underline{\sigma}_1 \tag{5-45}$$

and using equation (5-27)

$$d\underline{\sigma} = \left[D_2\right]d\underline{\varepsilon}_e + [dD]\underline{\varepsilon}_{e_1} \quad . \tag{5-46}$$

It is assumed that the total strain increment can be written as the sum of elastic, plastic and thermal components, *ie*

$$d\underline{\varepsilon} = d\underline{\varepsilon}_e + d\underline{\varepsilon}_p + d\underline{\varepsilon}_T \quad . \tag{5-47}$$

Premultiplying equation (5-47) by $\left[D_2\right]$

$$\left[D_2\right]d\underline{\varepsilon} = \left[D_2\right]d\underline{\varepsilon}_e + \left[D_2\right]d\underline{\varepsilon}_p + \left[D_2\right]d\underline{\varepsilon}_T \tag{5-48}$$

and utilizing equations (5-44), (5-46)

$$\left[D_2\right]d\underline{\varepsilon} = d\underline{\sigma} - [dD]\underline{\varepsilon}_{e_1} + \left[D_2\right]d\lambda\underline{a} + \left[D_2\right]d\underline{\varepsilon}_T \quad . \tag{5-49}$$

Premultiplying by $\underline{a}^T$ gives

$$\underline{a}^T\left[D_2\right]d\underline{\varepsilon} = \underline{a}^T d\underline{\sigma} - \underline{a}^T[dD]\underline{\varepsilon}_{e_1} + \underline{a}^T\left[D_2\right]d\lambda\ \underline{a} + \underline{a}^T\left[D_2\right]d\underline{\varepsilon}_T \quad . \tag{5-50}$$

From equation (5-43) $\underline{a}^T d\underline{\sigma} = Ad\lambda - \frac{\partial F}{\partial T}\,dT$ and hence, substituting into equation (5-50)

$$d\lambda = \frac{\underline{a}^T\left[D_2\right]d\underline{\varepsilon} + \underline{a}^T[dD]\underline{\varepsilon}_{e_1} + \frac{\partial F}{\partial T}\,dT - \underline{a}^T\left[D_2\right]d\underline{\varepsilon}_T}{A + \underline{a}^T\left[D_2\right]\underline{a}} \quad . \tag{5-51}$$

Rearranging equation (5-49)

$$d\underline{\sigma} = \left[D_2\right]d\underline{\varepsilon} + [dD]\underline{\varepsilon}_{e_1} - \left[D_2\right]d\lambda\ \underline{a} - \left[D_2\right]d\underline{\varepsilon}_T \quad . \tag{5-52}$$

Thus substituting for $d\lambda$ the elasto-plastic constitutive relationship becomes

$$d\underline{\sigma} = \left[\left[D_2\right] - \frac{\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]}{A + \underline{a}^T\left[D_2\right]\underline{a}}\right]d\underline{\varepsilon} + \left[[dD] - \frac{\left[D_2\right]\underline{a}\underline{a}^T[dD]}{A + \underline{a}^T\left[D_2\right]\underline{a}}\right]\underline{\varepsilon}_{e_1} - \frac{\left[D_2\right]\underline{a}\,\frac{\partial F}{\partial T}\,dT}{A + \underline{a}^T\left[D_2\right]\underline{a}}$$

$$- \left[\left[D_2\right] - \frac{\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]}{A + \underline{a}^T\left[D_2\right]\underline{a}}\right]d\underline{\varepsilon}_T \quad . \qquad \ldots\ldots \text{(5-53)}$$

It is now necessary to determine the value of the scalars A and $(\partial F/\partial T)dT$. If a uniaxial tensile test at constant temperature is considered then the yield criterion (equation (5-37)) can be written in the form

$$F(\underline{\sigma},\kappa) = f(\underline{\sigma}) - \sigma_y(\kappa) = 0 \qquad \text{(5-54)}$$

since the temperature is constant, where σ_y is the uniaxial yield stress.

Thus

$$\frac{\partial F}{\partial \kappa} = -\frac{d\sigma_y}{d\kappa} \qquad \text{(5-55)}$$

and from the definition of A and equation (5-55)

$$A = -\frac{1}{d\lambda}\frac{\partial F}{\partial \kappa}\,d\kappa = \frac{1}{d\lambda}\frac{d\sigma_y}{d\kappa}\,d\kappa \quad . \qquad \text{(5-56)}$$

From the flow rule and equation (5-39)

$$d\kappa = \underline{\sigma}^T d\underline{\varepsilon}_p = \underline{\sigma}^T d\lambda \frac{\partial F}{\partial \underline{\sigma}} = d\lambda\left(\frac{\partial F}{\partial \sigma}\right)^T \underline{\sigma} \quad . \qquad \text{(5-57)}$$

Substituting (5-57) in (5-56) gives

$$A = \frac{d\sigma_y}{d\kappa}\left(\frac{\partial F}{\partial \underline{\sigma}}\right)^T \underline{\sigma} \quad . \qquad \text{(5-58)}$$

For the uniaxial case $\underline{\sigma}$ is replaced by $\bar{\sigma} = \sigma_y$ and $d\underline{\varepsilon}_p$ is replaced by $d\bar{\varepsilon}_p$ where $\bar{\sigma}$ and $\bar{\varepsilon}_p$ are the Von Mises' effective stress and strain. Thus from equation (5-39)

$$d\kappa = \sigma_y d\bar{\varepsilon}_p \quad . \tag{5-59}$$

It can be shown using Euler's theorem of homogeneous functions that

$$\left(\frac{\partial F}{\partial \underline{\sigma}}\right)^T \underline{\sigma} = \sigma_y \quad . \tag{5-60}$$

and thus combining equations (5-58), (5-59) and (5-60)

$$A = \frac{d\sigma_y}{d\bar{\varepsilon}_p} \quad . \tag{5-61}$$

Thus A is given by the slope of the uniaxial stress versus plastic strain curve at the current stress level.

In order to determine the value of $(\partial F/\partial T)dT$ consider a uniaxial test where the temperature is changed at constant plastic strain. From Equation (5-39) $d\kappa = 0$ and hence from equation (5-41)

$$dF = \frac{\partial F}{\partial \underline{\sigma}} d\underline{\sigma} + \frac{\partial F}{\partial T} dT = 0 \quad . \tag{5-62}$$

It can be shown that, for the uniaxial case

$$\frac{\partial F}{\partial \underline{\sigma}} d\underline{\sigma} = d\sigma_y \quad . \tag{5-63}$$

Hence by equation (5-62)

$$\frac{\partial F}{\partial T} dT = - d\sigma_y \quad . \tag{5-64}$$

Thus $(\partial F/\partial T)dT$ is given by the change in effective stress from one temperature to another at constant plastic strain.

Equation (5-53) can be written as

$$d\underline{\sigma} = [Dep]d\underline{\varepsilon} + [dDep]\underline{\varepsilon}_{e_1} - d\underline{\sigma}_T - [Dep]d\underline{\varepsilon}_T \quad , \tag{5-65}$$

$$\text{where} \quad [\text{Dep}] = \left[[D_2] - \frac{[D_2]\underline{a}\underline{a}^T[D_2]}{A + \underline{a}^T[D_2]\underline{a}} \right] ,$$

$$[\text{dDep}] = \left[[\text{dD}] - \frac{[D_2]\underline{a}\underline{a}^T[\text{dD}]}{A + \underline{a}^T[D_2]\underline{a}} \right] ,$$

$$d\underline{\sigma}_T = \frac{[D_2]\underline{a}\,\frac{\partial F}{\partial T}\,dT}{A + \underline{a}^T[D_2]\underline{a}} .$$

It should be noted that because $\underline{a}$, A and $(\partial F/\partial T)dT$ are stress dependent, equation (5-65) is only valid for infinitesimal increments of stress and strain. In practice the load is applied incrementally. In order to relate the stress increment to the applied loads equation (5-25) is invoked. Thus

$$\int [B]^T \Delta\underline{\sigma} = \Delta\underline{f} . \qquad (5\text{-}66)$$

If equation (5-65) is assumed to be valid for finite increments of stress and strain then

$$\Delta\underline{\sigma} = [\text{Dep}]\Delta\underline{\varepsilon} + [\Delta\text{Dep}]\underline{\varepsilon}_e - \Delta\underline{\sigma}_T - [\text{Dep}]\Delta\varepsilon_T . \qquad (5\text{-}67)$$

Combining equations (5-66) and (5-67), and utilizing equation (5-16)

$$\int [B]^T[\text{Dep}][B]\,\Delta\underline{u} = \Delta\underline{f} - \int [B]^T[\Delta\text{Dep}]\underline{\varepsilon}_e + \int [B]^T\Delta\underline{\sigma}_T + \int [B]^T[\text{Dep}]\Delta\underline{\varepsilon}_T$$

..... (5-68)

or

$$[K_T]\Delta\underline{u} = \Delta\underline{f} + \Delta\underline{f}_p . \qquad (5\text{-}69)$$

Equation (5-69) enables the displacement increments resulting from an imposed temperature and external load change to be determined. The matrix $[K_T]$ which is known as the tangent stiffness matrix was first introduced by Pope[121]. The middle two terms on the right-hand side of equation (5-68) are 'pseudo' forces which account for the temperature dependence of the material properties. From the displacement increments

the total strain increment can be determined, and hence the stresses from equation (5-65).

Because equation (5-69) is used for finite increments of load it is found that the numerical solution can drift from the correct result. In order to overcome this it is normal to use an iterative procedure to ensure that the equilibrium equation (equation (5-66)) is satisfied at the end of each load step. Various iterative procedures are discussed in section 7.2.

In theory it is possible to adopt an alternative approach to overcome the disadvantages of equation (5-69). From equation (5-27)

$$\Delta\underline{\sigma} = \left[D_2\right]\Delta\underline{\varepsilon}_e + [\Delta D]\underline{\varepsilon}_e$$

$$= \left[D_2\right]\Delta\underline{\varepsilon} - \left[D_2\right]\Delta\underline{\varepsilon}_p + [\Delta D]\underline{\varepsilon}_e - \left[D_2\right]\Delta\underline{\varepsilon}_T \quad \ldots\ldots (5\text{-}70)$$

Hence, utilising equation (5-16) and (5-66)

$$\int [B]^T\left[D_2\right][B]\,\Delta\underline{u} = \Delta\underline{f} + \int [B]^T\left[D_2\right]\Delta\underline{\varepsilon}_p - \int [B]^T[\Delta D]\underline{\varepsilon}_e + \int [B]^T\left[D_2\right]\Delta\underline{\varepsilon}_T \quad \ldots\ldots (5\text{-}71)$$

The above equation, unlike equation (5-69) is exact for finite load increments, and is termed the initial strain method. However, the value of $\Delta\underline{\varepsilon}_p$ is not known at the start of the load increment, and therefore it must be estimated. This disadvantage led to the early rejection of the initial strain procedure for plasticity problems by Zienkiewicz, Valliappan and King[122]. However various workers at Grumman Aerospace[123-126] appear to have successfully used the procedure by estimating the plastic strain increment from that calculated in the previous step. It is felt however that the initial strain approach is not suitable for the blade analysis problem. Significant changes in the plastic strain increment per load step can be expected as a cyclic analysis is required. It was felt therefore that the use of the initial strain method could result in an overestimate of the plastic strains. As plastic deformation is nonconservative this would then lead to errors in the predicted stresses.

5.4 Creep analysis

The previous three parts of this chapter have considered time independent deformation. This section describes how creep problems can be analysed using the finite element method.

The phenomenon of creep in metals is manifested as a time dependent deformation under applied load. Generally three stages of creep may be observed, known as primary, secondary and tertiary creep, corresponding to reducing, steady and increasing deformation rates respectively. Creep rates are also strongly dependent on stress and temperature. In general therefore an equation of the form

$$\varepsilon_c = \varepsilon_c(\sigma, T, t) \tag{5-72}$$

will describe a family of constant stress and temperature creep curves. The first systematic attempt to derive an equation of this form was reported by Andrade[127] in 1910. Since that date many other more or less empirical equations have been proposed[128-133].

The form of equation (5-72) used in the default CRPLAW subroutine in the code is the so called θ' equation, introduced by Evans, Parker and Wilshire[133]. This takes the form

$$\varepsilon_c = \theta_1\left(1 - e^{-\theta_2 t}\right) + \theta_3\left(e^{\theta_4 t} - 1\right) \quad . \tag{5-73}$$

The terms θ_i are given by

$$\ln \theta_i = a_i + b_i T + c_i \sigma + d_i \sigma T \quad , \tag{5-74}$$

where a_i, b_i, c_i and d_i are constants. The values of these constants for the nickel superalloy Mar-M 002 were already available from previous work[134]. An assessment of the accuracy with which this equation fits the original data is given in Appendix C. The advantage of equation (5-73) for numerical work is that it is continuously differentiable for all values of time, in contrast to the Graham and Walles equation[131] which is of the form

$$\varepsilon_c = at^{\frac{1}{3}} + bt + ct^3 \tag{5-75}$$

and will thus produce an infinite creep rate at $t = 0$.

As is well known, equation (5-72) is insufficient to describe the creep behaviour of materials under time varying temperatures and stresses[135]. It is necessary to account for the dependence of creep rate on the prior history of the material. In essence, this can be done in two ways. The first of these uses the Boltzman superposition principle to integrate the past history. This approach is extremely cumbersome as it requires data on the whole of the stress and temperature history of the material. However it has been used successfully[136]. The simpler alternative approach, which is normally used, is to define an equation of state such that

$$\dot{\varepsilon}_c = \dot{\varepsilon}_c\left(\sigma, T, \kappa_1, \kappa_2, \ldots\right) \quad , \tag{5-76}$$

where κ_1, κ_2, ..., (as in plasticity) are a theoretically infinite number of state variables which describe the prior history of the material. The most common choice is to use one state variable, either the elapsed time t (so called time hardening) or the current creep strain ε_c (so called strain hardening) first introduced by Davenport[137]. Thus for time hardening

$$\dot{\varepsilon}_c = \dot{\varepsilon}_c(\sigma, T, t) \tag{5-77}$$

and for strain hardening

$$\dot{\varepsilon}_c = \dot{\varepsilon}_c\left(\sigma, T, \varepsilon_c\right) \quad . \tag{5-78}$$

As can be seen equation (5-77) can be derived directly by differentiating equation (5-72). However for strain hardening theory this in general is not possible and a numerical procedure must be used. Considering a change in stress and temperature from condition 1 to condition 2, a creep pseudo time t^* is found for which

$$\varepsilon_c\left(\sigma_2, T_2, t^*\right) = \varepsilon_c \quad . \tag{5-79}$$

The creep rate is then calculated as

$$\dot{\varepsilon}_c = \frac{\partial \varepsilon_c\left(\sigma_2, T_2, t^*\right)}{\partial t} \quad . \tag{5-80}$$

The disadvantage of time hardening theory lies in the fact that the creep rupture life of most materials varies markedly with stress and

temperature. Thus it is possible for the model to predict immediate rupture of the material under some stress and temperature changes. This is also true for the strain hardening theory, although because creep rupture strains are much less stress and temperature dependent, the problem is less likely to occur. This difficulty can be overcome by the use of a life or strain fraction hardening rule in which the current time or strain is normalized by the value at rupture. However the work of Harrison[138] established the validity of the strain hardening approach for superalloys and this has been confirmed by recent work on the cast blade alloy Mar-M 002 by Wilshire and Evans[139]. The strain hardening approach has therefore been used in the present work. Rashid[140] pointed out that equation (5-78) is not appropriate if stress changes from tensile to compressive loading are considered, and that more accurate results can be obtained if the strain hardening parameter is given by

$$\varepsilon_c^* = \int_0^t \left|\dot{\varepsilon}_c\right| dt \quad , \tag{5-81}$$

ie a monotonically increasing function. This improvement has been incorporated into the code. The user cannot change the hardening rule.

The first attempt to investigate creep under multiaxial stress conditions was conducted by Bailey[141] using tension-torsion tests of hollow cylinders. This technique was also used by Tapsell and Johnson[142]. An extensive series of investigations on a wide range of materials was conducted by Johnson and coworkers and these are reviewed in Ref 143. Reviews of multiaxial creep theory have been published by Boresi and Sidebottom[144] and Rees[145]. It is now generally accepted that the most appropriate theory to describe multiaxial creep behaviour is that due to Odqvist[146] which is based on the following assumptions:

(1) Creep occurs at constant volume.

(2) Creep rates are independent of a superimposed hydrostatic pressure.

(3) The direction of principal stress and strain rates coincide.

(4) A creep potential exists, *ie* the 'rate of dissipation' $\dot{W}$ defined as:

$$\dot{W} = \underline{\sigma}^T \cdot \underline{\dot{\varepsilon}}_c \tag{5-82}$$

is a scalar function of the applied stresses.

If the Von Mises effective stress is used for the creep potential these assumptions lead to an equation of the form

$$\dot{\underline{\varepsilon}}_c = \dot{\bar{\varepsilon}}_c \frac{\partial \bar{\sigma}}{\partial \underline{\sigma}} \tag{5-83}$$

where $\bar{\varepsilon}_c$ is the effective creep strain defined as

$$\bar{\varepsilon}_c = \sqrt{\frac{2}{3}} \left\{ \left(\varepsilon_{ij}\right)_c \left(\varepsilon_{ij}\right)_c \right\}^{\frac{1}{2}} \tag{5-84}$$

and $\bar{\sigma}$ is the Von Mises stress.

With these definitions of $\bar{\sigma}$ and $\bar{\varepsilon}_c$, the value of $\dot{\bar{\varepsilon}}_c$ is identical to the creep rate in a uniaxial creep test, and the hardening rules can then be used in the form

$$\dot{\bar{\varepsilon}}_c = \dot{\bar{\varepsilon}}_c \left(\bar{\sigma}, T, \bar{\varepsilon}_c^* \right) , \tag{5-85}$$

etc. To establish the validity of equation (5-83) a series of tension/torsion creep tests on Mar-M 002 were conducted at the National Engineering Laboratory under the supervision of Dr J. Henderson. Results from the tests are assessed in Appendix C. As indicated by Rees[145] it is probable that equation (5-85) should be modified as creep proceeds to account for developing anisotropy in the material. However the required tests to establish the degree of anisotropy have not, to the author's knowledge been performed at the temperature of interest and would indeed be extremely difficult. The simplifying assumption is made therefore that equation (5-85) is valid throughout the history of the material.

Zienkiewicz[57] in the first edition of his book published in 1967 showed how finite element creep analysis could be undertaken using an initial strain solution procedure. A timestepping solution scheme is employed. Assuming initially that plasticity does not occur and that the elastic properties of the material are unaffected by temperature, the change of stress on the nth timestep at any point in the structure will be related to the change in total and creep strains:

$$\Delta \underline{\sigma}^n = [D] \left(\Delta \underline{\varepsilon}^n - \Delta \underline{\varepsilon}_c^n \right) . \tag{5-86}$$

Hence, using the equilibrium equation (5-25) and the strain displacement relation (5-16)

$$\int [B]^T [D][B] \Delta \underline{u} = \Delta \underline{f} + \int [B]^T [D] \Delta \underline{\varepsilon}_c^n \quad . \tag{5-87}$$

Thus, if the creep strain increment can be determined the resulting displacement increment can be calculated, and hence the change in stress in the structure over the timestep can be evaluated.

The creep law, in order to include a hardening rule, will generally be written in rate form, and thus in order to determine the creep strain increment it is necessary to solve a first-order ordinary differential equation. In general a wide range of methods exist (see for example Van der Houwen[147]). However in the context of finite-element analysis, the number of possible methods is severely restricted. In general it is not practicable to store the value of $\underline{\varepsilon}_c$ at each integration point in the structure for more than the current condition, and if yielding or load cycling is occurring the value of $\dot{\underline{\varepsilon}}_c$ may change markedly from timestep to timestep. Thus the use of so called 'multi-step' methods[148] which estimate the new value of a variable on the basis of the values at a number of previous steps is generally impracticable. Runge-Kutta schemes require multiple evaluations of the creep rate within the timestep. This could be computationally expensive if the creep law is complicated - however a fourth order Runge-Kutta scheme was used successfully by Donea and Guiliani[149].

A family of possible single step methods is given by the expression

$$\Delta \underline{\varepsilon}_c^n = \Delta t_n \left[(1 - \theta) \dot{\underline{\varepsilon}}_c^n + \theta \dot{\underline{\varepsilon}}_c^{n+1} \right] \quad , \tag{5-88}$$

where Δt_n is the timestep, and $\dot{\underline{\varepsilon}}_c^n$ and $\dot{\underline{\varepsilon}}_c^{n+1}$ denote the creep rates at the start and end of the timestep respectively. θ is a constant which controls the type of solution scheme*. The case when $\theta = 0$ is the standard explicit or Euler scheme because the creep strain increment is completely determined from the existing conditions. This then leads directly to the initial strain method described above. If $\theta > 0$ it is necessary to determine the creep rate at the end of the timestep, which will be dependent on the changes in conditions over the timestep.

The first attempt to include the effect of stress change within the timestep on the creep strain increment was made by Cyr and Teter[150]. They used Soderberg's steady-state creep law, which was analytically

---

* NB It is important to distinguish between θ and the θ' creep equation.

integrated over the timestep, and hence produced a direct relationship between the increment of stress and the increment of total strain. A similar approach was adopted by Sharifi and Yates[151]. These analyses both included plasticity and thermal effects.

Clearly, for a general creep law involving hardening this approach is not adequate, as it will not in general be possible to integrate the creep law analytically. Zienkiewicz and Cormeau[152,153], overcame this difficulty by use of a predictor-corrector scheme to update the estimate of the creep strain increment. However each repetition of the corrector involved a resolution of the structural stiffness equations, and it does not appear that the scheme was very effective in practice.

As stated above if values of θ greater than zero are used in equation (5-88) it is clearly necessary to calculate or estimate $\dot{\underline{\varepsilon}}_c^{n+1}$. However $\dot{\underline{\varepsilon}}_c^{n+1}$ is a function of the stresses at the end of the timestep which of course cannot be determined without solving the structural equilibrium equations. This can be overcome by using the forward gradient method and estimating $\dot{\underline{\varepsilon}}_c^{n+1}$ using a Taylor series expansion. Thus

$$\dot{\underline{\varepsilon}}_c^{n+1} = \dot{\underline{\varepsilon}}_c^{n} + \left[\frac{\partial \dot{\underline{\varepsilon}}_c}{\partial \underline{\sigma}}\right] \Delta \underline{\sigma}^n \quad . \tag{5-89}$$

Hence

$$\Delta \underline{\varepsilon}_c^n = \left[\dot{\underline{\varepsilon}}_c^n + \theta \left(\frac{\partial \dot{\underline{\varepsilon}}_c}{\partial \underline{\sigma}}\right) \Delta \underline{\sigma}^n\right] \Delta t \quad . \tag{5-90}$$

This scheme (with $\theta = \frac{1}{2}$) was first used by Zienkiewicz[154]. He pointed out that if the creep rate was a linear function of the unknown stresses, etc the method would be identical to the well known trapezoidal or Crank-Nicholson formula, with its desirable convergence properties (it must be remembered however that this is not physically realistic for metallic creep).

Snyder and Bathe[155] developed a combined elasto-plastic and creep code. However the solution techniques were aimed at dynamic analysis and only the initial strain method was used for creep.

Cailletaud and Chaboche[156] used a second-order Euler scheme, and approximated the creep strain increment as

$$\Delta \underline{\varepsilon}_c = \dot{\underline{\varepsilon}}_{c,n} \Delta t + \ddot{\underline{\varepsilon}}_{c,n} \frac{\Delta t^2}{2} \quad . \tag{5-91}$$

Although this will represent an improvement over the standard Euler scheme, this method neglects the major source of nonlinearity in the problem, the dependence of creep rate on stress and temperature. In the opinion of the author therefore this is unlikely to represent a significant improvement over the Euler scheme. Chaudonneret *et al*[157] attempted to compare the scheme with the predictor-corrector method proposed by Zienkiewicz and Cormeau[152,153] but unfortunately the paper contains too little information to enable a valid comparison to be made.

Kim and Kuhlemeyer[158] used a time hardening creep theory, and interpolated between the creep rates at the start and the end of the timestep in log time. The creep rate at the end of the timestep was estimated using a two-term Taylor series. These assumptions enabled the creep strain increment over the timestep to be estimated by analytically integrating the creep rate over the timestep. Although this method represents an advance over the basic initial strain explicit theory, the use of a time hardening creep theory is not generally considered appropriate, as discussed above.

Kanchi, Zienkiewicz and Owen[159] used the forward gradient method to analyse materials where the creep rate is a function of damage parameters whose evolution is also rate dependent, for example the Kachanov damage equations[160], thus leading in theory to a set of coupled differential equations. However in the analysis they decoupled the equations, and thus neglected the effects of changes of the damage parameters within the timestep.

Bass, Johnson and McGrath[161] and Marques and Owen[162] derived the equations for the forward gradient technique with a strain hardening model. Marques and Owen[162] calculated the increment of effective visco-plastic strain (which governed the hardening of the material) using a simple Euler scheme. This simplifying assumption was not made by Bass, Johnson and McGrath[161]. In both cases it proved necessary to invert a tangent flexibility matrix in order to obtain the desired relationship for an increment of stress in terms of an increment of total strain.

Peirce, Shih and Needleman[163] in an important paper, showed that the inversion of the flexibility matrix is unnecessary. In the GANDALF program this approach has been extended to include plasticity and thermal effects, leading to the analysis presented in section 6.2.

If an explicit scheme is used ($\theta = 0$) it can be seen from equation (5-87) that the stiffness matrix remains constant for all time-steps (assuming that there is no plasticity). This considerably reduces

the amount of computation required on each step. However it can be shown that the scheme is mathematically unstable if the timestep size exceeds a certain critical value. Expressions for the critical time-step size were derived by Cormeau[164]. Cormeau's formula is

$$\Delta t \leqslant \frac{4(1+\nu)}{3E} \cdot \min\left(\frac{\bar{\sigma}}{\dot{\bar{\varepsilon}}_c}, \frac{1}{\partial \dot{\bar{\varepsilon}}_c / \partial \bar{\sigma}}\right), \qquad (5\text{-}92)$$

where the minimum is taken over all the integration points in the structure. It is noteworthy that this stability limit is independent of the finite-element discretization used in the analysis.

Bassani and McClintock[165] studied stress relaxation around a crack tip using an explicit timestepping scheme. They used the timestep limit given by Cormeau[164], but found that the solution was still stable and accurate if occasionally larger steps of 5 to 50 times the stability limit were used. However, in their analyses, the stresses in the region which control the size of the timestep (around the crack tip) are continuously decreasing, which might account for the success of this method.

As shown in section 6.2, if an implicit scheme ($\theta > 0$) is used the stiffness matrix must be reformulated on each timestep. Hughes and Taylor[166] investigated the stability of the scheme, and concluded that as with a linear system the algorithm was unconditionally stable for $\theta > \frac{1}{2}$. However, because in this case the stability of the system depends on the unknown stresses at the end of the timestep, they were unable to derive a general formula for a stable timestep for $0 < \theta < \frac{1}{2}$. Owen and Damjanic[167] derived a general stability criterion for problems where the displacements are fully constrained in the form

$$\Delta t < \frac{2}{W_\Gamma - \theta W_H}, \qquad (5\text{-}93)$$

where $W_\Gamma = \rho([D][\Gamma])$ and $W_H = \rho([D][H])$ are the maximum eigenvalues of the matrices $[D][\Gamma]$ and $[D][H]$. The matrices $[\Gamma]$ and $[H]$ are defined by the relationships

$$\dot{\underline{\varepsilon}}_c = [\Gamma]\underline{\sigma} \qquad \text{and} \qquad [H] = \left[\frac{\partial \dot{\underline{\varepsilon}}_c}{\partial \underline{\sigma}}\right]. \qquad (5\text{-}94)$$

It should be noted that equation (5-93) is only exact for fully constrained problems where the displacement increments are zero. According to the authors this represents the 'worst case' for stability. As the

criterion involves the evaluation of the eigenvalues W_Γ and W_H its use in practice is likely to prove rather expensive in computer time.

The forward gradient method provides a range of solution techniques within one basic framework, thus providing considerable flexibility. It was therefore chosen as the solution technique for the creep analysis. Chapter 6 describes how the technique was extended to handle the turbine blade problem.

Chapter 6

TURBINE BLADE ANALYSIS

6.1 Introduction

In order to formulate a suitable mathematical model for the turbine blade problem it is necessary to combine the elastic, plastic, thermal and creep analyses presented in sections 5.1 to 5.4. If the simplest Euler solution scheme is used for the creep analysis, the solution procedure is simply one of initial strain, as described in section 5.4. If tangent stiffness is used for the plasticity solution the two non-linearities are essentially decoupled, as shown below.

As

$$\Delta\underline{\varepsilon} = \Delta\underline{\varepsilon}_e + \Delta\underline{\varepsilon}_p + \Delta\underline{\varepsilon}_c + \Delta\underline{\varepsilon}_T \qquad (6\text{-}1)$$

and

$$\Delta\underline{\sigma} = [\mathrm{Dep}]\left\{\Delta\underline{\varepsilon}_e + \Delta\underline{\varepsilon}_p\right\} \qquad (6\text{-}2)$$

then using the equilibrium equation (5-25), and the strain displacement relation (5-16)

$$\int [B]^T[\mathrm{Dep}][B]\Delta\underline{u} = \Delta\underline{f} + \int [B]^T[\mathrm{Dep}]\Delta\underline{\varepsilon}_c + \int [B]^T[\mathrm{Dep}]\Delta\underline{\varepsilon}_T \cdot \qquad (6\text{-}3)$$

Thus as $\Delta\underline{\varepsilon}_c$ is calculated from the current conditions, $\Delta\underline{u}$ and hence the stress change over the timestep can be determined directly.

As will be seen in section 6.2 if the forward gradient method is used coupling occurs between the creep and plasticity nonlinearities and the tangent stiffness matrix must be modified to reflect this. Appropriate solutions techniques for the resulting equations are discussed in section 7.2.

6.2 Derivation of the elastic-plastic thermal creep model

In this section the governing equations for the elastic-plastic thermal creep model are derived. The forward gradient method is used to calculate the creep strain increment, and allowance is made for hardening due to strain and temperature. The elastic and plastic properties of the material are assumed to be temperature dependent. No matrix inversions are needed to derive the model.

It is convenient initially to set down the governing equations for the different material behaviours as follows:

(a) Temperature dependent stress/strain law

$$d\underline{\sigma} = \left[D_2\right] d\underline{\varepsilon}_e + [dD]\underline{\varepsilon}_e \quad . \tag{6-4}$$

(b) Plastic yield criterion

$$F(\underline{\sigma}, \kappa, T) = 0 \quad . \tag{6-5}$$

(c) Definition of plastic hardening criterion

$$d\kappa = \underline{\sigma}^T d\underline{\varepsilon}_p \quad . \tag{6-6}$$

as work hardening is assumed.

(d) Plastic flow law

$$d\underline{\varepsilon}_p = d\lambda \frac{\partial F}{\partial \underline{\sigma}} \quad . \tag{6-7}$$

(e) Strain hardening creep law

$$\dot{\bar{\varepsilon}}_c = \dot{\bar{\varepsilon}}_c\left(\bar{\sigma}, \bar{\varepsilon}_c, T\right) \quad . \tag{6-8}$$

(f) Creep flow law

$$d\underline{\varepsilon}_c = \frac{\partial \bar{\sigma}}{\partial \underline{\sigma}} d\bar{\varepsilon}_c \quad . \tag{6-9}$$

(g) Subdivision of strains

$$d\underline{\varepsilon} = d\underline{\varepsilon}_e + d\underline{\varepsilon}_p + d\underline{\varepsilon}_c + d\underline{\varepsilon}_T \quad . \tag{6-10}$$

(h) Equilibrium equation

$$\int [B]^T d\underline{\sigma} = d\underline{f} \quad . \tag{6-11}$$

In addition, it is assumed that the stress function used in the creep and plasticity models is identical, *ie* $\bar{\sigma} \equiv F$.

Using the forward gradient method the effective creep strain increment is defined as

$$d\bar{\varepsilon}_c = \left[(1 - \theta)\dot{\bar{\varepsilon}}_{c,t} + \theta\dot{\bar{\varepsilon}}_{c,t+dt}\right] dt \quad . \tag{6-12}$$

Using a Taylor expansion about the current conditions we assume that

$$\dot{\bar{\varepsilon}}_{c,t+dt} = \dot{\bar{\varepsilon}}_{c,t} + \left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}\right) d\bar{\sigma} + \left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}\right) d\bar{\varepsilon}_c + \left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{T}}\right) dT \quad (6\text{-}13)$$

Substituting for $\dot{\bar{\varepsilon}}_{c,t+dt}$ in equation (6-12),

$$d\bar{\varepsilon}_c = \left[\dot{\bar{\varepsilon}}_{c,t} + \theta\left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}\right) d\bar{\sigma} + \theta\left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}\right) d\bar{\varepsilon}_c + \theta\left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{T}}\right) dT\right] dt \quad . \quad (6\text{-}14)$$

$$\text{Therefore} \left[1 - \theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} dt\right] d\bar{\varepsilon}_c = \left[\dot{\bar{\varepsilon}}_c + \theta\left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}\right) d\bar{\sigma} + \theta\left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{T}}\right) dT\right] dt \quad (6\text{-}15)$$

and thus

$$d\bar{\varepsilon}_c = \frac{1}{C_1}\left[\dot{\bar{\varepsilon}}_c + \theta\left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}\right) d\bar{\sigma} + \theta\left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{T}}\right) dT\right] dt \quad (6\text{-}16)$$

where $C_1 = \left[1 - \theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} dt\right]$.

Differentiating the plastic yield criterion (equation (6-5))

$$dF = \left(\frac{\partial F}{\partial \underline{\sigma}}\right)^T d\underline{\sigma} + \left(\frac{\partial F}{\partial \kappa}\right) d\kappa + \left(\frac{\partial F}{\partial \bar{T}}\right) dT = 0 \quad . \quad (6\text{-}17)$$

Defining $\frac{\partial F}{\partial \underline{\sigma}} = \underline{a}$ and $-Ad\lambda = (\partial F/\partial \kappa) d\kappa$ then

$$\underline{a}^T d\underline{\sigma} - Ad\lambda + \left(\frac{\partial F}{\partial \bar{T}}\right) dT = 0 \quad . \quad (6\text{-}18)$$

Utilizing equations (6-4) and (6-10) the stress increment can be written as

$$d\underline{\sigma} = \left[D_2\right] d\underline{\varepsilon} - \left[D_2\right] d\underline{\varepsilon}_p - \left[D_2\right] d\underline{\varepsilon}_c - \left[D_2\right] d\underline{\varepsilon}_T + [dD]\underline{\varepsilon}_e \quad (6\text{-}19)$$

and substituting for $d\underline{\varepsilon}_p$

$$d\underline{\sigma} = \left[D_2\right]d\underline{\varepsilon} - \left[D_2\right]\underline{a}d\lambda - \left[D_2\right]d\underline{\varepsilon}_c - \left[D_2\right]d\underline{\varepsilon}_T + [dD]\underline{\varepsilon}_e \cdot \quad (6\text{-}20)$$

Noting that it is assumed that

$$\bar{\sigma} \equiv F$$

and utilising equation (6-9)

$$d\bar{\sigma} = \underline{a}^T d\underline{\sigma} = \underline{a}^T\left[D_2\right]d\underline{\varepsilon} - \underline{a}^T\left[D_2\right]\underline{a}d\lambda - \underline{a}^T\left[D_2\right]\underline{a}d\bar{\varepsilon}_c - \underline{a}^T\left[D_2\right]d\underline{\varepsilon}_T + \underline{a}^T[dD]\underline{\varepsilon}_e \cdot$$

..... (6-21)

Noting that

$$d\bar{\sigma} = Ad\lambda - \left(\frac{\partial F}{\partial T}\right)dT$$

$$d\bar{\varepsilon}_c = \frac{1}{C_1}\left[\dot{\bar{\varepsilon}}_c + \theta\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\left[Ad\lambda - \left(\frac{\partial F}{\partial T}\right)dT\right] + \theta\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial T}dT\right]dt \quad (6\text{-}22)$$

from equation (6-16).

Thus substituting for $d\bar{\varepsilon}_c$ in equation (6-21) and utilizing equation (6-18)

$$Ad\lambda - \left(\frac{\partial F}{\partial T}\right)dT = \underline{a}^T\left[D_2\right]d\underline{\varepsilon} - \underline{a}^T\left[D_2\right]\underline{a}d\lambda$$

$$- \frac{\underline{a}^T\left[D_2\right]\underline{a}}{C_1}\left[\dot{\bar{\varepsilon}}_c + \theta\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\left[Ad\lambda - \left(\frac{\partial F}{\partial T}\right)dT\right] + \theta\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial T}dT\right]dt$$

$$- \underline{a}^T\left[D_2\right]d\underline{\varepsilon}_T + \underline{a}^T[dD]\underline{\varepsilon}_e \cdot \quad (6\text{-}23)$$

Therefore

$$\left[A + \underline{a}^T\left[D_2\right]\underline{a}\left(1 + \frac{\theta A dt}{C_1}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}\right)\right] d\lambda = \underline{a}^T\left[D_2\right]d\underline{\varepsilon} - \underline{a}^T\left[D_2\right]d\underline{\varepsilon}_T + \underline{a}^T[dD]\underline{\varepsilon}_e$$

$$+ \frac{\partial F}{\partial T}dT + \frac{\underline{a}^T\left[D_2\right]\underline{a}}{C_1}\left[\left(\frac{\partial F}{\partial T}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} - \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial T}\right)\theta dT - \dot{\bar{\varepsilon}}_c\right]dt \equiv K_1 d\lambda$$

where $$K_1 = \left[A + \underline{a}^T\left[D_2\right]\underline{a}\left(1 + \frac{\theta A dt}{C_1}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}\right)\right] . \qquad (6\text{-}24)$$

Now

$$d\underline{\sigma} = \left[D_2\right]d\underline{\varepsilon} - \left[D_2\right]\underline{a}d\lambda - \left[D_2\right]\underline{a}d\bar{\varepsilon}_c - \left[D_2\right]d\underline{\varepsilon}_T + [dD]\underline{\varepsilon}_e \quad . \qquad (6\text{-}25)$$

Substituting for $d\bar{\varepsilon}_c$ from equation (6-22)

$$d\underline{\sigma} = \left[D_2\right]d\underline{\varepsilon} - \left[D_2\right]d\underline{\varepsilon}_T + [dD]\underline{\varepsilon}_e$$

$$+ \frac{\left[D_2\right]\underline{a}}{C_1}\left[\left(\frac{\partial F}{\partial T}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} - \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial T}\right)\theta dT - \dot{\bar{\varepsilon}}_c\right]dt - \left[D_2\right]\underline{a}\left[1 + \frac{\theta}{C_1}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}A dt\right]d\lambda \quad .$$

..... (6-26)

Substituting for $d\lambda$ and defining $C_2 \equiv \left(1 + \frac{\theta}{C_1}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}A dt\right)$

$$d\underline{\sigma} = \left[D_2\right]d\underline{\varepsilon} - \left[D_2\right]d\underline{\varepsilon}_T + [dD]\underline{\varepsilon}_e + \frac{\left[D_2\right]\underline{a}}{C_1}\left[\left(\frac{\partial F}{\partial T}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} - \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial T}\right)\theta dT - \dot{\bar{\varepsilon}}_c\right]dt$$

$$- \underline{\left[D_2\right]\underline{a}\,\frac{C_2}{K_1}}\left\{\underline{a}^T\left[D_2\right]d\underline{\varepsilon} - \underline{a}^T\left[D_2\right]d\underline{\varepsilon}_T + \underline{a}^T[dD]\underline{\varepsilon}_e + \frac{\partial F}{\partial T}dT\right.$$

$$\left. + \frac{\underline{a}^T\left[D_2\right]\underline{a}}{C_1}\left[\left(\frac{\partial F}{\partial T}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} - \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial T}\right)\theta dT - \dot{\bar{\varepsilon}}_c\right]dt\right\}. \qquad \ldots\ldots (6\text{-}27)$$

Therefore

$$d\underline{\sigma} = \left[\left[D_2\right] - \frac{C_2}{K_1}\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]\right]d\underline{\varepsilon} + \left[[dD] - \frac{C_2}{K_1}\left[D_2\right]\underline{a}\underline{a}^T[dD]\right]\underline{\varepsilon}_e$$

$$- \left[\left[D_2\right] - \frac{C_2}{K_1}\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]\right]d\underline{\varepsilon}_T - \frac{C_2\left[D_2\right]\underline{a}}{K_1}\frac{\partial F}{\partial T}dT$$

$$- \left[\left[D_2\right] - \frac{C_2}{K_1}\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]\right]\underline{a}\left[\dot{\bar{\varepsilon}}_c - \left(\frac{\partial F}{\partial T}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} - \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial T}\right)\theta dT\right]\frac{dt}{C_1}. \qquad \ldots\ldots (6\text{-}28)$$

Thus, utilizing the equilibrium equation (6-11) and writing

$$\left[D_{epc}\right] = \left[\left[D_2\right] - \frac{C_2}{K_1}\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]\right]$$

$$\int [B]^T \left[D_{epc}\right] [B]\, d\underline{u} = d\underline{f} - \int [B]^T \left[[dD] - \frac{C_2}{K_1} \left[D_2\right] \underline{a}\underline{a}^T [dD] \right] \underline{\varepsilon}_e + \int [B]^T \left[D_{epc}\right] d\underline{\varepsilon}_T + \int \frac{[B]^T C_2 \left[D_2\right] \underline{a}}{K_1} \frac{\partial F}{\partial T}\, dT + \int [B]^T \left[D_{epc}\right] \underline{a} \left[\dot{\underline{\varepsilon}}_c - \left(\frac{\partial F}{\partial T} \frac{\partial \dot{\underline{\varepsilon}}_c}{\partial \underline{\sigma}} - \frac{\partial \dot{\underline{\varepsilon}}_c}{\partial T} \right) \theta dT \right] \frac{dt}{C_1} \quad \ldots\ldots \quad (6\text{-}29)$$

or

$$\left[K_T\right] d\underline{u} = d\underline{f} + d\underline{f}_p \quad . \qquad (6\text{-}30)$$

Equation (6-30) thus provides the desired relationship between the known conditions at the start of the timestep and the unknown displacement increment $d\underline{u}$. Once this has been determined the stress increment and plastic and creep strain increments can be determined from equations (6-22), (6-24) and (6-28). Equation (6-30) is in principle only valid for infinitesimal increments, but, as in the case of plasticity analysis it is assumed to be valid for finite increments.

Snyder and Bathe[168] propose that the plastic strain increment should be evaluated at an intermediate stress $\sigma_{t+\theta\Delta t} = \underline{\sigma}_t + \theta \Delta \underline{\sigma}_t$, rather than at the initial stress $\underline{\sigma}_t$. In the author's opinion the approach is incompatible with the forward gradient method presented here. It is easy to see that the plastic yield criterion (equation (6-5)) and the hardening criterion can be rewritten in the form

$$\underline{\varepsilon}_p = \underline{\varepsilon}_p(\underline{\sigma}, T) \quad . \qquad (6\text{-}31)$$

Hence

$$d\underline{\varepsilon}_p = \frac{\partial \underline{\varepsilon}_p}{\partial \underline{\sigma}} d\underline{\sigma} + \frac{\partial \underline{\varepsilon}_p}{\partial T} dT \quad . \qquad (6\text{-}32)$$

The forward gradient approximation for $d\underline{\varepsilon}_c$ means that

$$d\underline{\varepsilon}_c = \theta \left[\frac{\partial \dot{\underline{\varepsilon}}_c}{\partial \underline{\sigma}} d\underline{\sigma} + \frac{\partial \dot{\underline{\varepsilon}}_c}{\partial \underline{\varepsilon}_c} d\underline{\varepsilon}_c + \frac{\partial \dot{\underline{\varepsilon}}_c}{\partial T} dT \right] dt + \dot{\underline{\varepsilon}}_c\, dt \quad . \qquad (6\text{-}33)$$

In both cases, therefore the approximation is limited to first-order terms in $d\underline{\sigma}$ and dT. The evaluation of the plastic strain increment at an intermediate stress would involve the inclusion of second-order terms which have already been neglected in the approximation for the creep strains. Snyder and Bathe propose an iterative scheme to calculate the plastic and creep strain increments, in which the first guess for $\underline{\sigma}_{t+\Delta t}$ is $\underline{\sigma}_t$. The method proposed here provides a consistent method for calculating the stress and inelastic strain increments, and avoids the need for iteration if an implicit solution scheme is used.

The solution procedure is shown on the flow diagram in Fig 8. The first step is to calculate the incremental pseudo loads $\Delta\underline{f}_p$ from equation (6-29). From these and the external load increment $\Delta\underline{f}$ the residual $\underline{\psi}_0$ is formed. This is the unbalanced force between the current state of stress in the structure and the external loads. The structural stiffness matrix is calculated using the current stresses, creep and plastic strains, and the displacement increment $\Delta\underline{u}_i$ evaluated by solving the structural equilibrium equations. Because equations (6-22), (6-24) and (6-28) relate the stress, plastic and creep strain increments to the total strain increment over the load step it is necessary to evaluate the total displacement increment $\Delta\underline{u}_{t,i+1}$. This is calculated by adding the current displacement increment $\Delta\underline{u}_i$ to $\Delta\underline{u}_{t,i}$ the estimate of the total displacement increment obtained on the previous iteration. The stress, plastic and creep strain increments are then evaluated, and a new estimate of the residual $\underline{\psi}_{i+1}$ is calculated. The pseudo loads do not appear in this calculation as they are implicitly accommodated in the calculation of the stress increment. If there was no nonlinearity in the structural response the residual would be identically zero (apart from roundoff errors). However as the behaviour of the structure has been linearised about the current conditions, the residual will in general be non zero. Thus an iterative solution procedure is used to reduce the residual to zero. One possibility is to reform the structural stiffness matrix on each iteration based on the current stresses and to calculate a further displacement increment using the residual as a force vector. It should be noted in this case that even for an implicit creep analysis the elastic or elasto-plastic stiffness matrix is used after the first iteration, because an estimate of the creep strain increment over the load step has already been calculated in determining the residual. The procedure then corresponds to an initial strain method for the creep analysis. Convergence is deemed to have been achieved, and the iterative procedure is terminated when the root mean square of the residual

is less than a given fraction (usually 1%) of the root mean square of the applied force vector. The remaining residual force is then carried forward to the next load step. This so called 'tangent stiffness' approach offers very rapid convergence, but for large problems such as the blade analysis the resolution of the structural stiffness equations on each iteration is very time consuming. Improved methods are discussed in section 7.2.

6.3 The subincrementation solution procedure

Up to this point it has been implicitly assumed that the magnitude of the steps used in solving the global equilibrium equations and calculating the stress and inelastic strain increments at the integration point level are identical. However this is not the optimum approach. Huffington[169] found that for elasto-plastic analysis considerable improvements in accuracy could be achieved by calculating the stress and plastic strain increments at each integration point in a number of steps. This procedure was also advocated by Nayak and Zienkiewicz[170]. For a large problem the solution of the global equilibrium equations will represent the most time consuming part of the calculation. It is thus sensible to gain as much advantage as possible from each equilibrium iteration by calculating the stresses and inelastic strains as accurately as possible. Bushnell[171] proposed an alternative scheme whereby a converged solution for the displacement increment was obtained from the global equilibrium equations before each evaluation of the inelastic strain increments. In the opinion of the author this represents the reverse of what is required. To obtain convergence of the global equilibrium equations at each estimate of inelastic strain would be extremely expensive, and the solution will in any case be inaccurate because of the incorrect inelastic strains.

In the solution scheme proposed the stress and inelastic strain increments are calculated from the total displacement increment over the load step $\Delta \underline{u}_{t,i+1}$. This is in contrast to a previous version of the code developed for elasto-plastic problems[172] where the displacement increment $\Delta \underline{u}_i$ from the current iteration is used. One of the reasons given by Bushnell for the development of his algorithm is that this procedure in its standard form can fail. This occurs primarily on elasto-plastic-thermal problems where the inclusion of the pseudo-load terms on the first iteration can cause an over-estimate of the plastic strains. On subsequent iterations, particularly in statically indeterminate problems integration points may then spuriously unload, leading to an inaccurate solution or divergence of the numerical procedure. In order to overcome

this difficulty Holt and Parsons[173] proposed a modification whereby if an integration point yields on the first iteration spurious unloading is prevented, and the point returns along the stress/strain curve, leading in a mathematical sense to recovery of plastic strain. The difficulty is also circumvented by the use of $\Delta \underline{u}_{t,i+1}$ in the strain to stress calculation as adopted in the GANDALF code although this is a less efficient solution, because the stress calculation has to be performed for the entire increment on each iteration. However, with the use of an implicit creep algorithm, it is difficult to see any alternative method.

The subincrementation procedure used in the code is similar to that suggested by Snyder and Bathe[168]. The total displacement increment $\Delta \underline{u}_{t,i+1}$ is divided into a number of subincrements and the stress and inelastic strain increments are calculated for each subincrement. It should be noted that there is no reason why the θ parameter controlling the timestepping scheme should be the same in the equilibrium solution loop as in the subincrementation loop. Snyder and Bathe assumed that the total strain increment occurred linearly over the load/time step. However, Ottosen and Gunneskov[174] showed that whilst this is a good assumption for elasto-plastic and secondary creep problems greater accuracy can be obtained for creep hardening problems by assuming a non-linear variation of total strain increment over the load step. A similar scheme has been developed for the GANDALF code, and details are presented in section 8.5.

Chapter 7

THE SOLUTION OF FINITE ELEMENT EQUATIONS

7.1 Linear equation systems

Finite element stress analysis involves the solution of sets of linear equations of the form (*cf* equation (5-21)):

$$[H]\underline{u} = \underline{f} \ , \tag{7-1}$$

where $[H]$ is an $N \times N$ matrix, $\underline{u}$ is the vector of N nodal displacements and $\underline{f}$ is the applied nodal force vector. In an elastic analysis $[H]$ is the structural stiffness matrix $[K]$. For a typical GANDALF analysis N would be approximately 15000. As in conventional static analysis, in order to prevent rigid body motion, a number of the structural degrees of freedom must be constrained. For clarity it will be assumed that the displacement and force vectors can be partitioned such that

$$\underline{u} = \begin{bmatrix} \underline{u}_u \\ \\ \underline{u}_c \end{bmatrix} , \qquad \underline{f} = \begin{bmatrix} \underline{f}_u \\ \\ \underline{f}_c \end{bmatrix} , \tag{7-2}$$

where subscript u denotes unconstrained degrees of freedom and subscript c denotes constrained degrees of freedom. The force vector in equation (7-2) is incomplete since to preserve static equilibrium there will be reactions at the constrained degrees of freedom. Thus equation (7-1) can be written

$$[H]\begin{bmatrix} \underline{u}_u \\ \\ \underline{u}_c \end{bmatrix} = \begin{bmatrix} \underline{f}_u \\ \\ \underline{f}_c \end{bmatrix} + \begin{bmatrix} \underline{0} \\ \\ \underline{R} \end{bmatrix} , \tag{7-3}$$

where $\underline{R}$ is the vector of reactions. The unknowns in this system of equations are the displacements of the unconstrained degrees of freedom and the reactions.

The stiffness matrix is symmetric and there is thus no need to store terms below the leading diagonal. As described in section 5.1 the stiffness matrix is obtained by summing the individual element contributions. Thus a term h_{ij} will only arise in the global stiffness matrix if the two degrees-of-freedom i, j both appear in the same element.

This means that by judicious numbering of the nodes non-zero terms can be confined to an area close to the leading diagonal. Such a matrix is described as banded. The maximum distance along a row between the leading diagonal and the last non-zero term is termed the bandwidth of the matrix and will be denoted by B. For matrices arising from GANDALF analyses B would be approximately 900. Since B is considerably less than N, the number of degrees of freedom in the structure, most of the upper triangle of the matrix will be zero, and need not be stored.

Gaussian elimination requires the minimum number of operations to solve equation (7-1) by a direct method[175]. The stiffness matrix is positive definite and thus the equations can be reduced in any order without the necessity of searching for the largest pivot at each stage[176]. At any stage in the elimination process, terms below the leading diagonal are either equal to the corresponding terms above the diagonal or zero. Thus no additional storage is required during the solution procedure. This makes the algorithm computationally very attractive since it is both reliable and efficient. The Gaussian elimination process falls into three distinct phases. The first two stages are to reduce the stiffness matrix, *ie* make every term below the leading diagonal zero and at the same time perform the same operations on the force vector. When this has been completed the final equation will contain only one term and the unknown associated with it. The unknowns are then determined by backsubstitution.

It can be shown that the number of operations required for forward reduction is of order NB^2 and for backsubstitution and reduction of the force vector of order NB. Thus the time taken to perform backsubstitution is much less than that required for forward reduction. The importance of this observation lies in the fact that only reduction of the force vector and backsubstitution are necessary when resolving the equations for different applied loads. This allows the equations to be solved for multiple force vectors relatively cheaply and is the basis of most nonlinear solution techniques.

There are two commonly used methods of implementing Gaussian elimination in a finite element program, a band solution and the more modern front solution. In the band solution the first step is to assemble the global stiffness matrix by adding together all the element contributions. The equations are then solved using the conventional Gaussian elimination technique, with each equation in turn being used as the pivotal row. The bandwidth of the matrix and hence the solution

time is determined by the node numbering. The band solution suffers from a number of disadvantages. The assembly of the structural stiffness matrix is an inefficient process as it involves adding terms in an unstructured manner to a large array. This is particularly noticeable if as in the present case not all the stiffness matrix can be in core simultaneously. As a result of the large number of nodes involved it can be difficult to achieve acceptable node numbering. An automatic scheme is essential and much work has been done in this field[62,63,177].

The above disadvantages can be largely overcome by use of the so called frontal solution and this technique has been adopted. The frontal solution is described in detail by Irons[60] and a code is given by Irons and Ahmed[61] and therefore only an outline will be presented here. The essential difference between the band solution and the frontal approach is that in the frontal solution the stiffness matrix [H] is never explicitly assembled or stored. Instead the assembly process is interlaced with the reduction and only terms relating to a subset of the degrees of freedom (the so-called front) are active at any stage. The front is equivalent to the triangle of terms lying below the pivotal row in the band solution. It can be shown that a degree of freedom can be eliminated from the front providing it does not occur in any subsequent element. This can easily be determined by examination of the structural topology before the solution is started. It follows that the order in which the elements are added to the front determines the number of degrees of freedom in the front at any stage, and hence the solution time. The node numbers are entirely irrelevant. The methods used to achieve an optimal element numbering, an inherently easier problem than optimising node numbering were discussed in section 3.2.

The solution proceeds as follows. The stiffness matrix for the current element is first calculated and its degrees of freedom are considered in turn. If the degree of freedom already exists in the front the corresponding terms in the stiffness matrix are simply added to the existing terms in the front. However if the degree of freedom is not present in the front a new location must be allocated for it before the stiffness matrix terms are added into the front. If the new location is at either end of the front the distance between the first and last term (the so called frontwidth) will increase. The element degrees of freedom are then re-examined to determine which if any of them can be eliminated. As the elimination proceeds the frontwidth is decreased if the degrees of freedom at either end of the front are reduced. This is

clearly a desirable feature as the number of operations required to reduce a given degree of freedom is proportional to the square of the current frontwidth. The lifetime of a degree of freedom is defined as the number of element additions for which it remains in the front. The addition of degrees of freedom with a short lifetime to the edges of the front rather than the centre will tend to promote the desired decreases in the frontwidth. This technique is utilised in the GANDALF code. A similar technique has been described by Yeo[178].

The frontal solution procedure described above has been implemented on a VAX 11/780 and FPS-164 attached processor computer system. Details on how this has been achieved are given in Appendix D.

7.2 Nonlinear equation systems

7.2.1 The tangent stiffness method

For the elastic-plastic-creep analysis a nonlinear equation of the type (*cf* equation (6-30)),

$$\left[K(\underline{u})\right]\underline{u} = \underline{f} \tag{7-4}$$

has to be solved for the unknown $\underline{u}$ for each increment of load $\underline{f}$.

Consider the equation,

$$\underline{\Psi}(\underline{u}) = \underline{f} - \left[K(\underline{u})\right]\underline{u} \ , \tag{7-5}$$

where $\underline{\Psi}$ is the residual force vector introduced in section 6.2.

The results of successive iterations will be denoted by $\underline{u}_0, \underline{u}_1, \cdots, \underline{u}_i$ and the solution by $\underline{u}_{i+1}$.

Thus

$$\underline{\Psi}\left(\underline{u}_{i+1}\right) = \underline{0} \ , \tag{7-6}$$

where

$$\underline{u}_{i+1} = \underline{u}_i + \Delta\underline{u}_i \ . \tag{7-7}$$

Therefore

$$\underline{\Psi}\left(\underline{u}_{i+1}\right) = \underline{\Psi}\left(\underline{u}_i + \Delta\underline{u}_i\right) \ . \tag{7-8}$$

Using Taylor's expansion

$$\underline{\Psi}\left(\underline{u}_i + \Delta\underline{u}_i\right) = \underline{\Psi}\left(\underline{u}_i\right) + \left[\frac{\partial\underline{\Psi}\left(\underline{u}_i\right)}{\partial\underline{u}}\right]\Delta\underline{u}_i + \left[\frac{\partial^2\underline{\Psi}\left(\underline{u}_i\right)}{\partial\underline{u}^2}\right]\frac{\Delta\underline{u}_i^2}{2} + \cdots \ . \tag{7-9}$$

Ignoring second-order terms in this expression gives:

$$\underline{\Psi}\left(\underline{u}_{i+1}\right) = \underline{\Psi}\left(\underline{u}_i\right) + \left[\frac{\partial \underline{\Psi}\left(\underline{u}_i\right)}{\partial \underline{u}}\right] \Delta \underline{u}_i = \underline{0} \quad . \tag{7-10}$$

Thus

$$\left[\underline{\Psi}'\left(\underline{u}_i\right)\right] \Delta \underline{u}_i = - \underline{\Psi}\left(\underline{u}_i\right) , \qquad \left[\underline{\Psi}'\right] = \left[\frac{\partial \underline{\Psi}}{\partial \underline{u}}\right] \quad . \tag{7-11}$$

But, from equation (7-5)

$$\underline{\Psi}\left(\underline{u}_i\right) = \underline{f} - \left[K\left(\underline{u}_i\right)\right] \underline{u}_i \quad .$$

Thus differentiating,

$$\left[\underline{\Psi}'\left(\underline{u}_i\right)\right] = - \left[K\left(\underline{u}_i\right)\right] - \left[\frac{\partial K\left(\underline{u}_i\right)}{\partial \underline{u}} \underline{u}_i\right] \tag{7-12}$$

and substituting (7-12) in (7-11) gives

$$\left\{\left[K_i\right]_T + \left[K_i\right]_T'\right\} \Delta \underline{u}_i = \underline{\Psi}\left(\underline{u}_i\right) , \tag{7-13}$$

where $\left[K_i\right]_T = \left[K\left(\underline{u}_i\right)\right]$ and $\left[K_i\right]_T' = \left[\frac{\partial K\left(\underline{u}_i\right)}{\partial \underline{u}}\right] \underline{u}_i$.

This technique is known as the Newton-Raphson method. In general equation (7-13) is not used as the term $[K_i]_T'$ is nonlinear and difficult to determine. Moreover it would introduce non-symmetric terms into the otherwise symmetric stiffness matrix $[K_i]_T$. For these reasons this term is normally discarded so that

$$\left[K_i\right]_T \Delta \underline{u}_i = \underline{\Psi}\left(\underline{u}_i\right) \quad . \tag{7-14}$$

This simplification of the Newton-Raphson procedure is termed the tangent stiffness method. The matrix $[K_i]_T$ can be interpreted physically as a measure of the current stiffness of the structure under analysis. It is easy to see, by equations (6-30) and (7-14), that $[K_i]_T$ is the tangent stiffness matrix given by

$$\left[K_i\right]_T = \int [B]^T \left[D_{epc}\right] [B] \tag{7-15}$$

with $\left[D_{epc}\right]$ evaluated at the current conditions. Equation (7-14) is a linear equation and is solved using the technique described in section section 7.1. Once the displacement increment $\Delta\underline{u}_i$ has been calculated, the total displacement increment $\Delta\underline{u}_{t,i+1}$ is evaluated, and the new stresses $\underline{\sigma}\left(\underline{u}_{i+1}\right)$ and inelastic strains are determined. The new value of the residual is then calculated as

$$\underline{\psi}\left(\underline{u}_{i+1}\right) = \underline{f} + \underline{R} - \int [B]^T \underline{\sigma}\left(\underline{u}_{i+1}\right) \quad , \tag{7-16}$$

where $\underline{R}$ are the reactions at the constrained nodes. If the root mean square of the residual divided by the root mean square of the applied loads has reduced to below a specified tolerance the iterative procedure is terminated. If not a further displacement increment is obtained from equation (7-14) using the new value of the residual and the updated tangent stiffness matrix and the procedure is repeated. Fig 9 depicts how the solution procedure would progress using the tangent stiffness method.

In the tangent stiffness algorithm the stiffness matrix has to be re-evaluated for each iteration. For a large problem with many degrees of freedom this method may require an unacceptable amount of computer time. For this reason other numerical solution techniques have been considered and six are included in the GANDALF code. These algorithms have been called 'modified Newton-Raphson' methods because they are all derived, in some sense, from the Newton-Raphson (or more accurately the tangent stiffness) method. The algorithms are described in the next section.

7.2.2 Modified Newton-Raphson methods

For a large problem the formulation and reduction of the stiffness matrix is the major computing task. In the tangent stiffness method this is repeated each iteration. A number of techniques, called modified Newton-Raphson methods, have been developed in which the stiffness matrix is not formed and reduced on each iteration, thus significantly reducing the computing time per iteration. These methods do not invalidate the analysis as to calculate the residual using equation (7-16) the stiffness matrix is not directly required. However, the number of iterations required per load increment will depend on how well the stiffness matrix is following the changing stiffness of the structure. In those load increments in which the stiffness changes markedly a

modified Newton-Raphson method will converge more slowly (if at all) than the tangent stiffness method. One strategy that has been found to be advantageous in some applications is to apply a combination of the tangent stiffness and one or more of the modified Newton-Raphson methods. For this strategy the solution algorithm would vary from load increment to load increment depending on the anticipated changes in the stiffness of the structure.

The stiffness of the structure changes with the extent and severity of the yielding in the structure under the application of a load. Thus a small amount of localised yielding over a load increment will cause minor changes to the stiffness of the structure, whereas large and extensive yielding will cause a significant change in the stiffness of the structure. The stiffness of the structure will also change significantly with temperature.

The choice of which method or indeed which combination of methods to use for a particular problem is a matter of judgement. Guiding remarks are given and the solution procedure for each of the six modified Newton-Raphson methods is compared to that of the tangent stiffness method presented in the previous section.

(i) The numerical solution procedure for the 'initial stiffness' method

For the 'initial stiffness' method the numerical solution procedure is similar to that of the tangent stiffness method except that the stiffness matrix evaluated at the beginning of the first load case is used throughout. Thus the stiffness matrix is only evaluated once, and the average computer time taken per itération, even for large problems, is relatively small. However, the number of iterations required to achieve convergence in each load increment will, in general, be large. This method corresponds to the 'initial stress' method first introduced for elasto-plastic problems by Zienkiewicz, Valliappan and King[122]. The method is illustrated in Fig 10.

(ii) The numerical solution procedure using the 'Hybrid Mk 0' method

This method is identical to the initial stiffness approach except that the stiffness is reformulated at the start of each loadcase. This solution technique is thus able to take account of the change in stiffness due to temperature changes, etc during the previous load case.

(iii) The numerical solution procedure for the 'Hybrid Mk 0 elastic' method

This procedure is identical to the 'Hybrid Mk 0' approach in that the stiffness matrix is reformulated at the start of each loadcase. However, unlike the 'Hybrid Mk 0' approach in which the current tangent stiffness matrix is calculated, this method calculates the elastic stiffness matrix at the current conditions. It was felt that this approach might be advantageous in situations where the majority of the structure was unloading elastically due for example to creep.

(iv) The numerical solution procedure for the 'Hybrid Mk 1' method

This procedure, together with the 'Hybrid Mk 2' method was first introduced by Nayak and Zienkiewicz[170]. For this method, the solution procedure is similar to that of the tangent stiffness method except that a new stiffness matrix is calculated only on the first iteration of each load increment. This method is suitable for problems where the stiffness of the structure changes significantly over the loadcase. However, if significant stiffness changes occur within a given load increment, the stiffness matrix will not be an adequate approximation to the stiffness of the structure, and convergence may not be achieved. This will limit the size of the load steps which can be utilised. The method is illustrated in Fig 11.

(v) The numerical solution procedure for the 'Hybrid Mk 2' method

For this method the solution procedure is similar to that of the tangent stiffness method, except that a new stiffness matrix is calculated only on the second iteration of each subsequent load increment. This strategy is similar to that used in the 'Hybrid Mk 1' method except that it can respond to a greater variation of the stiffness of the structure during a load increment, and therefore should, in general, allow larger load steps than the 'Hybrid Mk 1' algorithm. The method is illustrated in Fig 12.

(vi) The numerical solution procedure for the 'Hybrid Mk 3' method

For this method a new stiffness matrix is calculated on the first and second iterations of each load increment. It has been introduced because when solving an elasto-creep problem using the tangent stiffness method the elasto-creep constitutive matrix is

used to formulate the stiffness matrix on the first iteration, and the elastic constitutive matrix thereafter. The Hybrid Mk 3 method will thus produce identical results to the tangent stiffness method for problems of this type, but will avoid the expense of recalculating the stiffness matrix from the third iteration onwards. The method is illustrated in Fig 13.

7.2.3 Other nonlinear solution schemes

The algorithms introduced in sections 7.2.1 and 7.2.2 suffer from a number of disadvantages. As discussed above, the recalculation and reduction of the tangent stiffness matrix on each iteration can be prohibitively expensive for large problems. If the reformulation of the stiffness matrix is suppressed, as in the modified Newton strategies, the convergence of the iterative procedure can be badly affected. Recognition of these difficulties has led to the development of so called quasi- or secant Newton methods in recent years, which seek to update the stiffness matrix [K] or its inverse in a simple way after each iteration, rather than recompute it entirely (as in the tangent stiffness method) or leave it unchanged (as in the modified Newton-Raphson schemes). The first reference to this type of approach is given by Davidon[179] who introduced it for optimization problems. Broydon[180] applied these methods to systems of nonlinear equations, and the application of them to nonlinear finite element analysis was first given by Matthies and Strang[181]. A number of these methods have been developed and these are discussed below.

The basic assumption of quasi-Newton algorithms is that the new stiffness matrix $[K_i]$ is chosen so that it satisfies the so called 'secant condition', *ie*

$$\left[K_i\right]\left(\underline{u}_i - \underline{u}_{i-1}\right) = \underline{\psi}\left(\underline{u}_{i-1}\right) - \underline{\psi}\left(\underline{u}_i\right)$$

$$\equiv \underline{\psi}_{i-1} - \underline{\psi}_i \quad . \tag{7-17}$$

This is shown diagrammatically in Fig 14. For a one degree-of-freedom problem equation (7-17) uniquely determines $[K_i]$. However for a system of finite element equations with many degrees-of-freedom in principle any matrix $[K_i]$ satisfying equation (7-17) would be suitable. A survey of possible methods is given by Dennis and Moré[182]. In practice however, as explained by Matthies and Strang[181] it is desirable that the updated matrix $[K_i]$ should be symmetric and positive definite, and

that the new displacement increment $\Delta\underline{u}_i = [K_i]^{-1}\underline{\psi}_i$ should be cheap to compute. These requirements are satisfied by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update[183-186] which can be written

$$\left[K_i\right]^{-1} = \left[[I] + \underline{w}_i\underline{v}_i^T\right]\left[K_{i-1}\right]^{-1}\left[[I] + \underline{v}_i\underline{w}_i^T\right] , \qquad (7\text{-}18)$$

where $$\underline{v}_i = -\left[\frac{\Delta\underline{u}_{i-1}^T\left(\underline{\psi}_{i-1} - \underline{\psi}_i\right)}{\Delta\underline{u}_{i-1}^T\left[K_{i-1}\right]\Delta\underline{u}_{i-1}}\right]^{\frac{1}{2}}\left[K_{i-1}\right]\Delta\underline{u}_{i-1} + \underline{\psi}_i - \underline{\psi}_{i-1}$$

and $$\underline{w}_i = \frac{\Delta\underline{u}_{i-1}}{\Delta\underline{u}_{i-1}^T\left(\underline{\psi}_{i-1} - \underline{\psi}_i\right)} .$$

In practice since

$$\left[K_{i-1}\right]\Delta\underline{u}_{i-1} = \underline{\psi}_{i-1} \qquad (7\text{-}19)$$

the matrix multiplications in the definition of $\underline{v}_i$ can be simplified so that

$$\underline{v}_i = \underline{\psi}_i - \left[1 + \left[\frac{\Delta\underline{u}_{i-1}^T\left(\underline{\psi}_{i-1} - \underline{\psi}_i\right)}{\Delta\underline{u}_{i-1}^T\underline{\psi}_{i-1}}\right]^{\frac{1}{2}}\right]\underline{\psi}_{i-1} . \qquad (7\text{-}20)$$

It can be shown that these equations do in fact satisfy equation (7-17). In order to apply the method on successive iterations it is more convenient to expand equation (7-18) starting from the last stiffness matrix [K*] which was calculated using the modified Newton-Raphson scheme. Thus

$$\left[K_i\right]^{-1} = \left[\prod_{j=i}^{n+1}\left\{[I] + \underline{w}_j\underline{v}_j^T\right\}\right][K^*]^{-1}\left[\prod_{j=n+1}^{i}\left\{[I] + \underline{v}_j\underline{w}_j^T\right\}\right] \qquad (7\text{-}21)$$

where the stiffness matrix was last recalculated on iteration n and $\underline{v}_j$ and $\underline{w}_j$ are calculated using equations (7-18) and (7-20) with $\Delta\underline{u}_{j-1}$ replaced by $\Delta\underline{u}^*_{j-1}$ the value calculated using the BFGS procedure. The numerical procedure thus involves calculating a modified residual $\tilde{\underline{\psi}}_i$ as

$$\tilde{\underline{\psi}}_i = \left[\prod_{j=n+1}^{i} \left\{ [I] + \underline{v}_j \underline{w}_j^T \right\} \right] \underline{\psi}_i \quad . \tag{7-22}$$

$\Delta\tilde{\underline{u}}_i$ is then calculated by backsubstitution from

$$\Delta\tilde{\underline{u}}_i = [K^*]^{-1} \tilde{\underline{\psi}}_i \quad , \tag{7-23}$$

and finally the required displacement increment $\Delta\underline{u}_i^*$ is obtained from

$$\Delta\underline{u}_i^* = \left[\prod_{j=i}^{n+1} \left\{ [I] + \underline{w}_j \underline{v}_j^T \right\} \right] \Delta\tilde{\underline{u}}_i \quad . \tag{7-24}$$

As given above, the algorithm requires the calculation and storage of a new pair of vectors $\underline{v}$ and $\underline{w}$ on each iteration. In general it is desirable to set a limit on the number of vectors used to avoid degradation of algorithm efficiency. A limit of 15 is suggested by Matthies and Strang[181]. However, for a large problem such as the blade analysis the computation required to calculate the vectors is much less than that required for a backsubstitution. Thus in the GANDALF code no limit has been set on the number of vectors.

Matthies and Strang[181] recommend that the update should not be applied if the condition number (ratio of maximum to minimum eigenvalues) of the updating matrix $\left[[I] + \underline{v}_i \underline{w}_i^T \right]$ exceeds a given value, typically 10^5. The condition number c can easily be calculated as

$$c = \left[\frac{\Delta\underline{u}_{i-1}^T \underline{\psi}_{i-1}}{\Delta\underline{u}_{i-1}^T \left(\underline{\psi}_{i-1} - \underline{\psi}_i \right)} \right]^{\frac{1}{2}} \quad . \tag{7-25}$$

If c exceeds the tolerance $\underline{v}_i$ and $\underline{w}_i$ are set equal to zero and the update is not applied for the current iteration.

The need to store previous values of the vectors $\underline{v}$ and $\underline{w}$ can be seen as a disadvantage to the BFGS technique. Crisfield[187-189] developed a number of secant methods which do not suffer from this disadvantage. The simplest of these, given originally in Ref 187 has been incorporated into the code, as Crisfield reviewing his earlier work in a later paper[190] concluded that there was little further advantage to be gained from his more complex methods.

The numerical solution procedure is as follows. For the hybrid methods the equation which is solved to calculate the displacement increment is (*cf* equation (7-14))

$$[K^*]\Delta\underline{u}_i = \underline{\psi}_i \quad . \tag{7-26}$$

With Crisfield's acceleration the displacement increment $\Delta\underline{u}_i$ becomes $\Delta\underline{u}^*_i$ where

$$\Delta\underline{u}^*_i = a_i\Delta\underline{u}_i + b_i\Delta\underline{u}^*_{i-1} \tag{7-27}$$

with $a_i = -\alpha_i/\beta_i$

$$b_i = a_i\left(1 - \gamma_i/\beta_i\right) - 1$$

and $\alpha_i = \Delta\underline{u}^{*T}_{i-1}\underline{\psi}_{i-1}$,

$$\beta_i = \Delta\underline{u}^{*T}_{i-1}\left(\underline{\psi}_i - \underline{\psi}_{i-1}\right) ,$$

$$\gamma_i = \Delta\underline{u}^{T}_{i}\left(\underline{\psi}_i - \underline{\psi}_{i-1}\right) .$$

The new displacement vector then becomes

$$\underline{u}_{i+1} = \underline{u}_i + \Delta\underline{u}^*_i \quad . \tag{7-28}$$

The method assumes that the stiffness matrix is constant throughout the load increment. In the Hybrid Mk 2 and Hybrid Mk 3 methods the stiffness matrix is reformed on the second iteration of each load increment. Thus, in these cases, the acceleration is only applied on the third iteration onwards in each load increment. This means that $\Delta\underline{u}^*_i = \Delta\underline{u}_i$, for $i = 0,1$.

To apply Crisfield's acceleration method, two additional vectors $\Delta\underline{u}^*_{i-1}$ and $\underline{\psi}_{i-1}$, have to be stored, however, the cost of the extra computation involved is negligible. Unlike the BFGS method only information from the previous iteration is required. The technique would be expected to be of most benefit in cases where a large number of iterations would ordinarily be expected within the load increment. This will happen when significant nonlinearity occurs.

In certain circumstances the displacement increment, $\Delta\underline{u}^*_i$, obtained when using Crisfield's acceleration, leads to a less

accurate solution than the displacement increment $\Delta\underline{u}_i$. Numerical experiments have shown that this may occur if the coefficient a_i of equation (7-27) is much different from unity, or if the relative contribution from $\Delta\underline{u}^*_{i-1}$, in equation (7-27) becomes too large. Thus the accelerated solution is only used if

$$R_1 > a_i > \frac{1}{R_1} \tag{7-29}$$

and

$$R_2 > \frac{b_i}{a_i} > -\tfrac{1}{2}R_2 \ . \tag{7-30}$$

If either a_i or b_i/a_i is outside the prescribed range the GANDALF code automatically defaults to the displacement increment $\Delta\underline{u}_i$. Crisfield[189] suggested that R_1 should be between 2 and 3 and R_2 between 0.3 and 1.0. However the optimum choice for these parameters has not yet been determined.

An alternative method of improving the convergence of the iterative procedure is by the use of so called 'line searches'. The line search procedure involves multiplying the displacement increment $\Delta\underline{u}_i$ the so called search direction obtained from the modified Newton solution procedure by a scalar value η_i , to produce a new estimate of the displacement increment $\underline{\delta}_i$. Thus

$$\underline{\delta}_i = \eta_i \Delta\underline{u}_i \equiv \eta_i \left[[K^*]^{-1} \underline{\psi}_i \right] \ , \tag{7-31}$$

and hence

$$\underline{u}_{i+1} = \underline{u}_i + \underline{\delta}_i = \underline{u}_i + \eta_i \Delta\underline{u}_i \ . \tag{7-32}$$

The usual criterion for the selection of η_i is to choose a value such that

$$G(\eta_i) = \Delta\underline{u}_i^T \underline{\psi}(\underline{u}_i + \eta_i \Delta\underline{u}_i) \tag{7-33}$$

is minimised. If $G(\eta_i)$ can be made equal to zero for some value of η_i this is equivalent to the statement that the residual vector at the end of the iteration is acting normal to the direction of the current displacement increment, *ie* no further displacement in the current direction would be expected, and thus the current value of η_i is the optimum choice for the current iteration.

In practice it is unlikely that a zero value of $G(\eta_i)$ will be obtained and the procedure involves calculating a number of trial values of $G(\eta_i)$ to determine a minimum. $G(0)$ can easily be obtained because

$$G(0) = \Delta \underline{u}_i^T \underline{\psi}_i \quad . \tag{7-34}$$

Each successive calculation of $G(\eta_i)$ however requires the evaluation of a new residual vector, and hence the determination of a new trial set of stresses within the structure. This can prove very expensive particularly if a complex material model is being used. It is essential therefore to evaluate $G(\eta_i)$ at as few points as possible for the method to be viable. Generally the search procedure is terminated when

$$G(\eta_i) < GTOL \, . \, G(0) \quad , \tag{7-35}$$

where if a small value of GTOL is used it would be expected that the number of trials would increase and the number of iterations decrease, whereas the reverse behaviour will occur if a value of GTOL near unity is chosen. A value of 0.5 was used by Matthies and Strang[181] and Bathe and Cimento[191].

The search procedure incorporated into the GANDALF code proceeds as follows. The first value of η_i for which $G(\eta_i)$ is evaluated is unity corresponding to an unaccelerated solution. This then provides two trial solutions for η_i. Subsequent trial values are then obtained by linear interpolation between two previous values of $G(\eta_i)$. Care must be taken to ensure that the optimum pair of values are chosen, it being desirable to ensure that the values are as near zero as possible, and of different signs, thus bracketing the solution. Occasionally this procedure can break down, for example if the extrapolation between $G(0)$ and $G(1)$ has produced a value of η_i, $\eta_{i,3}$ greater than one with $G(\eta_{i,3}) > G(1)$. It is likely in this case that $G(\eta_i)$ has a minimum (not a zero) in the range 0 to $\eta_{i,3}$, and obvious that if this is the case linear interpolation is unlikely to find it. In cases of this kind therefore the code takes the next value of η_i as the minimum of a quadratic fitted through the last three points, and evaluates the corresponding value of $G(\eta_i)$. To save further expense in this case the routine is terminated and the optimum value of η_i selected from those available.

In addition to accelerating the displacements using the line search procedure it has been found to be beneficial[192] to accelerate the increment of reaction loads $\Delta\underline{R}_i$. It will be recalled that the structural equilibrium equations can be written (*cf* equation (7-3))

$$[K^*]\begin{pmatrix}\Delta\underline{u}_{i,u}\\ \\ \Delta\underline{u}_{i,c}\end{pmatrix} = \underline{\psi}_i + \begin{pmatrix}\underline{0}\\ \\ \Delta\underline{R}_i\end{pmatrix} . \qquad (7\text{-}36)$$

Thus after the line search has been applied

$$\underline{\delta}_i = \begin{pmatrix}\eta_i\Delta\underline{u}_{i,u}\\ \\ \Delta\underline{u}_{i,c}\end{pmatrix} , \qquad \Delta\underline{R}_i = \begin{pmatrix}\underline{0}\\ \\ \eta_i\Delta\underline{R}_i\end{pmatrix} . \qquad (7\text{-}37)$$

It should be noted that as the increment of reactions on the first iteration balances the applied loads, the reactions are only accelerated from the second iteration onwards.

The line search procedure can also be used in conjunction with either the BFGS or Crisfield methods; indeed this approach is advocated by Matthies and Strang[181] and Crisfield[187-190] . In these cases the BFGS or Crisfield method is applied first to obtain the search direction $\Delta\underline{u}_i^*$ and then $\underline{\delta}_i$ is obtained from $\Delta\underline{u}_i^*$ using the line search procedure. For the BFGS method $\underline{v}_i$ and $\underline{w}_i$ are given by

$$\underline{v}_i = \underline{\psi}_i - \left[1 + \left[\frac{\eta_{i-1}\underline{\delta}_{i-1}^T\left(\underline{\psi}_{i-1} - \underline{\psi}_i\right)}{\underline{\delta}_{i-1}^T\underline{\psi}_{i-1}}\right]^{\frac{1}{2}}\right]\underline{\psi}_{i-1} \qquad (7\text{-}38)$$

$$\underline{w}_i = \frac{\underline{\delta}_{i-1}}{\underline{\delta}_{i-1}^T\left(\underline{\psi}_{i-1} - \underline{\psi}_i\right)}$$

and for Crisfield

$$\Delta\underline{u}_i^* = a_i\Delta\underline{u}_i + b_i\underline{\delta}_{i-1} \qquad (7\text{-}39)$$

with

$$a_i = - \frac{\Delta \underline{u}_{i-1}^{*T} \underline{\psi}_{i-1}}{\Delta \underline{u}_{i-1}^{*T} \left(\underline{\psi}_i - \underline{\psi}_{i-1} \right)}$$

$$b_i = a_i \left[1 - \frac{\Delta \underline{u}_i^T \left(\underline{\psi}_i - \underline{\psi}_{i-1} \right)}{\underline{\delta}_{i-1}^T \left(\underline{\psi}_i - \underline{\psi}_{i-1} \right)} \right] - 1 \quad .$$

In order to reduce the storage requirement it is convenient to use $\underline{\delta}_{i-1}$ instead of $\Delta \underline{u}_{i-1}^*$ in the expression for a_i . This will produce identical results, because the η_{i-1} factor will cancel, except on the second iteration if a degree of freedom is constrained with a non-zero displacement. However the additional storage to deal with this special case does not seem justified.

Chapter 8

THE STRESS ANALYSIS PROGRAM GANDALF3P

8.1 The overall structure of the code

A flow diagram of the code is shown in Fig 15. The main structure consists of three nested loops. The outermost one controls which loadcase the program is currently analysing. A loadcase is a particular condition in the flight cycle for example takeoff or cruise. In a nonlinear finite element analysis as discussed in Chapter 5 the loads are applied in a series of increments. The middle loop in the code counts the number of load increments (LINC) applied for this loadcase. The size of each increment is either specified by the user as one of the loading parameters or is determined by the program. Within each load increment it is necessary to perform an iterative solution. The number of iterations (ITER) is controlled by the innermost loop.

The first task performed by the program is to read in the problem data. This is the mesh topology and geometry, the material properties and a number of control parameters. This information is not changed from loadcase to loadcase.

Subroutine MATRON

The name of this routine is taken from Irons' and Ahmed's[61] frontal solution code. It first calculates the lifetime of each node and hence determines the order in which the degrees of freedom of each element should be added to the front. The maximum frontwidth which will be required by the problem is calculated, to ensure that sufficient storage will be available. Finally the last occurrence of each node in the structural topology is marked to indicate that the degrees of freedom associated with the node can now be reduced.

The next stage in the main program is to read in the problem data which is specific to a particular loadcase. These are the rotational speed and temperature distribution of the structure and the external loads on it. The user can select in the data input program GANDALF1N whether manual or automatic loading is to be used. If manual loading has been selected the program reads in the preselected load levels at which a solution is required.

Subroutine INCREM

The function of this subroutine is to determine what fraction of the loading to be applied in the current loadcase should be applied on

the current loadstep. Control of the size of the loadsteps is essential to ensure stability of the solution procedure, and accurate prediction of the structural response. The user has three methods of controlling the size employed. Firstly, on all loadcases the user can specify particular load levels which the code is to use. This is useful if a solution at part load is required. Secondly, for transient loadcases, the user can specify the sizes of the first and subsequent load steps. Finally, for loadcases where creep occurs the limiting timestep given by Cormeau's stability criterion (see section 5.4) is evaluated. The user can apply a specified multiple of this timestep on the current increment. In addition to these user specified controls, an automatic control scheme for the implicit creep analysis based on changes in the structural stiffness matrix has been devised. This is described in more detail in section 8.2. The subroutine calculates the maximum allowable size of increment on the basis of these criteria, which is then used in the analysis.

It should be borne in mind that due to the nonlinear nature of the analysis it is not possible to calculate a trial solution using a standard load increment and then simply scale the solution back to an acceptable level. An alternative approach might be the use of so called 'arc-length' methods[193,194] whereby the size of the load increment is adjusted from iteration to iteration. However it is difficult to see how such a method could be implemented economically for an implicit creep analysis.

Subroutine ALGOR

The function of this routine is to determine whether or not a new stiffness matrix is required on the current iteration. This is dependent on the solution algorithm selected, and on whether the code is restarting from a previously stored condition on the current iteration, in which case a new stiffness matrix is always required.

Subroutine BFGS

The routine performs the BFGS update if required. It is called twice per iteration, once before the main equation solving routine FRONT in order to modify the residual, and then after the call to FRONT to modify the displacement increment.

Subroutine FRONT

This is the main solution routine for the structural stiffness equations. It is called on every pass through the iteration loop. The code can perform three different types of solution. A stiffness matrix

and a force vector can be created and the equations solved for the nodal displacements. Alternatively, as occurs when the tangent stiffness algorithm is employed, a new stiffness matrix, reflecting the current state of the structure, can be calculated, and the additional displacements due to the existing residual force vector from the previous iteration can be evaluated. The final option is to use the previously reduced stiffness matrix and the residual force vector to evaluate the incremental displacements. This option is used by the initial stiffness algorithm. The program automatically determines which option is required at any stage, depending on the solution algorithm requested.

Subroutine CRISFIELD

This routine, if required, updates the displacement increment calculated in FRONT using the Crisfield acceleration procedure.

Subroutine LINSEARCH

This routine carries out the linesearch procedure if required. This procedure is discussed in detail in section 7.2.3.

Subroutine ADDVEC

This subroutine adds the displacement increment calculated in subroutine FRONT to the previous total displacement increment $\Delta \underline{u}_{t,i}$ to obtain $\Delta \underline{u}_{t,i+1}$. If the Crisfield acceleration technique is being used the current residual and displacement increment are stored for use on the next iteration. If the BFGS acceleration technique is being used the displacement increment is also stored, and also the residual vector on the first iteration. However on subsequent iterations the residual will already have been modified by the BFGS routine and therefore in this case it is stored by the BFGS routine before modification.

Subroutine RESIDL

This subroutine calculates the stress and inelastic strain increments at the integration points in the structure from the total displacement increment $\Delta \underline{u}_{t,i+1}$. The procedure is described in more detail in section 8.4. The routine updates the total stress vector and calculates the equivalent nodal forces corresponding to it.

Subroutine CONVER

The residual vector is calculated as the difference between the equivalent nodal forces and the applied nodal forces. The root mean square of the residual vector is divided by the root mean square of the applied nodal forces and the resulting quantity is compared to a pre-set

tolerance. If the quantity is less than the pre-set tolerance, then convergence is deemed to have been achieved for the current load increment. It has been found that convergence can still eventually be achieved even if the size of the quantity has increased between iterations 1 and 2. This is particularly true in creep problems, where the formulation is changed from tangent stiffness to initial strain at this point in the calculation. The iterative procedure is thus not assumed to be diverging unless the magnitude of the quantity is larger than both the values calculated on iterations 1 and 2. If the quantity is decreasing but is larger than the pre-set tolerance, then the process is converging although convergence has not yet been achieved.

Once convergence has been achieved the next task is to output the results for the current load increment. The results are copied from a scratch file into the restart database file RUNnnn.RST. This not only serves as a restart file, but can also be interrogated by the user after the analysis has been completed using a separate program RDDI. If graphical output is required to the FEMVIEW package it is processed by routine OUTPUT.

Subroutine OUTPUT

The function of this subroutine is to output the results to the graphics package FEMVIEW. The interface to the graphics software is described in Appendix B. The graphics programs require nodal stresses, which are extrapolated from Gauss point values using a trilinear function of the element local coordinate system. Results for nodes which are common to more than one element are obtained by taking the average of the element contributions.

If the whole load has been applied the program moves on to the next loadcase, otherwise the next load increment is applied. It is possible to halt program execution before the whole load has been applied if required. Once the last loadcase has been analysed program execution is terminated.

8.2 The calculation of the allowable timestep size

As discussed in section 8.1 it is necessary to determine what fraction of the loading to be applied in the current loadcase should be applied in the current step. It was originally intended to control the size of steps to be used in creep problems by using Cormeau's criterion[164]. Although this is only valid as a stability criterion for explicit creep analysis it was hoped that it would nonetheless provide a useable control parameter for implicit analysis. As the forward gradient

method is unconditionally stable for $\theta > \frac{1}{2}$ the user is able to set the selected timestep to a multiple of that given by the Cormeau criterion, the objective being to control the accuracy of the analysis.

However it has been found that numerical difficulties arise with this approach, because for an implicit analysis the structural stiffness matrix can become singular or negative definite. The reason for this can be seen by examining the expression for $[D_{epc}]$ (*cf* equation (6-28))

$$\left[D_{epc}\right] = \left[\left[D_2\right] - \frac{C_2}{K_1}\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]\right] \tag{8-1}$$

where $C_2 = 1 + \frac{\theta A \Delta t}{C_1}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}$

and $K_1 = A + \underline{a}^T\left[D_2\right]\underline{a} + \frac{\underline{a}^T\left[D_2\right]\underline{a}\theta A \Delta t}{C_1}\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}$

and $C_1 = 1 - \theta \Delta t \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}$.

This expression has a similar form to that for [Dep] (equation (5-65)). As is well known [Dep] is semidefinite for $A = 0$ and negative definite for $A < 0$ and hence it is not unexpected that $[D_{epc}]$ can also display these properties.

It is useful to examine the reduction in stiffness caused by the inclusion of the creep terms in the tangent stiffness formulation. Utilising equation (8-1)

$$\underline{a}^T\left[D_{epc}\right]\underline{a} = \underline{a}^T\left[D_2\right]\underline{a}\left[1 - \frac{C_2}{K_1}\underline{a}^T\left[D_2\right]\underline{a}\right] \tag{8-2}$$

and as

$$[Dep] = \left[\left[D_2\right] - \frac{\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]}{A + \underline{a}^T\left[D_2\right]\underline{a}}\right] \tag{8-3}$$

$$\underline{a}^T[Dep]\underline{a} = \underline{a}^T\left[D_2\right]\underline{a}\left[1 - \frac{\underline{a}^T\left[D_2\right]\underline{a}}{A + \underline{a}^T\left[D_2\right]\underline{a}}\right] . \tag{8-4}$$

Thus, a measure of the reduction in stiffness caused by creep can be obtained by defining a "creep stiffness parameter" χ_{epc} as

$$\chi_{epc} \equiv \frac{\underline{a}^T\left[D_{epc}\right]\underline{a}}{\underline{a}^T[Dep]\underline{a}} = \frac{\left[1 - \frac{C_2}{K_1}\,\underline{a}^T\left[D_2\right]\underline{a}\right]}{\left[1 - \frac{\underline{a}^T\left[D_2\right]\underline{a}}{A + \underline{a}^T\left[D_2\right]\underline{a}}\right]} \quad . \tag{8-5}$$

Substituting for K_1

$$\chi_{epc} = \frac{A}{A + C_2\underline{a}^T\left[D_2\right]\underline{a}} \cdot \frac{A + \underline{a}^T\left[D_2\right]\underline{a}}{A} \quad . \tag{8-6}$$

Hence, substituting for C_2

$$\chi_{epc} = \frac{1 - \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}\theta\Delta t}{1 + \left(\frac{\underline{a}^T\left[D_2\right]\underline{a}}{A + \underline{a}^T\left[D_2\right]\underline{a}} A \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}} - \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}\right)\theta\Delta t} \quad . \tag{8-7}$$

The limiting value of χ_{epc} as Δt tends to ∞ can easily be obtained using l'Hôpital's rule as

$$\underset{\Delta t\to\infty}{Lt}\left(\chi_{epc}\right) = \frac{-\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}}{\left(\frac{\underline{a}^T\left[D_2\right]\underline{a}}{A + \underline{a}^T\left[D_2\right]\underline{a}} A \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}} - \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}\right)} \quad . \tag{8-8}$$

For a strain hardening material A is greater than zero. Hence, for primary creep where $\partial\dot{\bar{\varepsilon}}_c/\partial\bar{\varepsilon}_c$ is less than zero it can be seen that χ_{epc} will always be positive whatever timestep size is chosen. If tertiary creep is considered however, where $\partial\dot{\bar{\varepsilon}}_c/\partial\bar{\varepsilon}_c$ is greater than zero then χ_{epc} will be negative for values of Δt greater than a critical value Δt_0, which is given by

$$\Delta t_0 = \frac{1}{\theta\,\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}} \quad . \tag{8-9}$$

It has been found that the occurrence of negative values of χ_{epc} is well correlated with the structural stiffness matrix becoming non positive definite. An impression of the variation of χ_{epc} with Δt is given in Fig 16.

In practice it has been found to be necessary to control the timestep by setting the minimum permitted positive value of χ_{epc}. Denoting this value by $\bar{\chi}_{epc}$ the maximum timestep Δt_{max} is given by

$$\Delta t_{max} = \frac{1 - \bar{\chi}_{epc}}{\left(\frac{\underline{a}^T \left[D_2\right] \underline{a}}{A + \underline{a}^T \left[D_2\right] \underline{a}} \right) A\theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} \bar{\chi}_{epc} + \theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} \left(1 - \bar{\chi}_{epc}\right)} \quad \ldots\ldots \quad (8\text{-}10)$$

It can be shown that for all stress states (with the exception of hydrostatic tension or compression for which $\underline{a}$ is indeterminate)

$$\underline{a}^T \left[D_2\right] \underline{a} = \frac{3E_2}{2(1 + \nu)} \quad (8\text{-}11)$$

for an isotropic material where E_2 is the Young's modulus at the end of the increment. Thus equation (8-10) can be simplified to

$$\Delta t_{max} = \frac{1 - \bar{\chi}_{epc}}{\left(\frac{3E_2 A}{2(1 + \nu)A + 3E_2} \right) \theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} \bar{\chi}_{epc} + \theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} \left(1 - \bar{\chi}_{epc}\right)} \quad \ldots\ldots \quad (8\text{-}12)$$

A difficulty arises with equation (8-12) if creep occurs during a transient loadcase, because E_2 cannot be determined until the size of the timestep has been selected. In order to overcome this problem E_2 is simply approximated by the value of Young's modulus at the start of the increment. It is not felt that this will be a significant problem in practice, because the expected transients are relatively rapid, and it is unlikely that sufficient creep will occur during a transient to require a reduction in the timestep size. The condition (8-12) is evaluated for the Gauss point which has the minimum allowable timestep using the Cormeau criterion. If necessary the timestep is then reduced. It is desirable to make $\bar{\chi}_{epc}$ as small as possible, whilst still retaining a positive definite stiffness matrix, as this will give the

largest timestep. A value of 0.2 for $\bar{\chi}_{epc}$ has been found to be appropriate.

If plasticity is not present, then a similar analysis to that presented in section 6.2 leads to an expression for $\Delta\underline{\sigma}$ of the form

$$\Delta\underline{\sigma} = \left[[D_2] - \frac{[D_2]\underline{a}\underline{a}^T[D_2]\theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\Delta t}{C_1C_3}\right]\Delta\underline{\varepsilon} - \left[[D_2] - \frac{[D_2]\underline{a}\underline{a}^T[D_2]\theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\Delta t}{C_1C_3}\right]\Delta\underline{\varepsilon}_T$$

$$+ \left[[\Delta D] - \frac{[D_2]\underline{a}\underline{a}^T[\Delta D]\theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\Delta t}{C_1C_3}\right]\underline{\varepsilon}_e - \frac{[D_2]\underline{a}}{C_1C_3}\left[\dot{\bar{\varepsilon}}_c + \theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial T}\Delta T\right]\Delta t \qquad \ldots\ldots \text{(8-13)}$$

where $C_3 = 1 + \dfrac{\theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\underline{a}^T[D_2]\underline{a}\Delta t}{C_1}$.

Hence the elasto-creep constitutive matrix $[D_{ec}]$ is given by

$$[D_{ec}] = \left[[D_2] - \frac{[D_2]\underline{a}\underline{a}^T[D_2]\theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\Delta t}{C_1C_3}\right] . \qquad \text{(8-14)}$$

Thus defining χ_{ec} in a similar manner to χ_{epc}, it can be shown that

$$\chi_{ec} \equiv \frac{\underline{a}^T[D_{ec}]\underline{a}}{\underline{a}^T[D_2]\underline{a}} = \frac{1 - \theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}\Delta t}{1 + \left(\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\underline{a}^T[D_2]\underline{a} - \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}\right)\theta\Delta t} . \qquad \text{(8-15)}$$

As with the elasto-plastic creep case χ_{ec} can only be negative in tertiary creep. The maximum allowable timestep is given as a function of the limiting value of χ_{ec}, $\bar{\chi}_{ec}$ as

$$\Delta t_{max} = \frac{1 - \bar{\chi}_{ec}}{\underline{a}^T[D_2]\underline{a}\theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\bar{\chi}_{ec} + \theta \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}\left(1 - \bar{\chi}_{ec}\right)} \qquad \text{(8-16)}$$

and utilising equation (8-11)

$$\Delta t_{max} = \frac{1 - \bar{\chi}_{ec}}{\frac{3E_2}{2(1+\nu)}\theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} \bar{\chi}_{ec} + \theta \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}\left(1 - \bar{\chi}_{ec}\right)} \quad . \qquad (8\text{-}17)$$

Results obtained using the above algorithm are presented in Chapter 9.

8.3 The calculation of the centrifugal load

The calculation of the centrifugal loads follows the standard finite element procedure as given in Zienkiewicz[57] or Bathe[80] but is given here for completeness.

The virtual work statement given in equations (5-18) and (5-19) can be written as follows:

$$\int_{ve} \left[[B]\Delta\underline{\delta}_{el}\right]^T [D][B]dv\, \underline{u}_{el} = \Delta\underline{\delta}_{el}^T \underline{f}_{el} \quad . \qquad (8\text{-}18)$$

If body forces, such as centrifugal loads enter the analysis however the virtual work statement above is incomplete because it neglects the work done by the body forces during the small arbitrary displacement. The complete statement can be written as follows:

$$\int_{ve} \left[[B]\Delta\underline{\delta}_{el}\right]^T [D][B]dv\, \underline{u}_{el} = \Delta\underline{\delta}_{el}^T \underline{f}_{el} + \int_{ve} \Delta\underline{\delta}^T \underline{f}_b \, dv \quad , \qquad (8\text{-}19)$$

where $\underline{f}_b$ is the body force at a point in the element and $\Delta\underline{\delta}$ is the displacement at that point caused by the virtual displacement of the nodes. Using the shape functions $\underline{N}$

$$\Delta\underline{\delta} = \begin{bmatrix} \underline{N} & \underline{0} & \underline{0} \\ \underline{0} & \underline{N} & \underline{0} \\ \underline{0} & \underline{0} & \underline{N} \end{bmatrix} \begin{bmatrix} \Delta\underline{\delta}_x \\ \Delta\underline{\delta}_y \\ \Delta\underline{\delta}_z \end{bmatrix} = [\tilde{N}]\Delta\underline{\delta}_{el} \quad . \qquad (8\text{-}20)$$

Hence, substituting in equation (8-19) and dividing by the arbitrary displacement $\Delta\underline{\delta}_{el}$

$$\int_{ve} [B]^T[D][B]dv\ \underline{u}_{el} = \underline{f}_{el} + \int_{ve} [\tilde{N}]^T \underline{f}_b\ dv \qquad (8\text{-}21)$$

or

$$\left[K_e\right] \underline{u}_{el} = \underline{f}_{el} + \underline{f}_{b,el} \ . \qquad (8\text{-}22)$$

In practice the integral required to calculate the body force is evaluated numerically using Gaussian quadrature as used for the stiffness matrix. Thus

$$\underline{f}_{b,el} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \left[\tilde{N}(\xi,\eta,\zeta)\right]^T \underline{f}_b\ \det[J]\ d\xi d\eta d\zeta \qquad (8\text{-}23)$$

and $\underline{f}_b$ is the centrifugal load caused by a unit volume of material at each of the quadrature sampling points.

8.4 The strain to stress calculation

As stated in section 8.1 the function of subroutine RESIDL is to calculate the stress and inelastic strain increments for each Gauss point from the total strain increment. The subincrementation procedure is used, whereby the strain increment is divided into a number of substeps, and the stresses and inelastic strains obtained for each one in succession. Techniques for carrying out these calculations for elasto-plastic material behaviour are exhaustively reviewed by Marques[195]. It has naturally proved necessary to modify these extensively to accommodate both the temperature dependence and creep behaviour involved in the blade problem.

The variation in total strain over the increment can be assumed to be either linear or nonlinear. It has been found that greater accuracy and improved convergence of the iterative procedure are achieved if a nonlinear variation is used. The method by which this is achieved is described in section 8.5.

The user can optionally update the calculation of the creep strain increment in the strain to stress calculation. This means that on each subincrement the creep law is re-evaluated using the current stress, temperature, etc and the creep strain increment is then calculated using the desired timestepping scheme. This leads to improved accuracy at the expense of additional computational effort. If the creep strain increment is calculated using the assumptions made when evaluating the stiffness matrix and the pseudo loads, then for an elasto-creep analysis

convergence of the iterative procedure will be obtained on the first iteration, albeit with some loss in accuracy. The first task of the subroutine therefore, if the creep strain calculation is not to be updated, is to calculate the creep strain increment using the assumptions made when evaluating the stiffness matrix.

The next stage is to determine an elastic trial solution, *ie* a trial value of the effective stress at the end of the subincrement assuming that plasticity does not occur. This is necessary in order to be able to distinguish between the three possibilities which can occur, *viz* (i) the Gauss point is elastic or just plastic at the start of the subincrement, and behaves entirely elastically, (ii) the Gauss point is elastic at the start of the subincrement and becomes plastic, or (iii) the Gauss point is initially plastic and remains so. It is assumed (possibly pessimistically) that the behaviour of the Gauss point on the previous subincrement is no guide to its behaviour on the current one.

The size of each subincrement is initially set equal to the remainder of the load increment. If creep occurs on the load increment the size is reduced to a user specified factor of the ratio between the Cormeau timestep limits for the particular Gauss point and the Gauss point which controls the timestep for the whole structure. This has the effect of allowing the user to specify the number of subincrements required for those parts of the structure with the highest creep rate, whilst not wasting an unnecessary number of subincrements on those parts of the structure where little creep is occurring. If large timesteps are used it is possible that a significant increase in the creep rate could occur over an increment. It is important in this case not to use too large a subincrement size initially. This is achieved by estimating the stress and creep strain at the end of the increment without using subincrementation. Hence an estimate of the final creep rate is obtained and the subincrement size is then reduced if necessary.

If the Gauss point started the subincrement elastic and becomes plastic it is necessary to determine when yield occurs in order to calculate the plastic strains accurately. For non-transient load increments without updating of the creep strain calculation this is a relatively simple procedure. Following Marques[195] the fraction R of the subincrement at which yield first occurs is given by

$$R = \frac{\bar{\sigma}_E - \sigma_y}{\frac{\partial F}{\partial \underline{\sigma}} \cdot \Delta \underline{\sigma}_E} , \qquad (8\text{-}24)$$

where $\bar{\sigma}_E$ is the effective stress at the end of the subincrement calculated elastically, σ_y is the yield stress, and $\Delta\underline{\sigma}_E$ is the increment of stress calculated elastically. For transient subincrements or those with creep strain updating the position is more complicated, as there is no analytic expression for R. A numerical solution procedure has therefore been adopted. The desired value of R is obtained when the ratio $\bar{\sigma}_E/\sigma_y$ is equal to unity, and the value of the ratio is already available for two trial values $R = 0$ and $R = 1$. The next trial value of R is obtained by linear interpolation between successive values of $\bar{\sigma}_E/\sigma_y$ obtained from previous trials. It is found that this procedure usually converges within a few trials. Once the value of R has been obtained, the size of the subincrement is reduced by this factor so that only elastic behaviour occurs within it. It is thus expected that on the next subincrement the behaviour will be entirely plastic.

If plasticity occurs it is necessary for accuracy to reduce the previously calculated size of the subincrement. The size of an entirely plastic subincrement is determined using a similar criterion to that given by Owen and Hinton[196]. Thus

$$\text{MSTEP} = \text{INT}\left\{\frac{8.0\left(\bar{\sigma}_E - \sigma_{y,i+1}\right)}{\sigma_{y,i+1}} + 1\right\} , \qquad (8\text{-}25)$$

where $\sigma_{y,i+1}$ is the yield stress at the current plastic strain and the final temperature of the subincrement. The size of the subincrement is then reduced by a factor 1/MSTEP.

For plastic subincrements it is necessary to calculate the value of the plastic multiplier $d\lambda$. This will be a function of the stress state which will change over the subincrement and cannot be determined *a priori*. It is possible to determine the plastic multiplier using the stresses at the end of the subincrement calculated elastically, the so called radial return method[197]. However the additional complexity does not appear to be justified in practice[198] and therefore in the GANDALF code $d\lambda$ is evaluated using the conditions at the start of the subincrement.

It is sometimes found that at the end of the subincrement the computed stresses do not lie exactly on the yield surface. This is because the behaviour of the material has been linearized over the subincrement. The error can most readily be removed by a simple scaling of the stresses and the elastic and plastic strains. It should be noted that this does not exactly bring the stress point to the yield surface, as the yield

stress will be altered by scaling the plastic strains. However the additional expense of invoking an iterative procedure to perform the correction does not seem justified.

8.5 The nonlinear subincrementation procedure

An overview of the subincrementation solution procedure was presented in section 6.3. The objective of the method is to achieve greater accuracy by calculating the stresses and inelastic strain increments at the integration point level in a number of steps. The total displacement increment $\Delta \underline{u}_{t,i+1}$ is divided into a number of subincrements and the stress and inelastic strain increments are calculated successively for each subincrement. The simplest approach is to assume that the displacement increment and hence the total strain varies linearly over the load/timestep. However it is easy to see that for some analyses this is not a particularly realistic assumption. For example in an analysis of a constant stress creep test the strain will vary nonlinearly with time in both primary and tertiary creep. This deficiency was noted by Ottosen and Gunneskov[174] who developed a subincrementation procedure where the total strain varied nonlinearly over the load/timestep. However their method relies on a number of additional assumptions. In particular it is assumed that the same variation over the load/timestep occurs in each component of strain. As, for example, the creep strain increment will have a zero hydrostatic component, whereas the deviatoric component of the thermal strain increment will be zero, this will be an incorrect assumption for some conditions. In addition in order to estimate the creep strain increment the authors assumed that the effective stress remained constant over the timestep, thus allowing the creep law to be integrated analytically. These two limitations have been overcome in the analysis presented below.

The objective of the method is to devise an expression for an estimate of the total strain increment up to any point in the load/timestep. It is clear that at the start of the timestep the total strain increment will be zero, and at the end of the timestep the estimate of the total strain increment must be equal to $\Delta \underline{\varepsilon}$, the predicted total strain increment. Denoting the total strain increment up to the current subincrement by $\delta \underline{\varepsilon}$, it is assumed that as before the total strain increment can be partitioned into elastic, plastic, creep and thermal strains. Thus

$$\delta \underline{\varepsilon} = \delta \underline{\varepsilon}_e + \delta \underline{\varepsilon}_p + \delta \underline{\varepsilon}_c + \delta \underline{\varepsilon}_T \quad . \tag{8-26}$$

If a flexibility matrix [F] is defined such that

$$\underline{\varepsilon}_e = [F]\underline{\sigma} \tag{8-27}$$

then it can be shown that (*cf* equation (5-27))

$$\delta\underline{\varepsilon}_e = [\delta F]\underline{\sigma}_t + \left[F_{t+\delta t}\right]\delta\underline{\sigma} \tag{8-28}$$

where $\underline{\sigma}_t$ is the stress at the start of the increment.

The plastic strain increment can be estimated using the plasticity equations presented in section 5.3. If it is assumed that these are valid for finite increments then (*cf* equation (5-43))

$$\underline{a}^T\delta\underline{\sigma} = A\delta\lambda - \frac{\partial F}{\partial T}\,\delta T \tag{8-29}$$

and

$$\delta\underline{\varepsilon}_p = \underline{a}\delta\lambda \quad . \tag{8-30}$$

Hence combining equations (8-29) and (8-30), assuming that $F \equiv \bar{\sigma}$ and noting that $\delta\bar{\sigma} = \underline{a}^T\delta\underline{\sigma}$

$$\delta\underline{\varepsilon}_p = \frac{\underline{a}}{A}\left[\delta\bar{\sigma} + \frac{\partial\bar{\sigma}}{\partial T}\,\delta T\right] \quad . \tag{8-31}$$

The creep strain increment can be estimated using the forward gradient method given in section 6.2 as

$$\delta\underline{\varepsilon}_c = \frac{\underline{a}}{C_1^*}\left[\dot{\bar{\varepsilon}}_c + \theta\,\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\,\delta\bar{\sigma} + \theta\,\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial T}\,\delta T\right]\delta t \tag{8-32}$$

where $C_1^* = 1 - \theta\,\dfrac{\delta\dot{\bar{\varepsilon}}_c}{\delta\bar{\varepsilon}_c}\,\delta t$.

Thus combining equations (8-26), (8-28), (8-31) and (8-32) the increment of total strain up to the current point in the timestep can be expressed as

$$\delta\underline{\varepsilon} = [\delta F]\underline{\sigma}_t + \left[F_{t+\delta t}\right]\delta\underline{\sigma} + \frac{\underline{a}}{A}\left[\delta\bar{\sigma} + \frac{\partial\bar{\sigma}}{\partial T}\,\delta T\right]$$

$$+ \frac{\underline{a}}{C_1^*}\left[\dot{\bar{\varepsilon}}_c + \theta\,\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\,\delta\bar{\sigma} + \theta\,\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial T}\,\delta T\right]\delta t + \delta\underline{\varepsilon}_T \quad .$$

..... (8-33)

In order to utilise equation (8-33) it is necessary to assume a variation in the stresses over the timestep so that $\delta\underline{\sigma}$ and $\delta\bar{\sigma}$ can be evaluated. If the variation is assumed to be linear then

$$\delta\underline{\sigma} = \frac{\delta t}{\Delta t}\Delta\underline{\sigma} \qquad \text{and} \qquad \delta\bar{\sigma} \quad \frac{\delta t}{\Delta t}\Delta\bar{\sigma} \quad . \qquad (8\text{-}34)$$

As the total strain increment over the timestep is known $\Delta\underline{\sigma}$ can be found directly from the elasto-plastic creep constitutive equation (6-28). As no additional assumptions have been made in deriving equation (8-33) using equation (6-28) to give a value for $\Delta\underline{\sigma}$ means that at the end of the timestep, when δt equals Δt , the estimate of the total strain increment $\delta\underline{\varepsilon}$ will automatically be equal to $\Delta\underline{\varepsilon}$. Hence the assumed increment of total strain over each subincrement can be obtained by evaluating $\delta\underline{\varepsilon}$ for the values of δt corresponding to the start and end of each subincrement. As with Ottosen and Gunneskov's method[174] a non-linear variation in strain over the increment is predicted when creep hardening occurs. Additionally a nonlinear variation will occur if the material properties vary nonlinearly with temperature. Equation (8-33) cannot be used if ideal plasticity occurs, when $A = 0$. In this case the total strain is simply assumed to vary linearly over the load/timestep. It should be noted that in cases where the total strain increment over the timestep is zero equation (8-33) will not necessarily give a zero strain increment at intermediate points within the load step. Results for the method are presented in Chapter 9.

8.6 The restart capability

If the restart facility is required then at the end of each successfully completed load increment information is written to the restart database RUNnnn.RST shown in Fig 5. This is done by the subroutine WRTDB. The database contains three types of data:

(i) Data that describe the database itself;

(ii) A number of loading parameters which determine the current state of the analysis; and

(iii) The solution data at nodes and Gauss points.

The solution is stored for the current load increment and can also be stored at a previous load increment which has been specified by the user. This is useful if, for example, the user wishes to change the solution algorithm after a particular load increment.

At the beginning of a restart analysis, in addition to the current state of the structure, which is stored in the restart database, information is also required on the structural geometry and topology and the material properties. To avoid copying these data they are obtained from the original GDATAnnn.DAT file. Data is read from the restart database by GANDALF3P using two subroutines RDLODRST and RDNODRST. RDLODRST reads the loading parameters and RDNODRST the solution data. The information which describes which describes the database enables the code to locate the correct data for RDLODRST and RDNODRST.

On a restart analysis, since the structural stiffness matrix is not stored, it must be created no matter what solution algorithm is employed (*eg* Initial Stiffness, Hybrid Mk 2, etc). In general the stiffness matrix will reflect the current state of the structure. However, it is possible to use the original elastic stiffness matrix if required at the temperature distribution after the first load increment of either the first or last loadcase. This enables the initial stiffness and Hybrid Mk 0 elastic solution algorithms to be restarted.

Chapter 9

APPLICATION OF GANDALF3P TO TEST PROBLEMS

9.1 Introduction to the test problems

In order to decide on an appropriate test strategy for a finite element code, it is first necessary to consider the various aspects of an analysis from which errors can arise. In the general case the various sources of error arise from three categories of approximation: in the method, in the models in the code and in the analysis. As will be seen, each of these groups of approximations requires a different treatment.

At its most basic level any finite element code can be considered as a numerical solver of partial differential equations. Errors in an analysis are inevitable because assumptions are made about the spatial and temporal variations of displacement, stress, etc. The method also discretises a continuum, so that the solution is only accurately available at a finite number of points, the Gauss points rather than throughout the region of interest. A finite element solution is thus inevitably an approximation to the behaviour of the real system, however accurately the boundary conditions are known. The accuracy of the approximate solution will depend on such factors as the fineness of the mesh, the degree of numerical round-off in the solution, and the stability and accuracy of the solution method. A good impression of the importance of these effects can be obtained by comparing the predictions of the code with analytic solutions, although this is limited to relatively simple problems.

The second category of approximation arises because a finite element analysis will employ some mathematical models of the real world to enable the problem to be solved numerically. The chief area where this is required in GANDALF is in models of material behaviour, to include such effects as creep, plasticity, anisotropy, damage, etc. Inevitably these models are approximations to reality and may not be well correlated with experimental observations due to the technical and logistical constraints on conducting appropriate testing. Each new level of sophistication in a model not only leads to more complex coding, but also to progressively more difficult and expensive test programmes to determine the material constants. The sensible strategy is to start with a simple model and gradually increase its complexity, and also to ensure that a given model is not used outside its known range of validity.

In order to validate such models it is often not essential to actually conduct a finite element analysis (apart from analyses to check the coding of the model). The majority of test specimens are statically

determinate, and so the predicted response to a prescribed testing cycle can often be determined directly from the model. It is suggested that this is in general the most desirable method as it avoids introducing the additional errors inherent in a finite element analysis into the validation procedures.

The third category of approximation arises because in conducting a given finite element analysis the engineer will inevitably be forced into making some assumptions and approximations. These may arise firstly because the required input data is unknown or uncertain. In the blade analysis perhaps the major uncertainty is the metal temperatures. The engineer may also attempt justifiably to simplify the analysis by neglecting effects which he believes to be second order. In the GANDALF code for example pressure loads and centrifugal stiffening which is caused by the change in the line of action of the centrifugal load due to the bending of the blade, are neglected. The scope of the analysis may also be limited by cost considerations. It is difficult to suggest a universal panacea to overcome the potential difficulties in this category, other than the application of engineering judgement and the use of sensitivity studies. The use of a more complex analysis is often impractical.

As the development of an effective solution method for the turbine blade problem was a major objective of the research programme, it was considered that the most appropriate aim for the test exercise would be to assess the errors arising from the solution methods in the code. The objective was to examine the various solution methods available with a view to identifying the fastest and most accurate techniques. Although some work has been carried out on validating the material model contained in the GANDALF code, and this is discussed in Appendix C, further work would have required more extensive material test data which was not available. The final category of approximations those appertaining to particular analyses, are obviously strongly problem dependent. An investigation aimed at solving a range of problems with a similar solution method whilst potentially enabling the accuracy achievable by a given method to be assessed would not have allowed the various methods in the GANDALF code to be fully tested. The test programme thus involved the running of a number of problems using a wide range of solution techniques in order to determine the most appropriate scheme.

9.2 Creep test cases

9.2.1 Introduction

The time stepping solution algorithm introduced in Chapter 6 provides a range of solution schemes. The scheme employed depends on the values of a number of control parameters. In the displacement calculation these are

θ_s the implicit/explicit factor and S_s the factor by which the Cormeau timestep limit is multiplied. In the strain to stress calculation the parameters are whether the creep strain calculation is updated, θ_r the implicit/explicit factor and S_r the factor by which the Cormeau timestep limit is multiplied.

It was considered important to examine what effect these parameters had on the accuracy and convergence rate of the solution. This has been achieved by examining two uniaxial problems, one being creep at constant stress, the other modelling stress relaxation due to creep at constant total strain. In both cases a four element mesh was used. Although these two problems are both very simple, it was hoped that they would provide bounds on the behaviour of the solution scheme. The stresses and temperatures chosen were representative of blade conditions. A range of control parameters was examined using both the initial stiffness and Hybrid Mk 3 solution schemes. The effects of the non-linear subincrementation scheme described in section 8.5 were also studied. Plastic behaviour was neglected. In this case as discussed in section 6.2 the results obtained from the tangent stiffness and Hybrid Mk 3 schemes are identical. The Hybrid Mk 3 scheme was expected to require less CPU time than the tangent stiffness method and was thus chosen in preference to it. The two schemes chosen thus represent the extremes of possible modified Newton-Raphson strategies. Results are presented in sections 9.2.2 and 9.2.3.

9.2.2 The stress relaxation problem

Loading for this problem was applied in two loadcases. In the first a prescribed displacement was applied, together with an instantaneous increase in temperature to 900°C. The displacement was selected so that a uniaxial tensile stress of 180 MPa was established at the end of the loadcase. The loading was applied in two steps, the first of 80% and the second of 20% of the full load. In the second loadcase the displacement was held constant and the stresses were allowed to relax due to creep. The response over a total of 500 hours of creep was calculated, although in practice it was found that the stresses had relaxed sufficiently after approximately 50 hours for the remaining time to be analysed in one load step.

Values of θ_s and θ_r of 0.0, 0.5 and 1.0 were analysed. S_s was kept constant at 1.0 and values of S_r of 0.1 and 0.5 were used, to enable the effects of differing numbers of subincrements to be examined. This relatively low value of S_s in combination with a figure of 0.2 for $\bar{\chi}_{ec}$ the control value of the creep stiffness parameter meant that the timestep

sizes were solely determined by the Cormeau criteria, and the additional timestep control scheme described in section 8.2 was not activated. In nearly all of the computer runs it was decided to update the creep strain during the strain to stress calculation. If updating is not used identical results are obtained for elasto-creep problems whether or not subincrementation is used and so in this case S_r was set equal to S_s. Identical results are obtained for both the linear and nonlinear subincrementation scheme when θ_s equals zero as in this case the nonlinear scheme degenerates to the linear one. Results for the linear scheme are given in Tables 1 to 3 and those for the nonlinear scheme in Tables 4 to 5. Results for both the initial stiffness algorithm (using the elastic stiffness matrix at the end of the first load increment of loadcase one), and the Hybrid Mk 3 algorithm are tabulated.

The results show that the initial stiffness method has generally only a slightly inferior rate of convergence than the Hybrid Mk 3 algorithm for both the linear and nonlinear schemes. In most cases only about a third more iterations are required. An exception to this arises in the runs with no updating of the creep strain when the initial stiffness method requires two or three times the number of iterations used by the Hybrid Mk 3 method. It is clear therefore that updating the creep strain calculation improves the convergence rate of the initial stiffness algorithm relative to Hybrid Mk 3. The good convergence rate of the initial stiffness algorithm for this problem is also probably due to the fact that the tangent stiffness matrix will not change a great deal during the second loadcase. It will be recalled that the elasto-creep constitutive matrix is given by (*cf* equation 8-14).

$$[\text{Dec}] = \left[[D] - \frac{\left[D_2\right]\underline{a}\underline{a}^T\left[D_2\right]\theta\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\Delta t}{C_1 C_3} \right] \qquad (9\text{-}1)$$

where $C_1 = 1 - \theta\Delta t\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}$

and $C_3 = 1 + \frac{\theta\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\underline{a}^T\left[D_2\right]\underline{a}\Delta t}{C_1}$.

As the timestep is controlled by the Cormeau criterion the factor $\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\Delta t$ remained constant throughout the analysis. Thus the stiffness matrix is only modified by changes in C_1 and consequent changes in C_3. It was found that the structure always remained in primary creep and so $\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}$ was

always negative. As creep progressed the absolute value of $\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}$ reduced but its effect on the value of C_1 was counteracted by the increasing value of Δt . Thus the stiffness parameter χ_{ec} between the tangent and elastic stiffnesses at the full load temperature typically only varied over the loadcase between 0.4 and 0.7 even for those runs where θ_s equalled 1.0. At 900°C, the full load temperature, the elastic modulus of the material is approximately 91% of its value at 724°C, the temperature used to calculate the initial stiffness. This suggests that the change in the stiffness matrix due to the change in temperature is relatively insignificant compared to the change caused by the inclusion of the implicit creep terms in the stiffness matrix formulation. The total difference in stiffness between the initial and tangent stiffnesses in this problem is however low enough to produce only a relatively low number of iterations if the initial stiffness method is used.

For the initial stiffness runs with the linear code with updating the best convergence rates are in general produced when θ_s equals 1.0. This is somewhat unexpected as the initial and tangent stiffnesses matrices are closest in magnitude when θ_s equals 0.0 (and thus the tangent stiffness matrix is simply the elastic stiffness matrix at the full load temperature). However as discussed above the change in stiffness due to the inclusion of the creep terms is relatively small. It is probable that θ_s equals 1.0 gives a closer approximation to the creep strain increment which is calculated in the subincrementation procedure than either θ_s equals 0.0 or θ_s equals 0.5, and this accounts for the good convergence. For the nonlinear code with initial stiffness and updating however the best convergence rates are produced when θ_s equals 0.5. The explanation is probably that the estimated increment of total strain at intermediate points in the timestep calculated using equation (8-33) will be more in error (*ie* more non zero) when θ_s equals 1.0 than when θ_s equals 0.5. For the Hybrid Mk 3 algorithm the best convergence was provided when θ_s equals 0.5 for both the linear and nonlinear codes.

An analytical solution is not possible for this problem and thus the accuracy of the results was assessed by comparing them with another run using the initial stiffness algorithm, and the nonlinear code, number 414. This had a value of S_s of 0.01, θ_s of 0.5 and S_r and θ_r of 0.001 and 0.5 respectively. The run required 260 increments. The root mean square percentage and maximum errors in effective stress and creep strain for the various runs are shown in Tables 1 to 5. It was considered that run 414 was acceptably close to the true solution because

another run using double the timestep size in both the displacement and strain to stress calculations gave virtually identical results. It is clear from the results that in general updating significantly reduces the magnitude of the errors. If updating is not used then the smallest errors occur when θ_s equals 0.5 for both algorithms and of course both the linear and nonlinear schemes produce identical results.

With updating the value of θ_s and the choice of solution algorithm have almost no effect on the final accuracy achieved with the linear code. This is to be expected, because as the displacement increment is constrained to be zero the original assumptions used in the displacement loop to calculate the creep strain are overwritten by those used in the strain to stress loop. The most accurate subincrementation method is obtained for the linear scheme when θ_r equals 0.5. Even when the larger of the two subincrement sizes is used it is still nearly as accurate as either of the other two schemes with the smaller size of subincrement. The explicit subincrementation scheme (θ_r equals 0.0) performs badly. Reducing S_r from 0.5 to 0.1 reduces the error by approximately a factor of 3 in all cases.

For the nonlinear scheme updating again in general significantly reduces the magnitude of the errors. However for the nonlinear scheme the results are more complex than those for the linear scheme because increasing the number of subincrements does not improve the accuracy achieved when θ_r equals 0.5 and θ_s is greater than zero. In general for a given subincrementation scheme the accuracy is decreased as θ_s is increased. The results produced by the subincrementation scheme will clearly depend on the variation of total strain over each subincrement. As the nonlinear code erroneously assumes a nonzero total strain increment halfway through the increment, this will automatically cause an error in the predicted creep strains. In general therefore, however good the subincrementation scheme the accuracy achieved will to some extent be dependent on the assumptions made in the displacement loop. For the nonlinear code, therefore, the most accurate solution is obtained when θ_s equals 0.0 (and θ_r equals 0.5) because this assumes a zero total strain increment throughout the step. This result is identical to that achieved with the linear code.

The times taken for the analyses are given in Tables 1 to 5. The mesh contained only four elements and so it is not appropriate to draw any conclusions about the fastest solution method for the blade problem from these results. However as the time taken for the forward reduction varies as the cube of the problem size, and that for backsubstitution as the square of problem size it would be expected that the time taken for the Hybrid Mk 3

algorithm would substantially increase relative to that taken for the initial stiffness scheme.

9.2.3 The constant stress problem

The loading for this problem was applied in two loadcases. In the first a load was applied together with an instantaneous increase in temperature to 900°C. The load was selected so that a uniaxial stress of 180 MPa was established at the end of the loadcase. As with the stress relaxation problem the loading was applied in two steps, the first of 80% and the second of 20% of the full load. In the second loadcase the applied load was held constant and the structure was allowed to creep at a nominally constant stress. The response over a total of 500 hours of creep was predicted. The same sets of values of the creep solution control parameters as were used in the stress relaxation problem were employed. A value of 1.0 for S_s was used throughout, together with a value of 0.2 for $\bar{\chi}_{ec}$, which meant that as with the stress relaxation problem only the Cormeau criterion was used to control the timestep size. Results for the linear subincrementation scheme are given in Tables 6 to 8 and those for the nonlinear scheme in Tables 9 to 10. Results for both the initial stiffness algorithm (using the elastic stiffness matrix at the end of the first load increment of loadcase one) and the Hybrid Mk 3 algorithm are tabulated.

Unlike the stress relaxation problem discussed in section 9.2.2 the initial stiffness algorithm requires substantially more iterations than Hybrid Mk 3. This is because during the 500 hours of creep time the structure passes through primary, secondary and tertiary creep. This leads to substantial changes in the tangent stiffness, and is reflected in the values of χ_{ec} calculated during the analyses. Typically, for θ_s equals 1.0 these varied between 0.3 and 0.6 and hence the initial stiffness can be a relatively poor approximation to the tangent stiffness. The effect was particularly marked as in the analysis the majority of the increments were in tertiary creep where χ_{ec} was close to the lower figure.

If updating of the creep strains is not used the results for the linear and nonlinear schemes are of course identical, and as expected the best convergence rate for the initial stiffness scheme is obtained when θ_s equals 0.0 as this scheme does not modify the stiffness matrix to include creep effects. The ratio between the number of iterations taken for the initial stiffness and Hybrid Mk 3 schemes varied between approximately 3:1 for θ_s equals 0.0 and 12.5:1 for θ_s equals 1.0.

If updating is used the best convergence for the Hybrid Mk 3 algorithm was obtained when θ_s equals 0.5, and for the initial stiffness algorithm when θ_s equals 0.0. This applied to both the linear and nonlinear subincrementation schemes. These results are not unexpected, as the initial stiffness matrix will be closest to the tangent stiffness matrix when θ_s equals 0.0, and the Hybrid Mk 3 algorithm when θ_s equals 0.5 is probably the best approximation to the various subincrementation schemes.

An analytical solution is available for this problem as the creep took place at nominally constant stress. The root mean square percentage and maximum errors in creep strain are given in Tables 6 to 10. The errors in stress were very small in all cases and will not be further discussed. Unlike the stress relaxation problem different accuracies for the creep strains are obtained for the initial stiffness and Hybrid Mk 3 algorithms, as would be expected due to the different displacement increments calculated for each method. However, there does not appear to be any systematic variation between the accuracies achieved for the two methods.

If updating is not used then the linear and nonlinear subincrementation schemes will of course produce identical results. The most accurate solution, as with the stress relaxation problem, is obtained when θ_s equals 0.5, for both solution algorithms.

The most interesting feature of the results using updating is the remarkably good performance of the subincrementation scheme with θ_r equals 0.0 and $S_r = 0.5$. This scheme performs well with the linear subincrementation scheme for all values of θ_s and with both solution algorithms. However it does not perform well with the nonlinear subincrementation scheme, and also produced very poor results for the stress relaxation problem. It is clear that the good performance of this scheme is problem dependent. Possibly for the constant stress problem the errors produced on the two subincrements within each increment cancel out, thus accounting for the good results.

For the linear subincrementation scheme the most accurate results were in general produced when θ_s equals 0.5 although the results for θ_s equals 0.0 were nearly as good, particularly for the tangent stiffness algorithm. Increasing the number of subincrements was often not beneficial, as was observed for the stress relaxation problem with the nonlinear code. It is assumed that the inaccuracy arising from the assumption of linear strain variation over the increment is sufficient to negate any increase in accuracy arising from an improved subincrementation scheme. The most

accurate results were produced with θ_r equals 0.0, as discussed above. The series of runs with θ_r equals 1.0 produced relatively poor results.

The nonlinear subincrementation scheme produced generally more accurate results than the linear scheme. The best results were in general produced with θ_s equals 0.5. In most cases increasing the number of subincrements was beneficial. On average approximately a factor of 3 improvement in accuracy was obtained when S_r was reduced from 0.5 to 0.1. This is similar to the improvement obtained for the stress relaxation problem. The most accurate solution was produced when θ_r equals 1.0 and S_r equals 0.1. Some good results were also produced when θ_r equalled 0.5.

As with the stress relaxation problem it is not possible to draw any precise conclusions from the times taken for the analyses, because of the small size of the mesh. The comments made in the previous section concerning the effect of problem size on the times taken for the initial stiffness and Hybrid Mk 3 algorithms will also apply in this case.

9.2.4 Tests of the creep timestep control algorithm

The size of timesteps which are employed during a creep analysis is controlled using the scheme described in section 8.2. Two control parameters are available to the user, the multiple of the Cormeau timestep to be used, and $\bar{\chi}_{epc}$ or $\bar{\chi}_{ec}$ the control value of the creep stiffness parameter. In order to investigate the performance of the creep stiffness parameter in controlling the timestep size the constant stress creep problem described in section 9.2.3 was re-analysed. The Cormeau timestep multiple was set at 5.0 and a range of values of $\bar{\chi}_{ec}$ from 0.0 to 0.3 was examined. As the solution scheme was expected to be mathematically unstable for this timestep for values of θ_s of less than 0.5 values of 0.5, 0.667, 0.878 and 1.0 were used. A value of 0.667 was advocated by Zlamel[199] and a value of 0.878 by Lambert[200] and Liniger[201] on the basis that this most accurately fits the exponential solution to a linear first order differential equation. Ten subincrements in each increment were used throughout and θ_r was set equal to θ_s. Results for the nonlinear subincrementation scheme are presented in Table 11, and for the linear subincrementation scheme in Table 12.

During each analysis the observed values of χ_{ec} (if the timestep was not reduced by the control) followed the same general pattern. At the start of the analysis a value of 0.2 was typical. This then increased during primary creep to approximately 0.5. There was then a rapid reduction at the onset of tertiary creep to typically 0.01. This

corresponded to the change in sign of $\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}$. Finally during tertiary creep the value increased to about 0.1. The creep stiffness control will only operate and reduce the timestep when $\bar{\chi}_{ec}$ is greater than χ_{ec} . Thus the proportion of the analysis for which the control operates is dependent on the choice of $\bar{\chi}_{ec}$.

A number of trends can clearly be distinguished from the results. As $\bar{\chi}_{ec}$ is reduced the number of increments taken for each analysis with the nonlinear code is reduced, because the creep stiffness control acts less frequently to reduce the size of the timestep. The total number of iterations required is in general diminished as $\bar{\chi}_{ec}$ is reduced and the increment size increases. This effect was observed by the author in the analysis of elasto-plastic problems[172]. It is suggested that this is caused by the fact that a relatively large number of iterations are required to obtain convergence on each timestep and so a few larger timesteps are more advantageous than a large number of small ones. The only exceptions to this effect are the runs when θ_s equals 0.5 and this may be caused by the large maximum number of iterations per increment required in this case for low $\bar{\chi}_{ec}$. The linear subincrementation scheme generally required substantially more iterations than the nonlinear scheme. Indeed for low values of $\bar{\chi}_{ec}$ convergence was not achieved as the number of iterations per increment exceeded 120 the maximum permissible.

As expected variations in $\bar{\chi}_{ec}$ had the most effect for the largest values of θ_s . This is because large values of θ_s will produce the most change in the stiffness matrix and hence produce the smallest values of χ_{ec} . Thus the stiffness control will be activated more frequently to increase χ_{ec} to the desired value $\bar{\chi}_{ec}$. For $\bar{\chi}_{ec}$ equals 0.0 the control was not activated for any of the values of θ_s examined.

It is instructive to examine the average and maximum number of iterations per increment. These data are given in Tables 11 and 12. It will be observed that for large values of $\bar{\chi}_{ec}$ the average and maximum values are similar. Thus the stiffness control is ensuring that roughly the same number of iterations are used for each increment. As $\bar{\chi}_{ec}$ is reduced a much larger ratio between the maximum and average values is observed, because the stiffness control is less effective.

The accuracy of the solutions progressively deteriorates as $\bar{\chi}_{ec}$ is reduced. This is not unexpected as the size of the increments is increasing. Generally the nonlinear subincrementation scheme produces substantially more accurate results. The most accurate solution for a fixed number of timesteps appears to be obtained when θ_s equals 0.667.

It was considered worthwhile to investigate whether a more accurate solution for a given number of increments could be obtained with timesteps selected using the creep stiffness control, a multiple of the Cormeau criterion or a combination of both as used in the results presented in Tables 11 and 12. As the nonlinear subincrementation scheme produced both faster convergence rates and a more accurate solution it was used for this study. It did not prove possible to find a value of $\bar{\chi}_{ec}$ which provided adequate control of the increment size throughout the analysis. However by adjusting the value of S_s with $\bar{\chi}_{ec}$ set to zero it proved possible to generate runs with approximately the same number of increments as had been used for the runs with S_s of 5.0 and $\bar{\chi}_{ec}$ of 0.3. These runs are tabulated in Table 13. In general the accuracies of the two sets of runs are similar. However the use of the χ_{ec} parameter slightly reduces the total number of iterations required in most cases (the exception is run 961 where the χ_{ec} parameter is used least). The number of iterations used per increment is also slightly more constant from increment to increment when the χ_{ec} parameter is used. Examination of intermediate timings showed that for the same total number of steps in the analysis the runs using the χ_{ec} parameter used larger steps during primary creep than those without, but that this effect was reversed during tertiary creep.

The major advantage of the χ_{ec} parameter lies in its ability to detect the onset of tertiary creep. Because of the reduction in creep rate which occurs as the structure moves out of primary creep, the Cormeau criterion allows an increasing timestep size at this point in the analysis. The χ_{ec} parameter, because it contains the term $\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}$ can detect the additional creep which will occur due to the increasing creep rate over the projected timestep. The use of the parameter as a control criterion allows the timestep to be reduced to prevent an excessive number of iterations being required at this point, and a consequent deterioration in the accuracy of the solution. Although it is possible to obtain as accurate a solution in tertiary creep without using χ_{ec} by using a smaller value of S_s the results suggest that this leads to unnecessarily conservative timestep sizes in the primary creep stage of the analysis. In primary creep it is not practicable to use $\bar{\chi}_{ec}$ as the only control on the timestep size. This is because as noted in section 9.2.2 χ_{ec} does not vary substantially during primary creep, despite the rapid reduction in creep rate occurring at this stage of an analysis. The substantially reduced creep rate near the end of primary creep allows too large timesteps to be taken for a given value of $\bar{\chi}_{ec}$. However the

the use of S_s as the control parameter in primary creep would appear to give an acceptable solution.

The θ' creep equation used in the GANDALF code does not have a secondary creep term, and the performance of the timestepping scheme has not been investigated in this case. However an indication of its performance can be obtained by considering the expressions for χ_{ec}, and the Cormeau timestep Δt_c. It will be recalled (*cf* equation (8-15)) that

$$\chi_{ec} = \frac{1 - \theta_s \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} \Delta t}{1 + \left(\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} \underline{a}^T [D_2] \underline{a} - \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \varepsilon_c} \right) \theta_s \Delta t} \tag{9-2}$$

and for practical creep laws the Cormeau criterion gives an allowable timestep Δt_c of (*cf* equation (5-92))

$$\Delta t_c = \frac{4(1 + \nu)}{3E} \cdot \frac{1}{\partial \dot{\bar{\varepsilon}}_c / \partial \bar{\sigma}} \quad . \tag{9-3}$$

For an isotropic material with a Von Mises effective stress (*cf* equation (8-11))

$$\underline{a}^T [D_2] \underline{a} = \frac{3E_2}{2(1 + \nu)} \quad . \tag{9-4}$$

Thus combining equations (9-2) to (9-4) and noting that Δt equals $S_s \Delta t_c$

$$\chi_{ec} = \frac{1 - \theta_s \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} S_s \Delta t_c}{1 - \theta_s \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} S_s \Delta t_c + 2\theta_s S_s} \tag{9-5}$$

where it is assumed that the elastic modulus of the material remains constant over the timestep. Thus for secondary creep, where $\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c}$ is zero χ_{ec} is given by the relatively simple expression

$$\chi_{ec} = \frac{1}{1 + 2\theta_s S_s} \quad . \tag{9-6}$$

Thus in view of the close relationship between the two parameters in this case it is probable that they would be equally effective in controlling the timestep size. Overall, it appears that the use of the two parameters S_s and χ_{ec} provides good control of the analysis throughout the various stages of creep, with S_s the most effective for primary creep and χ_{ec} for the tertiary stage.

9.2.5 Investigation of the optimum number of subincrements for creep problems

In order to examine the performance of the code when different numbers of subincrements per timestep are chosen, the constant stress creep problem described in section 9.2.3 was re-analysed. The Cormeau timestep multiplying factor S_s was set at 5.0. The same values of θ_s as were used in section 9.2.4 were chosen. These were 0.5, 0.667, 0.878 and 1.0. The control value of the creep stiffness parameter $\bar{\chi}_{ec}$ was adjusted to give approximately the same number of increments in the analysis for each value of θ_s. Two further series of runs with θ_s equals 1.0 and 0.5 and $\bar{\chi}_{ec}$ equals 0.0 to give larger timesteps were also undertaken. The number of subincrements per timestep was varied by using values of 0.05 to 5.0 for S_r. It should be noted that the ratio S_s/S_r represents the maximum number of subincrements which can be used on any step. Thus the range of values examined corresponded to 1-100 subincrements per main step. However, the subincrementation scheme will increase the size of each subincrement if the creep rate falls during the timestep, for example during primary creep, and therefore fewer subincrements will be used. The nonlinear subincrementation scheme was used throughout, together with the initial stiffness method. Results are presented in Table 14.

All the runs produced very low errors in the predicted stresses. Typically the root mean square error in the stress was 0.03% which was substantially less than the convergence tolerance of 1%.

Examination of the strain errors suggests that for each value of θ_s there is a minimum achievable error, however many subincrements are taken. This would appear to be dependent on the timestep size and the value of θ_s. As would be expected for a given θ_s smaller timesteps give a smaller minimum achievable error. For a given timestep size the most accurate achievable solution is provided when θ_s equals 0.667 with θ_s equals 0.5 and 0.878 providing similar but slightly less good accuracies. The worst accuracy occurs when θ_s equals 1.0. (It is suggested that run 800 with θ_s equals 0.5 and S_r equals 1.0 is

anomalous because the maximum error in creep strain does not occur on the last increment.)

For the series of runs with $\bar{\chi}_{ec}$ greater than zero a significant improvement in accuracy was achieved through subincrementation when θ_s equals 0.667 or 0.878, but there was little improvement for values of θ_s of 0.5 and 1.0. However a worthwhile improvement in accuracy was achieved by subincrementation when θ_s equals 0.5 and $\bar{\chi}_{ec}$ equals 0.0. No improvement in accuracy was achieved when θ_s equals 1.0 and $\bar{\chi}_{ec}$ equals 0.0, but subincrementation did allow convergence of the iterative procedure to be achieved.

For the nonlinear subincrementation scheme to be effective the estimated creep strain variation within the increment (equation (8-33)) must accurately reflect the true variation of the strain. It is suggested that the good performance achieved when θ_s equals 0.667 and 0.878 arises because these values of θ_s give a good estimate of the variation in creep strain. Run 802 with θ_s equals 0.5 and $\bar{\chi}_{ec}$ equals 0.3 produced a very low error and thus there is little scope for further improvement through subincrementation. If however, the size of the timesteps is increased by reducing $\bar{\chi}_{ec}$ it then becomes possible to reduce the error using subincrementation for this value of θ_s. Subincrementation does not appear effective when θ_s equals 1.0 for either of the increment sizes chosen, and it is suggested that this is due to the inaccuracy in the assumed creep strain variation over the increment.

As with the results for the constant stress and stress relaxation problems with S_s equals 1.0 discussed in sections 9.2.2 and 9.2.3, for those series of runs where subincrementation was effective there was generally a reduction in the number of iterations required as the subincrement size was reduced. The exception is the series of runs with θ_s equals 0.5 and $\bar{\chi}_{ec}$ equals 0.0 and this may be due to the differing number of increments used by the various runs and the somewhat erratic performance of the subincrementation scheme in this case. The reduction in the number of iterations counterbalanced the increased amount of time required for the strain to stress calculation, and so for a given value of θ_s a similar time was taken for each run for all subincrement sizes down to a value of S_r of 0.5, although there was usually a slight reduction compared to the time taken for the corresponding runs with no subincrementation. However, the runs with S_r of 0.05, *ie* potentially 100 subincrements required substantially more computer time in all cases with only limited further reductions in the number of iterations required. It is suggested therefore that the use of 10 subincrements per main step

is likely to provide close to the minimum number of iterations. For the size of problem examined the time saving achieved by the reduction in the number of iterations was comparable to the additional time required for the strain to stress calculation. However for larger problems an overall time saving can be expected because of the nonlinear dependence of solution time on problem size.

Little improvement in accuracy was achieved by increasing the number of subincrements beyond two. This agrees with the results of Ottosen and Gunneskov[174] who obtained approximately a 65% reduction in error using two subincrements, only a further 5% reduction using 10 subincrements and virtually no further improvement in using 100 subincrements. It is suggested therefore that the use of 10 subincrements per main step is likely to provide both a significant reduction in the total number of iterations and virtually the greatest achievable accuracy.

It is unclear why these results and the results presented in sections 9.2.2 and 9.2.3 do not always show a monotonic increase in the accuracy achieved as the subincrement size is reduced. This is in contrast to the results reported by Ottosen and Gunneskov, who however, only assessed the accuracy of their scheme at the end of each analysis rather than using an RMS value as in the results presented here. Ottosen and Gunneskov based their method on the creep equation

$$\varepsilon_c = A\sigma^q t^p \qquad (9\text{-}7)$$

and evaluated the estimated variation in creep strain over the timestep by assuming that the stress remained constant. This is in contrast to the method used in the GANDALF code which attempts to allow for changes in the stress over the timestep using equation (8-33). The explanation for the different performance of the two methods may lie in the fact that the Ottosen and Gunneskov technique may consistently under or over predict the estimated creep strain at all intermediate points within the timestep depending on whether the stress is increasing or decreasing. This would suggest that the error would decrease monotonically as the subincrement size is reduced. However it is possible that the GANDALF method under or overpredicts the estimated creep strain at differing points within each timestep. It is conceivable therefore that the errors in creep strain cancel out for some values of S_r leading to a much lower error in these cases. In addition, the increments chosen by the automatic timestepping scheme depend slightly on the subincrementation procedure

chosen, as shown by the varying number of increments required for the various runs in each series of analyses, and this may also have some effect on the accuracy achieved.

9.2.6 Application of quasi-Newton methods to creep problems

A number of runs of the constant stress creep problem described in section 9.2.3 were undertaken to examine the performance of the various quasi-Newton solution methods in the code, with and without linesearches, when applied to a creep problem. The quasi-Newton methods were used in conjunction with the initial stiffness algorithm. Values of S_s of 5.0, θ_s of 0.5, S_r of 0.5 and θ_r of 0.5 were used throughout, together with the nonlinear subincrementation scheme. The creep strain was always updated in the strain to stress loop. A value of 0.3 for $\bar{\chi}_{ec}$ was used. Results are presented in Table 15. Little difference was observed between the accuracy of the various methods.

All of the quasi-Newton methods converged apart from the BFGS scheme without linesearch and gave a substantial reduction in the total number of iterations. It is suggested that the nonconvergence of the BFGS scheme is due to the use of the tangent stiffness matrix in the formulation for iteration 1. As the tangent stiffness matrix reverts back to the elastic matrix on the second iteration the quasi-Newton extrapolation will underestimate the required stiffness, leading to potential divergence of the solution. It should be noted that this will occur even though the initial stiffness algorithm is used, because the BFGS method characterises the required stiffness of the structure using the displacement increment and the change in the residual. In order to overcome this difficulty a modified version of the BFGS scheme was coded, known as "BFGS-creep", in which the update was only applied from the third iteration onwards, rather than the second as in the standard scheme. This method was found to converge successfully.

The values of the control parameters for the Crisfield run were set at 5.0 for R_1 and 0.65 for R_2. The value for R_1 is outside the range suggested by Crisfield[189] but with this choice of parameters the acceleration was applied on every load increment from the third iteration onwards. In general the values of b_i/a_i were less than 0.15 indicating that only a small proportion of the displacement increment from the previous iteration was contributing to the accelerated increment. On the second iteration, however, the acceleration was only applied on two increments. In most cases a_i exceeded the specified control bounds substantially with a peak positive value of 1240.9 and negative value of -2399.5 being

recorded. These values are clearly inappropriate. It is probable therefore that only the control bounds on R_1 were enabling convergence of the acceleration scheme to be achieved, and relaxing the value of R_1 might well cause the method to diverge. The poor performance of the Crisfield scheme on the second iteration is probably for the same reason that the unmodified BFGS scheme diverges; the quasi-Newton assumption is not appropriate on the second iteration. Overall, however, the Crisfield method provides a 70% drop in the total number of iterations used compared to the initial stiffness algorithm.

The "BFGS-creep" algorithm performed almost as well as the Crisfield method. The condition number of the updating matrix never fell below 0.65 and so the update was applied from the third iteration onwards on all timesteps. Despite the fact that the "BFGS-creep" algorithm uses data from iteration two onwards to estimate the updated stiffness matrix rather than only from the previous iteration as in the Crisfield scheme, it does not appear to offer any benefit over the Crisfield algorithm. This is perhaps not surprising in view of the low values for b_i/a_i found for the Crisfield algorithm, which suggests that the displacement from the current iteration is much the most helpful in achieving convergence.

The linesearch algorithm achieved better convergence than either the "BFGS-creep" scheme or the Crisfield method, probably partly due to the fact that it can be applied on the first iteration, whereas the quasi-Newton schemes can only be applied from the second iteration onwards. Indeed the algorithm used fewer iterations than the Hybrid Mk 3 algorithm, which is also included in Table 15. As previously noted the Hybrid Mk 3 algorithm is equivalent to tangent stiffness for elasto-creep problems. A maximum value of η_i of 5.0 and a maximum of five searches were allowed. The linesearch convergence tolerance GTOL was set at 0.1. It was found that if the search procedure was started GTOL could always be satisfied in a maximum of three searches. On timesteps where high creep rates or rapid changes of creep rate with creep strain occurred, particularly at the transition between primary and secondary creep, it was found that on the first iteration G(1.0) exceeded G(0.0), thus indicating that as might be expected the modified Newton procedure had produced a poor search direction. In this case the linesearch procedure is abandoned and the unaccelerated displacement increment is used. On subsequent iterations however, the linesearch procedure was used successfully throughout the analysis. In general, a smaller number of searches on each iteration was required as the iteration number increased. This

is presumably because on successive iterations the solution (calculated from the total displacement increment) is nearer convergence and it is easier to obtain a value of η_i for which $G(\eta_i)$ is acceptably reduced. Overall the linesearch scheme achieved an 84% reduction in the total number of iterations used compared to the initial stiffness algorithm. It is, however, likely that this extremely good performance is in part due to the relative simplicity of the test problem. A more complex mesh with widely varying stiffness and creep rates might produce less improvement.

Little further improvement in convergence rate was obtained by combining the linesearch algorithm with the quasi-Newton schemes, although the use of linesearch enabled convergence to be achieved for the unmodified BFGS algorithm.

The run with the linesearch and "BFGS-creep" algorithm required nearly the same number of iterations as the run using linesearch alone. This was not unexpected as the linesearch algorithm converged in a maximum of three iterations, and hence the "BFGS-creep" algorithm is only used on very few iterations. The BFGS update was applied on the third iteration on all time-steps where three iterations were required. As with the linesearch algorithm convergence was achieved in a maximum of three iterations. The only reason that overall the scheme required two fewer iterations than the linesearch algorithm was that on two increments convergence was achieved on the second iteration, rather than the third as had been the case with the linesearch algorithm. This is due to slight changes in the numerical value of the solution obtained.

The analysis using the unmodified form of the BFGS algorithm and linesearch performed slightly better than the run using linesearch alone and also slightly better than the "BFGS-creep" with linesearch method. In order for the BFGS algorithm to be applied on the second iteration it was essential for the linesearch procedure to operate on the first iteration. If this did not occur (because G(1.0) was greater than G(0.0)) the BFGS algorithm was not applied on the second iteration because the square root in the evaluation of v_{-i} (equation (7-20)) was negative. No improvement was obtained in the number of iterations required for these increments. For increments where the linesearch procedure was successful on the first iteration, the BFGS acceleration was always applied on the second and all subsequent iterations. For these iterations the linesearch tolerance GTOL was satisfied in most cases with a value of unity for η_i and therefore no additional searches were performed. It would thus seem worthwhile to investigate the possibility of dispensing with the linesearch procedure entirely after the

first iteration. The current algorithm appears to have the potential to work well but it is clear that the stiffness matrix must be sufficiently accurate to provide a good search direction on the first iteration.

The linesearch with Crisfield acceleration algorithm produced an identical solution to the standard BFGS with linesearch algorithm. This result was unexpected and further numerical experiments showed that it was due to the relative simplicity of the test problem. It would be expected however that the results would be similar as Crisfield[190] derived his method from the nonproduct form of the BFGS update which can be written

$$\left[K_i\right]^{-1} = \left[K_{i-1}\right]^{-1} - \frac{\Delta u^*_{i-1}\left(\psi_{i-1} - \psi_i\right)^T\left[K_{i-1}\right]^{-1}}{\Delta u^{*T}_{i-1}\left(\psi_{i-1} - \psi_i\right)} - \frac{\left[K_{i-1}\right]^{-1}\left(\psi_{i-1} - \psi_i\right)\Delta u^{*T}_{i-1}}{\Delta u^{*T}_{i-1}\left(\psi_{i-1} - \psi_i\right)}$$

$$+ \left(1 + \frac{\left(\psi_{i-1} - \psi_i\right)^T\left[K_{i-1}\right]^{-1}\left(\psi_{i-1} - \psi_i\right)}{\delta^T_{i-1}\left(\psi_{i-1} - \psi_i\right)}\right)\left(\frac{\delta_{i-1}\Delta u^{*T}_{i-1}}{\Delta u^{*T}_{i-1}\left(\psi_{i-1} - \psi_i\right)}\right). \quad \ldots\ldots (9\text{-}8)$$

Crisfield assumed that $\left[K_{i-1}\right]^{-1}$ was equal to $\left[K^*\right]^{-1}$ the stiffness matrix used in the unaccelerated modified Newton-Raphson scheme, and, to derive the method used in the GANDALF code made the additional simplification

$$\Delta u_{i-1} \equiv \left[K^*\right]^{-1}\psi_{i-1} \simeq \delta_{i-1} \qquad (9\text{-}9)$$

which will be exactly true if only the standard modified Newton-Raphson scheme has been used on the previous iteration. As can be seen therefore the two methods would not in general be expected to produce identical results.

It is clear that the good performance of the combined quasi-Newton and linesearch schemes is crucially dependent on whether a successful linesearch can be carried out on the first iteration, which in turn depends on whether a good search direction has been obtained. It is possible that the use of the initial stiffness algorithm may not provide a good enough search direction in some cases as indeed was observed in the problem considered here, and more frequent updating of the stiffness matrix may be required. This will, of course, degrade the speed of the algorithm.

Little difference was observed between the times taken for the various quasi-Newton schemes whether or not linesearch was used, although they all required substantially less time than the initial stiffness algorithm. It is clear therefore that when the linesearch algorithm is used the time saved by the reduction in the number of iterations is lost due to the time taken for the linesearch procedure. It is possible that more promising results might be obtained for a larger problem, as the time taken for each iteration will be dependent on the square of the problem size, whereas the time taken for the linesearch procedure will be linear with problem size. However, the linesearch procedure might not be able to produce as large a percentage drop in the number of iterations for a larger and more complex problem. Slightly the fastest solution was obtained with the Crisfield with linesearch algorithm, as this method produced the same results as the BFGS with linesearch method without the need to recalculate the update procedure from the start of each increment.

9.3 Elasto-plastic test cases

9.3.1 Analysis of a hollow sphere under internal pressure loading

The deformation of a hollow sphere of ideally plastic material obeying Von Mises yield criterion under internal pressure is one of the few three-dimensional elasto-plastic test problems for which an analytic solution can be derived. The solution is given in Refs 98 to 99 and 202. The problem was therefore chosen for assessment of the GANDALF code.

The finite element mesh utilised for the analyses is shown in Fig 17. Only one eighth of the sphere need be modelled as due to symmetry the remainder of the structure can be simulated by constraining the nodes on the $x = 0$, $y = 0$ and $z = 0$ planes. The mesh was created using an automatic mesh generator with parabolic interpolation between control points. As a result the nodes do not lie exactly on the spherical surface. However the maximum error in radius is less than 1% and it is considered that this has negligible influence on the results. The mesh has 162 elements, 922 nodes and the maximum frontwidth during the reduction was 537 degrees of freedom. The inner and outer radii were 100 and 200 metres respectively. The material properties used were a Young's modulus of 2.14×10^{11} N/m^2, a Poisson's ratio of 0.3 and a yield stress of 2.14×10^{8} N/m^2.

In order to perform the analysis it is first necessary to transform the pressure loading into an equivalent set of forces acting at the nodes on the internal surface of the sphere. The internal surface is made up of a number of element faces, and each face will be defined by a constant

value of one of the element's local coordinates ξ, η, ζ. Only the forces due to the pressure load on one face will be considered, as the forces due to pressure over the entire surface can simply be obtained by summation of the forces on each face. It will be assumed without loss of generality that the face is defined by ζ = constant. As the global coordinates x, y, z of any point on the element face are a function of ξ, η, ζ, by the usual rules of partial differentiation

$$\begin{aligned} dx &= \frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \eta} d\eta + \frac{\partial x}{\partial \zeta} d\zeta \\ dy &= \frac{\partial y}{\partial \xi} d\xi + \frac{\partial y}{\partial \eta} d\eta + \frac{\partial y}{\partial \zeta} d\zeta \\ dz &= \frac{\partial z}{\partial \xi} d\xi \quad \frac{\partial z}{\partial \eta} d\eta + \frac{\partial z}{\partial \zeta} d\zeta \quad . \end{aligned} \tag{9-10}$$

The elementary area of the face defined by the infinitesimal increments $d\xi$ and $d\eta$ is thus a parallelogram with sides given by the vectors

$$\vec{\underline{\xi}} = \frac{\partial x}{\partial \xi}, \frac{\partial y}{\partial \xi}, \frac{\partial z}{\partial \xi} \qquad \vec{\underline{\eta}} = \frac{\partial x}{\partial \eta}, \frac{\partial y}{\partial \eta}, \frac{\partial z}{\partial \eta} \quad . \tag{9-11}$$

It can be shown that the resultant force $\underline{F}$ due to a pressure p on the elementary area is given by the vector product of $\vec{\underline{\xi}}$ and $\vec{\underline{\eta}}$

$$\underline{F} = -p\left[\vec{\underline{\xi}} \times \vec{\underline{\eta}}\right] = -p \begin{vmatrix} \underline{i} & \underline{j} & \underline{k} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{vmatrix} \tag{9-12}$$

where $\underline{i}$, $\underline{j}$ and $\underline{k}$ are unit vectors in the directions x, y and z respectively.

Consider an arbitrary displacement $(\delta u_x, \delta u_y, \delta u_z)$ of the ith node of the element. The displacement in the x direction of a point ξ, η, ζ on the face is given by (*cf* equation (5-7))

$$u_x(\xi,\eta,\zeta) = N_i(\xi,\eta,\zeta)\delta u_x \tag{9-13}$$

with similar expressions for $u_y(\xi,\eta,\zeta)$ and $u_z(\xi,\eta,\zeta)$. Thus the work done by the displacement of the entire face is given by

$$W = -\int_S pN_i(\xi,\eta,\zeta)\left[\delta u_x \delta u_y \delta u_z\right]\begin{vmatrix} \underline{i} & \underline{j} & \underline{k} \\ \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \end{vmatrix} d\xi d\eta \quad . \tag{9-14}$$

This must equal the work done by the forces on the ith node F_x, F_y and F_z. This is given by

$$W = F_x \delta u_x + F_y \delta u_y + F_z \delta u_z \quad . \tag{9-15}$$

By considering only δu_x, δu_y and δu_z to be non zero in turn it is easily seen that

$$F_x = -\int_S pN_i(\xi,\eta,\zeta)\left[\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi}\right] d\xi d\eta \tag{9-16}$$

with similar expressions for F_y and F_z. The integral in equation (9-16) is evaluated numerically using Gaussian quadrature. It should be noted that even for a square face under uniform pressure equation (9-16) does not produce an allocation of equal fractions of the total force on the element face to each node. In fact one-third of the total force on the face will be allocated to each midside node and minus one-twelfth to the corner nodes.

The use of equation (9-16) is relatively complicated, and for the analysis of the sphere a simplified procedure was adopted. Each element face was assumed to be planar and the position of its centroid was calculated from the positions of the corner nodes. The total force on the face was then calculated, and resolved into F_x, F_y and F_z by considering the centroid position. Finally the forces were apportioned to the nodes using the ratios described above.

As the pressure inside the sphere is increased yielding begins at the inside surface. The pressure required to cause yielding to spread to a radius r_c is given by

$$p = \sigma_y\left[2 \ln \frac{r_c}{a} + \frac{2}{3}\left(1 - \left(\frac{r_c}{b}\right)^3\right)\right] \tag{9-17}$$

where a and b are the inner and outer radii of the sphere respectively and σ_y is the yield stress of the material. For the sphere considered in this study initial yield occurs when $r_c = a$ at a pressure $p = 0.583\sigma_y$ and plastic collapse occurs when $r_c = b$ at a pressure $p = 1.385\sigma_y$.

The full applied load corresponded to an internal pressure of $4.143\sigma_y$ and it was therefore anticipated that initial yield would occur at 14.1% of full load, and plastic collapse at 33.4% of full load.

In order to examine the speed and convergence rates of the various algorithms in the code the problem was analysed using the same loading history for each technique. The loading history adopted was an initial step of 15% of the full load, followed by three equal steps of 5% of the full load. The first step was thus just into the plastic range and the final load level of 30% of full load was a little below the theoretical collapse load. Results are presented in Table 16.

The theoretical solution for the stresses within the sphere is given by

$$\sigma_{rr} = 2\sigma_y \ln\left(\frac{r}{a}\right) - p \,, \qquad \sigma_{\theta\theta} = \sigma_y\left(1 + 2\ln\left(\frac{r}{a}\right)\right) - p \qquad (9\text{-}18)$$

for the plastic region $r \leq r_c$ and

$$\sigma_{rr} = -\frac{2\sigma_y}{3}\left(\frac{r_c}{r}\right)^3\left(1 - \frac{r^3}{b^3}\right) \,, \qquad \sigma_{\theta\theta} = \frac{\sigma_y}{3}\left(\frac{r_c}{r}\right)^3\left(1 + \frac{2r^3}{b^3}\right) \qquad (9\text{-}19)$$

for the elastic region $r > r_c$. The elastic-plastic transition radius r_c can be determined from

$$p = \frac{2\sigma_y}{3}\left[1 - \left(\frac{r_c}{b}\right)^3 + 3\ln\left(\frac{r_c}{a}\right)\right] \,. \qquad (9\text{-}20)$$

The accuracy of the predicted stresses was very similar for all the algorithms. The root mean square error taken over all the Gauss points in the structure varied from run to run from 2.16% to 2.37% for the radial stress and 5.00% to 5.72% for the hoop stresses.

The number of iterations required for each of the solution methods differed significantly. For the modified Newton-Raphson schemes the number of iterations required by each method was reduced as the stiffness matrix was reformulated more often, as would be expected. Thus the

fewest number of iterations was required by the Hybrid Mk 3 scheme and the largest by the initial stiffness scheme. Even the Hybrid Mk 3 scheme required slightly more iterations than the tangent stiffness method. The Crisfield solution scheme used the same values for R_1 and R_2 as had been previously used for the creep problem. The effectiveness of the quasi-Newton schemes can be assessed by examining the percentage reduction in the number of iterations required compared to the corresponding modified Newton-Raphson method. For both the BFGS and Crisfield algorithms the greatest reduction occurred when they were used in conjunction with the initial stiffness method, and the least when used in conjunction with the Hybrid Mk 2 method. This is as expected, as if the Hybrid Mk 2 method is used the quasi-Newton schemes are not applied until the third iteration as the stiffness matrix has been updated on the second iteration and there will be little scope for further improvement. In every case the selected quasi-Newton update was applied on every iteration where it was available. Generally the BFGS algorithm produced slightly better results than the Crisfield method. Of the quasi-Newton schemes the fewest iterations were required by the Hybrid Mk 2 scheme with the BFGS or Crisfield method.

The linesearch solution procedure used the same values of the control parameters as those previously used for the creep problem. As might be expected the linesearch algorithm produced no reduction in the number of iterations required when used in conjunction with the tangent stiffness scheme, although an improved value of η_i was found on two iterations. When the linesearch algorithm was used in conjunction with the modified Newton-Raphson schemes approximately the same percentage reduction in the number of iterations required was achieved for each scheme. Hence the smallest number of iterations were required when the procedure was used in conjunction with the Hybrid Mk 2 algorithm. However approximately twice as many searches per iteration were performed when the procedure was used in conjunction with the initial stiffness or Hybrid Mk 1 algorithm as were required when the procedure was used with the Hybrid Mk 2 algorithm. This effect was not unexpected as it is more difficult for the search procedure to find a low value of $G(\eta_i)$ if the search direction is poor.

When the linesearch procedure was added to the quasi-Newton solution methods a significant reduction in the number of iterations required was achieved for both the quasi-Newton with initial stiffness schemes. However little further improvement was obtained for the quasi-Newton with Hybrid Mk 1 or Hybrid Mk 2 algorithms. This is not surprising as the number of iterations taken for these schemes without linesearch is quite close to that required for tangent stiffness.

The number of searches per iteration required for the quasi-Newton with linesearch schemes was similar to those required for the corresponding basic modified Newton methods, although there was a slight reduction for the Hybrid Mk 1 based schemes. It would appear therefore that for this problem the quasi-Newton schemes do not substantially alter the search direction. It is assumed therefore that the observed improvement in convergence over the standard modified Newton methods which occurs when the quasi-Newton schemes are used without linesearch arises mostly from the acceleration of the modified Newton-Raphson displacement increments $\Delta \underline{u}_i$. This is borne out by the relatively low values of b_i/a_i encountered when the Crisfield algorithm was used, the highest value being 0.365 on the initial stiffness with Crisfield run. However it is significant that for the initial stiffness based schemes the quasi-Newton methods (without linesearch) perform better than the linesearch method, suggesting that in this case the search direction is being somewhat improved by the quasi-Newton updates. However at the other extreme, the Hybrid Mk 2 based schemes, the linesearch method performs better than the quasi-Newton methods indicating that little further improvement in the search direction is being achieved by using the quasi-Newton update.

The assessment of the time taken by each solution method is complicated by the use of the attached processor. Unlike the creep problem where the time taken on the attached processor was insignificant due to the small size of the mesh, the sphere problem does use a significant amount of attached processor time. Due to the parallel algorithm adopted however (see Appendix D) this time will be hidden in the Vax processor time. Thus, for the particular hardware configuration employed, an assessment of the best solution algorithm can be made by considering the time taken on the Vax alone. Using this criterion the results show that the tangent stiffness algorithm produces the most rapid solution because it makes the most use of the attached processor and requires the fewest iterations.

It is however instructive and also more generally relevant to consider how the code would perform on a single serial processor. Based on the results for the "cube of cubes" problem described in Appendix D it was estimated that the attached processor was running approximately eighteen times faster than the Vax for a problem of this size. The time taken for the code to run on a Vax alone has been estimated using this assumption and is shown in Table 16. It is significant that the tangent stiffness algorithm is now estimated to take the longest time of any of

of the algorithms examined. As expected in general the modified Newton-Raphson schemes perform better the fewer the number of stiffness matrix reformulations required. However the Hybrid Mk 2 scheme is estimated to be slightly better than the initial stiffness algorithm because of its better convergence properties. The large time taken for the Hybrid Mk 3 scheme suggests that it is not a promising solution method, either on its own or in conjunction with a quasi-Newton method.

Both quasi-Newton schemes reduced the estimated time taken for the analysis. As expected the greatest reduction occurred when they were used in conjunction with the initial stiffness scheme, as this resulted in the largest reduction in the number of iterations. The most rapid solution scheme is estimated to be the initial stiffness with BFGS method, which takes about a third of the time required for the tangent stiffness method.

The time taken by the modified Newton with linesearch schemes was somewhat disappointing. Little improvement in solution time was achieved by the addition of linesearch indicating that the benefit achieved by reducing the number of iterations was substantially eroded by the time taken to perform the searches. The addition of linesearch to the quasi-Newton schemes led to generally slightly worse run times. It is clear therefore that whilst the linesearch procedure does reduce the number of iterations required, its use does not appear to achieve significant reductions in computer time. This effect was also found for the creep problem discussed in section 9.2.6. It is noteworthy that for the problem described in this section, in general only about three-quarters of the number of searches per iteration were used compared to the creep problem. However, this was offset by the generally lower percentage reduction in the number of iterations achieved. The particularly low number of searches required with the Hybrid Mk 2 algorithm is reflected in the relatively short solution time predicted.

9.3.2 Analysis of a thermally loaded infinite plate

It was considered desirable to devise a test case for the temperature dependent plasticity model given in Chapter 5. A numerical solution method to calculate the stresses in a plastically deforming infinite plate with temperature dependent material properties was given by Mendelson and Spero[203], and so this geometry was chosen for analysis. The plate considered is infinite in the x direction. Plane stress conditions are assumed. If a temperature gradient $T(y)$ is established across the width of the plate then the only nonzero stress will be σ_{xx}, which will vary with y.

In the test analysis 400 elements were used. Because only 20 node brick elements are available in the code plane stress conditions were simulated by reducing the thickness of the brick elements. The plate analysed was 5×10^{-2}m long, 10^{-3}m wide and 2.5×10^{-5}m thick. This geometry enabled a good approximation to plane stress conditions to be obtained, and allowed end effects to be neglected at the centre of the plate. The 400 elements were arranged in a uniform mesh with 8 elements across the plate width, 50 along the length and one through the thickness. Constraints were applied to prevent rigid body motion or bending of the plate.

It was decided to use a temperature distribution which was symmetric about the plate centreline, as this meant that no external loads had to be applied. The 20 node brick element with 'serendipity' shape functions when mapped into a cuboid in real world geometry can only model a linear variation of total strain across the element. The correct solution could not be expected therefore unless the variation in thermal strain across each element was linear. To achieve this, a constant expansion coefficient was used, together with a linear variation in temperature from 20°C at the centre of the plate to 1050°C at the edges. Modulus data and stress-plastic strain curves for the plate material, the nickel superalloy Mar-M 002, were taken from Ref 204 and were temperature dependent. Between 20°C and 1050°C the modulus of the material drops by a factor of approximately 2½ and the yield stress by a factor of 4½. At 1050°C the 0.5% proof stress is a factor of approximately 2½ times greater than the yield stress.

The problem was analysed using two solution strategies. In the first the tangent stiffness solution method was used to analyse the problem in 20 equal increments. In the second the initial stiffness algorithm was used to analyse the problem in five equal increments. The initial stiffness run was undertaken first with a convergence tolerance of 1%. However it was realised that for this problem the applied load which is used to normalise the residual is essentially zero, because it consists only of the spurious reactions which arise due to end effects and inaccuracies in the solution. A larger tolerance is therefore appropriate in this case and a value of 10% was chosen for the tangent stiffness run. Results at various stations across the plate are given in Table 17, together with the alternative numerical solution obtained using Mendelson and Spero's method.

It is clear that the agreement between the Mendelson and Spero method and the finite element procedure is extremely close. The root mean square error between the two methods is about 0.55% in stress and about 1% in plastic strain for both finite element analyses. The tangent stiffness solution was slightly the more accurate of the two finite element

algorithms. The good agreement between the predictions of the finite element code and the alternative numerical solution method gives confidence both in the temperature dependent plasticity model given in Chapter 5 and also in the solution techniques employed.

Chapter 10

ANALYSIS OF THE RB211 "RACETRACK" BLADE

In order to complete validation of the code it was applied to the analysis of the RB211-524 "Racetrack" blade. The RB211-524 engine manufactured by Rolls-Royce plc is the largest of the RB211 family of engines having a maximum thrust of 56000 lbs. It entered airline service in 1977. The blade was chosen partly as a representative design and also because the in-service failure sites were well established.

Data describing the blade were obtained from the Rolls-Royce TACITUS system and these were processed using the program suite described in Chapter 3. The TACITUS geometry data consisted of information describing five cross-sections through the blade aerofoil. Unfortunately no data were available for the blade shroud or root. However, it proved possible to add an approximate representation of the root geometry during the construction of the mesh with the FEMGEN interactive graphics program. The mesh contained 839 20-node brick elements and 5438 nodes and is depicted in Fig 18. The blade aerofoil span is approximately 55 mm. Constraints were applied to the blade root to prevent rigid body motion.

Temperature data were available from TACITUS at the engine cruise condition for the five cross-sections through the blade aerofoil. These were interpolated onto the aerofoil mesh using the TEMPINT program described in Chapter 3. An approximate temperature distribution was applied to the blade root as no data were available from TACITUS.

As the shroud geometry was not available it was not possible to model correctly the distribution of the centrifugal load on the blade due to the shroud. However, the total load due to the shroud was known from Ref 205. The load was apportioned according to the area of each element face on the upper surface of the blade and then divided so that minus one-twelfth of the load on each element was applied to each element corner and one-third to each mid-side node. As discussed in section 9.3.1 these proportions are required to apply uniform pressure to a rectangular face of a "serendipity" 20-node brick element. It was felt that this procedure would give a good approximation to a uniform load applied at the blade tip. Gas pressure loading was neglected.

The blade is manufactured from the nickel superalloy Mar-M 002 which is conventionally cast. The θ' creep constitutive equation described in Appendix C which had a small primary region and a large tertiary stage was used. Additional material property data were obtained from Ref 204.

As data were only available for one engine condition it was not possible to analyse a complete flight cycle and therefore a simpler procedure was adopted. The blade was instantaneously loaded with the known centrifugal and thermal loads at cruise and a total of approximately 50 hours of steady creep were then analysed. A full elasto-plastic creep analysis was performed throughout. The objective of the analysis was to attempt to predict the stresses in the blade which would occur after creep relaxation had taken place, using the rule of thumb that this would correspond to the time taken to develop creep strains of the same order as the elastic strains. It should be noted that as a creep law with tertiary creep was used no true steady state solution was expected.

The "Hybrid Mk 0 elastic" solution method was used throughout in conjunction with the Crisfield acceleration method. The Hybrid Mk 0 elastic scheme was used as the basic modified Newton-Raphson scheme as it was felt that for a significant proportion of the second (steady state) loadcase the blade would be behaving elastically with unloading occurring from previously yielded regions. The results discussed in Chapter 9 suggested that reasonable convergence would be likely to still be obtained using the initial stiffness method with an acceleration scheme, rather than what for this size of problem would be one of the substantially more expensive hybrid schemes. As it was planned to restart the analysis at the end of the first loadcase and this would necessitate the recalculation of the stiffness matrix, it was sensible to choose the Hybrid Mk 0 elastic scheme, which would accommodate the change in elastic modulus after the loading up of the blade, rather than the initial stiffness method. The Crisfield acceleration scheme was used in preference to the BFGS method in view of its generally better performance for the creep problems described in Chapter 9. The previously used values of the control parameters (R_1 = 5.0 and R_2 = 0.65) were employed. The linesearch scheme was not used in view of its limited effectiveness in reducing the overall solution time.

The first loadcase (the loading up of the blade) was analysed using six increments, which were specified manually. The first increment applied half of the load, and the remaining increments 10% each. A total of 18 iterations were required. The Crisfield acceleration was applied on all iterations where it was available. As expected the values of a_i increased during the loadcase as the initial stiffness became a worse approximation to the true stiffness of the structure. Yielding first occurred on the fifth load increment at between 80 and 90% of full load. The largest value of

b_i/a_i encountered was 7.54×10^{-2} on the last iteration of the last increment, suggesting that the search direction is not being changed significantly by the acceleration procedure.

Results after the end of the first loadcase are shown in Figs 19 and 20. Fig 19 shows the Von Mises stress in the blade at a cross-section approximately one-third of the distance up the aerofoil. In general the regions of highest stress lie towards the centre of the aerofoil section as these are in general the coolest parts of the blade. A region of high stress is shown on the pressure and suction surfaces near the leading-edge hole. This is caused by the thermal expansion of the thin wall at the leading edge being constrained by the cold centre of the blade. The distribution of maximum principal stress over the same cross-section is shown in Fig 20. In general the maximum principal stress acts in the radial direction as would be expected due to the high centrifugal loads. The only exception is around the leading-edge hole, where the high level of constraint means that the maximum principal stress runs around the hole, leading to the peak values depicted on the inner surface. In general, however, the results show high values of principal stress in cold regions of the blade and low (or even compressive) values in hot regions. The reason for this is that the hot regions of the blade will attempt to expand radially and will be constrained and hence forced into compression by the cold core of the blade. The reverse effect will occur in cold regions and this in combination with the centrifugal load leads to the observed stress distribution.

Once the full load stress distribution had been found the blade was then allowed to creep at constant external load. In view of the good performance achieved with the test cases in Chapter 9 when θ_s equals 0.667 this value was chosen for the analysis. S_s was set to 5.0 and S_r to 0.5 thus giving approximately 10 subincrements per main step. The creep strain was updated in the strain to stress calculation, and θ_r was set to 0.667. The control value of the creep stiffness parameter $\bar{\chi}_{epc}$ was set to 0.2. The results in Chapter 9 suggested that this combination of solution parameters would provide acceptable accuracy and good convergence of the solution.

A total of approximately 50 hours of creep were analysed using the automatic timestepping scheme. A total of 101 increments were required. In all cases the allowable timestep was controlled by a Gauss point in primary creep. Thus as might be expected from the results in Chapter 9, the creep stiffness parameter was never used to reduce the size of the timestep.

Initially, as might be expected, the timestep was controlled by Gauss points at high temperature and moderately high effective stresses. After approximately three-quarters of an hour of creep however, the timestep began to be controlled by Gauss points at low temperature and with high stress, suggesting that the hot parts of the blade had relaxed onto the cooler regions.

A total of 181 iterations were required on the second loadcase. The maximum number of iterations required on any increment was seven, for the third load increment. As might be expected, there was a general reduction in the number of iterations required for each increment as the analysis proceeded. Indeed, after the eighth increment a maximum of two iterations per increment were used during the remainder of the analysis, suggesting that the choice of timestep was somewhat conservative.

The performance of the Crisfield acceleration algorithm was substantially better than it had been for the creep test problem discussed in section 9.2.6. In general the acceleration was applied on every iteration where it was available in contrast to the test problem where, for example, the acceleration was only applied on the second iteration on two increments out of 16. The only occasions when the acceleration was not applied were on the second iteration of the first increment and the third iteration of the second and third increments. In all cases the acceleration was not applied because the bounds on b_i/a_i were exceeded. On the first increment the value of b_i/a_i was slightly too large at 0.696. On the second and third increments, however, the acceleration was not applied because b_i/a_i was less than $-\frac{1}{2}R_2$. This was probably due to the substantial increase in the residual which occurred during iteration two on these two increments as the creep strain was recalculated using the subincrementation procedure. The increase in the residual will increase the magnitude of $\Delta\underline{u}_i$ compared to $\Delta\underline{u}^*_{i-1}$ and will thus potentially give negative values of b_i (*cf* equation (7-27)). A further investigation into the performance of the Crisfield acceleration was conducted by reanalysing the first three increments of the second loadcase with the values of R_1 and R_2 relaxed so that the acceleration was applied in all cases from iteration 2 onwards. This solution method required fewer iterations on increments 2 and 3, but more on increment 1. The result therefore confirms the conclusions in section 9.2.6 that the application of a quasi-Newton acceleration on iteration 2 with this formulation is not necessarily beneficial to convergence. Further work is required to determine the optimum values of the control parameters R_1 and R_2.

Results at the end of the creep period are shown in Figs 21 to 24 which depict the same aerofoil cross-section as shown in Figs 19 and 20. The Von Mises stress is shown in Fig 21. As expected the distribution of stress is much more even after creep has occurred. In particular the stress peak around the leading edge hole has completely relaxed. However, the maximum value of the Von Mises stress on the cross-section increases slightly during creep. The maximum principal stress after creep is shown in Fig 22. As with the Von Mises stress the distribution has become much more even. As might be expected the peak value has decreased due to stress relaxation. The explanation for the differing results for the peak values of the Von Mises and maximum principal stresses lies in the behaviour of the other principal stresses. At the start of creep there were significant tensile intermediate principal stresses acting in the plane of the blade cross section. During the creep period these stresses relaxed almost completely, and so by the end of the analysis the blade was virtually in a uniaxial stress state acting in the radial direction. This reduction in the intermediate principal stresses accounts for the observed increase in the peak Von Mises stress.

Figs 23 and 24 depict the creep hardening parameter $\bar{\varepsilon}_c^*$ and the maximum principal creep strain. The values for both parameters are similar, suggesting that in general creep has occurred more or less under uniaxial stress. The greatest discrepancies are observed around the leading edge hole, where the initial biaxial stress distribution appears to have caused some creep. By far the largest amount of creep however was predicted on the inside of the trailing edge hole on the suction surface, and this location agrees with one of the failure sites observed in service. Cracking is also observed in service on the inside of the leading edge hole. Although this location is predicted as a region of relatively high stress at the end of the transient stage of the analysis, relatively little creep strain is predicted to have accumulated in this region by the end of the analysis. In service the cracking is observed to run radially up and down the leading edge hole, suggesting that it is caused by stresses acting in the plane of the blade cross-section. The analysis predicts that these stresses will relax to zero relatively rapidly under steady state conditions. It is suggested therefore that the cracking in the leading edge hole is probably caused primarily by engine transients which are not accurately included in the analysis presented in this chapter. A more detailed cyclic analysis should therefore be conducted with the code to investigate this hypothesis.

Further work will be required to develop reliable damage criteria to assess blade life from the stress/strain history predicted by the code. In addition, further analysis as discussed above is required. However, the encouraging correlation between the location where the maximum creep strain accumulation was predicted and an observed in-service failure site suggests that the analysis is at least qualitatively correct.

Chapter 11

SUGGESTIONS FOR FURTHER WORK

The current version of the GANDALF code is capable of performing cyclic elasto-plastic creep analyses, including the effects of changes in temperature on material properties. As shown in chapter 10 analyses of blades can be undertaken with the current code. However during the course of the work a number of possible areas for future development have been identified and these are listed in this chapter.

The most important area for further work is in improving the material model incorporated into the code. The major deficiency in the current model is that it only caters for isotropic materials. Increasing use is being made of directionally solidified and more recently single crystal turbine blades in modern engines. Both of these materials pose difficulties for the stress analyst because they are anisotropic.

If the simplifying assumption is made that there are a reasonable number of grains across the blade cross section the directionally solidified material can be analysed as if it were orthotropic, *ie* has one principal axis of anisotropy and is isotropic in all directions orthogonal to this axis. To model the elastic behaviour of such a material simply requires the modification of the [D] matrix given in section 5.1. The form of this modification is well established and is given for example by Zienkiewicz[57]. Hill[90] developed a plasticity model for anisotropic materials by modifying Von Mises' yield criterion so that differing weight is given to each of the stress terms in the yield criterion. As might be expected only one independent constant is required when the model is applied to an orthotropic material, the ratio between the properties in the principal direction and those orthogonal to it. This model for anisotropic plasticity can also be used for creep in a similar manner to the isotropic creep model given in section 5.4 by modifying the definitions of effective stress and effective creep strain used in equations (5-83) and (5-84). However, the degree of anisotropy in the material may change during creep and this will change the values of the effective stress and strain used in the model. It thus may be difficult to establish appropriate values of the anisotropy constant from test data.

The single crystal material presents special difficulties as its axes of anisotropy the <100>, <110> and <111> crystallographic directions are not orthogonal. Indeed, relative to the principal axes of the crystal the material is actually isotropic. Nevertheless an appropriate

[D] matrix for such a material has been derived[206]. Not surprisingly in view of the three differing axes of anisotropy, three independent elastic constants are required, rather than the two in an isotropic material. To the author's knowledge however no well established model for plasticity or creep behaviour of single crystal materials is available at the time of writing, and further research would be required to develop such a formulation.

Recent American work on models of high temperature materials behaviour has concentrated on the development of so-called 'unified' theories[207-208] in which a combined model for both creep and plasticity is employed. These models are claimed to have certain computational advantages, in that for example the complexity of a yield surface is dispensed with. It is clear that an interaction between the two behaviours does exist at high temperature, with for example prior creep tending to raise the yield stress of the material. Further research would be required to investigate the magnitude of these effects and determine whether a unified theory would provide a more accurate or less computationally expensive approach than the present separate models.

The ultimate objective of the research is to develop a capability to predict the life of a given design of blade under a known duty cycle. In order to be able to do this it will be necessary to develop an appropriate damage criteria to derive the life from the predicted stress/strain history. It will be necessary to consider both creep and fatigue damage.

The simplest method of assessing creep damage is to use the accumulated effective creep stain, modified for use in a strain hardening calculation. Failure is then assumed to have occurred when the ductility of the material is exhausted. However this takes no account of the influence of state of stress on rupture strain. An alternative strategy would be the use of the continuum damage mechanics approach introduced by Kachanov[160] and Rabotnov[209] and extended to multiaxial situations by Chaboche[210]. However it is considered that it would be quite difficult to implement an implicit solution scheme for the damage parameter ω into the existing code.

The primary source of fatigue damage is the severe thermal cycling experienced by the blade during startup and shutdown. If significant inelastic strain only occurs in localized regions of the blade then these regions will be constrained by the surrounding material which remains elastic. The localized regions will thus be cycled between approximately constant strain limits. As is well known, the magnitude

of the cyclic strain range in a laboratory low cycle fatigue test is well correlated with the observed life. Hence this parameter could, in principle, be used for blade life prediction. Indeed this approach forms the basis of the strain range partitioning method[43], in which it is assumed that the various types of inelastic strain cycle contribute different amounts of damage to the material. Clearly it would be necessary to devise a system to extract the required inelastic strain ranges from the multiaxial strain history of the blade, and such a technique is described by Arvanitis *et al*[211]. The assessment of damage due to crack growth by fatigue could in principle be accomplished using fracture mechanics techniques[212], and that due to creep crack growth by the C* method[213]. Such methods are well established for the assessment of existing cracks in components. A practical difficulty arises however in the application of these methods to turbine blades. As in any cast component a large number of potential crack initiation sites will exist distributed randomly throughout the component. It is unclear (until an analysis is performed) in what areas of the blade an initial defect will grow most rapidly, and thus the use of these methods will potentially require the examination of a large number of possible failure sites.

Due to the complex geometry of a modern blade it has on occasion proved quite difficult to generate a suitable mesh using only 20 node brick elements. This problem could be overcome if a wider range of element types, particularly the 15 node wedge element, were incorporated into the code.

Currently the code does not have the capability to model pressure loading or centrifugal stiffening or unstiffening. The incorporation of pressure loading could be achieved using standard finite element techniques[61]. Centrifugal stiffening and unstiffening is caused by the changes in the magnitude and relative line of action of the centrifugal load as the blade deforms in the centrifugal field. This would be more difficult to incorporate as it would probably require the use of a large strain formulation to relate the forces and displacements to the current deformed position.

The existing frontal solution method on the Vax 11/780 and attached processor works well, and during the forward reduction phase at least, the workload is sensibly divided between the two machines. A disadvantage of the frontal solution method however, and indeed of all direct solution schemes, is the requirement to form the stiffness matrix for the entire structure and then store it after the forward reduction. As

discussed in section 7.1 and Appendix D, the process of adding each element stiffness to the structural stiffness matrix or the front array is extremely inefficient particularly on a modern computer employing a "pipeline" architecture. Thus some research has been carried out in recent years on the use of iterative solution schemes, in which the entire structural stiffness matrix is never formulated[214]. The most popular of these appears to be conjugate gradient method with element by element preconditioning, in which only the leading diagonal of the structural stiffness matrix is required. Mixed success appears to have been achieved to date using these methods for practical problems, chiefly due to poor convergence of the iterative scheme. However, they appear well suited to modern computer architectures and would seem to be worthy of some investigation for the blade lifing problem.

From the results given in chapter 9, various extensions to the code would appear to be worth investigation. The controls on the Crisfield acceleration scheme could be altered to allow the acceleration to still be applied for a larger value of a_i . The acceleration is not applied when a_i is small and hence the unaccelerated displacement increment is used, which corresponds to a_i equals 1.0 and b_i equals zero. This does not appear to the author to represent an optimum strategy. If a_i is less than unity this is presumably because the current displacement increment is in a poor direction. It would seem more sensible in this case to apply a modified form of the acceleration (in fact a reduction in the size of the displacement increment) by using the calculated value of a_i and setting b_i to zero.

The results given in chapter 9 suggest that the BFGS scheme is not particularly useful for creep problems unless linesearch is used. The major difficulty was the occurrence of negative square roots in the evaluation of $\underline{v}_i$. This problem can clearly not be overcome as it indicates that the quasi-Newton assumption is completely inappropriate in this case. However, it is suggested that it might be useful to try to devise some additional control parameters, akin to the R_1 and R_2 controls used in the Crisfield method, to determine whether or not the BFGS scheme will produce a good update to the stiffness matrix. It should be noted that a_i could be calculated before the residual is modified by the BFGS scheme, but b_i cannot be calculated as it contains $\Delta\underline{u}_i$.

The use of linesearch in conjunction with the BFGS scheme appears essential if convergence is to be achieved for creep problems. However in most cases the linesearch procedure simply produces a value of unity

for η_i after the first iteration. It would seem worthwhile therefore to investigate the possibility of dispensing with the linesearch procedure entirely after the first iteration in this case.

The current timestep control procedure using a multiple of the Cormeau timestep limit and the creep stiffness parameter appears to work well. However currently the code has no method of controlling the size of the load steps for plasticity problems and this is left to the user. A previous version of the code[172] incorporated a control based on the "current stiffness parameter" introduced by Bergan and Soriede[215]. The parameter measures the current stiffness of the structure using the load and displacement increments on the previous load step, and this information is then used to determine what the load increment should be on the current step. The implementation described in Ref 172 used a trial load increment to produce a value for the current stiffness parameter for the current step, and the load increment was then scaled when the required size had been determined. This approach was abandoned when the addition of temperature dependent material properties and an implicit solution scheme for creep problems to the code made such scaling impractical. However Bergan and Soriede's original concept could still be incorporated into the code.

Chapter 12

CONCLUDING REMARKS

The primary objective of the work described in this Thesis was to develop an optimised analysis code to predict the stresses in cooled gas turbine blades, and this has been achieved. The predictions of the code will be used to enable life assessments of these components to be undertaken. The code, known as GANDALF, incorporates elasto-plastic and creep behaviour with temperature dependent material properties, and allows analysis of any arbitrary flight cycle to be undertaken. A restart facility is also available.

In order to generate the data required for the GANDALF code, a suite of input programs has been written to interface the code to the Rolls-Royce plc blade cooling design system TACITUS. The suite enables the three dimensional finite element mesh of the blade to be generated, and an optimised element numbering for the frontal solution method used in the code to be determined. The temperature data from TACITUS can be interpolated onto the blade mesh.

A creep constitutive equation for the conventionally cast nickel superalloy Mar-M 002 has been incorporated into the code. This uses the θ' equation developed at Swansea University in conjunction with the strain hardening rule. An assessment has been made of the accuracy with which the supplied values of the constants in the θ' equation enable the original set of creep data to be regenerated. The constitutive equation appeared generally to fit the data to within the material scatter expected in creep testing. The errors appeared to be worst in primary creep. Multiaxial creep behaviour is described using the Von Mises effective stress and the normality principle. The use of the Von Mises stress appears reasonable as a series of tension/torsion creep tests gave a good correlation between this stress and rupture life.

The creep timestepping scheme in the code has been based on the well established single step θ method. The forward gradient method is used to estimate the creep rate at the end of the timestep. The effect of variations in effective stress, effective creep strain and temperature over the timestep have all been included, thus leading to a solution method which is fully consistent with the assumption of strain hardening behaviour.

Using the forward gradient method for the creep behaviour, and the isotropic hardening model for plasticity, the tangent elasto-plastic-creep

constitutive matrix has been derived. In the formulation it has proved possible to avoid the inversion of the tangent flexibility matrix inherent in earlier approaches.

A range of solution methods has been incorporated into the code. The frontal solution method is used as the linear equation solver. Elements are packed into the front using a variant of the degree of freedom lifetime technique first introduced by Yeo[178].

The availability of the tangent elasto-plastic creep constitutive matrix leads naturally to the use of the tangent stiffness solution procedure for the non-linear problem. However, as is well known, this procedure is impracticably slow, particularly for large problems and so a range of modified Newton schemes have been incorporated into the code. A quasi-Newton scheme due to Crisfield[187-190] and the well known BFGS quasi-Newton method[183-186] have also been included. A linesearch procedure is available.

As expected the number of iterations required by the various modified Newton schemes was dependent on the number of times the stiffness matrix was reformulated. Thus the initial stiffness algorithm, in which the stiffness matrix is never reformulated, required the largest number of iterations, whereas the hybrid Mk 3 algorithm, in which the stiffness matrix is reformulated on the first and second iterations of every load increment required the fewest iterations. For the elasto-plastic sphere problem described in section 9.3.1, the modified Newton method requiring the least CPU time was the Hybrid Mk 2 algorithm, in which the stiffness matrix is reformulated on the second iteration of each load increment.

The performance of the BFGS quasi-Newton scheme on the creep problem investigated was disappointing as convergence of the iterative scheme was not achieved. This is because the quasi-Newton assumption is not appropriate on the second iteration as due to the formulation used for the creep equations the tangent stiffness matrix stiffens at this point in the analysis. An alternative version of the BFGS scheme for creep problems was investigated in which the quasi-Newton update is only applied from the third iteration onwards. This converged successfully but it gave slightly less improvement in the total number of iterations required than the Crisfield scheme. The controls on the application of the Crisfield acceleration meant that convergence could be achieved for the creep problem, because the acceleration was seldom applied on the second iteration. Despite this a worthwhile reduction

in the total number of iterations required was obtained using the Crisfield acceleration. Both quasi-Newton schemes worked well for the elasto-plastic sphere problem with slightly the better results being produced by the BFGS algorithm.

The use of linesearches resulted in a significant reduction in the total number of iterations required. Indeed for the creep problem, all the algorithms tested which used linesearch required fewer iterations than the tangent stiffness method. The use of linesearch also enabled convergence of the BFGS scheme to be achieved for creep problems. However no significant reduction in CPU time was achieved using linesearch because the time saved by the reduction in the number of iterations was offset by the time required to calculate the trial solutions.

A subincrementation solution procedure has been devised to calculate the stresses and inelastic strains in the structure at the Gauss point level from the predicted total strain increment. In creep problems an updated estimate of the creep strain increment within each subincrement can be calculated using the single step θ method, and the test problems discussed in chapter 9 indicate that this gives more accurate results, albeit at the expense of additional computational effort.

The variation in total strain over the increment for creep problems can be assumed to be either linear or non-linear. The non-linear variation is controlled by the value of θ_s, the value of θ used in the displacement iteration. Perhaps not surprisingly the assumption of a non-linear variation of strain proved to be a poor one for the stress relaxation problem discussed in chapter 9. However, this is a special case, as the most important total strain increment is zero for this problem. In general the non-linear scheme gave greater accuracy and required fewer iterations than the linear one, particularly when the constant stress creep problem was run with S_s equals 5.0, the larger of the two timestep sizes investigated. It is recommended therefore that the non-linear scheme is adopted.

An assessment has been made of the most appropriate value for θ_s the value of θ to be used in the displacement iteration. If the creep strains are not updated in the strain to stress calculation then, as expected, a value of 0.0 for θ_s gives the most rapid convergence of the modified Newton iterative procedure, because in this case the tangent stiffness matrix is not modified by the inclusion of creep terms. If the creep strains are updated then a value of 0.5 for θ_s gave the best convergence for the Hybrid Mk 3 algorithm. However, for

the initial stiffness algorithm the optimum choice of θ_s was problem dependent. For the constant stress problem, where substantial changes in the tangent stiffness matrix occur during the analysis, the best convergence is provided when θ_s equals 0.0 because this makes the tangent stiffness matrix equal to the elastic (or elastic-plastic) stiffness. For the stress relaxation problem however, where little change in the tangent stiffness matrix occurs, the best convergence is provided when the assumed creep strain increment in the displacement iteration is closest to that evaluated during the strain to stress calculation. A value of 1.0 for θ_s was found to give the best convergence when a linear variation of total strain was assumed, and a value of 0.5 when a non-linear variation was used.

The effect of the value of θ_s on the accuracy of the solution has also been assessed. The stress relaxation problem appears to be a special case because the displacement increments are constrained to be zero. This means that for the updating scheme with a linear variation of total strain over the increment the accuracy of the solution is independent of the value of θ_s. If a non-linear variation in total strain over the increment is used the variation is controlled by the value of θ_s, and in this case it was found that the most accurate solution was obtained when θ_s equals zero. This is because the total strain increment is then zero at all intermediate points within the timestep. For the constant stress problem analysed with S_s equals 1.0 the most accurate of the values of θ_s investigated was 0.5 for all the solution algorithms and both updating schemes. When a larger timestep and a wider range of values of θ_s were used on this problem a value of 0.667 was found to be optimal.

It is assumed that there are two sources of error in the time-stepping scheme. The first of these is the necessity (for an implicit scheme) of estimating the creep rate at the end of the timestep using the forward gradient method. It is reasonable to assume that the importance of this error will increase as the value of θ_s is increased, as this means that a greater proportion of the creep rate at the end of the timestep is used to estimate the creep strain increment. The second source of error arises because of the potential oscillations in the solution inherent in the single step θ method. As is well known only a value of 1.0 for θ_s guarantees that no oscillation will occur, and as θ_s is reduced a progressively more oscillatory solution will be predicted, with eventually potential instability for values of θ_s less than 0.5. It is suggested that the value of 0.667 for θ_s provides

acceptable control of both of these sources of error, thus leading to the accurate solution given by the code.

It is unclear what value should be adopted for θ_r, the value of θ to be used in the subincrementation procedure. It is reasonable to suggest that in most cases the choice of this parameter is of less importance than the value of θ_s, the value of θ in the displacement iteration, because there will, in general, be less variation in creep rate over a subincrement than over the whole timestep. For the non-linear scheme, a value of θ_r equals 0.667 appears to be the optimum of those investigated, based on the good performance achieved with this value in the results for the constant stress problem with S_s equals 5.0, and the generally quite good results achieved for θ_r equals 0.5 for the constant stress and stress relaxation problems with S_s equals 1.0. For the linear subincrementation scheme, the position is more complex, as the performance of the explicit subincrementation scheme was so variable. It is suggested therefore that a value of 0.5 for θ_r is adopted as this produced good results for the stress relaxation problem, and an acceptable performance for the constant stress problem.

An investigation has been conducted into the optimum number of subincrements which should be used within each main step. It has been found that ten subincrements gives optimum accuracy and good convergence of the iterative procedure without requiring an excessive amount of CPU time. The maximum accuracy achieveable was a function of the size of the timesteps and the value of θ_s. For some combinations of θ_s and timestep size, particularly when θ_s equals 1.0 it was found not to be possible to increase the accuracy of the analysis by subincrementation.

A novel method of automatically controlling the size of the time-steps employed during a creep analysis has been devised, based on the magnitude of the change in stiffness caused by the inclusion of the creep terms in the tangent elasto-plastic-creep constitutive matrix. It has been found that the method is most advantageous during tertiary creep, because it is able to detect the additional creep which will occur due to the increasing creep rate over the projected timestep. During primary creep the timestep is controlled using a multiple of the limiting timestep for stability of an explicit solution method, given originally by Cormeau[164].

The code has been implemented on a Vax 11/780 computer with a Floating Point Systems FPS-164 attached processor. A parallel algorithm has been developed for the frontal solution, in which the element

stiffness matrices are calculated on the host Vax 11/780 at the same time as the structural stiffness matrix is reducted on the attached processor. The method is well suited to the blade problem as the times taken for these two processes on the two machines are comparable.

The results of the analysis can be assessed using the commercial interactive graphics program FEMVIEW. Data is stored in the FEMVIEW database using a preprocessor program PREVIEW. Software has been written to allow the stress analysis program to initiate a run of PREVIEW whilst executing. This enables the results of the analysis to be examined graphically whilst it is in progress.

The most important area for further work is in improving the material model incorporated into the code. Modern blade designs increasingly employ directionally solidified or single crystal materials. Both of these materials are anisotropic, and thus require special treatment in a stress analysis. Although a suitable description of the elastic behaviour of these materials exists, further research will be required to develop appropriate models of creep and plasticity behaviour. The ultimate objective of the research is to develop a method to predict the life of a given design of blade under a known duty cycle, and this will require further work to develop a suitable damage criterion.

In order to complete validation of the code an analysis of the Rolls-Royce RB211-524 "Racetrack" blade has been conducted. The loading up of the blade and approximately 50 hours of creep at the cruise condition were analysed. A full elasto-plastic creep analysis was performed throughout. The objective of the analysis was to attempt to predict the stresses in the blade which would occur after stress relaxation had taken place. The Crisfield acceleration scheme was used in conjunction with the 'Hybrid Mk 0 elastic' modified Newton algorithm in which the stiffness matrix is reformulated at the start of each loadcase as the elastic stiffness. The acceleration scheme was applied on nearly every iteration where it was available. The major creep deformation was predicted on the inside of the trailing edge hole on the suction surface, and this location is well correlated with one of the failure sites observed in service. Failure is also observed on the inside of the leading edge hole and this was not predicted by the analysis. It is felt however that failure in this region of the blade is primarily due to engine transients, which were not included in the analysis. A more detailed cyclic analysis should therefore be conducted with the code. However the correlation between

the location where the major creep deformation was predicted and an observed inservice failure site suggests that the analysis is qualitatively correct.

This encouraging result increases confidence in the algorithms and models used in the code, and suggests that it will act as a firm basis for the development of a complete life prediction system for turbine blades.

Appendix A

THE DATA INPUT FILES

This Appendix lists the data structures in two sets of input files, those used as input to GANDALF1N and those produced by GANDALF1N which are used as input to GANDALF3P.

A.1 The input files to GANDALF1N

(a) GMESHnnn.DAT containing data on the finite element mesh.

	Item	Meaning
(i)	NELZ, NODMAX, LIMFRO	Number of elements, number of nodes, maximum frontwidth.
(ii)	(LTYPE(J), J=1,NELZ)	Element type (always 1 (20 node brick) at present).
(iii)	DO 100 J=1,NELZ (TOPOL(K,J),K=1,20) 100 CONTINUE	Element topology.
(iv)	((COORD(J,K),J=1,3),K=1, NODMAX)	Nodal coordinates.

(b) MCONSnnn.DAT containing data on the nodal constraints.

	Item	Meaning
(i)	NFIX	Number of constrained nodes.
(ii)	DO 100 J=1,NFIX NODFIX(J),(KODFIX(K,J), K=1,3),(VFIX(L,J),L=1,3) 100 CONTINUE	Node number constrained, 3 flags indicating if the x,y or z degrees-of-freedom are constrained (1=fixed, 0=free), amount of displacement in x,y and z.

(c) MATERnnn.DAT containing material properties

	Item	Meaning
(i)	(EM(J),J=1,3), (ALPHA(J), J=1,3),NU,RHO	Coefficients of quadratics giving variation of Young's Modulus and expansion coefficient with temperature, Poisson's ratio, density.
(ii)	NTEMPS	Number of temperatures at which stress/plastic strain curves are defined.
	DO 200 J=1,NTEMPS	
(iii)	TEMPS(J)	Temperature at which stress/plastic strain curve defined.
(iv)	NCURPTS(J)	Number of points on curve.
	DO 100 K=1,NCURPTS(J)	

(v)	STRESSDAT(K,J), STRAINDAT(K,J) 100 CONTINUE 200 CONTINUE	Stress and plastic strain for each point on the curve (the first point is taken as the yield stress).

Creep data are passed to GANDALF3P using the subroutine CRPLAW which includes an equation of state for the creep curves and carries out the strain hardening procedure.

(d) GCYCLnnn.DAT containing load cycle data

	Item	Meaning
(i)	NLOADCASE	Number of loadcases.
(ii)	IRAD	Axis of rotation (0,1,2,3 means none, x,y or z axis respectively).
	DO 100 J=1, NLOADCASE	
(iii)	J, DELTIME	Loadcase number, duration of loadcase (secs).
(iv)	LCASENAME	Name of loadcase (20 characters).
(v)	IF(IRAD.GT.0)NEWSPEED	Flag for new speed on this loadcase (1=true, 0=false).
(vi)	IF(IRAD.GT.0.AND. NEWSPEED)REVPERSEC	Rotational speed (revolutions/second).
	100 CONTINUE	

(e) MTEMPnnn.DAT containing nodal temperature data

	Item	Meaning
	DO 100 J=1,NLOADCASE	
(i)	J,NEWTEMP IF(NEWTEMP,EQ.1)THEN	Loadcase number, flag for new temperatures (1=true, 0=false).
(ii)	(NODE,GTEMP(NODE), NODE=1,NODMAX) ENDIF	Node number, nodal temperature.
	100 CONTINUE.	

(f) MPRESnnn.DAT containing nodal load data

	Item	Meaning
(i)	NLOADMAX	Maximum number of nodes in any loadcase with applied load.
	DO 200 J=1,NLOADCASE	
(ii)	J,NEWLOAD	Loadcase number, flag for new loads on this loadcase (1=true, 0=false).

	IF(NEWLOAD.EQ.1)THEN	
(iii)	NLOAD	Number of nodes with applied load
	DO 100 K=1,NLOAD NODLOD(K),(VLOAD(L,K), L=1,3)	Number of node with load applied, amount of load in x,y and z directions.
100	CONTINUE	
	ENDIF	
200	CONTINUE	

A.2 The input files to GANDALF3P

(a) GCONTnnn.DAT containing analysis control data. All flags in this file are 'true' if equal to one and 'false' if equal to zero.

	Item	Meaning
(i)	ISRSTON,ICRST	Flag for restart analysis, flag for create restart tape.
	IF(ISRSTON.EQ.1)THEN	File number of old restart tape, flag for which data is to be used on the restart tape (1=initial, 2=current), flag for restarting with elastic stiffness matrix, flag for existence of load file.
(ii)	aaa,NFLAG,ISFLAG,ILODFLG ENDIF	
(iii)	bbb,ccc	Main data (GDATA) input file number, output restart file number.
(iv)	IF(ILODFLG.EQ.1)ddd	Input load (GLOAD) file number.
(v)	IF(ICRST.EQ.1)IPLCOUT, LINCOUT	Loadcase and load increment numbers for which data is to be written to output restart file in 'initial data' space. If zero no data is written.
(vi)	IPREV	Flag for graphics output.
(vii)	IF(IPREV.EQ.1)eee	FEMVIEW database number.
(viii)	IPLAST,ICREEP,IGEOM	Flags for plastic, creep and geometrically nonlinear analysis.
(ix)	CUTOFF,ITERMAX,NALGO, RFACT1,RFACT2,LBFGSFLG, LCRISFLG,LSEARCH	Convergence tolerance (%), maximum permitted number of iterations per increment, MNR solution type, fraction of load to be applied on first and subsequent increments of transient loadcase, flags for BFGS, Crisfield and linesearch algorithms.

(x)	IF(LSEARCH.EQ.1)ETMAX, NTIMESMAX,GTOL	Line search parameters:- Maximum value of η_i allowed, maximum number of searches, value of $G(\eta_i)/G(0)$ at which searches should be terminated.
(xi)	IF(ICREEP.EQ.1)IMPLICS, THETAS,STABFACTS,IUPDAT, IMPLICR,THETAR,STABFACTR	Creep solution parameters:- Implicit solution flag, implicit/explicit factor, Cormeau timestep factor for displacement iteration, update flag for calculation of creep strains in strain to stress routine, implicit solution flag, implicit/explicit factor, Cormeau timestep factor for strain to stress routine.
(xii)	IPLCSTOP,LINCSTOP	Loadcase and load increment number to halt analysis (if zero not used).
(xiii)	MLF	'Manual' loading flag.
(xiv)	IF(MLF.EQ.1)THEN DO 100 J=1,NLOADCASE NLEVELS(J)	Number of load levels for which solution is required.
	(FLEVELS(K),K=1,NLEVELS(J)) 100 CONTINUE ENDIF	Load levels required (negative values halt analysis at ABS(FLEVEL)).

(b) GLOADnnn.DAT containing loading data

Item		Meaning
(i)	NLOADMAX,NLOADCASE	Maximum number of nodes in any loadcase with applied load, number of loadcases in this file.
(ii)	IRAD	Axis of rotation (0,1,2,3 means none, x,y or z axes respectively).
	DO 300 J=1,NLOADCASE	
(iii)	LCASENAME	Loadcase name (20 characters).
(iv)	ITRANS,DELTIME	Transient loadcase flag (1=transient, 0=steady state). Duration of loadcase (seconds).
	IF(ITRANS.EQ.1)THEN	
(v)	NLOAD DO 100 J=1,NLOAD	Number of nodes with applied load.
(vi)	NODLOD(J),(VLOAD(K,J), K=1,3) 100 CONTINUE	Number of node with load applied, amount of load in x,y and z directions.
	DO 200 J=1,NELZ	

(vii)	(TEMP(K),K=1,20) 200 CONTINUE ENDIF 300 CONTINUE	Temperatures of nodes in element by element format.

(c) GDATAnnn.DAT containing geometry, material and constraint data.

	Item	Meaning
(i)	NELZ,NODMAX,NFIX,LIMFRO	Number of elements, number of nodes, number of constrained nodes, maximum frontwidth during reduction.
(ii)	(EM(J),J=1,3),(ALPHA(J),J=1,3),NU,RHO	Coefficients of quadratics giving variation of Young's Modulus and expansion coefficient with temperature, Poisson's ratio, density.
(iii)	IPLASTD,ICREEPD	Flags for plasticity and creep data.
	IF(IPLASTD.EQ.1)THEN	
(iv)	NTEMPS	Number of temperatures at which stress/plastic strain curves are defined.
	DO 100 J=1,NTEMPS	
(v)	TEMPS(J)	Temperature at which stress/plastic strain curve is defined.
(vi)	NCURPTS(J)	Number of points on curve.
(vii)	(STRESSDAT(K,J),STRAINDAT(K,J),K=1,NCURPTS(J)) 100 CONTINUE END IF DO 200 J=1,NFIX	Stress and plastic strain for each point on the curve (the first point is taken as the yield stress).
(viii)	NODFIX(J),(KODFIX(K,J),K=1,3),(VFIX(K,J),K=1,3) 200 CONTINUE	Node number constrained, 3 flags indicating if the x,y or z degrees-of-freedom are constrained (1=fixed, 0=free), amount of displacement in x,y and z.
(ix)	(LTYPE(J),J=1,NELZ)	Element type (always 1 (20 node brick) at present).
(x)	((TOPOL(I,J),I=1,20),J=1,NELZ) DO 300 J=1,NELZ	Element topology.
(xi)	((XYZ(K,L),K=1,3),L=1,20) 300 CONTINUE	Coordinate data in element by element format.

Appendix B

THE INTERFACE TO THE GRAPHICS PACKAGE

B.1 Introduction

One of the chief difficulties in using a large finite element scheme is the interpretation of the analysis results. A run of GANDALF can be expected to generate up to five million numbers as output and therefore graphical methods of assessing the data are essential. A number of software packages are available commercially for this task and after a survey of the market, it was decided to procure FEMVIEW. This package had the considerable advantage of making full use of interactive graphics techniques to specify views of the structure, rather than requiring the use of keyboard commands, and also two features which were unique, the ability to contour on an arbitrary cross-section through the structure, and a facility for removing 'hidden lines' from the idealisation.

The FEMVIEW system consists of two programs. The first, PREVIEW, takes as input a character file produced by the stress analysis program, and loads the data in it into two direct access files. One of these contains the actual results data and the second contains a set of pointers to the data in the first file. These two files will be referred to collectively as the database. The second program, FEMVIEW, is run interactively on a Tektronix 4014 terminal and enables the user to assess the results graphically.

The normal mode of operation of this system for a small problem would be to run the analysis program to produce the character file, run PREVIEW and then assess the results with FEMVIEW. For a lengthy analysis such as GANDALF this approach suffers from the disadvantage that it is impossible to look at the results as the program is running. In order to allow 'on-line' monitoring of the analysis the PREVIEW and FEMVIEW codes were modified to incorporate a number of VAX/VMS operating system routines. These modifications are described in this Appendix.

The first requirement was to load the geometry and topology of the mesh into the database so that they could be examined before running the stress analysis. This required the development of a link between the stress analysis input program GANDALF1N and the graphics software.

The second problem was that PREVIEW and a FEMVIEW user might both require simultaneous access to the database, with PREVIEW adding additional analysis results, and the user assessing the current

information. Multiple access to a single file can be controlled on a record by record basis. However, this approach is inappropriate when, as in the present situation there are two files involved with one acting as an index to the data in the other. This is because it is essential to ensure that the index always exactly reflects the contents of the data file, even though this is being augmented. The technique adopted regards both files as a single resource and only allows one program to access it at any one time.

The final consideration was the requirement for the stress analysis program to continue running when the user is examining the results in the database using FEMVIEW.

In order to satisfy these requirements the structure shown in Fig 25 was developed. The system utilises two features of the VAX/VMS operating system. The first of these is known as the create process system service. When a user logs into the machine a process is created. During a typical terminal session the user will then run a number of images such as an editor, the compiler or an application program. The create process system service provides a method of creating another process from a program and also of specifying the image which is to be executed by the new process.

The second feature of the operating system which is used is known as a common event flag cluster. This is a group of bit flags which as the name implies are common to more than one image and hence to more than one process. A group of cooperating processes can manipulate the flags and hence control the execution of the various images.

In order to use a common event flag cluster a process must first associate with it. If the specified cluster does not exist already, the operating system will create it. A process can then set or clear specific flags, test their current values and wait pending the setting of a particular flag or group of flags. A list of the event flags used by the system and their functions is shown in Fig 26.

B.2 The link between GANDALF1N and the graphics software

As previously described, the main function of GANDALF1N is to create the various input files for the analysis. These are shown as GDATAnnn.DAT, GLOADnnn.DAT and GCONTnnn.DAT in Fig 5 , where 'nnn' is the analysis number. If a new FEMVIEW database is to be created the program also creates two other files which are used by the graphics interface. The first of these is the character file input to PREVIEW, called G1RUNnnn.DAT. This contains the mesh geometry and

topology. The second file is called RUNnnn.NM1. This contains two items of information, the name of the input file to PREVIEW, *ie* G1RUNnnn.DAT, and the database name. This corresponds to what would normally be typed at a terminal in response to prompts from PREVIEW when running the program interactively. The create process system service is used to create another process to run PREVIEW. This process is called PREVIEW1. Part of the system service call assigns the RUNnnn.NM1 file so that the PREVIEW image uses it as though it were terminal input.

Because the input file to this run of PREVIEW contains mesh geometry and topology PREVIEW will automatically create a new database. Clearly it is essential to ensure that this task is complete before PREVIEW3 (the process created by GANDALF3P to run PREVIEW) attempts to add results to it. This is done by using three event flags, numbers 71, 73 and 74.

Flag 73 when set forces GANDALF3P to wait until the database has been created. If it is clear it is assumed that this has been done and execution continues immediately. The flag is set by GANDALF1N before it creates the process to run PREVIEW, thus ensuring that GANDALF3P will be blocked before PREVIEW starts creating the database. When a cluster is created all the flags in it are initially clear and the cluster will be deleted when the last image associated with it terminates. If GANDALF1N was to terminate before PREVIEW1 had associated with the cluster, the first cluster would be deleted and PREVIEW1 would be associating with a new one. Thus the fact that flag 73 had been set would be lost. To overcome this problem, after GANDALF1N has started PREVIEW1 it waits for flag 71 to be set. This is done by PREVIEW1. Obviously in order to set flag 71 PREVIEW1 will have had to associate with the cluster and so the fact that flag 73 is set will be passed on correctly.

Once PREVIEW1 has completed the database the final task is to restart the analysis program. Flag 74 is used for this purpose and it is set by PREVIEW1 to signal to GANDALF3P that the analysis can continue.

B.3 Accessing the graphics database

The system described in section B.2 ensures that it is not possible for PREVIEW3 and PREVIEW1 to be attempting to access the database simultaneously, as the analysis, which creates PREVIEW3, cannot proceed until PREVIEW1 has finished. However, steps must be taken to control access by one of the PREVIEW processes and possibly a number of processes running FEMVIEW. The control of multiple users wishing to access a

single resource, in this case the graphics database, is a classic real time computing problem. It arises for example in airline booking systems where a number of travel agents simultaneously wish to access information on seat availability.

What is required is a technique to allow each process to determine whether the resource is available and if it is to acquire it before another process makes it unavailable. This can be achieved through the use of an event flag as a mutual exclusion semaphore or lock. The flag is originally set, and the various competing processes try and clear it. Only one process will be able to clear the flag from the set position, as once this has been done all other processes will find that the flag was clear. The first process then proceeds to access the resource, and the other processes wait for the flag to be set again. This is done by the first process when it has finished accessing the resource. The competition to clear the event flag then begins again.

In the GANDALF implementation of this system flag 66 is used as the lock. It is clearly essential to ensure that this flag is actually set before the competition to clear it begins. To do this two other flags numbers 64 and 65 are used. Each process first sets flag 64. If it was originally clear then the process is the first one to execute (because when a cluster is created all the flags are initially clear). It therefore sets the lock flag 66 and then sets flag 65 to signal that the lock has been set up. If the process was not the first one to execute, *ie* it found that flag 64 had already been set, then it merely waits for flag 65 to be set. This will already have happened unless two processes are executing almost simultaneously. When flag 65 has been set flag 66, the lock, will also be set ready for use and the competition to access the database can begin.

B.4 The link between GANDALF3P and the database

The interface to the graphics software from GANDALF3P is implemented as follows. GANDALF3P creates a process called PREVIEW3 to run PREVIEW which loads the analysis results into the database. PREVIEW3 may have to wait whilst the database is used by FEMVIEW. Clearly it is undesirable in such a situation to force GANDALF3P to wait as well. This is avoided by using 'double buffering' between GANDALF3P and PREVIEW3. Two files of results are employed, with GANDALF3P writing into one file whilst PREVIEW3 is reading the contents of the other. The files are swopped only when GANDALF3P is able to write to the file which it is not currently using, and therefore the analysis is never held up.

Two event flags for each file are needed to implement this system. Flags 67 and 68 respectively allow GANDALF3P to write to, and PREVIEW3 to read from file one, and flags 69 and 70 perform the same functions for the second file. Each process is only given access to a particular file by the other process and then only when that process has completed I/O to or from the file. This ensures that it is impossible for both processes to be attempting to access the same file simultaneously.

A third file RUNnnn.NM3 is used to pass the name of the results files and the database file to PREVIEW3. This corresponds to terminal input when PREVIEW is running interactively.

Appendix C

VALIDATION OF THE GANDALF CREEP MODEL

C.1 Uniaxial behaviour

As stated in section 5.4 the creep equation used in the GANDALF code is the so called θ' equation. It will be recalled that this takes the form

$$\varepsilon_c = \theta_1\left(1 - e^{-\theta_2 t}\right) + \theta_3\left(e^{\theta_4 t} - 1\right) , \qquad (C\text{-}1)$$

where the terms θ_i are given by

$$\ln \theta_i = a_i + b_i T + c_i \sigma + d_i \sigma T . \qquad (C\text{-}2)$$

In order to generate the constants in these equations a set of 63 constant stress creep tests were performed at Swansea University under the supervision of Professor B.Wilshire. Both conventionally cast and directionally solidified Mar M-002 material was used. The DS material was cast as solid and hollow specimens, with the axis of the specimen in the <100> direction. All the conventionally cast specimens were hollow. The wall thickness of the hollow specimens was 1.05 mm. All the test pieces were heat treated to simulate the blade manufacturing process.

Creep rupture maps for the test series are shown in Figs 27 and 28 and a list of tests is given in Tables 18 and 19. These demonstrate the clear superiority of the directionally solidified material with life being increased by a factor of about 2½ over the conventionally cast alloy. The effect is particularly marked in the higher temperature tests. There is some evidence of a size effect in the directionally solidified material, as the solid specimen generally have lives slightly greater than the hollow specimens.

The supplied values of the constants in the θ' equation were obtained using the procedure documented in Ref 216. This involves performing a nonlinear least squares procedure on each creep curve to calculate θ_1, θ_2, θ_3 and θ_4, and then carrying out a linear least squares procedure using the θ_i parameters from all the curves to determine a_i, b_i, c_i and d_i for each θ_i.

In order to examine the accuracy of the resulting creep equation comparisons were made between the predictions of equation (C-1) and the

experimental data. For the conventionally cast tests the average ratio between the measured and predicted strains taken over all the data points varied from test to test between 1.36 and 8.44. It is probable that these figures give a somewhat pessimistic impression as large values of measured/predicted strain ratio were sometimes obtained early in primary creep after only a few seconds of the test. The corresponding range for the directionally solidified material was 1.44 to 5.78. Typical figures for the variation in time to a given strain from test to test under the same nominal conditions range from a factor of 4 in primary creep to about 1.6 in tertiary creep. If these are translated into errors in strain at a given time assuming a $t^{1/3}$ creep strain/time relationship in primary and a t^{3} relationship in tertiary, then these correspond to ratios of 1.6 in primary creep and 4 in tertiary. It can be seen therefore that the errors expected in the data due to testing variability are of the same order as differences between the measurements obtained from the test programme and the predictions of equation (C-1). It seems reasonable to assume therefore that the equation represents the material behaviour adequately.

C.2 Multiaxial behaviour

As stated in section 5.4 the multiaxial creep behaviour of the blade material is related to the unaxial behaviour by an equation of the form

$$\underline{\dot{\varepsilon}}_c = \dot{\bar{\varepsilon}}_c \frac{\partial \bar{\sigma}}{\partial \underline{\sigma}} \qquad \text{(C-3)}$$

and $\bar{\sigma}$ is assumed to be equal to the Von Mises stress. To establish the validity of equation (C-3) a series of tension/torsion creep tests was performed on conventionally cast Mar-M 002 at the National Engineering Laboratory under the supervision of Dr J. Henderson. A total of 29 tests were performed, together with two pure tension tests on solid specimens. Tests 1-24 and 25-31 were on two different batches of material, although no discernible difference could be detected in the results. The tension/torsion test pieces were machined from solid and had an inside diameter of 6.35 mm, an outside diameter of 8.25 mm and a gauge length of 51 mm. The results are listed in Table 20 and a plot of Von Mises effective stress against rupture life is given in Fig 29. As can be seen, apart from one or two results which displayed abnormally low lives, possibly due to defects in the castings, a reasonable correlation is obtained. In contrast in Fig 30, which depicts the rupture lives in terms of maximum principal stress the correlation is very poor. It is clear therefore that the Von Mises stress is the appropriate parameter

to correlate rupture life for this material. It is interesting to note that no size effect was observed, probably due to the fact that the specimens were machined from solid rather than cast as hollow tubes.

It was hoped to measure axial and shear creep strains during the testing and hence to validate equation (C-3) directly. Unfortunately the quality of the measurements obtained was very poor and it was impossible to draw any definitive conclusions from them. If equation (C-3) is valid then it would be expected that the ratio between axial and shear creep strain rates would be $\sigma/3\tau$ where σ and τ are the tensile and shear stresses respectively. Reasonable agreement was obtained with this ratio on most tests for times greater than half the rupture life; however for low times the measured ratios exceeded those predicted. It is not known whether this is a real effect or due to experimental error.

The only conclusion which can be drawn at this stage is that there is currently no evidence to disprove the validity of equation (C-3). Indeed it can be argued that the correlation obtained between Von Mises stress and rupture life suggests that it is at least a reasonable approximation. It is probable that the correct criterion is more complex, such as that proposed by Hayhurst[217], which includes the Von Mises, hydrostatic and maximum principal stresses. However the establishment of such a theory will depend on the availability of sufficient experimental data of adequate quality.

C.3 Calculation of the strain hardening constants

It will be recalled that in section 6.2 it is shown that the use of strain hardening theory in conjunction with an implicit creep algorithm leads to an expression for the creep strain increment over the timestep of the form (*cf* equation (6-14))

$$d\bar{\varepsilon}_c = \left[\dot{\bar{\varepsilon}}_{c,t} + \theta\left(\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}}\right)d\bar{\sigma} + \theta\left(\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\varepsilon}_c}\right)d\bar{\varepsilon}_c + \theta\left(\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial T}\right)dT\right]dt \quad . \qquad (C\text{-}4)$$

It is thus necessary to calculate the values of the partial derivatives in the above expression. If the strain hardening procedure is used, a creep pseudo time t^* is found for which (*cf* equation (5-79))

$$\bar{\varepsilon}_c\left(\sigma_2, T_2, t^*\right) = \bar{\varepsilon}_c \quad . \qquad (C\text{-}5)$$

Thus, if the θ' equation is used

$$\bar{\varepsilon}_c\left(\theta_1,\theta_2,\theta_3,\theta_4,t^*\right) = \bar{\varepsilon}_c \tag{C-6}$$

and hence

$$\dot{\bar{\varepsilon}}_c = \dot{\bar{\varepsilon}}_c\left(\theta_1,\theta_2,\theta_3,\theta_4,t^*\right) . \tag{C-7}$$

Thus, by the chain rule

$$\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\bar{\sigma}} = \frac{\partial\theta_1}{\partial\bar{\sigma}}\cdot\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\theta_1} + \frac{\partial\theta_2}{\partial\bar{\sigma}}\cdot\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\theta_2} + \frac{\partial\theta_3}{\partial\bar{\sigma}}\cdot\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\theta_3} + \frac{\partial\theta_4}{\partial\bar{\sigma}}\cdot\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial\theta_4} + \frac{\partial t^*}{\partial\bar{\sigma}}\cdot\frac{\partial\dot{\bar{\varepsilon}}_c}{\partial t^*} . \tag{C-8}$$

Expanding equation (C-6)

$$\bar{\varepsilon}_c = \theta_1\left(1 - e^{-\theta_2 t^*}\right) + \theta_3\left(e^{\theta_4 t^*} - 1\right) . \tag{C-9}$$

Differentiating twice with respect to t^*

$$\frac{\partial^2\bar{\varepsilon}_c}{\partial t^{*2}} \equiv \frac{\partial\dot{\bar{\varepsilon}}_c}{\partial t^*} = -\theta_1\theta_2^2 e^{-\theta_2 t^*} + \theta_3\theta_4^2 e^{\theta_4 t^*} . \tag{C-10}$$

Differentiating equation (C-9) with respect to $\bar{\sigma}$

$$\frac{\partial\bar{\varepsilon}_c}{\partial\bar{\sigma}} = \frac{\partial\theta_1}{\partial\bar{\sigma}}\left(1 - e^{-\theta_2 t^*}\right) + \theta_1 t^*\frac{\partial\theta_2}{\partial\bar{\sigma}} e^{-\theta_2 t}$$

$$+ \frac{\partial\theta_3}{\partial\bar{\sigma}}\left(e^{\theta_4 t^*} - 1\right) + \theta_3 t^*\frac{\partial\theta_4}{\partial\bar{\sigma}} e^{\theta_4 t^*} + \dot{\bar{\varepsilon}}_c\frac{\partial t^*}{\partial\bar{\sigma}} . \tag{C-11}$$

But, because strain hardening theory is being used $\partial\bar{\varepsilon}_c/\partial\bar{\sigma} = 0$. Hence

$$\frac{\partial t^*}{\partial\bar{\sigma}} = \frac{1}{\dot{\bar{\varepsilon}}_c}\left\{\left[\frac{\partial\theta_1}{\partial\bar{\sigma}} - \frac{\partial\theta_2}{\partial\bar{\sigma}}\theta_1 t^*\right] e^{-\theta_2 t^*} - \left[\frac{\partial\theta_3}{\partial\bar{\sigma}} + \frac{\partial\theta_4}{\partial\bar{\sigma}}\theta_3 t^*\right] e^{\theta_4 t^*}\right.$$

$$\left. - \frac{\partial\theta_1}{\partial\bar{\sigma}} + \frac{\partial\theta_3}{\partial\bar{\sigma}}\right\} . \tag{C-12}$$

Now

$$\frac{\partial \bar{\varepsilon}_c}{\partial t^*} = \theta_1\theta_2 e^{-\theta_2 t^*} + \theta_3\theta_4 e^{\theta_4 t^*} \qquad \text{(C-13)}$$

and hence

$$\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \theta_1} = \theta_2 e^{-\theta_2 t^*} , \qquad \text{etc.} \qquad \text{(C-14)}$$

Thus, substituting equations (C-12) and (C-14) into equation (C-8)

$$\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}} = \left[\frac{\partial \theta_1}{\partial \bar{\sigma}} \theta_2 + \frac{\partial \theta_2}{\partial \bar{\sigma}} \theta_1 \left(1 - \theta_2 t^*\right) + A\left(\frac{\partial \theta_1}{\partial \bar{\sigma}} - \frac{\partial \theta_2}{\partial \bar{\sigma}} \theta_1 t^*\right) \right] e^{-\theta_2 t^*}$$

$$+ \left[\frac{\partial \theta_3}{\partial \bar{\sigma}} \theta_4 + \frac{\partial \theta_4}{\partial \bar{\sigma}} \theta_3 \left(1 + \theta_4 t^*\right) - A\left(\frac{\partial \theta_3}{\partial \bar{\sigma}} + \frac{\partial \theta_4}{\partial \bar{\sigma}} \theta_3 t^*\right) \right] e^{\theta_4 t^*}$$

$$+ A\left[\frac{\partial \theta_3}{\partial \bar{\sigma}} - \frac{\partial \theta_1}{\partial \bar{\sigma}} \right] ,$$

where $$A \equiv \ddot{\bar{\varepsilon}}_c / \dot{\bar{\varepsilon}}_c . \qquad \text{(C-15)}$$

Equation (C-15) thus gives the first of the required partial differentials for equation (C-4). The $\partial\theta_i/\partial\bar{\sigma}$ can be obtained directly from equation (C-2). It is easy to see that by following a similar procedure but differentiating with respect to temperature rather than stress

$$\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial T} = \left[\frac{\partial \theta_1}{\partial T} \theta_2 + \frac{\partial \theta_2}{\partial T} \theta_1 \left(1 - \theta_2 t^*\right) + A\left(\frac{\partial \theta_1}{\partial T} - \frac{\partial \theta_2}{\partial T} \theta_1 t^*\right) \right] e^{-\theta_2 t^*}$$

$$+ \left[\frac{\partial \theta_3}{\partial T} \theta_4 + \frac{\partial \theta_4}{\partial T} \theta_3 \left(1 + \theta_4 t^*\right) - A\left(\frac{\partial \theta_3}{\partial T} + \frac{\partial \theta_4}{\partial T} \theta_3 t^*\right) \right] e^{\theta_4 t^*}$$

$$+ A\left[\frac{\partial \theta_3}{\partial T} - \frac{\partial \theta_1}{\partial T} \right] . \qquad \text{(C-16)}$$

The final partial differential $\partial\dot{\bar{\varepsilon}}_c/\partial\bar{\varepsilon}_c$ can be obtained very simply because the θ_i are not functions of $\bar{\varepsilon}_c$. Hence

$$\frac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\varepsilon}_c} = \frac{\partial t^*}{\partial \bar{\varepsilon}_c} \cdot \frac{\partial \dot{\bar{\varepsilon}}_c}{\partial t^*} = A \quad . \tag{C-17}$$

C.4 Control of the time stepping scheme for low stresses

As stated in section 5.4 the stable timestep for an explicit creep time stepping scheme is given by Cormeau (equation (5-92)) as

$$\Delta t \leqslant \frac{4(1+\nu)}{3E} \cdot \min\left(\frac{\bar{\sigma}}{\dot{\bar{\varepsilon}}_c}, \frac{1}{\partial \dot{\bar{\varepsilon}}_c / \partial \bar{\sigma}}\right) \quad . \tag{C-18}$$

It is easy to see (for example by considering Norton's power law formula for creep) that in general the second of these criteria is the more stringent and indeed this is observed in practice. For very low stresses however it is found that the timestep given by equation (C-18) tends to zero. This is because the θ' equation will produce a small positive value for $\dot{\bar{\varepsilon}}_c$ at zero stress, and hence $\bar{\sigma}/\dot{\bar{\varepsilon}}_c$ tends to zero rather than infinity as the stress is reduced. Thus the timestep chosen by the analysis is controlled by the least highly loaded Gauss points. This is clearly physically absurd as the allowable timestep should continuously increase as the stress is reduced. To overcome this difficulty the code evaluates the differential

$$\frac{\partial \Delta t}{\partial \bar{\sigma}} = \frac{\dot{\bar{\varepsilon}}_c - \bar{\sigma} \dfrac{\partial \dot{\bar{\varepsilon}}_c}{\partial \bar{\sigma}}}{\left(\dot{\bar{\varepsilon}}_c\right)^2} \quad . \tag{C-19}$$

For physically realistic situations the differential should always be negative. Creep is automatically switched off for the current time-step for Gauss points where a positive value is encountered.

Appendix D

IMPLEMENTATION OF THE FRONTAL SOLUTION PROCEDURE ON THE VAX 11/780/FPS-164 ATTACHED PROCESSOR COMPUTER SYSTEM

D.1 Introduction

The frontal solution procedure for finite element equations is described in detail in section 7.1. This Appendix describes how the algorithm has been implemented on a Floating Point Systems FPS-164 attached processor (AP) with a VAX 11/780 host machine. In essence the approach used is to calculate the element stiffness matrices on the host and reduce the resultant equations on the AP.

The AP does not have its own disc drive or the FPS SJE (Single Job Executive) software. Therefore the AP cannot operate 'stand alone' and must be accessed by the host sending code and data to it. The AP program is then executed and the results are retrieved by the host. The host and AP can operate in parallel and the host can perform I/O to the AP whilst the latter is executing. The hardware connection between the FPS-164 and the host is a Unibus link.

The FPS-164 processor hardware consists of a two-stage pipelined adder and a three-stage pipelined multiplier; both units being 64 bits wide. The clock rate is 5.5 MHz and thus as both arithmetic units can operate in parallel the maximum possible computational speed is 11 Mflops. The system used for the work described in this Thesis has 5.5 Mbytes of memory on the VAX 11/780 and 12.5 Mbytes of memory on the FPS-164.

D.2 Design of algorithm

The frontal solution as implemented on a serial machine is shown in Fig 31 . The forward reduction phase consists of a loop through the elements in the structure. The stiffness matrix for each element is evaluated and it is then added into the front. Degrees of freedom in the front which do not appear in any subsequent elements can then be reduced. In principle therefore the equations corresponding to any number from zero to the total number of degrees of freedom in the current element can be reduced and written to backing storage. Once the reduction of all the degrees of freedom is complete, the unknown displacements are found by backsubstitution.

The first task in deciding how to implement the algorithm on the AP was to estimate the time that would be taken for each part of the solution. It was essential to store the reduced structural stiffness matrix on the host due to the lack of a disc on the AP. It was assumed that the effective transfer rate between the VAX and AP would be

500 kbytes/s and that a realistic computational speed for the AP would be 3Mflops. Both these values were confirmed by benchmark tests.

The size of analysis used in the initial assessment involved 500 20-node brick elements with a total of 9000 degrees-of-freedom and a maximum frontwidth of 700. These figures produced the time estimates shown below.

Time estimates in seconds

Forward reduction	
Fomulation of one element stiffness matrix on VAX 11/780	1.72
Reduction of 1 degree-of-freedom from the front on VAX 11/780	3.12
Transfer of element stiffness matrix to AP	0.029
Reduction of 1 degree-of-freedom from the front on AP	0.163
Retrieval of one pivotal row from AP	0.011
Backsubstitution	
Backsubstitution of 1 degree-of-freedom on VAX 11/780	0.017
Transfer of one equation to AP	0.011
Backsubstitution of 1 degree-of-freedom on AP	4.7×10^{-4}

It was assumed that the assembly of the stiffness matrix into the front and the various 'housekeeping' activities associated with the frontal solution would consume negligible time when compared to the actual reduction. For the size of problem under consideration an average of 18 degrees-of-freedom are reduced from the front for each element addition. Taking this average figure it was estimated that to reduce one element on the AP would take 2.93 seconds. In addition it would require 0.03 seconds to transfer the stiffness matrix to the AP and 0.20 seconds to retrieve the results giving a total of 3.16 seconds. As the calculation of an element stiffness matrix on the host took only 1.72 seconds it was clear that it would be possible to arrange for the host machine to calculate the element stiffness matrices in parallel with reduction on the AP. There would thus be no advantage in transferring the calculation of the element stiffness matrices to the AP. Indeed this would

degrade the performance of the algorithm, particularly as this code is relatively complex and would be difficult to vectorise for the AP. The estimated solution time per element using the AP appeared to contrast very favourably with the host machine time of 56 seconds.

For the backsubstitution phase the situation was less clear. However this phase requires much less CPU time than the reduction and is thus much less important. If this task is performed on the AP it would obviously be heavily I/O bound. In addition the estimate of the time taken to transfer the data to the AP was similar to the time taken to perform the backsubstitution on the VAX. It thus appeared that there would be little benefit in performing this part of the calculation on the AP.

As a result of this initial assessment the algorithm design shown in Fig 32 was devised. Each element stiffness matrix is created in the host and is then sent to the AP. The front array resides in the AP, and the AP assembles each stiffness matrix into it. The AP then reduces the equations. The reduced equations are retrieved from the AP by the host, which then writes them to a host disc file. The host then performs the backsubstitution.

Two sets of buffers are provided on the AP, one for element stiffness matrices and one for reduced equations. The buffers must be situated on the AP as only the host can initiate data transfer between the two machines, and thus a buffer on the host could not be accessed by the AP. The buffers are provided in order to overlap AP execution with I/O between the host and AP. The number of degrees of freedom which can be reduced from the front per element addition varies during the reduction as does the number of degrees of freedom in the front. Thus the ratio between the AP and host CPU times required per element also varies. Therefore the element buffers are used to provide a queue of element stiffness matrices for the AP, to enable it to keep running even when the front contains only a few degrees of freedom, or only a small number of degrees of freedom per element are currently being reduced from the front. Adequate performance has been obtained from the present code which has 20 stiffness matrix buffers and 360 buffers for reduced equations. The minimum acceptable number of buffers has yet to be determined, but will obviously be dependent on the topology of the particular structure being analysed. Access to each buffer is controlled by an associated flag which is used to provide the necessary 'hand shaking' between the host and AP.

The main part of the host code consists of a loop through all the elements in the structure. On each pass through the loop the first task is to calculate the stiffness matrix for the current element. The code then sends the stiffness matrix to the AP if a buffer is available. If no buffer is available the code attempts to clear out reduced equations from the AP before trying to send the stiffness matrix to the AP again. Once the stiffness matrix has been sent, a further attempt is made to clear out reduced equations from the AP. The objective is to ensure that under normal conditions, when not all the element buffers are full, one clearance of the equation buffers occurs for every stiffness matrix sent to the AP. Once all the element stiffness matrices have been sent to the AP the host code then retrieves the remaining reduced equations from the AP, The back-substitution is then performed on the host.

The AP code also consists of a loop through all the elements in the structure. The code initially waits for the next stiffness matrix to become available in the buffer. The stiffness matrix is then added to the front. The code then reduces the degrees of freedom which can be eliminated from the front after the current element addition. Before each degree of freedom is reduced the code checks the appropriate buffer flag to ensure that a results buffer is available. Once all the degrees of freedom have been reduced AP execution terminates.

D.3 Implementation of the code on the attached processor

Code for the FPS-164 can be written in three ways. At the simplest level a FORTRAN compiler is available. Secondly a library of FORTRAN callable subroutines is provided which perform operations such as matrix multiply, vector add, etc. These are designed to use the pipe-lined architecture of the AP CPU efficiently. Although it is possible to program the machine in the assembler language APAL, this option has not been used in the present work. It was felt that the use of the maths library would provide an acceptable compromise between execution speed and development effort.

In order to overlap execution on the host and AP the user directed call (UDC) method was used to transfer data between the host and AP. This requires the specification of AP main data addresses. These were obtained from the AP link map and used as FORTRAN constants in the host code. Unfortunately if the AP compiler is used at an optimisation level higher than zero the AP code which causes the AP to wait for a buffer to become available does not execute correctly. This is almost certainly because the compiler assigns the buffer flag to an AP register where it can no longer be accessed by the host. In order to overcome this

deficiency a separate AP subroutine was written to wait for a buffer. This was compiled at optimisation level zero, whilst the remainder of the code was compiled at level 2.

Examination of the code disclosed three areas where it might be possible to use the maths library.

(i) The assembly of the stiffness matrix into the front.
(ii) Reduction of the structural force vector.
(iii) Reduction of the front.

It proved impossible to find a way of using the maths library routines to perform the assembly of the stiffness matrix into the front. The difficulty arises from the fact that the terms in the element stiffness matrix are added into essentially random locations in the front. All the maths library routines require that arrays are sequentially accessed (or at least accessed in equal steps) and are thus not appropriate for this type of calculation.

The reduction of the structural force vector can be represented by the following FORTRAN statements:

```
      FACT=EQLOD/PIVOT
      DO 10 J=FROSTART,FROFINISH
      SUBLOD(J)=SUBLOD(J)-EQ(J)*FACT
10    CONTINUE
```

where FROSTART and FROFINISH are integer variables which are the current start and end of the front, SUBLOD is the structural force vector and EQ the current pivotal row. EQLOD is the force corresponding to the current pivotal row. The VSMA (Vector Scalar Multiply and Add) library routine performs the following vector operation:

```
D(M)=A(M)*B+C(M) for M = 1 to N.
```

The routine is accessed using a statement of the form

```
CALL VSMA(A,I,B,C,K,D,L,N)
```

where I, K and L are the element steps for the vectors A, C and D respectively. Thus the above FORTRAN loop can easily be rewritten to use the maths library in the following manner:

```
FACT=-EQLOD/PIVOT
LENFRO=FROFINISH-FROSTART+1
CALL VSMA(EQ(FROSTART),1,FACT,SUBLOD(FROSTART),1,
SUBLOD (FROSTART),1,LENFRO)
```

The next and most important task was to use the Maths Library for the frontal reduction itself. The initial FORTRAN code was as follows:

```
      DO 20 I=FROSTART,FROFINISH
      FACT=EQ(I)/PIVOT
      IF(FACT.NE.0.0)THEN
      IPOS=I*(I+1)/2
      SIGDIG(I)=SIGDIG(I)+SUSTIF(IPOS)*SUSTIF(IPOS)
      JDEL=I*(I-1)/2
      JA=JDEL+FROSTART
      JZ=JDEL+I
      DO 10 J=JA,JZ
10    SUSTIF(J)=SUSTIF(J)-FACT*EQ(J-JDEL)
      ENDIF
20    CONTINUE
```

In the above code SUSTIF is the front vector and the DO 20 loop moves along the pivotal row and the DO 10 loop down each column. SIGDIG contains the squares of terms on the diagonal of the front, summed every time a degree of freedom is reduced. This is used later to determine the degree of roundoff in the solution.

The first task was to determine how much of the array index calculations could be removed from the loops. It is clear that IPOS and JDEL can be calculated once for every possible value of I between 1 and the maximum number of degrees of freedom in the front. The results of these calculations are stored in IPOSVEC and JDELVEC, respectively. The calculation of JA can then be vectorised as it simply involves the addition of the current value of FROSTART to every term in JDELVEC. The DO 10 loop can clearly be rewritten in a similar manner to the force vector reduction using the VSMA routine. This requires the length of each vector to be specified as an argument. The length of the Ith vector is given by JZ - JA + 1 *ie* I + 1 - FROSTART. This calculation can also be vectorised. The final improvement was to vectorise the division of the pivotal row by the pivot. This led to the code shown below:

```
      C
      C  Calculate JAVEC (starting position of each
      C     column)
      C
           CALL VIADD(JDELVEC(FROSTART),1,FROSTART,0,JAVEC(FROSTART),1,
           LENFRO)
```

```
C
C  Calculate LENCOL (length of each column = I + 1 - FROSTART)
C
      CALL VISUB(FROSTART,0,IPLUS1(FROSTART),LENCOL(FROSTART),1,
     LENFRO)
C
C  Create PIVNO (vector of pivotal row divided
C  by minus pivot)
C
      DPIV=-1.0/PIVOT
      CALL VSMUL(EQ(FROSTART),1,DPIV,PIVNO(FROSTART),1,LENFRO)
C
C  Loop along pivotal row
C
      DO 20 I=FROSTART, FROFINISH
      IF(PIVNO(I).NE.0.0)THEN
      SIGDIG(I)=SIGDIG(I)+SUSTIF(IPOSVEC(I))*SUSTIF(IPOSVEC(I))
C
C  Do reduction for column I
C
      CALL VSMA(EQ(FROSTART),1,PIVNO(I),SUSTIF(JAVEC(I)),1,
     SUSTIF(JAVEC(I)),1,LENCOL(I))
      ENDIF
20    CONTINUE
```

As with the addition of the element stiffness matrix to the front it proved impossible to vectorise the calculation of SIGDIG because of the non-sequential access to SUSTIF required for this calculation. A disadvantage of the code shown above is the need to call the VSMA routine each time a column is reduced. There will be an overhead associated with each call due to the need to fill the CPU pipelines, the so called 'setup' time. Ref 218 indicates that the setup time for the VSMA routine is 15.0 μs and that the execution time per loop is 0.5 μs. Thus the setup overhead ranges from 37.5% of the total reduction time for a frontwidth of 99 to 5.0% at a frontwidth of 1140, which were the smallest and largest test problems. The routine thus gives overall speeds of 2.5 Mflops and 3.8 Mflops, respectively.

D.4 Test results

In order to investigate the performance of the code over a range of problem sizes a set of benchmark analyses were run. These consisted

of cubes of 20 node brick elements. The stiffness matrices were evaluated using 2 × 2 × 2 Gaussian quadrature. The benchmarks are listed below.

Benchmark tests

Number of elements	Number of nodes	Maximum frontwidth	Number of operations for reduction
8	81	99	1.56×10^6
27	208	174	1.40×10^7
64	425	258	6.76×10^7
125	756	360	2.44×10^8
216	1225	480	7.24×10^8
343	1856	618	1.86×10^9
512	2673	774	4.26×10^9
729	3700	948	8.97×10^9
1000	4961	1140	1.76×10^{10}

The operation counts were estimated by assuming that the frontwidth remains constant at its maximum value throughout the reduction. This gives a reasonable estimate for a structure as regular as a cube, although the frontwidth is of course zero at the start and end of the reduction.

These benchmarks were run on the VAX 11/780 alone using the original FORTRAN code and also using the new code on the VAX/AP combination. It proved impractical to run the two largest analyses on the VAX as the front array did not fit into the available real memory, thus resulting in an excessive number of page faults. The results are depicted in Fig33. The host and AP CPU times and also the elapsed time are shown for the runs using both processors. The AP times do not include time spent waiting for the VAX. However, time spent by the host in waiting for the AP has been included. The VAX CPU time is also shown for the runs using the host alone.

D.5 Discussion

Fig 33 shows that as expected the AP CPU time is nearly proportional to the number of inner loop operations. This is almost exactly true for the larger problems (7 × 7 × 7 elements and above). The relationship becomes less accurate as the problem size is diminished.

This is due in part to the increasing relative importance of additional tasks such as element addition to the front, and reduction of the structural force vector. However, the main reason is that performance of the reduction code diminishes due to the setup delays as the problem size is decreased. The observed degradation in AP performance with decreasing problem size is depicted in Fig 34. The ratio in speed between the smallest and largest problems studied is in good agreement with the ratio expected due to the setup delay.

The host CPU time is as expected greater than the AP time for small problems. As the amount of host CPU time required is proportional to the number of elements it is also roughly proportional to the cube root of the number of inner loop operations. Thus as the problem size is increased the AP time increases more rapidly than the host time until for large problems the host waits for the AP. This waiting time is also accrued to the host CPU and explains the kink in the graph above $7 \times 7 \times 7$ elements.

Comparison of the elapsed times with the AP and VAX CPU times gives some measure of how I/O bound the algorithm is. For the smallest problem the ratio of elapsed time to VAX CPU time is 1.18. In this case the AP is waiting for the VAX for a significant period, and thus this figure is really a measure of how I/O bound the VAX process is. For large problems the elapsed/AP CPU time ratio falls to 1.03. This indicates that virtually all the I/O between the VAX and AP as well as the I/O required by the host to retrieve the element properties from disc and store the reduced stiffness matrix is hidden in the AP CPU time. For the $7 \times 7 \times 7$ problem the ratio increases to 1.25 which was the highest observed. For this size of problem a similar amount of CPU time was taken by the host and AP. It is thus likely that both processors have on occasion to wait for data from the other machine, which will cause an increase in the elapsed/CPU time ratio. However it is clear that in general the time taken for I/O does not significantly degrade the performance of the algorithm.

The speedup obtained using the algorithm is depicted in Fig 35. For the smallest problem the reduction on the host takes only about half the time taken to formulate the element stiffness matrices. The improvement in speed when the AP is used is therefore only modest. As the problem size is increased the relative importance of the reduction increases and thus so does the speed improvement. Eventually for large problems the improvement reaches a maximum of about 25 times the host speed. The relatively poor performance of the AP reduction code for

small problems due to setup delays is thus in a sense irrelevant as in this case the time taken to generate the element stiffnesses on the host is the controlling factor.

An improvement in speed of greater than one order of magnitude is obtained for problems requiring more than 4×10^8 operations. This corresponds to approximately 160 20-node brick elements. A problem of this magnitude requires approximately 45 minutes of CPU time if only the host is used. For the GANDALF blade problem approximately 10^{10} operations are required and the algorithm is thus operating at maximum efficiency.

The algorithm can reduce the solution time for forward reduction for a problem of the magnitude of the blade analysis by a factor of 25 compared to the VAX alone. However it appears that it is not possible to reduce the backsubstitution time significantly. The implementation will thus favour solution algorithms which reformulate the stiffness matrix more often, and this is reflected in the results presented in chapter 9.

REFERENCES

No.	Authors	Title, etc
1	D.G. Ainley G.C.R. Mathieson	A method of performance estimation for axial turbines. Aeronautical Research Council R&M 2974 (1951)
2	S.F. Smith	A simple correlation of turbine efficiency. Jnl. Roy. Aero. Soc., Vol 69, pp 467-470 (1965)
3	J.H. Horlock	Axial flow turbines. Butterworths, London (1966)
4	G.F.C. Rogers Y.R. Mayhew	Engineering thermodynamics, work and heat transfer. 2nd Edition, Longmans, London (1967)
5	J. Dunham P.M. Came	Improvements to the Ainley-Mathieson method of turbine performance prediction. ASME Paper 70-GT-2, ASME Gas Turbine Conference and Products Show, Brussels (1970)
6	J.F. Barnes	Advances in engine technology: cooled turbines. Aircraft Engineering, Vol 41, pp 18-21 (1969)
7	E. Elovic W.K. Koffel	Some considerations in the thermal design of turbine airfoil cooling systems. Int. Jnl. Turbo and Jet Engines, Vol 1, No.1, pp 45-65 (1984)
8	M.J. Holland T.F. Thake	Rotor blade cooling in high pressure turbines. Jnl. Aircraft, Vol 17, No.6, pp 412-418 (1980)
9	A.J. Dennis T.A. Cruse	Cost benefits from improved hot section life prediction technology. AIAA/SAE/ASME 15th Joint Propulsion Conference, Las Vegas (1979)
10	M.T. Loferski	Turbine airfoil life prediction by mission analysis. Jnl. Aircraft, Vol 12, No.4, pp 400-402 (1975)
11	M.L. Roberts	Engine life, usage and cycle selection. Jnl. Aircraft, Vol 15, No.4, pp 240-245 (1978)

REFERENCES (continued)

No.	Author	Title, etc
12	E.A. White M.J. Holland	Project optimization of military gas turbines with respect to turbine life. High temperature problems in gas turbine engines, Ankara, Turkey. AGARD Conference Proceedings, No.229 (1978)
13	R.E. Jenkins	Creep and aero gas turbine design. Inst. of Mech. Engineers Int. Conf. on Engineering Aspects of Creep, Sheffield (1980)
14	G.A. Halls S.G. Baker	Turbine blade cooling - the global picture. 9th Int. Aeronautical Congress, Paris (1969)
15	W.R. Taylor J.S. Ogg	Accelerated mission testing of gas turbine engines. AIAA/SAE 13th Propulsion Conference, Orlando, USA (1977)
16	W.R. Taylor	Accelerated mission endurance testing (AMET). Engine cyclic durability by analysis and testing, Lisse, Netherlands. AGARD Conference Proceedings No.368 (1984)
17	F.S. Stepka	Analysis of uncertainties in turbine metal temperature predictions. NASA TP 1593 (1980)
18	R.C. Haubert E. Hsia H.M. Maclin M.E. Noe R.O. Brooks	High pressure turbine blade life sensitivity. AIAA/SAE/ASME 16th Joint Propulsion Conference, Hartford, Connecticut, USA (1980)
19	E.S. Hsia	Sensitivity of blade life to changes in design variables. ASME Gas Turbine Conference and Products Show, Houston, USA (1981)
20	R.A. Rudey R.W. Graham	A review of NASA combustor and turbine heat transfer research. ASME Int. Gas Turbine Conference, Amsterdam (1984)
21	B.A. Boley J.H. Weiner	Theory of thermal stresses. John Wiley, New York (1960)

REFERENCES (continued)

No.	Author	Title, etc
22	B.L. Lewis L.R. Beckwith	A unified approach to turbine blade life prediction. SAE Paper 821439, Aerospace Congress, Anaheim, California (1982)
23	J.C. Freche J. Acurio	Materials and structural aspects of advanced gas turbine helicopter engines. NASA TM 79100 (1979)
24	J.-L. Chaboche	Stress calculations for lifetime prediction in turbine blades. Int. Jnl. Solids and Structures, Vol 10, pp 473-481 (1974)
25	A. Kaufman	Steady state stress relaxation analysis of turbine blade cooling designs. NASA TN D-5282 (1969)
26	H. Policella J.-P. Culie	A method for predicting the lifetime of gas turbine blades. Fatigue of Engineering Materials and Structures, Vol 4, No.2, pp 157-172 (1981)
27	U. Hall U. Olsson	Compressible flow, temperature and life calculations with turbine blades. Proc. 3rd Int. Symposium on Air Breathing Engines, Munich, Eds. D.K. Hennecke and G. Winterfeld (1976)
28	W. Peschel R. Schreieck	A contribution on thermal fatigue in cooled turbine blades. Stresses, vibrations, structural integration and engine integrity, Cleveland, USA. AGARD Conference Proceedings, No.248 (1978)
29	J.-L. Chaboche J.P. Culie	On turbine blade creep and fatigue analysis by special kinematic assumptions. Inst. of Mech. Engineers Int. Conf. on Engineering Aspects of Creep, Sheffield (1980)
30	A. Kaufman	Analysis of the effects on life of leading edge holes in an airfoil subjected to arbitrary spanwise and chordwise temperature distributions. NASA TM X-3257 (1975)

REFERENCES (continued)

No.	Author	Title, etc
31	MacNeal-Schwendler Corporation	MSC/NASTRAN user's manual. Report No.MSR-46, MacNeal-Schwendler Corp., Los Angeles, California (1978)
32	D.S. Wolf	Stress analysis of a first stage turbine vane using a three dimensional model with non-linear behaviour. AIAA/SAE/ASME Joint Propulsion Conference, Colorado Springs (1981)
33	C.B. Shung J.X. Xiao	Three dimensional stress analysis for an air cooled turbine blade. AIAA/SAE/ASME 17th Joint Propulsion Conference, Colorado Springs (1981)
34	T. Maya I. Katsumata M. Itoh	A study of thermal fatigue life prediction of air cooled turbine blades. ASME Gas Turbine Conference, London (1978)
35	J.M. Allen	Effect of temperature dependent mechanical properties on thermal stress in cooled turbine blades. ASME Int. Gas Turbine Conference and Products Show, Houston (1981)
36	A. Kaufman	Development of a simplified procedure for cyclic structural analysis. NASA TP 2243 (1984)
37	A. Kaufman	A simplified method for elastic-plastic-creep structural analysis. NASA TM 83509 (1984)
38	A. Kaufman	Comparison of elastic and elasto-plastic structural analyses for cooled turbine blade airfoils. NASA TP 1679 (1980)
39	Marc Analysis Research Corporation	MARC general purpose finite element program. Vol. A, User Information Manual, Marc Analysis Research Corporation, Palo Alto, California (1983)

REFERENCES (continued)

No.	Author	Title, etc
40	A. Kaufman R.E. Gaugler	Cyclic structural analyses of air cooled gas turbine blades and vanes. SAE Aerospace Engineering and Manufacturing Meeting, San Diego (1976)
41	S. Utku M.S.M. Rao G.J. Dvorak	Elas 65 computer program for equilibrium problems of elastic-thermoplastic solids and structures. Structural Mechanics Series 15, Duke University, North Carolina (1973)
42	S. Utku J.Q. Tarn G.J. Dvorak	Elas 55 computer program for equilibrium problems of thermo-viscoelastic-plastic solids. Structural Mechanics Series 21, Duke University, North Carolina (1974)
43	G.R. Halford M.H. Hirschberg S.S. Manson	Temperature effects on the strain range partitioning approach for creep fatigue analysis. Fatigue at elevated temperatures, American Society for Testing and Materials, STP 520 (1973)
44	E.L. Robinson	Effect of temperature variation on the creep strength of steels. Trans. ASME, Vol 60, pp 253-259 (1938)
45	A. Kaufman R.E. Gaugler	Thermal-structural mission analysis of air cooled gas turbine blades. NASA TM 78963 (1979)
46	A. Kaufman R.E. Gaugler	Nonlinear three dimensional finite element analysis of air cooled gas turbine blades. NASA TP 1669 (1980)
47	R.L. McKnight J.H. Laflen G.T. Spamer	Turbine blade tip durability analysis. NASA CR 165268 (1981)
48	R.L. McKnight J.H. Laflen G.R. Halford A. Kaufman	Turbine blade nonlinear structural and life analysis. Jnl. Aircraft, Vol 20, No.5, pp 475-480 (1983)

REFERENCES (continued)

No.	Author	Title, etc
49	P.C. Kohnke	ANSYS engineering analysis systems theoretical manual. Swanson Analysis Systems Inc., Elizabeth, Pennsylvania, USA (1977)
50	J.M. Manderscheid A. Kaufman	Cyclic structural analyses of anisotropic turbine blades for reusable space propulsion systems. NASA TM 86990 (1985)
51	V. Moreno	Combustor liner durability analysis. NASA CR 165250 (1981)
52	V. Moreno G.J. Meyers A. Kaufman G.R. Halford	Nonlinear structural and life analyses of a combustor liner. Computers and Structures, Vol 16, Nos.1-4, pp 509-515 (1983)
53	D.E. Andress	An analytical study of thermal barrier coated first stage blades in an F100 engine. NASA CR 135359 (1978)
54	W.R. Sevcik B.L. Stoner	An analytical study of thermal barrier coated first stage blades in a JT9D engine. NASA CR 135360 (1978)
55	M.R. Peterson R.G. Anderson R.J. Stockton D.J. Tree	Three dimensional finite element techniques for gas turbine blade life prediction. Stresses, vibrations, structural integration and engine integrity, Cleveland, USA. AGARD Conference Proceedings, No.248 (1978)
56	Fegs Limited	FEMGEN User Manual version 8.5. Fegs Limited, Oakington, Cambridge (1984)
57	O.C. Zienkiewicz	The finite element method. 3rd Edition McGraw Hill, London (1977)
58	R. Courant	Differential and integral calculus. Blackie and Son, Glasgow (1936)
59	W.B. Jordan	The plane isoparametric structural element. Report No.KAPL-M-7112, Knolls Atomic Power Lab., Schenectady, New York (1970)

REFERENCES (continued)

No.	Author	Title, etc
60	B.M. Irons	A frontal solution program for finite element analysis. Int. J. Num. Meth. Engng., Vol 2, pp 5-32 (1970)
61	B.M. Irons S. Ahmad	Techniques of finite elements. Ellis Horwood, Chichester (1980)
62	N.E. Gibbs W.G. Poole P.K. Stockmeyer	An algorithm for reducing the bandwidth and profile of a sparse matrix. SIAM Jnl. Numer. Anal., Vol 13, pp 236-250 (1976)
63	A. Razzaque	Automatic reduction of frontwidth for finite element analysis. Int. Jnl. Num. Meth. Engng., Vol 15, pp 1315-1324 (1980)
64	W.C. Clifford R.T. Whitworth A.L. Kersey	Frontwidth reduction for the finite element method. National Gas Turbine Establishment Technical Memorandum M81013 (1981)
65	J.G. Griffiths	Eliminating hidden edges in line drawings. Computer Aided Design, Vol 11, pp 71-78 (1979)
66	Femview Limited	Femview User Manual version 4.0. Femview Limited, Leicester (1986)
67	R.V. Southwell	Relaxation methods in theoretical physics. Clarendon Press, Oxford (1946)
68	D.N. de G. Allen	Relaxation methods. McGraw Hill (1955)
69	S.H. Crandall	Engineering analysis. McGraw Hill (1956)
70	B.A. Findlayson	The method of weighted residuals and variational principles. Academic Press (1972)
71	W. Ritz	*Uber eine neue methode zur losung gewisser variationsprobleme der mathematischen physik.* Zeitschrift fur Angewandte Mathematik und Mechanik, Vol 135, pp 1-61 (1908)

REFERENCES (continued)

No.	Author	Title, etc
72	D. McHenry	A lattice analogy for the solution of plane stress problems. Jnl. Inst. Civ. Eng., Vol 21, pp 59-82 (1943)
73	A. Hrenikoff	Solutions of problems in elasticity by the framework method. ASME Jnl. Appl. Mech., Vol A8, pp 169-175 (1941)
74	N.M. Newmark	Numerical methods of analysis in bars, plates and elastic bodies. In "Numerical Methods in Analysis in Engineering", Ed. L.E. Grinter, Macmillan (1949)
75	J.H. Argyris	Energy theorems and structural analysis. Butterworths (1960)
76	M.J. Turner R.W. Clough H.C. Martin	Stiffness and deflection analysis of complex structures. Jnl. Aeronautical Sciences, Vol 23, pp 805-823, (1956)
77	R.W. Clough	The finite element in plane stress analysis. Proc. 2nd ASCE Conf. on Electronic Computation, Pittsburgh, Pennsylvania (1960)
78	J. Szmelter	The energy method of networks of arbitrary shape in problems of the theory of elasticity. Proc. IUTAM Symposium on nonhomogeneity in elasticity and plasticity, Ed. W. Olszak, Pergamon Press (1959)
79	R.W. Clough	The finite element method in structural mechanics. In "Stress analysis", Eds. O.C. Zienkiewicz and G.S. Holister, John Wiley (1965)
80	K.-J. Bathe	Finite element procedures in engineering analysis. Prentice Hall, Englewood Cliffs, New Jersey (1982)
81	I.C. Taig	Structural analysis by the matrix displacement method. English Electric Aviation Report S017 (1961)

REFERENCES (continued)

No.	Author	Title, etc
82	B.M. Irons	Numerical integration applied to finite element methods. Conf. Use of Digital Computers in Structural Engineering, University of Newcastle (1966)
83	B.M. Irons	Engineering application of numerical integration in stiffness methods. AIAA Journal, Vol 4, pp 2035-2037 (1966)
84	I. Ergatoudis B.M. Irons O.C. Zienkiewicz	Curved, isoparametric quadrilateral elements for finite element analysis. Int. Jnl. Solids and Structures, Vol 4, pp 31-42 (1968)
85	C. Marples	A stiffness method of stress analysis using numerical integration. Report No.ASM 1405, Rolls-Royce plc, Derby (1966)
86	O.C. Zienkiewicz B.M. Irons J. Ergatoudis S. Ahmad F.C. Scott	Isoparametric and associated element families for two and three dimensional analysis. In "Finite element methods in stress analysis", Eds. I. Holand and K. Bell, Tapir Press, Trondheim, Norway (1969)
87	E.A.H. Love	The mathematical theory of elasticity. 4th Edition, Cambridge University Press (1927)
88	S. Timoshenko J.N. Goodier	Theory of elasticity. 2nd Edition, McGraw Hill (1951)
89	F. Schied	Numerical analysis. McGraw Hill (1968)
90	R. Hill	The mathematical theory of plasticity. Oxford University Press (1950)
91	J.C. Nagtegaal D.M. Parks J.R. Rice	On numerically accurate finite element solutions in the fully plastic range. Comp. Meth. in Appl. Mech. Engng., Vol 4, pp 153-178 (1974)
92	B.M. Irons	Economical computer techniques for numerically integrated finite elements. Int. Jnl. Num. Meth Engng., Vol 1, pp 201-203 (1969)

REFERENCES (continued)

No.	Author	Title, etc
93	A.K. Gupta B. Mohraz	A method of computing numerically integrated stiffness matrices. Int. Jnl. Num. Meth. Engng., Vol 5, pp 113-135 (1972)
94	S.W. Sloan	A fast stiffness formulation for finite element analysis of two dimensional solids. Int. Jnl. Num. Meth. Engng., Vol 17, pp 1313-1323 (1981)
95	J. Barlow	Optimal stress locations in finite element models. Int. Jnl. Num. Meth. Engng., Vol 10, pp 243-251 (1976)
96	D.C. Drucker	Plasticity. Proc. 1st Int. Symp. on Naval Structural Mechanics, Pergamon Press, Oxford (1960)
97	P.M. Naghdi	Stress strain relations in plasticity and thermoplasticity. Proc. 1st Int. Symp. on Naval Structural Mechanics, Pergamon Press, Oxford (1960)
98	W. Johnson P.B. Mellor	Plasticity for mechanical engineers. Van Nostrand (1962)
99	A. Mendelson	Plasticity: theory and application. Macmillan, New York (1968)
100	H. Tresca	*Sur l'ecoulement des corps solides soumis a de fortes pression.* Comptes Rendus Aca. Sci., Paris, Vol 59, p.754 (1864)
101	B. de Saint-Venant	*Memoire sur l'etablissement des equations differentielles des mouvements interieurs operes dans les corps solides ductiles au dela des limits ou l'elasticite pourrait les ramener a leur premier etat.* Comptes Rendus Acad. Sci., Paris, Vol 70, pp 473-480 (1870)

REFERENCES (continued)

No.	Author	Title, etc
102	M. Levy	*Memoire sur les equations generales des mouvements interieurs des corps solides ductile au dela limites ou l'elasticite pourrait les ramener a leur premier etat.* Comptes Rendus Acad. Sci., Paris, Vol 70, pp 1323-1325 (1870)
103	R. von Mises	*Mechanik der festen koerper in plastich deformablem zustand.* Gottinger Nachrichten, Math. - Phys. Klasse, pp 582-593 (1913)
104	L. Prandtl	*Spannungsverteilung in plastichen koerpern.* Proc. 1st Int. Cong. Appl. Mech., Delft, Netherlands, pp 43-54 (1924)
105	A. Reuss	*Beruecksichtigung der elastischen formaenderungen in der plastizitaetstheorie.* Zeits. Ang. Math. Mech., Vol 10, pp 266-274 (1930)
106	H. Hencky	*Zur theorie plastischer deformation und der hierdurch im material hervorgerufenen nachspannungen.* Zeits. ang. Math. Mech., Vol 4, pp 323-334 (1924)
107	D.R. Bland	The two measures of workhardening. Proc. 9th Int. Conf. of Applied Mechanics, Brussels (1956)
108	W. Prager	The theory of plasticity: a summary of recent achievements. James Clayton Lecture, Proc. Inst. Mech. Eng., Vol 169, pp 41-57 (1955)
109	H. Ziegler	A modification of Prager's hardening rule. Quart. Appl. Math., Vol 17, pp 55-65 (1959)

REFERENCES (continued)

No.	Author	Title, etc
110	P.G. Hodge	Discussion on "W. Prager - a new method of analysing stresses and strains in workhardening solids. (ASME Jnl. Appl. Mech., Vol 23, pp 493-496 (1956))", ASME Jnl. Appl. Mech., Vol 24, pp 481-484 (1957)
111	Z. Mroz	On the description of anisotropic workhardening. Jnl. Mech. Phys. Solids, Vol 15, pp 163-175 (1967)
112	M.A. Eisenberg	A generalization of plastic flow theory with application to cyclic hardening and softening phenomena. ASME Jnl. Engng. Mat. and Tech., Vol 98, pp 221-228 (1976)
113	D.C. Drucker	A more fundamental approach to plastic stress-strain relations. Proc. 1st Conf. US National Congress of Applied Mechanics (1951)
114	B. Hunsaker D.K. Vaughan J.A. Stricklin	A comparison of the capability of four hardening rules to predict a material's behaviour. ASME Paper 75-PVP-43 2nd National Congress on Pressure Vessels and Piping, San Francisco (1975)
115	W. Prager	Non isothermal plastic deformation. Proc. Koninklijke Nederlandse Akademie van Wetenschappen, Vol 61, pp 176-182 (1958)
116	P.W. Bridgeman	Studies in large plastic flow and fracture. McGraw Hill, New York (1958)
117	W. Lode	*Versuche ueber den einfluss der mittleren hauptspannung auf das fliessen der metalle eisen kupfer und nickel.* Z. Physik., Vol 36, pp 913-939 (1926)
118	G.I. Taylor H. Quinney	The plastic distortion of metals. Phil. Trans. Royal Society Series A, Vol 230, pp 323-362 (1931)

REFERENCES (continued)

No.	Author	Title, etc
119	M.R. Winstone	Influence of prestress on the yield surface of the cast metallic superalloy Mar-M 002. National Gas Turbine Establishment, Pyestock Memorandum M83006 (1983)
120	D.R. Bland	The associated flow rule of plasticity. Jnl. Mech. Phys. of Solids, Vol 6, pp 71-78 (1957)
121	G.G. Pope	A discrete element method for analysis of plane elasto-plastic strain problems. Royal Aircraft Establishment Technical Report 65028 (1965)
122	O.C. Zienkiewicz S. Valliappan I.P. King	Elasto-plastic solution of engineering problems "initial stress" finite element approach. Int. Jnl. Num. Meth. Engng., Vol 1, pp 75-100 (1969)
123	H. Armen A.B. Pifko H.S. Levine G. Isakson	Plastic analysis of structures. Grumman Aerospace Research Department, Report RE 380J, Bethpage, New York (1970)
124	H. Armen A.B. Pifko H.S. Levine	Finite element analysis of structures in the plastic range. NASA Contract Report CR 1649 (1971)
125	H. Armen H. Levine A.B. Pifko A. Levy	Nonlinear analysis of structures. NASA Contract Report CR 2351 (1974)
126	A.B. Pifko H.S. Levine H. Armen	PLANS - a finite element program for non-linear analysis of structures. NASA Contract Report CR 2568 (1975)
127	E.N. da C. Andrade	On the viscous flow of metal and allied phenomena. Proc. Royal Society Series A, Vol 84, p 1 (1910)
128	F.H. Norton	The creep of steel at high temperatures. McGraw Hill, New York (1929)

REFERENCES (continued)

No.	Author	Title, etc
129	P.G. McVetty	Working stresses for high temperature service. Mechanical Engng., Vol 56, pp 149-154 (1934)
130	J.E. Dorn	Some fundamental experiments on high temperature creep. Jnl. Mech. Phys. Solids, Vol 3, pp 85-116 (1955)
131	A. Graham K.F.A. Walles	Relationships between long and short term creep and tensile properties of commercial alloys. Jnl. Iron and Steel Inst., Vol 179, pp 105-120 (1955)
132	F. Garofalo	Fundamentals of creep and creep rupture in metals. Macmillan, London and New York (1965)
133	R.W. Evans J.D. Parker B. Wilshire	An extrapolation procedure for long term creep strain and creep life prediction with special reference to ½Cr½Mo¼V ferritic steels. In "Recent Advances in Creep and Fracture of Engineering Materials and Structures", Eds. B. Wilshire and D.R.J. Owen, Pineridge Press, Swansea (1982)
134	B. Wilshire R.W. Evans	Private communication (1984)
135	J.T. Boyle J. Spence	Stress analysis for creep. Butterworths, London (1983)
136	Y.R. Rashid	Part 1: Theory report for CREEP-PLAST computer program: analysis of two dimensional problems under simultaneous creep and plasticity. General Electric Co., AEC Research and Development Report (1972)
137	C.C. Davenport	Correlation of creep and relaxation properties of copper. Trans. ASME Jnl. Appl. Mech., Vol 5, pp 55-60 (1938)
138	G.F. Harrison	Deformation and fracture behaviour of some nickel based gas turbine alloys. PhD Thesis, University of London (1975)

REFERENCES (continued)

No.	Author	Title, etc
139	B. Wilshire R.W. Evans	Private communication (1986)
140	Y.R. Rashid	Analysis of multiaxial flow under variable load and temperature. Int. Conf. on Creep and Fatigue in Elevated Temperature Applications, Philadelphia and Sheffield. Inst. Mech. Eng. Conf. Publ. 13 (1973)
141	R.W. Bailey	The utilization of creep test data in engineering design. Proc. Inst. Mech. Eng., Vol 131, pp 186 *et seq* (1935)
142	H.J. Tapsell A.E. Johnson	Creep under combined tension and torsion. Engineering, pp 24 *et seq* (1940)
143	A.E. Johnson J. Henderson B. Khan	Complex stress creep, relaxation and fracture of metallic alloys. HMSO, Edinburgh (1962)
144	A.P. Boresi O.M. Sidebottom	Creep of metals under multiaxial states of stress. Nuclear Engineering and Design, Vol 18, pp 415-456 (1972)
145	D.W.A. Rees	The potential theory of creep. In "Engineering Approaches to High Temperature Design", Eds. B. Wilshire and D.R.J. Owen, Pineridge Press, Swansea (1983)
146	F.K.G. Odqvist	Mathematical theory of creep and creep rupture. 2nd Edition, Oxford University Press (1974)
147	P.J. Van der Houwen	Construction of integration formulas for initial value problems. North Holland Series in Applied Mathematics and Mechanics, Vol 19 (1977)
148	C.W. Gear	The automatic integration of ordinary differential equations. Communications of ACM, Vol 14, pp 176-179 (1971)

REFERENCES (continued)

No.	Author	Title, etc
149	J. Donea S. Giuliani	Creep analysis of transversely isotropic bodies subjected to time dependent loading. Nuclear Engineering and Design, Vol 24, pp 410-419 (1973)
150	N.A. Cyr R.D. Teter	Finite element elastic-plastic-creep analysis of a two dimensional continuum with temperature dependent material properties. Computer and Structures, Vol 3, pp 849-863 (1973)
151	P. Sharifi D.N. Yates	Nonlinear thermo-elastic-plastic and creep analysis by the finite element method. AIAA Journal, Vol 12, pp 1210-1215 (1974)
152	O.C. Zienkiewicz I.C. Cormeau	Viscoplasticity, plasticity and creep in elastic solids - a unified numerical solution approach. Int. Jnl. Num. Meth. Engng., Vol 8, pp 821-845 (1974)
153	O.C. Zienkiewicz I.C. Cormeau	Viscoplasticity and plasticity - an alternative for finite element solution of material nonlinearities. Int. Symp. on Computing Methods in Applied Science and Engineering, Versailles, 1973. Lecture Notes in Computer Sciences No.10, Springer Verlag, Berlin (1974)
154	O.C. Zienkiewicz	Viscoplasticity, plasticity, creep and visco-plastic flow (problems of small, large and continuing deformation). Int. Conf. on Computational Methods in Non-linear Mechanics, Austin, Texas, 1974. Lecture Notes in Mathematics No.461, Springer Verlag, Berlin (1975)
155	M.D. Snyder K.-J. Bathe	Formulation and numerical solution of thermo-elastic plastic and creep problems. Rpt. No.82448-3, Massachusetts Inst. of Technology (1977)

REFERENCES (continued)

No.	Author	Title, etc
156	G. Cailletaud J.-L. Chaboche	*Prevision de duree de vie a haute temperature sous chargement complex.* Engine cyclic durability by analysis and testing, Lisse, Netherlands, 1984. AGARD Conference Proceedings, No. 368 (1984)
157	M. Chaudonneret M. Cristescu G. Loubignac B. Maine	Comparison of algorithms and validation of behaviour laws in elasto-viscoplasticity and plasticity. In "Numerical Methods for nonlinear Problems", Eds. C. Taylor, E. Hinton, D.R.J. Owen, Pineridge Press, Swansea (1980)
158	H.O. Kim R.L. Kuhlemeyer	A finite element formulation for creep analysis. Int. Jnl. Num. Meth. Engng., Vol 11, pp 1865-1877 (1977)
159	M.B. Kanchi O.C. Zienkiewicz D.R.J. Owen	The viscoplastic approach to problems of plasticity and creep involving geometric nonlinear effects. Int. Jnl. Num. Meth. Engng., Vol 12, pp 169-181 (1978)
160	L.M. Kachanov	Time of the rupture process under creep conditions. Izv. Ak. Nauk. SSR No.8, pp 26-31 (1958)
161	B.R. Bass W.G. Johnson S.V. McGrath	An implicit strain hardening formulation for creep analysis. Union Carbide Corp., Nuclear Division (1979)
162	J.M.M.C. Marques D.R.J. Owen	Strain hardening representation for implicit quasi-static elasto-viscoplastic algorithm. Computer and Structures, Vol 17, pp 301-304 (1984)
163	D. Peirce C.F. Shih A. Needleman	A tangent modulus method for rate dependent solids. Computers and Structures, Vol 18, pp 875-887 (1984)

REFERENCES (continued)

No.	Author	Title, etc
164	I.C. Cormeau	Numerical stability in quasistatic elasto-viscoplasticity. Int. Jnl. Num. Meth. Engng., Vol 9, pp 109-127 (1975)
165	J.L. Bassani F.A. McClintock	Creep relaxation of stress around a crack tip. Int. Jnl. Solids and Structures, Vol 17, pp 479-492 (1981)
166	T.J.R. Hughes R.L. Taylor	Unconditionally stable algorithms for quasi-static elasto-viscoplastic finite element analysis. Computers and Structures, Vol 8, pp 169-173 (1978)
167	D.R.J. Owen F. Damjanic	Viscoplastic analysis of solids: stability considerations. In "Recent advances in nonlinear computational mechanics", Eds. E. Hinton, D.R.J. Owen, C. Taylor, Pineridge Press, Swansea (1982)
168	M.D. Snyder K.-J. Bathe	A solution procedure for thermo-elastic-plastic and creep problems. Nuclear Engineering and Design, Vol 64, pp 49-80 (1981)
169	N.G. Huffington	Numerical analysis of elastoplastic stresses. Memo. Report No.2006, Ballistic Research Labs., Aberdeen Proving Ground, Maryland (1969)
170	G.C. Nayak O.C. Zienkiewicz	Elasto-plastic stress analysis: a generalization for various constitutive equations including strain softening. Int. Jnl. Num. Meth. Engng., Vol 5, pp 113-135 (1972)
171	D. Bushnell	A strategy for the solution of problems involving large deflections, plasticity and creep. Int. Jnl. Num. Meth. Engng., Vol 11, pp 683-708 (1977)

REFERENCES (continued)

No.	Author	Title, etc
172	C.G. Burton T.J.W. Ward	A computer suite for turbine blade stress analysis: the elasto-plastic analysis. Unpublished Royal Aircraft Establishment Report (1984)
173	J.S. Holt B. Parsons	Special problems in incremental finite element plasticity analysis. In "The mathematics of finite elements and applications III", Ed. J.R. Whiteman, Academic Press (1979)
174	N.S. Ottosen O. Gunneskov	Nonlinear subincremental method for determination of elasto-plastic creep behaviour. Int. Jnl. Num. Meth. Engng., Vol 21, pp 2237-2256 (1985)
175	V.V. Klyuyev N.I. Kokovin-Shcherbak	On the minimization of the number of arithmetic operations for the solution of linear algebraic systems of equations. Trans. G.J. Tee, Stanford University Report No.CS24 (1965)
176	J.H. Wilkinson	The algebraic eigenvalue problem. Oxford University Press (1965)
177	E. Cuthill J. McKee	Reducing the bandwidth of sparse symmetric matrices. Proc. ACM Conf., Assoc. for Comp. Machinery, New York (1969)
178	M.F. Yeo	A more efficient front solution: allocating assembly locations by longevity considerations. Int. Jnl. Num. Meth. Engng., Vol 7, pp 570-573 (1973)
179	W.C. Davidon	Variable metric method for minimization. Argonne Nat. Lab. Report ANL-5990 (1959)
180	C.G. Broyden	A class of methods for solving nonlinear simultaneous equations. Math. Comp., Vol 19, pp 577-593 (1965)

REFERENCES (continued)

No.	Author	Title, etc
181	H. Mathies G. Strang	The solution of nonlinear finite element equations. Int. Jnl. Num. Meth. Engng., Vol 14, pp 1613-1626 (1979)
182	J.E. Dennis J. More	Quasi-Newton methods - motivation and theory. SIAM Review, Vol 19, pp 46-89 (1977)
183	C.G. Broyden	A new double-rank minimization algorithm. Notices Amer. Math. Soc., Vol 16, p 670 (1969)
184	R. Fletcher	A new approach to variable metric algorithms. Computer Jnl., Vol 13, pp 317-322 (1970)
185	D. Goldfarb	A family of variable metric methods derived by variational means. Math. Comp., Vol 24, pp 23-26 (1970)
186	D.F. Shanno	Conditioning of quasi-Newton methods for function minimization. Math. Comp., Vol 24, pp 647-656 (1970)
187	M.A. Crisfield	Iterative procedures for linear and nonlinear analysis. Report LR 900, Transport and Road Research Laboratory, Crowthorne, Berks (1979)
188	M.A. Crisfield	A faster modified Newton-Raphson iteration. Computer Methods in Applied Mechanics and Engineering, Vol 20, pp 267-278 (1979)
189	M.A. Crisfield	Solution procedures for nonlinear structural problems. In "Recent advances in nonlinear computational mechanics", Eds. E. Hinton, D.R.J. Owen, C. Taylor, Pineridge Press, Swansea (1982)
190	M.A. Crisfield	Accelerating and damping the modified Newton-Raphson method. Computers and Structures, Vol 18, pp 395-407 (1984)
191	K.-J. Bathe A.P. Cimento	Some practical procedures for the solution of nonlinear finite element equations. Computer Methods in Applied Mechanics and Engineering, Vol 22, pp 59-85 (1980)

REFERENCES (continued)

No.	Author	Title, etc
192	A. Neil C.G. Burton	Solution algorithms for finite element stress analysis of turbine blades. Royal Aircraft Establishment Technical Report 86008 (1986)
193	E. Riks	An incremental approach to the solution of snapping and buckling problems. Int. Jnl. Solids and Structures, Vol 15, pp 529-551 (1979)
194	J. Padovan S. Tovichakchaikul	Algorithms for elasto-plastic-creep post buckling. ASME Jnl. Eng. Mechanics, Vol 110, pp 911-929 (1984)
195	J.M.M.C. Marques	Stress computation in elastoplasticity. Engineering Computations, Vol 1, pp 42-51 (1984)
196	D.R.J. Owen E. Hinton	Finite elements in plasticity: theory and practice. Pineridge Press, Swansea (1980)
197	R.D. Kreig D.B. Kreig	Accuracies of numerical solution methods for the elastic-perfectly plastic model. ASME Jnl. Pressure Vessel Tech., Vol 99, pp 510-515 (1977)
198	H.L. Schreyer R.F. Kulak J.M. Kramer	Accurate numerical solutions for elasto-plastic models. ASME Jnl. Pressure Vessel Tech., Vol 101, pp 226-234 (1979)
199	M. Zlamel	Finite element methods in heat conduction problems. In "The mathematics of finte elements and applications II", Ed. J.R. Whiteman, Academic Press (1977)
200	T.D. Lambert	Computational methods in ordinary differential equations. John Wiley, London (1973)

REFERENCES (continued)

No.	Author	Title, etc
201	W. Liniger	Optimization of a numerical integration method for stiff systems of ordinary differential equations. IBM Research Report RC 2198 (1968)
202	B. Venkatramen S.A. Patel	Structural mechanics with introductions to elasticity and plasticity. McGraw Hill, New York (1970)
203	A. Mendelson S.W. Spero	A general solution for the elasto-plastic thermal stresses in a strain hardening plate with arbitrary material properties. ASME Jnl. Appl. Mech., Vol 29, pp 151-158 (1962)
204	Rolls-Royce plc	Unpublished data (1976)
205	Rolls-Royce plc	Private communication (1986)
206	S.G. Lekhnitskii	Theory of elasticity of an anisotropic body. Trans. P. Fern, Holden Day, San Francisco (1963)
207	NASA	Nonlinear constitutive equations for high temperature applications. (Proceedings of Symposium held at Univeristy of Arizona, 1982), NASA CP 2271 (1983)
208	NASA	Nonlinear constitutive equations for high temperature applications. (Proceedings of Symposium held at NASA Lewis Research Centre, Cleveland, Ohio, 1984), NASA CP 2369 (1985)
209	Y.N. Rabotnov	Creep problems in structural members. North Holland (1969)
210	J.-L. Chaboche	Anisotropic creep damage in the framework of continuum damage mechanics. Nuclear Engineering and Design, Vol 79, pp 309-319 (1984)

REFERENCES (concluded)

No.	Author	Title, etc
211	S.T. Arvanitis Y.B. Symko R.N. Tadros	Multiaxial life prediction system for turbine components. ASME Jnl. Engng. for Power, Vol 109, pp 107-114 (1984)
212	J.F. Knott	Fundamentals of fracture mechanics. Butterworths, London (1973)
213	G.A. Webster	Crack growth at high temperatures. In "Engineering approaches to high temperature design", Eds. B. Wilshire, D.R.J. Owen, Pineridge Press, Swansea (1983)
214	W.K. Liu T. Belytschko K.C. Park (Editors)	Proc. Int. Conf. on Innovative Methods for Nonlinear Problems, New Orleans, 1984, Pineridge Press, Swansea (1984)
215	P.A. Bergan T.H. Soreide	Solution of large displacement and instability problems using the current stiffness parameter. Int. Conf. on Finite Elements in Nonlinear Solid and Structural Mechanics, Geilo, Norway (1977)
216	R.W. Evans B. Wilshire	Creep of metals and alloys. Institute of Metals, London (1985)
217	D.R. Hayhurst	Creep rupture under multiaxial states of stress. Jnl. Mech. Phys. Solids, Vol 20, pp 381-390 (1972)
218	Floating Point Systems Inc	APMATH64 Manual release E volumes 1-4. Floating Point Systems Inc., Portland, Oregon (1984)

Run No.	θ_S	S_S	UPDATE	θ_T	S_T	No. incs	No. iter	CPU m:s	Stress (N/m^2) Max Err.	Step	RMS%	Creep Strain Max Err.	Step	RMS%
INITIAL STIFFNESS ALGORITHM														
102	0.0	1.0	N	-	1.0	3	8	1:30	4.18×10^{7}	1	33.10	2.72×10^{-4}	1	30.88
105	0.0	1.0	Y	0.0	0.1	6	20	3:49	3.00×10^{6}	2	2.63	1.96×10^{-5}	2	1.87
140	0.0	1.0	Y	0.0	0.5	5	16	2:56	1.32×10^{7}	2	11.40	8.58×10^{-5}	2	8.87
718	0.0	1.0	Y	0.5	0.1	7	25	4:47	1.39×10^{6}	6	0.83	9.00×10^{-6}	6	0.53
719	0.0	1.0	Y	0.5	0.5	7	25	4:29	5.46×10^{6}	6	3.08	3.55×10^{-5}	6	2.00
106	0.0	1.0	Y	1.0	0.1	7	24	4:43	3.72×10^{6}	2	3.35	2.42×10^{-5}	2	2.20
141	0.0	1.0	Y	1.0	0.5	8	31	5:30	1.29×10^{7}	8	12.10	8.39×10^{-5}	8	7.11
HYBRID Mk 3 ALGORITHM														
202	0.0	1.0	N	-	1.0	3	3	0:55	4.18×10^{7}	1	33.10	2.72×10^{-4}	1	30.88
205	0.0	1.0	Y	0.0	0.1	6	16	4:05	3.01×10^{6}	2	2.63	1.96×10^{-5}	2	1.87
240	0.0	1.0	Y	0.0	0.5	5	12	3:03	1.32×10^{7}	2	11.40	8.58×10^{-5}	2	8.87
318	0.0	1.0	Y	0.5	0.1	7	18	4:43	1.38×10^{6}	6	0.83	9.00×10^{-6}	6	0.53
319	0.0	1.0	Y	0.5	0.5	7	19	4:39	5.46×10^{6}	6	3.08	3.55×10^{-5}	6	2.00
206	0.0	1.0	Y	1.0	0.1	7	18	4:46	3.72×10^{6}	2	3.34	2.42×10^{-5}	2	2.20
241	0.0	1.0	Y	1.0	0.5	8	24	5:49	1.29×10^{7}	8	12.10	8.40×10^{-5}	8	7.11

Table 1 Stress relaxation problem - linear subincrementation scheme

Run No.	θ_s	S_s	UPDATE	θ_r	S_r	No. incs	No. iter	CPU m:s	Stress (N/m^2) Max Err.	Step	RMS%	Creep Strain Max Err.	Step	RMS%
INITIAL STIFFNESS ALGORITHM														
601	0.5	1.0	N	-	1.0	6	18	3:10	1.46×10^{7}	6	8.41	9.47×10^{-5}	6	5.57
602	0.5	1.0	Y	0.0	0.1	6	17	3:05	3.01×10^{6}	2	2.63	1.95×10^{-5}	2	1.86
603	0.5	1.0	Y	0.0	0.5	5	16	2:46	1.32×10^{7}	2	11.46	8.57×10^{-5}	2	8.87
604	0.5	1.0	Y	0.5	0.1	7	18	3:21	1.38×10^{6}	6	0.82	8.98×10^{-6}	6	0.53
605	0.5	1.0	Y	0.5	0.5	7	18	3:09	5.45×10^{6}	6	3.07	3.55×10^{-5}	6	2.00
606	0.5	1.0	Y	1.0	0.1	7	18	3:24	3.72×10^{6}	2	3.34	2.42×10^{-5}	2	2.21
607	0.5	1.0	Y	1.0	0.5	8	21	3:51	1.29×10^{7}	8	12.11	8.41×10^{-5}	8	7.12
HYBRID Mk 3 ALGORITHM														
801	0.5	1.0	N	-	1.0	6	6	1:50	1.46×10^{7}	6	8.41	9.47×10^{-5}	6	5.57
802	0.5	1.0	Y	0.0	0.1	6	12	3:22	3.00×10^{6}	2	2.63	1.95×10^{-5}	2	1.86
803	0.5	1.0	Y	0.0	0.5	5	11	2:52	1.32×10^{7}	2	11.46	8.57×10^{-5}	2	8.87
804	0.5	1.0	Y	0.5	0.1	7	13	3:46	1.38×10^{6}	6	0.82	8.99×10^{-6}	6	0.53
805	0.5	1.0	Y	0.5	0.5	7	13	3:39	5.46×10^{6}	6	3.07	3.55×10^{-5}	6	2.00
806	0.5	1.0	Y	1.0	0.1	7	13	3:49	3.72×10^{6}	2	3.34	2.42×10^{-5}	2	2.20
807	0.5	1.0	Y	1.0	0.5	8	22	5:25	1.29×10^{7}	8	12.11	8.39×10^{-5}	8	7.11

Table 2 Stress relaxation problem - linear subincrementation scheme

Run No.	θ_s	S_s	UPDATE	θ_Y	S_Y	No. incs	No. iter	CPU m:s	Stress (N/m^2) Max Err.	Step	RMS%	Creep Strain Max Err.	Step	RMS%
									INITIAL STIFFNESS ALGORITHM					
120	1.0	1.0	N	-	1.0	11	28	4:49	2.37×10^{7}	3	21.71	1.55×10^{-4}	3	12.23
125	1.0	1.0	Y	0.0	0.1	6	20	3:36	3.01×10^{6}	2	2.63	1.95×10^{-5}	2	1.86
148	1.0	1.0	Y	0.0	0.5	5	16	2:48	1.32×10^{7}	2	11.40	8.57×10^{-5}	2	8.87
716	1.0	1.0	Y	0.5	0.1	7	23	4:16	1.38×10^{6}	6	0.82	8.98×10^{-6}	6	0.53
717	1.0	1.0	Y	0.5	0.5	7	23	3:55	5.45×10^{6}	6	3.07	3.55×10^{-5}	6	2.00
126	1.0	1.0	Y	1.0	0.1	7	23	4:14	3.72×10^{6}	2	3.34	2.42×10^{-5}	2	2.21
149	1.0	1.0	Y	1.0	0.5	8	28	4:46	1.29×10^{7}	8	12.10	8.38×10^{-5}	8	7.11
									HYBRID Mk 3 ALGORITHM					
220	1.0	1.0	N	-	1.0	11	11	3:06	2.38×10^{7}	3	21.72	1.55×10^{-4}	3	12.23
225	1.0	1.0	Y	0.0	0.1	6	16	3:56	3.01×10^{6}	2	2.63	1.96×10^{-5}	2	1.87
248	1.0	1.0	Y	0.0	0.5	5	12	2:56	1.32×10^{7}	2	11.40	8.58×10^{-5}	2	8.87
316	1.0	1.0	Y	0.5	0.1	7	17	4:24	1.38×10^{6}	6	0.82	8.99×10^{-6}	6	0.53
317	1.0	1.0	Y	0.5	0.5	7	17	4:07	5.46×10^{6}	6	3.07	3.55×10^{-5}	6	2.00
226	1.0	1.0	Y	1.0	0.1	7	17	4:19	3.72×10^{6}	2	3.34	2.42×10^{-5}	2	2.21
249	1.0	1.0	Y	1.0	0.5	8	22	5:09	1.29×10^{7}	8	12.10	8.39×10^{-5}	8	7.11

Table 3 Stress relaxation problem - linear subincrementation scheme

Run No.	θ_S	S_S	UPDATE	θ_Y	S_Y	No. incs	No. iter	CPU m:s	Stress (N/m^2) Max Err.	Step	RMS%	Creep Strain Max Err.	Step	RMS%
INITIAL STIFFNESS ALGORITHM														
108	0.5	1.0	N	-	1.0	6	18	13:13	1.46×10^{7}	6	8.41	9.47×10^{-5}	6	5.57
113	0.5	1.0	Y	0.0	0.1	6	19	3:51	7.39×10^{6}	2	6.38	4.81×10^{-5}	2	4.40
142	0.5	1.0	Y	0.0	0.5	4	13	2:24	1.67×10^{7}	2	14.00	1.09×10^{-4}	2	12.70
714	0.5	1.0	Y	0.5	0.1	6	17	3:37	4.35×10^{6}	2	3.74	2.83×10^{-5}	2	2.63
715	0.5	1.0	Y	0.5	0.5	6	18	3:20	2.38×10^{6}	2	2.25	1.55×10^{-5}	2	1.60
114	0.5	1.0	Y	1.0	0.1	6	16	3:27	1.20×10^{6}	1	1.11	7.66×10^{-6}	1	0.78
143	0.5	1.0	Y	1.0	0.5	8	21	3:53	9.33×10^{6}	2	8.29	6.07×10^{-5}	2	5.17
HYBRID Mk 3 ALGORITHM														
208	0.5	1.0	N	-	1.0	6	6	1:45	1.46×10^{7}	6	8.42	9.47×10^{-5}	6	5.57
213	0.5	1.0	Y	0.0	0.1	6	14	3:50	7.39×10^{6}	2	6.38	4.81×10^{-5}	2	4.40
242	0.5	1.0	Y	0.0	0.5	4	9	2:19	1.67×10^{7}	2	14.00	1.09×10^{-4}	2	12.70
314	0.5	1.0	Y	0.5	0.1	6	13	3:40	4.36×10^{6}	2	3.74	2.83×10^{-5}	2	2.63
315	0.5	1.0	Y	0.5	0.5	6	14	3:35	2.38×10^{6}	2	2.25	1.55×10^{-5}	2	1.60
214	0.5	1.0	Y	1.0	0.1	6	12	3:38	1.20×10^{6}	1	1.10	7.65×10^{-6}	1	0.78
243	0.5	1.0	Y	1.0	0.5	8	19	4:55	9.32×10^{6}	2	8.29	6.06×10^{-5}	2	5.17

Table 4 Stress relaxation problem - nonlinear subincrementation scheme

Run No.	θ_S	S_S	UPDATE	θ_T	S_T	No. incs	No. iter	CPU m:s	Stress (N/m^2) Max Err.	Step	RMS%	Creep Strain Max Err.	Step	RMS%
INITIAL STIFFNESS ALGORITHM														
120	1.0	1.0	N	-	1.0	11	28	5:02	2.38×10^{7}	3	21.70	1.55×10^{-4}	3	12.23
125	1.0	1.0	Y	0.0	0.1	6	19	3:58	7.58×10^{6}	2	6.50	4.95×10^{-5}	2	4.49
148	1.0	1.0	Y	0.0	0.5	4	13	2:28	1.72×10^{7}	2	14.60	1.12×10^{-4}	2	13.21
716	1.0	1.0	Y	0.5	0.1	6	20	4:18	5.12×10^{6}	2	4.39	3.33×10^{-5}	2	3.06
717	1.0	1.0	Y	0.5	0.5	6	20	3:40	2.87×10^{6}	2	2.61	1.86×10^{-5}	2	1.85
126	1.0	1.0	Y	1.0	0.1	6	20	4:20	2.75×10^{6}	2	2.29	1.79×10^{-5}	2	1.62
149	1.0	1.0	Y	1.0	0.5	8	26	4:49	8.99×10^{6}	2	8.05	5.85×10^{-5}	2	5.01
HYBRID Mk 3 ALGORITHM														
220	1.0	1.0	N	-	1.0	11	11	3:11	2.38×10^{7}	3	21.70	1.55×10^{-4}	3	12.23
225	1.0	1.0	Y	0.0	0.1	6	15	4:05	7.58×10^{6}	2	6.50	4.95×10^{-5}	2	4.49
248	1.0	1.0	Y	0.0	0.5	4	10	2:30	1.72×10^{7}	2	14.60	1.12×10^{-4}	2	13.21
316	1.0	1.0	Y	0.5	0.1	6	15	4:16	5.12×10^{6}	2	4.38	3.33×10^{-5}	2	3.06
317	1.0	1.0	Y	0.5	0.5	6	15	3:46	2.87×10^{6}	2	2.61	1.87×10^{-5}	2	1.85
226	1.0	1.0	Y	1.0	0.1	6	15	4:13	2.75×10^{6}	2	2.29	1.79×10^{-5}	2	1.62
249	1.0	1.0	Y	1.0	0.5	8	20	5:02	8.99×10^{6}	2	8.05	5.85×10^{-5}	2	5.01

Table 5 Stress relaxation problem - nonlinear subincrementation scheme

Run No.	θ_S	S_S	UPDATE	θ_T	S_T	No. incs	No. iter	CPU time min:sec	Creep Strain Max Error	Step	RMS Error%
								INITIAL STIFFNESS ALGORITHM			
102	0.0	1.0	N	-	1.0	31	92	15:45	1.78×10^{-3}	31	7.81
105	0.0	1.0	Y	0.0	0.1	36	267	50:36	4.50×10^{-4}	36	2.06
140	0.0	1.0	Y	0.0	0.5	35	226	38:08	1.89×10^{-5}	35	0.09
718	0.0	1.0	Y	0.5	0.1	36	274	53:38	5.55×10^{-4}	36	2.53
719	0.0	1.0	Y	0.5	0.5	36	268	45:08	4.18×10^{-4}	36	1.91
106	0.0	1.0	Y	1.0	0.1	36	279	54:42	6.44×10^{-4}	36	2.91
141	0.0	1.0	Y	1.0	0.5	37	319	53:41	1.14×10^{-3}	37	5.08
								HYBRID Mk 3 ALGORITHM			
202	0.0	1.0	N	-	1.0	31	31	8:40	1.72×10^{-3}	31	7.56
205	0.0	1.0	Y	0.0	0.1	36	182	50:54	4.42×10^{-4}	36	2.04
240	0.0	1.0	Y	0.0	0.5	35	153	40:20	2.69×10^{-5}	35	0.11
318	0.0	1.0	Y	0.5	0.1	36	193	44:06	5.57×10^{-4}	36	2.54
319	0.0	1.0	Y	0.5	0.5	36	184	37:29	4.06×10^{-4}	36	1.86
206	0.0	1.0	Y	1.0	0.1	36	199	57:44	6.50×10^{-4}	36	2.92
241	0.0	1.0	Y	1.0	0.5	37	235	62:25	1.14×10^{-3}	37	5.10

Table 6 Constant stress problem - linear subincrementation scheme

Run No.	θ_s	S_s	UPDATE	θ_r	S_r	No. incs	No. iter	CPU time min:sec	Creep Strain Max Error	Step	RMS Error%
								INITIAL STIFFNESS ALGORITHM			
108	0.5	1.0	N	-	1.0	34	309	48:59	7.60×10^{-4}	34	3.57
113	0.5	1.0	Y	0.0	0.1	36	334	60:41	4.14×10^{-4}	36	1.92
142	0.5	1.0	Y	0.0	0.5	35	284	46:12	4.79×10^{-5}	35	0.17
714	0.5	1.0	Y	0.5	0.1	36	350	66:12	5.26×10^{-4}	36	2.39
715	0.5	1.0	Y	0.5	0.5	36	347	57:23	3.94×10^{-4}	36	1.79
114	0.5	1.0	Y	1.0	0.1	36	357	67:18	6.09×10^{-4}	36	2.75
143	0.5	1.0	Y	1.0	0.5	37	417	68:18	1.12×10^{-3}	37	4.99
								HYBRID Mk 3 ALGORITHM			
208	0.5	1.0	N	-	1.0	34	34	9:35	6.17×10^{-4}	34	2.98
213	0.5	1.0	Y	0.0	0.1	36	93	26:32	4.49×10^{-4}	36	2.07
242	0.5	1.0	Y	0.0	0.5	35	44	12:37	5.61×10^{-5}	35	0.27
314	0.5	1.0	Y	0.5	0.1	36	108	31:44	5.60×10^{-4}	36	2.55
315	0.5	1.0	Y	0.5	0.5	36	95	23:03	4.12×10^{-4}	36	1.89
214	0.5	1.0	Y	1.0	0.1	36	120	34:50	6.36×10^{-4}	36	2.87
243	0.5	1.0	Y	1.0	0.5	37	164	43:33	1.14×10^{-3}	37	5.06

Table 7 Constant stress problem - Linear subincrementation scheme

Run No.	θ_S	S_S	UPDATE	θ_Y	S_Y	No. incs	No. iter	CPU time min:sec	Creep Strain Max Error	Step	RMS Error%
INITIAL STIFFNESS ALGORITHM											
120	1.0	1.0	N	-	1.0	37	480	77:16	5.44×10^{-4}	37	2.18
125	1.0	1.0	Y	0.0	0.1	36	352	65:31	4.25×10^{-4}	36	1.94
148	1.0	1.0	Y	0.0	0.5	35	304	49:01	3.46×10^{-5}	34	0.13
716	1.0	1.0	Y	0.5	0.1	36	360	70:03	5.24×10^{-4}	36	2.40
717	1.0	1.0	Y	0.5	0.5	36	356	60:37	3.96×10^{-4}	36	1.81
126	1.0	1.0	Y	1.0	0.1	36	384	74:39	6.32×10^{-4}	36	2.83
149	1.0	1.0	Y	1.0	0.5	37	428	72:01	1.11×10^{-3}	37	4.93
HYBRID Mk3 ALGORITHM											
220	1.0	1.0	N	-	1.0	38	38	11:14	7.46×10^{-4}	38	3.02
225	1.0	1.0	Y	0.0	0.1	36	153	36:08	7.63×10^{-4}	36	3.36
248	1.0	1.0	Y	0.0	0.5	35	152	32:22	2.45×10^{-4}	35	1.07
316	1.0	1.0	Y	0.5	0.1	36	150	37:04	8.38×10^{-4}	36	3.67
317	1.0	1.0	Y	0.5	0.5	36	152	32:52	7.32×10^{-4}	36	3.21
226	1.0	1.0	Y	1.0	0.1	37	150	36:16	9.39×10^{-4}	37	4.12
249	1.0	1.0	Y	1.0	0.5	38	108	25:22	1.48×10^{-3}	38	6.47

Table 8 Constant stress problem - Linear subincrementation scheme

Run No.	θ_s	S_s	UPDATE	θ_r	S_r	No. incs	No. iter	CPU time min:sec	Creep Strain Max Error	Step	RMS Error%
								INITIAL STIFFNESS ALGORITHM			
108	0.5	1.0	N	-	1.0	34	309	49:05	7.60×10^{-4}	34	3.57
113	0.5	1.0	Y	0.0	0.1	35	283	53:46	2.82×10^{-4}	35	1.21
142	0.5	1.0	Y	0.0	0.5	34	225	38:15	8.52×10^{-4}	34	3.71
714	0.5	1.0	Y	0.5	0.1	35	290	59:37	1.84×10^{-4}	35	0.76
715	0.5	1.0	Y	0.5	0.5	35	295	50:47	2.34×10^{-4}	35	1.01
114	0.5	1.0	Y	1.0	0.1	35	314	63:49	1.55×10^{-5}	3	0.06
143	0.5	1.0	Y	1.0	0.5	37	383	65:31	5.59×10^{-4}	37	2.45
								HYBRID Mk 3 ALGORITHM			
208	0.5	1.0	N	-	1.0	34	34	9:44	6.17×10^{-4}	34	2.98
213	0.5	1.0	Y	0.0	0.1	35	52	15:51	2.60×10^{-5}	34	0.12
242	0.5	1.0	Y	0.0	0.5	34	92	22:54	5.67×10^{-4}	34	2.55
314	0.5	1.0	Y	0.5	0.1	35	48	14:52	5.71×10^{-5}	35	0.26
315	0.5	1.0	Y	0.5	0.5	35	47	13:09	2.59×10^{-5}	35	0.13
214	0.5	1.0	Y	1.0	0.1	35	48	14:42	2.61×10^{-5}	35	0.12
243	0.5	1.0	Y	1.0	0.5	37	131	29:53	5.55×10^{-4}	37	2.45

Table 9 Constant stress problem - Nonlinear subincrementation scheme

Run No.	θ_s	S_s	UPDATE	θ_r	S_r	No. incs	No. iter	CPU time min:sec	Creep Strain Max Error	Step	RMS Error%
					INITIAL STIFFNESS ALGORITHM						
120	1.0	1.0	N	-	1.0	37	480	76:18	5.44×10^{-4}	37	2.18
125	1.0	1.0	Y	0.0	0.1	35	284	54:32	4.75×10^{-4}	35	2.07
148	1.0	1.0	Y	0.0	0.5	33	217	36:27	1.10×10^{-3}	33	4.86
716	1.0	1.0	Y	0.5	0.1	35	286	57:26	4.15×10^{-4}	35	1.81
717	1.0	1.0	Y	0.5	0.5	35	289	49:38	4.72×10^{-4}	35	2.09
126	1.0	1.0	Y	1.0	0.1	35	317	64:06	2.29×10^{-4}	35	1.01
149	1.0	1.0	Y	1.0	0.5	36	382	65:20	3.68×10^{-4}	36	1.59
					HYBRID Mk 3 ALGORITHM						
220	1.0	1.0	N	-	1.0	38	38	10:47	7.46×10^{-4}	38	3.02
225	1.0	1.0	Y	0.0	0.1	35	156	37:38	2.24×10^{-4}	35	1.03
248	1.0	1.0	Y	0.0	0.5	34	136	29:57	8.48×10^{-4}	34	3.81
316	1.0	1.0	Y	0.5	0.1	35	157	39:29	1.12×10^{-4}	35	0.56
317	1.0	1.0	Y	0.5	0.5	35	155	33:54	1.47×10^{-4}	35	0.74
226	1.0	1.0	Y	1.0	0.1	35	158	39:07	3.61×10^{-5}	35	0.13
249	1.0	1.0	Y	1.0	0.5	37	146	32:20	6.80×10^{-4}	37	2.90

Table 10 Constant stress problem -
Nonlinear subincrementation scheme

Run No.	θ_S	S_S	UPDATE	θ_Y	S_Y	No. incs	No. iter	CPU min:sec	Creep Strain Max Err	Step	RMS Err%	Iter/inc Mean	Max
								INITIAL STIFFNESS ALGORITHM					
								$\bar{X}_{ec}$ = 0.3					
961	0.500	5.0	Y	0.500	0.5	16	256	48:41	6.16×10^{-4}	16	3.04	16.00	21
620	0.667	5.0	Y	0.667	0.5	20	254	52:52	3.32×10^{-4}	20	1.46	12.70	14
610	0.878	5.0	Y	0.878	0.5	26	269	55:58	3.49×10^{-4}	26	1.53	10.35	11
600	1.000	5.0	Y	1.000	0.5	29	280	58:09	3.35×10^{-4}	29	1.47	9.66	10
								$\bar{X}_{ec}$ = 0.2					
951	0.500	5.0	Y	0.500	0.5	11	278	53:14	2.57×10^{-3}	11	12.35	25.27	52
811	0.667	5.0	Y	0.667	0.5	13	234	44:35	2.83×10^{-4}	13	1.20	18.00	21
611	0.878	5.0	Y	0.878	0.5	17	243	50:59	5.34×10^{-4}	17	2.41	14.29	17
601	1.000	5.0	Y	1.000	0.5	19	250	51:57	5.19×10^{-4}	19	2.34	13.16	15
								$\bar{X}_{ec}$ = 0.1					
632	0.500	5.0	Y	0.500	0.5	8	296	59:21	3.65×10^{-3}	8	17.59	37.00	75
622	0.667	5.0	Y	0.667	0.5	8	223	44:34	8.22×10^{-4}	8	3.67	27.88	42
612	0.878	5.0	Y	0.878	0.5	10	217	43:16	1.26×10^{-3}	10	5.78	21.70	33
602	1.000	5.0	Y	1.000	0.5	11	222	46:01	1.11×10^{-3}	11	5.09	20.18	29
								$\bar{X}_{ec}$ = 0.0					
633	0.500	5.0	Y	0.500	0.5	8	296	59:29	3.65×10^{-3}	8	17.59	37.00	75
623	0.667	5.0	Y	0.667	0.5	8	216	42:59	1.58×10^{-3}	8	7.25	27.00	46
613	0.878	5.0	Y	0.878	0.5	7	166	33:16	4.36×10^{-3}	7	20.24	23.71	41
603	1.000	5.0	Y	1.000	0.5	7	150	29:53	5.36×10^{-3}	7	24.95	21.43	39

Table 11 Investigation of variation in $\bar{X}_{ec}$ -

Constant stress problem - nonlinear subincrementation

Run No.	θ_S	S_S	UPDATE	θ_Y	S_Y	No. incs	No. iter	CPU min:sec	Creep Strain Max Err	Step	RMS Err%	Iter/inc Mean	Max
								INITIAL STIFFNESS ALGORITHM					
								$\bar{X}_{ec} = 0.3$					
961	0.500	5.0	Y	0.500	0.5	17	389	74:29	3.14×10^{-3}	17	14.23	22.88	39
620	0.667	5.0	Y	0.667	0.5	22	428	80:27	4.76×10^{-3}	22	22.28	19.45	81
610	0.878	5.0	Y	0.878	0.5	26	344	66:03	4.15×10^{-4}	26	1.75	13.23	14
600	1.000	5.0	Y	1.000	0.5	30	362	66:33	1.05×10^{-4}	29	0.52	12.07	13
								$\bar{X}_{ec} = 0.2$					
951	0.500	5.0	Y	0.500	0.5	13	538	106:11	8.97×10^{-3}	13	40.97	41.38	102
621	0.667	5.0	Y	0.667	0.5	15	428	81:43	4.44×10^{-3}	15	20.07	28.53	48
611	0.878	5.0	Y	0.878	0.5	18	391	74:05	2.79×10^{-3}	18	12.62	21.72	38
601	1.000	5.0	Y	1.000	0.5	Did not converge on inc. 4 after 120 iterations							
								$\bar{X}_{ec} = 0.1$					
632	0.500	5.0	Y	0.500	0.5	Did not converge on inc. 4 after 120 iterations							
622	0.667	5.0	Y	0.667	0.5	Did not converge on inc. 4 after 120 iterations							
612	0.878	5.0	Y	0.878	0.5	Did not converge on inc. 4 after 120 iterations							
602	1.000	5.0	Y	1.000	0.5	Did not converge on inc. 4 after 120 iterations							

Table 12 Investigation of variation in $\bar{X}_{ec}$ - Constant stress problem - Linear subincrementation scheme

Run No.	θ_s	S_s	UPDATE	θ_r	S_r	No. incs	No. iter	CPU m:s	Creep Strain Max Err	Step	RMS Err%	Iter/inc Mean	Max
									INITIAL STIFFNESS ALGORITHM				
									$\bar{X}_{ec} = 0.0$				
653	0.500	2.50	Y	0.500	0.250	15	245	46:53	2.97×10^{-4}	15	1.31	16.33	24
652	0.667	2.00	Y	0.667	0.200	18	255	49:15	6.74×10^{-5}	11	0.46	14.17	18
651	0.878	1.35	Y	0.878	0.135	26	278	53:31	3.18×10^{-4}	26	1.39	10.69	12
650	1.000	1.20	Y	1.000	0.120	29	288	57:53	3.12×10^{-4}	29	1.37	9.93	11

Table 13 Constant stress problem - Results using only S_s to control timestep size

Run No.	θ_s	S_s	UPDATE	θ_Y	S_Y	No. incs	No. iter	CPU time min:sec	Creep Strain Max Error	Step	RMS Error%
					INITIAL STIFFNESS ALGORITHM						
					$\bar{X}_{ec}$ = 0.30						
975	0.500	5.0	Y	0.500	0.05	16	258	137:16	8.11×10^{-4}	16	3.98
961	0.500	5.0	Y	0.500	0.50	16	256	48:41	6.16×10^{-4}	16	3.04
800	0.500	5.0	Y	0.500	1.00	16	248	42:30	3.38×10^{-5}	3	0.16
801	0.500	5.0	Y	0.500	2.50	15	250	40:10	4.66×10^{-4}	15	2.34
802	0.500	5.0	Y	0.500	5.00	15	262	41:38	4.37×10^{-4}	15	2.18
					$\bar{X}_{ec}$ = 0.20						
810	0.667	5.0	Y	0.667	0.05	13	235	125:49	9.33×10^{-5}	3	0.60
811	0.667	5.0	Y	0.667	0.50	13	234	44:35	2.83×10^{-4}	13	1.20
812	0.667	5.0	Y	0.667	1.00	13	236	40:43	7.52×10^{-4}	13	3.46
813	0.667	5.0	Y	0.667	2.50	13	260	41:13	3.52×10^{-4}	13	1.74
814	0.667	5.0	Y	0.667	5.00	12	288	45:22	1.35×10^{-3}	12	6.95
					$\bar{X}_{ec}$ = 0.15						
820	0.878	5.0	Y	0.878	0.05	13	221	114:04	8.54×10^{-4}	13	3.82
821	0.878	5.0	Y	0.878	0.50	13	229	43:13	9.17×10^{-4}	13	4.19
822	0.878	5.0	Y	0.878	1.00	13	243	41:33	1.05×10^{-3}	13	4.93
823	0.878	5.0	Y	0.878	2.50	13	305	49:22	9.23×10^{-4}	13	4.08
824	0.878	5.0	Y	0.878	5.00	13	343	53:18	2.38×10^{-3}	13	12.70

Table 14 Constant stress problem - Investigation of optimum number of subincrements

Run No.	θ_s	S_s	UPDATE	θ_r	S_r	No. incs	No. iter	CPU time min:sec	Creep Strain Max Error	Step	RMS Error%
						INITIAL STIFFNESS ALGORITHM					
						$\bar{X}_{ec} = 0.14$					
830	1.000	5.0	Y	1.000	0.05	14	220	115:26	9.95×10^{-4}	14	4.49
831	1.000	5.0	Y	1.000	0.50	14	234	45:41	9.53×10^{-4}	14	4.38
832	1.000	5.0	Y	1.000	1.00	14	250	44:20	7.29×10^{-4}	14	3.44
833	1.000	5.0	Y	1.000	2.50	14	302	49:37	1.29×10^{-3}	14	6.69
834	1.000	5.0	Y	1.000	5.00	16	440	68:42	1.19×10^{-3}	16	5.01
						$\bar{X}_{ec} = 0.0$					
856	0.500	5.0	Y	0.500	0.25	8	288	66:24	3.17×10^{-3}	8	15.26
633	0.500	5.0	Y	0.500	0.50	8	296	59:29	3.65×10^{-3}	8	17.59
857	0.500	5.0	Y	0.500	1.00	8	255	43:54	8.44×10^{-4}	8	4.22
858	0.500	5.0	Y	0.500	2.50	7	247	39:53	5.98×10^{-5}	3	0.28
859	0.500	5.0	Y	0.500	5.00	6	175	27:29	6.61×10^{-3}	6	31.35
						$\bar{X}_{ec} = 0.0$					
851	1.000	5.0	Y	1.000	0.05	7	140	72:11	5.58×10^{-3}	7	25.85
852	1.000	5.0	Y	1.000	0.10	7	141	47:29	5.58×10^{-3}	7	25.89
853	1.000	5.0	Y	1.000	0.25	7	144	33:33	5.66×10^{-3}	7	26.29
603	1.000	5.0	Y	1.000	0.50	7	150	29:53	5.36×10^{-3}	7	24.95
854	1.000	5.0	Y	1.000	2.50	Did not converge on inc. 6 after 250 iters.					
855	1.000	5.0	Y	1.000	5.00	Did not converge on inc. 6 after 250 iters.					

Table 14 (continued) Constant stress problem - Investigation of optimum number of subincrements

Run No.	No. incs	No. iter	CPU min:sec	Creep Strain Max. Error	Step	RMS Error%	Algorithm
			$\theta_S = 0.5$, $S_S = 5.0$, $\theta_Y = 0.5$, $S_Y = 0.5$, $\bar{X}_{ec} = 0.30$				
961	16	256	48:41	6.16×10^{-4}	16	3.04	Initial Stiffness
963	16	75	14:36	6.31×10^{-4}	16	3.08	IS + Crisfield
964	16	76	14:53	6.24×10^{-4}	16	3.05	IS + BFGS (creep)
965			Diverged on increment 4				IS + BFGS
966	16	41	14:18	7.48×10^{-4}	16	3.58	IS + line search
967	16	37	12:16	8.02×10^{-4}	16	3.77	IS + Crisfield + ls
968	16	39	14:05	6.98×10^{-4}	16	3.34	IS + BFGS (creep) + ls
969	16	37	13:17	8.02×10^{-4}	16	3.77	IS + BFGS + ls
970	16	55	13:32	9.46×10^{-4}	16	4.44	Hybrid Mk 3

Table 15 Constant stress problem - Performance of quasi-Newton algorithms

Run No.	Algorithm	No. incs	No. iter	VAX CPU h:m:s	AP CPU m:s	Est. stand alone VAX time (hr)
800	Initial Stiffness	4	44	3:58:56	2:17	4.7
810	Tangent Stiffness	4	9	1:17:42	20:33	7.5
820	Hybrid Mk 1	4	27	2:43:34	9:08	5.5
830	Hybrid Mk 2	4	16	1:42:46	9:08	4.5
840	Hybrid Mk 3	4	12	1:29:55	15:59	6.3
850	IS + BFGS	4	18	1:45:50	2:17	2.4
860	IS + Crisfield	4	19	1:49:19	2:17	2.5
870	IS + line search	4	23	3:35:41	2:17	4.3
880	IS + BFGS + line search	4	12	1:53:38	2:17	2.6
885	IS + Crisfield + line search	4	13	2:01:15	2:17	2.7
900	Tangent Stiffness + l.s.	4	9	1:25:56	20:33	7.6
910	Hybrid Mk 1 + BFGS	4	15	1:38:53	9:08	4.4
915	Hybrid Mk 1 + Crisfield	4	17	1:50:46	9:08	4.6
920	Hybrid Mk 1 + line search	4	15	2:35:38	9:08	5.3
930	Hyb. Mk 1 + BFGS + l.s.	4	13	1:59:40	9:08	4.7
935	Hyb. Mk 1 + Crisfield + l.s.	4	13	2:04:53	9:08	4.8
950	Hybrid Mk 2 + BFGS	4	13	1:26:49	9:08	4.2
955	Hybrid Mk 2 + Crisfield	4	13	1:25:58	9:08	4.2
960	Hybrid Mk 2 + line search	4	9	1:21:27	9:08	4.1
970	Hyb. Mk 2 + BFGS + l.s.	4	10	1:26:14	9:08	4.2
975	Hyb. Mk 2 + Crisfield + l.s.	4	10	1:30:32	9:08	4.2

Table 16 Elasto-plastic analysis of hollow sphere under internal pressure

Temp. (°C)	Stress (N/m^2) Mendelson & Spero	Initial Stiffness	Tangent Stiffness	Plastic Strain Mendelson & Spero	Initial Stiffness	Tangent Stiffness
995.6	-3.721×10^8	-3.721×10^8	-3.729×10^8	-3.737×10^{-3}	-3.716×10^{-3}	-3.723×10^{-3}
846.9	-5.846×10^8	-5.836×10^8	-5.845×10^8	-0.800×10^{-4}	-7.960×10^{-4}	-7.957×10^{-4}
738.1	-5.295×10^8	-5.276×10^8	-5.287×10^8	0.0	0.0	0.0
589.4	-2.410×10^8	-2.389×10^8	-2.401×10^8	0.0	0.0	0.0
480.6	-0.050×10^8	-0.030×10^8	-0.043×10^8	0.0	0.0	0.0
331.9	3.454×10^8	3.481×10^8	3.468×10^8	0.0	0.0	0.0
223.1	5.955×10^8	5.902×10^8	5.897×10^8	1.124×10^{-4}	1.448×10^{-4}	1.410×10^{-4}
74.4	7.913×10^8	7.872×10^8	7.871×10^8	1.051×10^{-3}	1.080×10^{-3}	1.074×10^{-3}

Table 17 Solution for thermal plate problem at various positions across plate width

Test No.	Stress (MPa)	Temperature (^{0}C)	Rupture life (hrs)	Specimen type
1	400.0	850	137.2	Hollow
2	492.0	900	4.3	Hollow
3	309.0	900	62.2	Hollow
4	207.0	950	62.5	Hollow
5	238.0	950	20.0	Hollow
6	400.0	900	9.6	Hollow
7	194.0	950	136.4	Hollow
8	237.0	950	43.0	Hollow
9	262.0	950	23.2	Hollow
10	309.0	950	7.3	Hollow
11	400.0	900	10.9	Hollow
12	140.0	950	411.1	Hollow
13	405.0	950	0.8	Hollow
14	400.0	950	0.7	Hollow
15	566.0	850	17.6	Hollow
16	492.0	850	37.7	Hollow
17	154.0	950	416.6	Hollow
18	350.0	900	40.0	Hollow
19	400.0	850	172.2	Hollow
20	400.0	800	2027.7	Hollow
21	400.0	900	4.9	Hollow
22	330.0	850	502.7	Hollow
23	450.0	900	4.8	Hollow
24	564.0	800	109.4	Hollow

Table 18 Uniaxial creep tests on conventionally cast Mar-M 002 material

Test No.	Stress (MPa)	Temperature (^{0}C)	Rupture life (hrs)	Specimen type
1	181.0	950	552.7	Solid
2	185.0	950	419.4	Solid
3	230.0	950	124.1	Solid
4	300.0	950	28.0	Solid
5	275.0	950	65.8	Solid
6	250.0	950	39.7	Hollow
7	350.0	950	5.3	Hollow
8	400.0	900	23.7	Solid
9	400.0	950	3.1	Solid
10	250.0	950	108.3	Solid
11	400.0	950	0.9	Hollow
12	400.0	850	283.3	Hollow
13	400.0	900	10.1	Hollow
14	250.0	950	85.0	Hollow
15	185.0	950	347.2	Hollow
16	450.0	850	155.6	Hollow
17	275.0	950	57.7	Hollow
18	500.0	850	65.5	Hollow
19	140.0	950	1380.5	Solid
20	200.0	950	160.8	Hollow

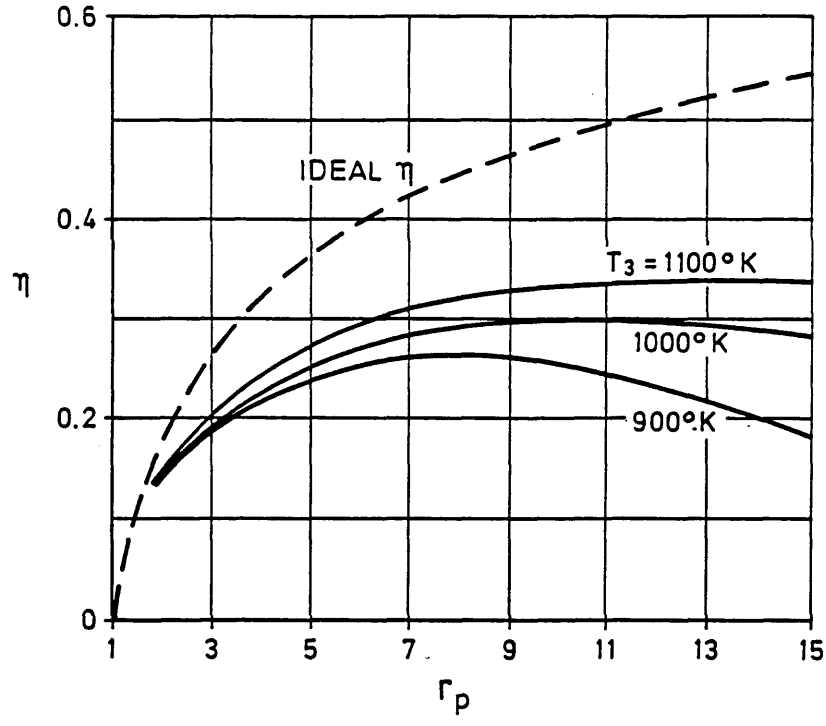
Table 19 Uniaxial creep tests on directionally solidified Mar-M 002 material

Test No.	Stress (MPa)	Temperature (^{0}C)	Rupture life (hrs)	Specimen type
21	160.0	950	1405.5	Solid
22	300.0	850	245.5	Hollow
23	800.0	760	127.2	Solid
24	700.0	760	154.2	Solid
25	950.0	760	3.2	Solid
26	950.0	760	0.6	Solid
27	140.0	950	2447.0	Hollow
28	300.0	950	25.6	Hollow
29	400.0	850	544.4	Solid
30	350.0	900	50.8	Hollow
31	450.0	900	7.3	Hollow
32	500.0	900	3.3	Hollow
33	350.0	850	527.7	Hollow
34	250.0	900	700.0	Hollow
35	400.0	800	3833.3	Hollow
36	500.0	800	413.8	Hollow
37	300.0	850	1019.4	Hollow
38	500.0	800	522.2	Hollow

Table 19 (continued) Uniaxial creep tests on directionally solidified Mar-M 002 material

Test No.	Temp (°C)	Tensile (MPa)	Shear (MPa)	Stress Von Mises (MPa)	Maximum Principal (MPa)	Rupture Life (hr)	Remarks
1	950	120.0	0.0	120	120	174	
2	950	120.0	0.0	120	120	2460	Solid specimen
3	950	0.0	120.0	208	120	291	Solid specimen
4	950	0.0	109.0	189	109	250	
5	950	0.0	120.0	208	120	224	
6	950	33.9	101.6	179	120	387	
7	950	90.0	60.0	137	120	1345	
8	950	58.0	87.0	161	121	1095	
9	950	120.0	0.0	120	120	1984	
10	950	39.2	117.6	207	139	247	
11	950	75.0	112.5	209	156	214	
12	950	136.0	90.7	208	181	141	
13	950	0.0	173.0	300	173	24	
14	1050	30.7	46.0	85	64	141	
15	950	56.7	170.0	300	200	115	
16	950	300.0	0.0	300	300	10	
17	950	196.0	131.0	300	262	20	
18	950	56.7	170.0	300	200	28	
19	1050	0.0	49.0	85	49	188	
20	1050	85.0	0.0	85	85	29	
21	1050	16.0	48.0	85	57	-	Furnace failure
22	950	180.0	60.0	208	198	209	
23	950	191.0	48.0	208	202	54	
24	950	202.0	28.6	208	206	135	
25	1050	60.0	0.0	60	60	628	
26	1050	24.6	73.8	130	87	37	
27	1050	0.0	34.6	60	35	3194	
28	1050	78.0	19.5	85	83	375	
29	1050	119.3	29.8	130	126	1.5	
30	1050	130.0	0.0	130	130	60	
31	1050	119.3	29.8	130	126	60	

Table 20 Tension-torsion creep test series on conventionally cast Mar-M 002



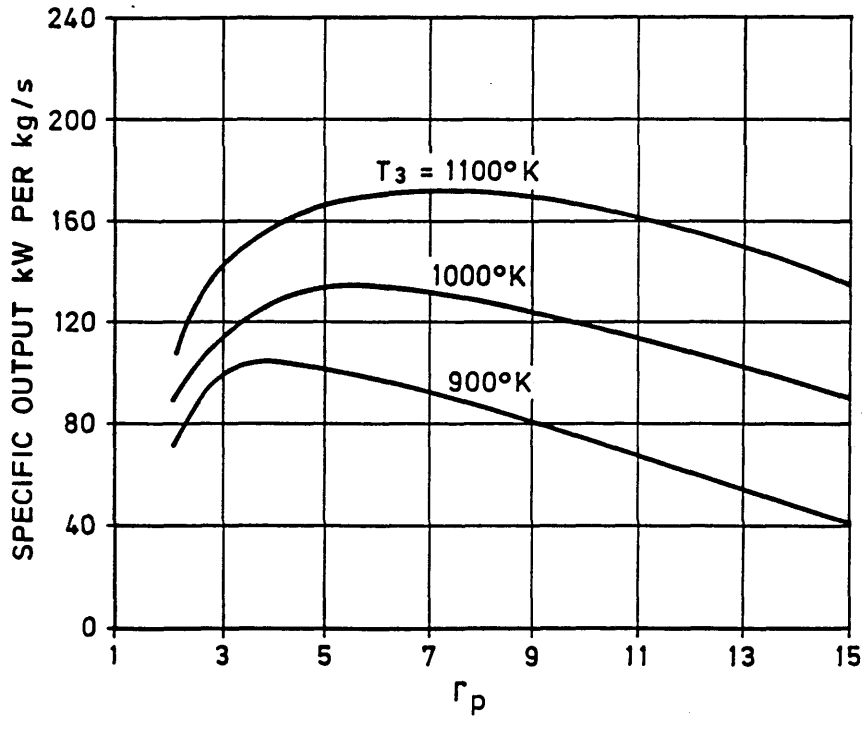


Fig. 1 The simple gas turbine cycle

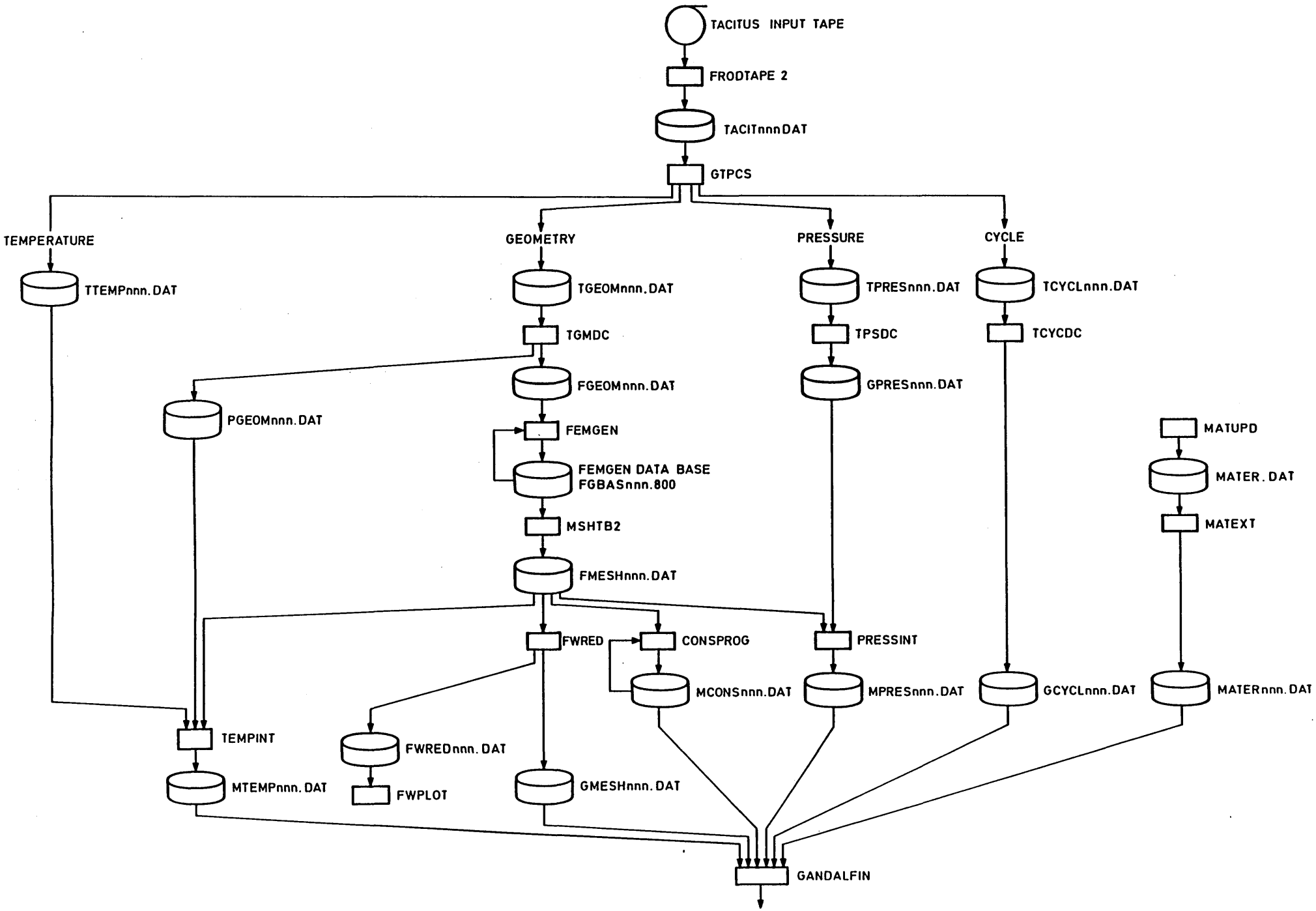


Fig. 2 Structure of GANDALF data input system

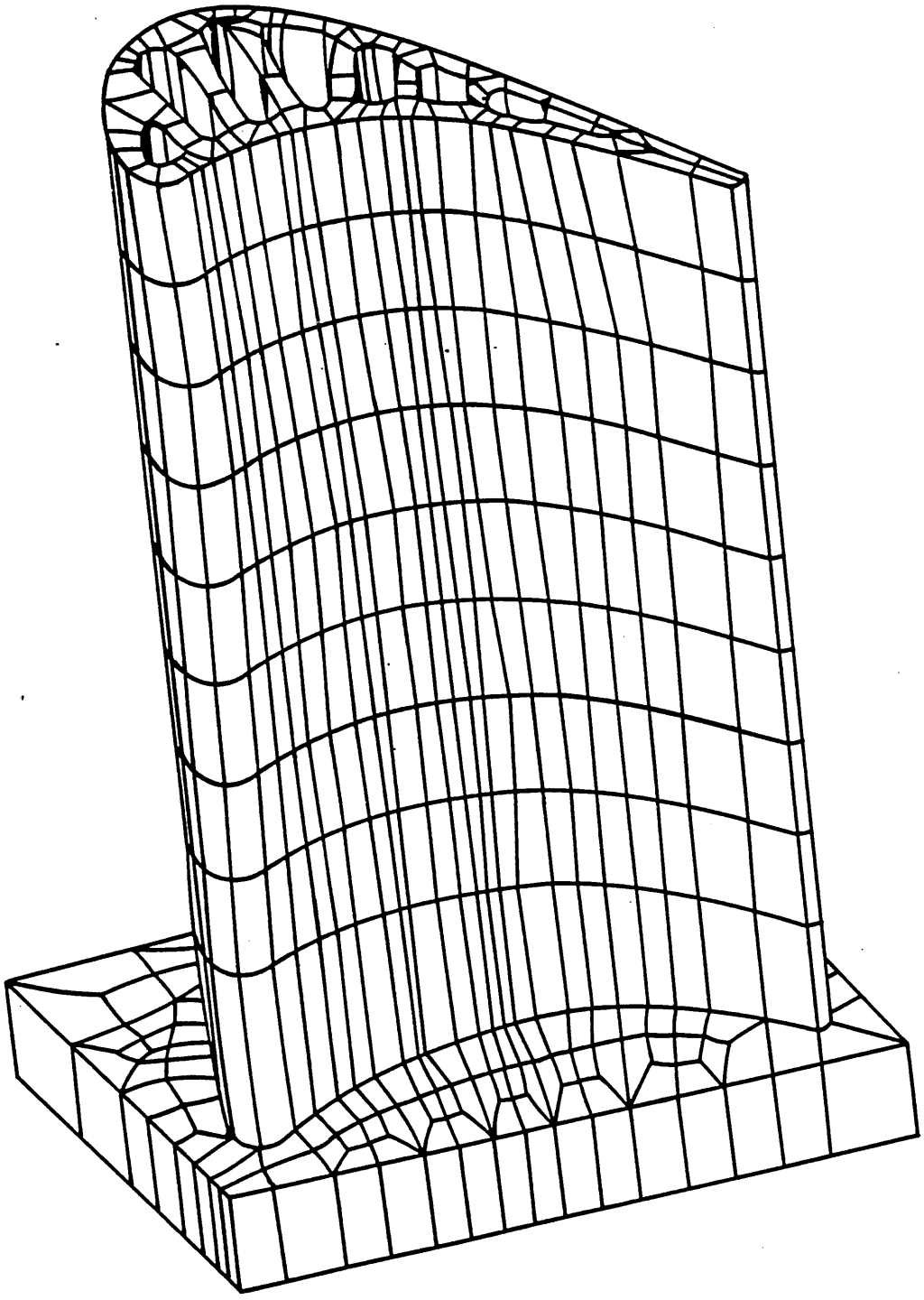
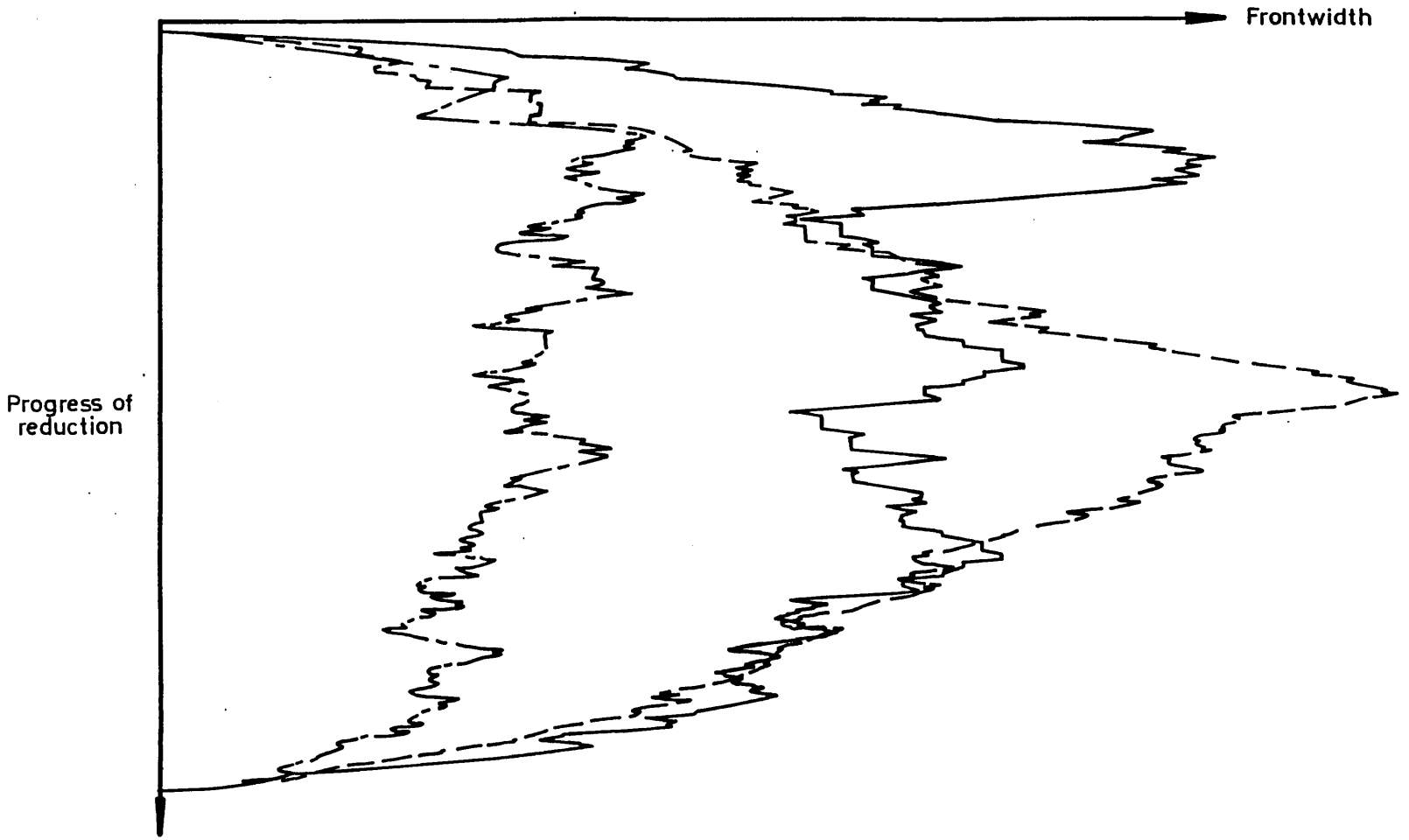


Fig. 3 Typical turbine blade finite element mesh



Frontwidth statistics - structure has 584 elements with 3832 nodes

Original numbering gives 6.827E+09 cycles with maximum Frontwidth of 1629
G.P.S. numbering gives 1.600E+09 cycles with maximum Frontwidth of 768
Razzaque numbering gives 7.307E+09 cycles with maximum Frontwidth of 1911

G.P.S. NUMBERING SELECTED

Fig 4 Variation of frontwidth during reduction as plotted by FWPLOT

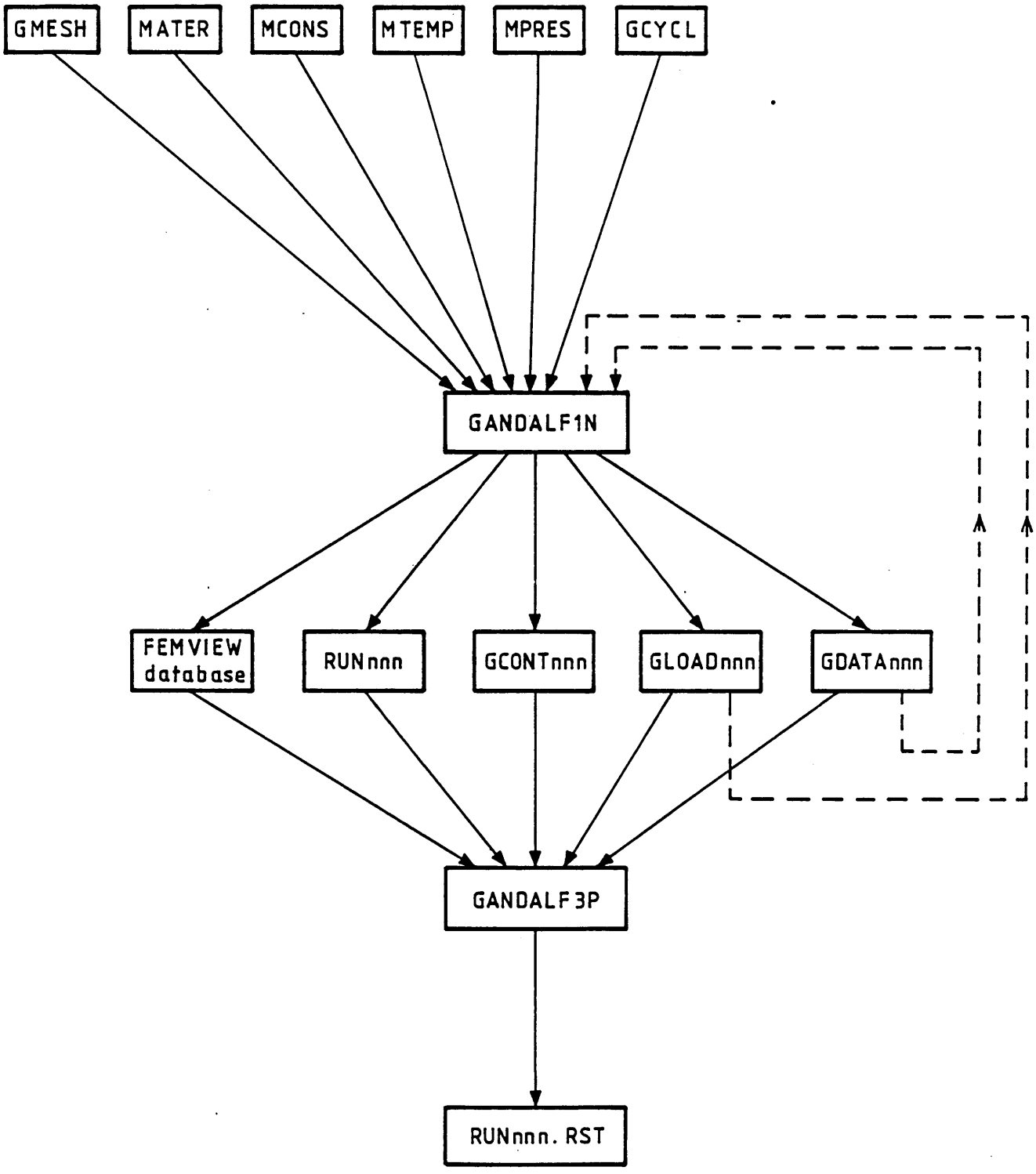


Fig. 5 The GANDALF file system

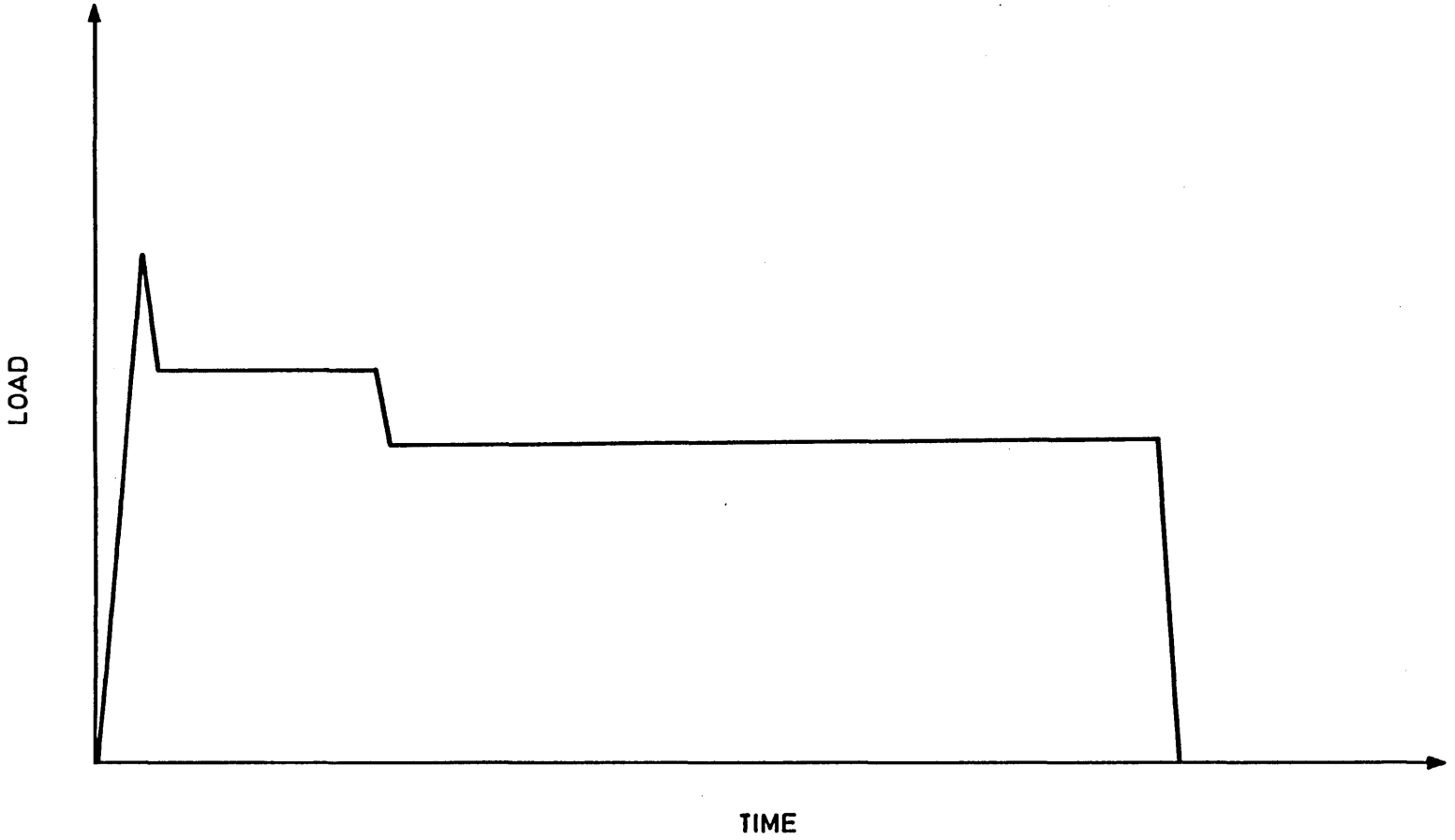


Fig.6 Flight cycle made up of transient and steady state loadcases

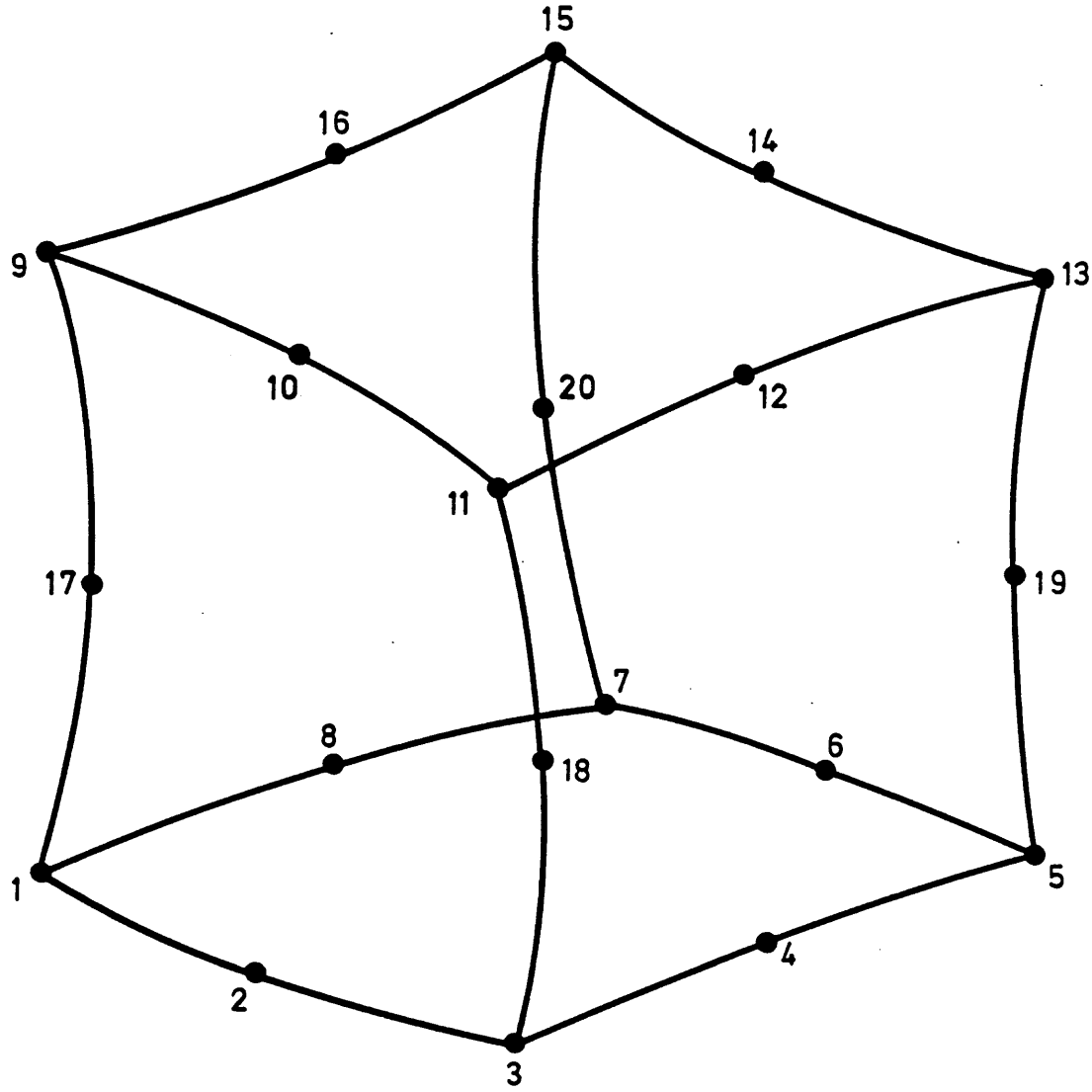


Fig. 7 The GANDALF element

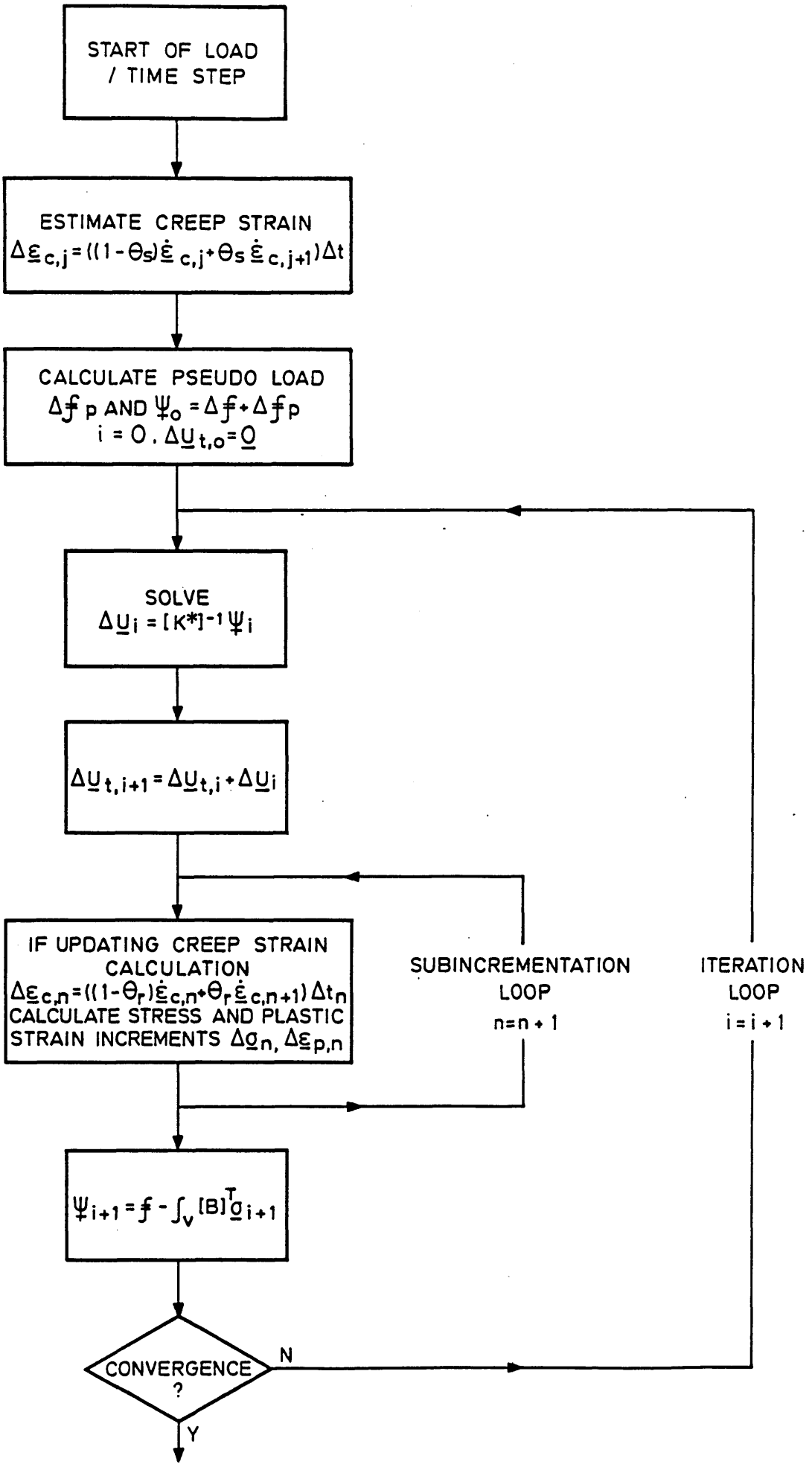


Fig. 8 The nonlinear solution procedure

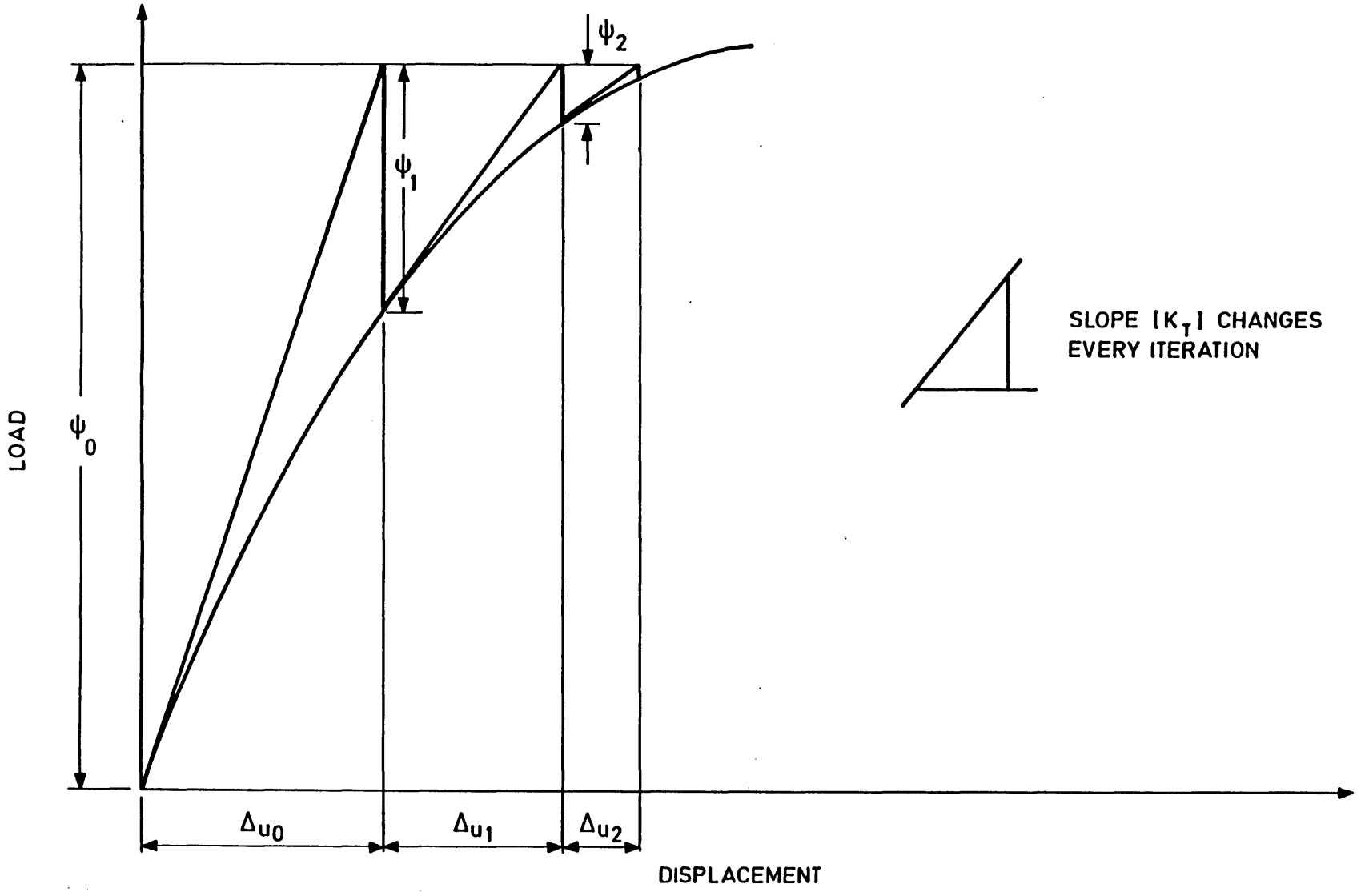


Fig. 9 The tangent stiffness method

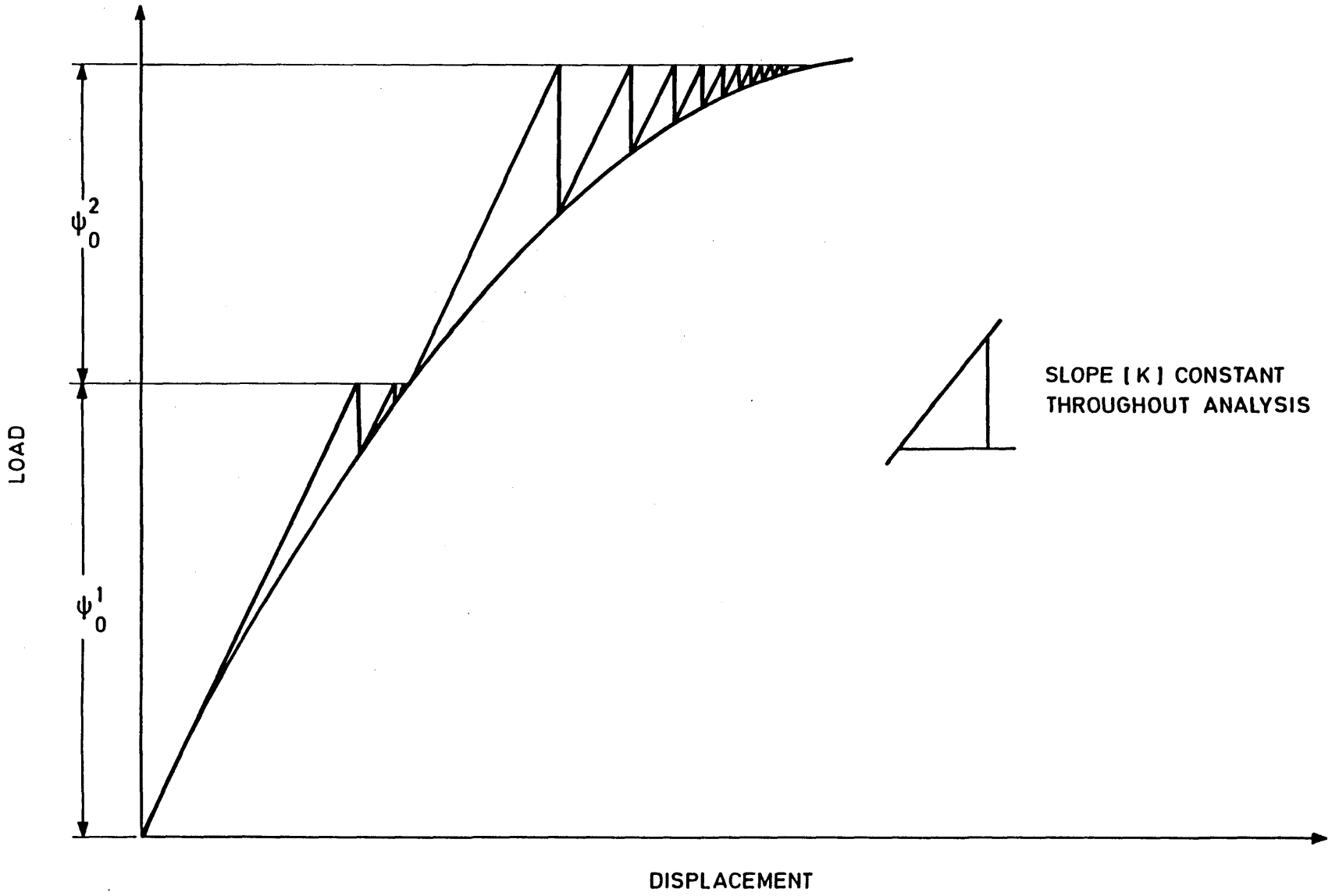


Fig. 10 The initial stiffness method

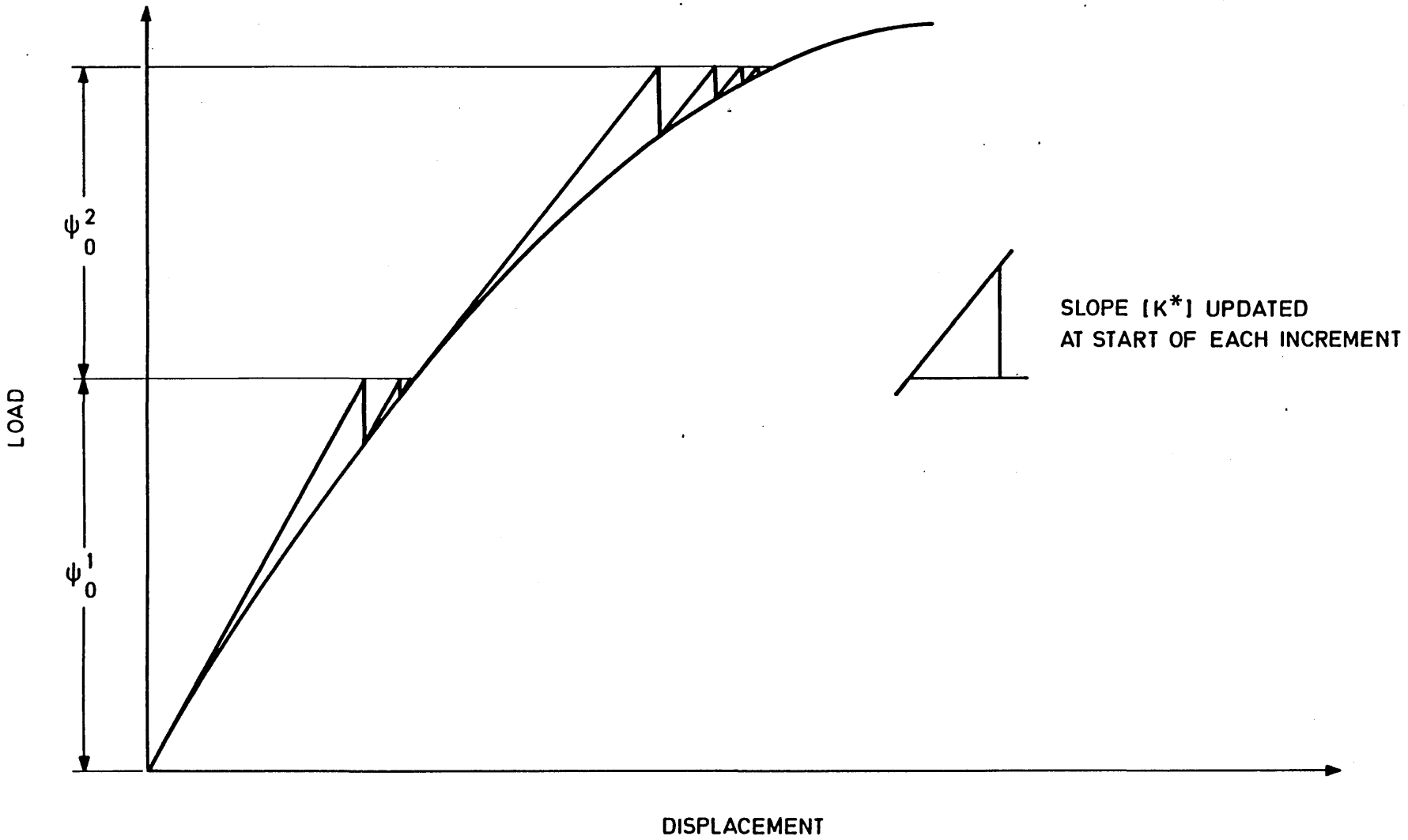


Fig. 11 The Hybrid Mark 1 method

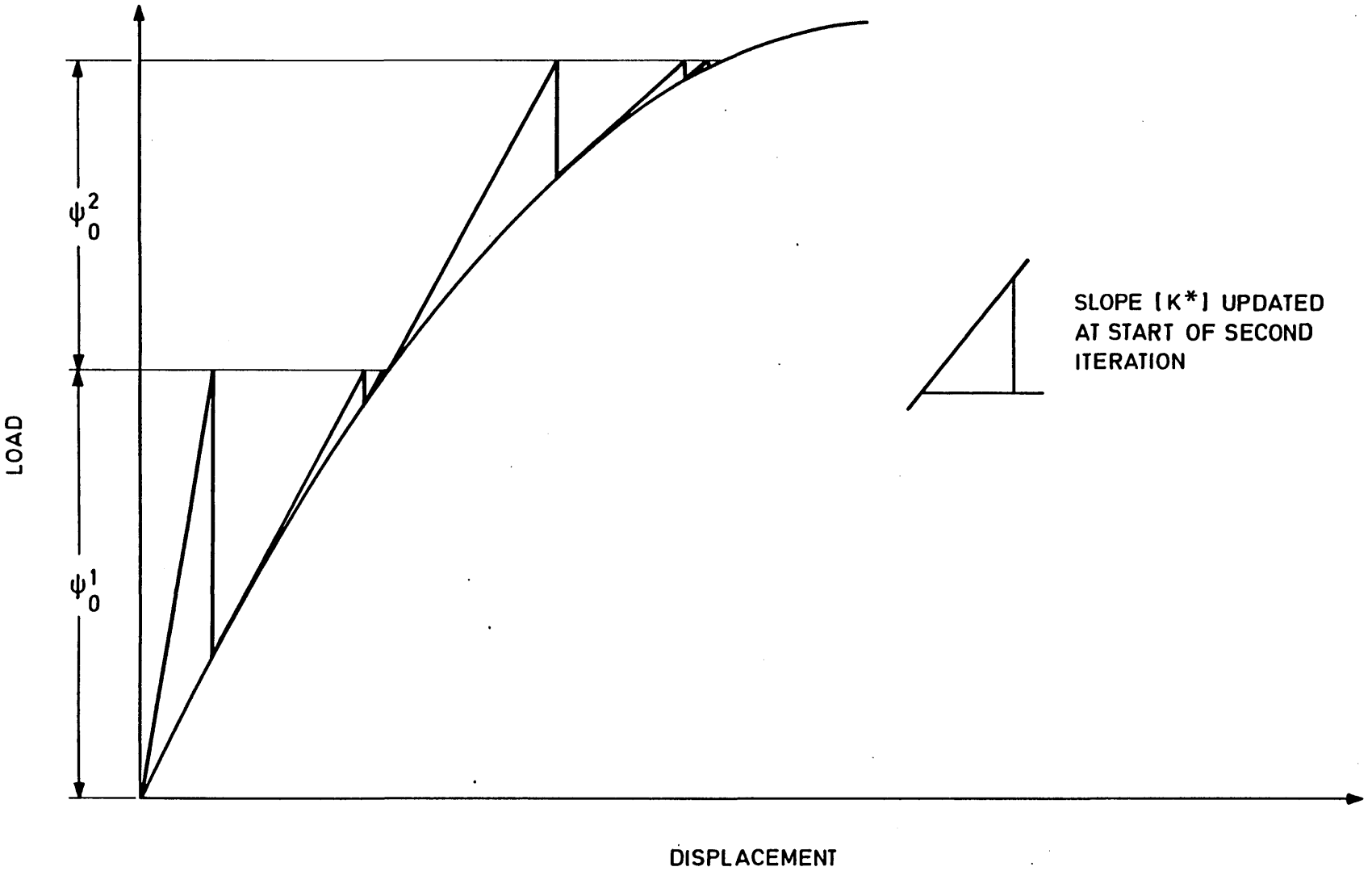


Fig.12 The Hybrid Mark 2 method

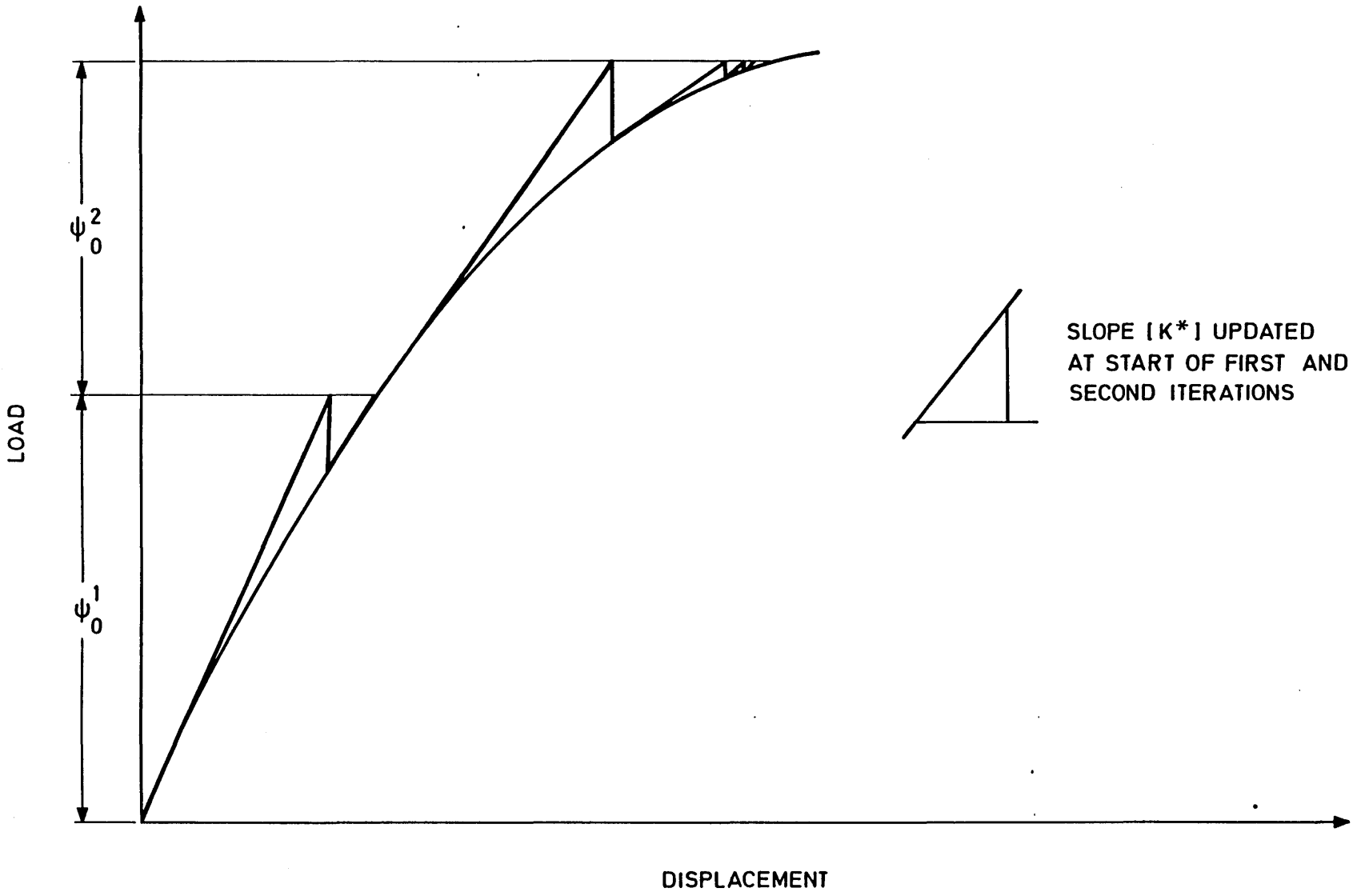


Fig.13 The Hybrid Mark 3 method

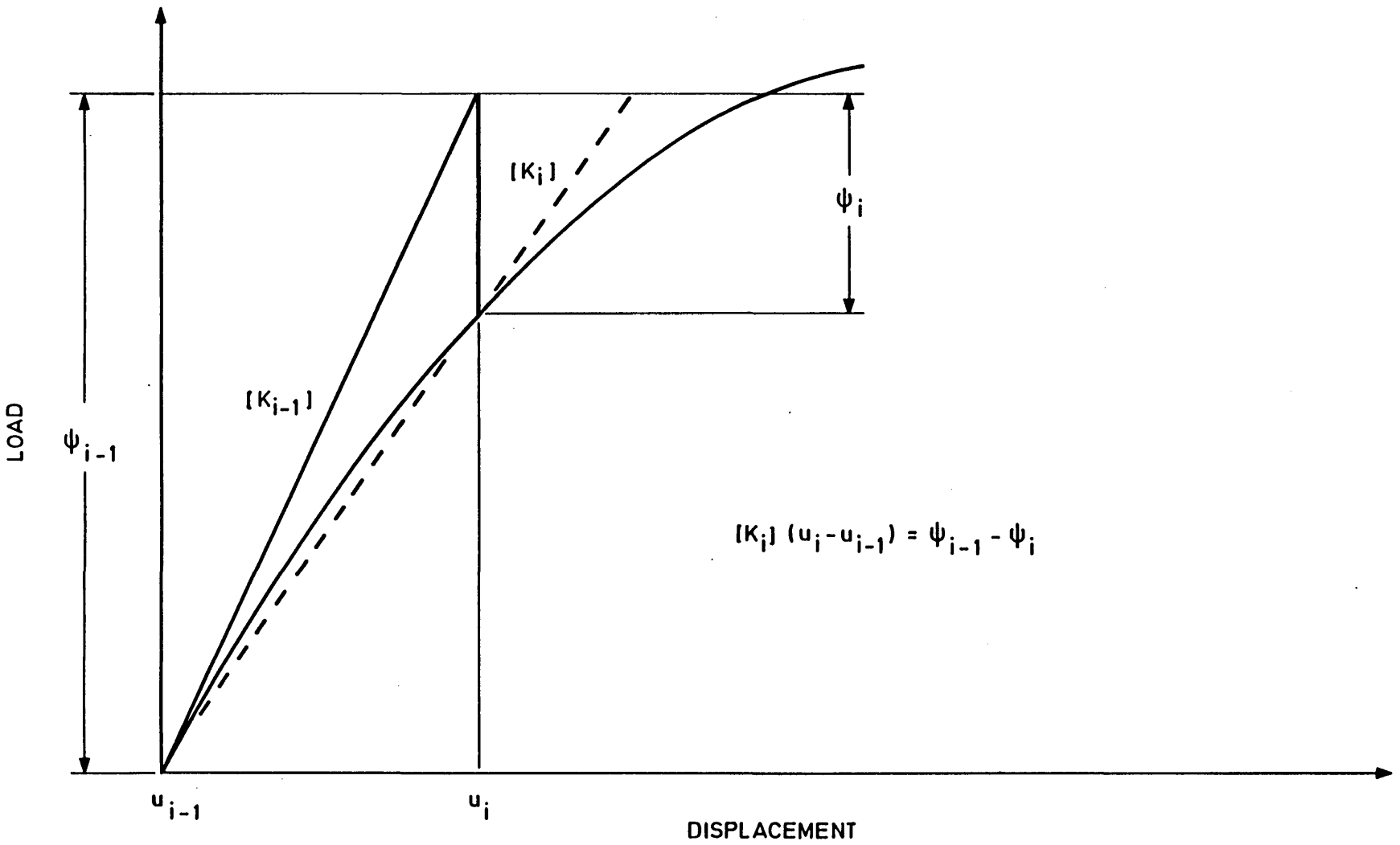


Fig.14 The secant condition

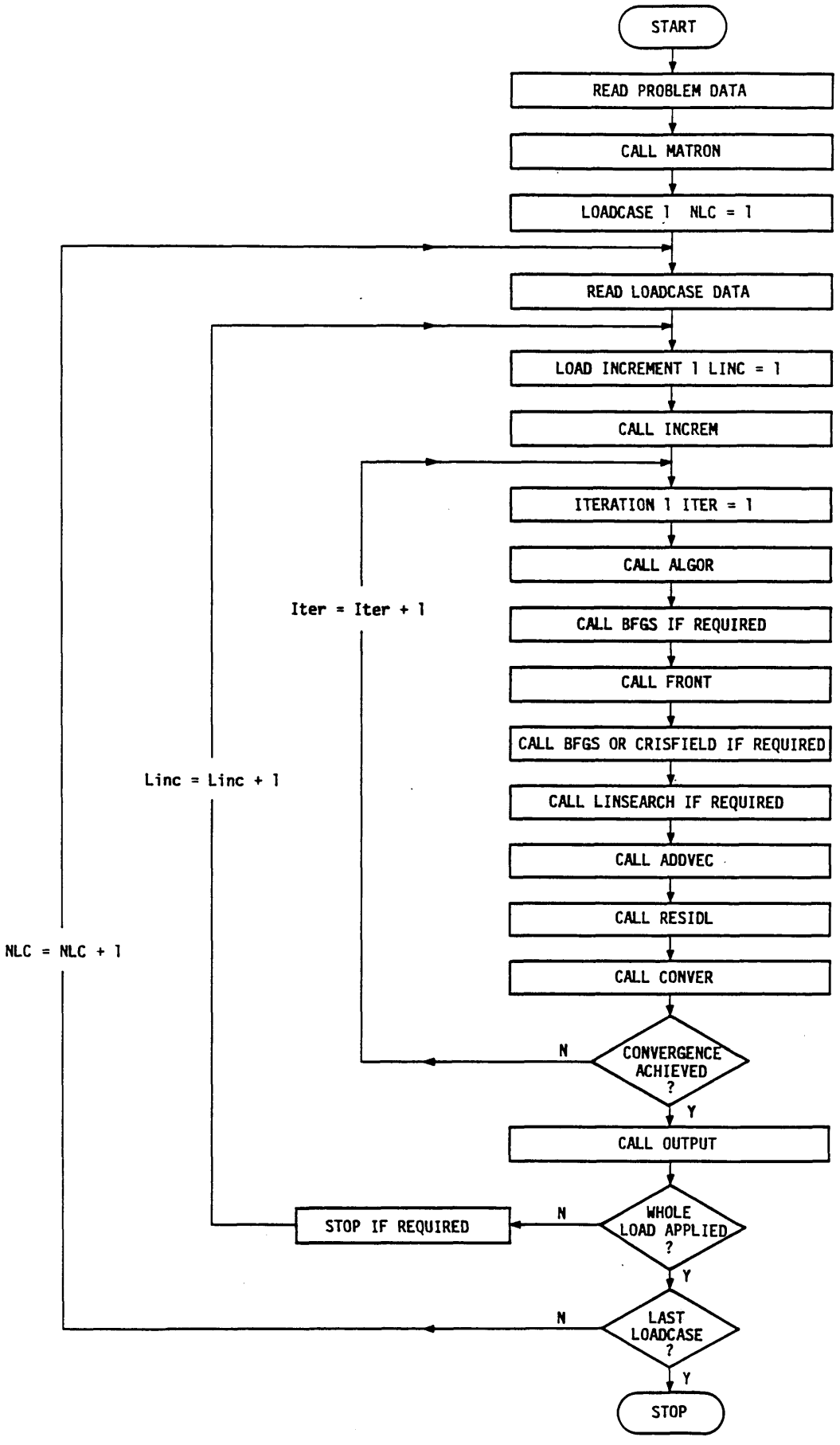


Fig. 15 Flow diagram for GANDALF 3P

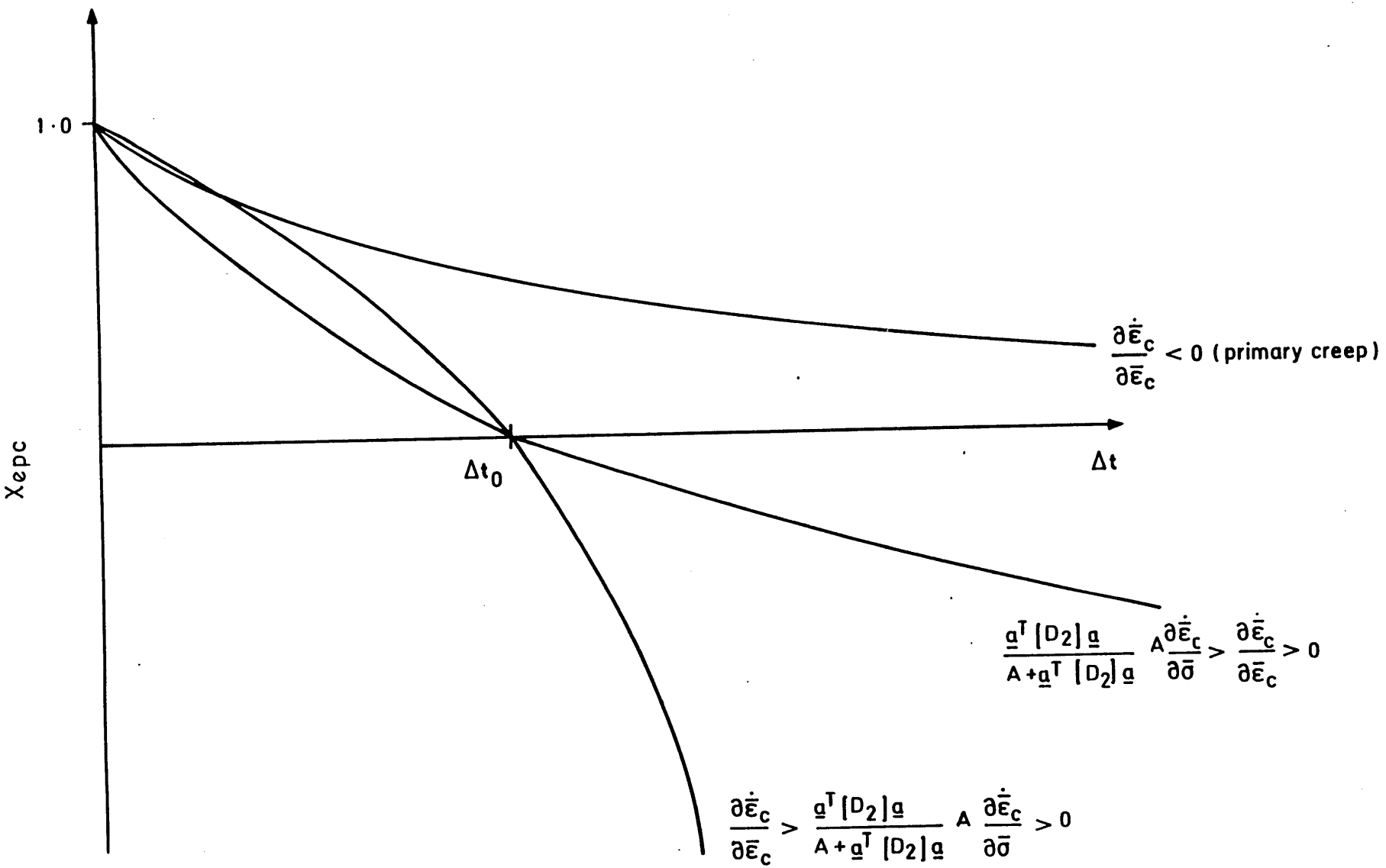


Fig.16 Variation of χ_{epc} with Δt

162 Elements
922 Nodes
537 Maximum frontwidth

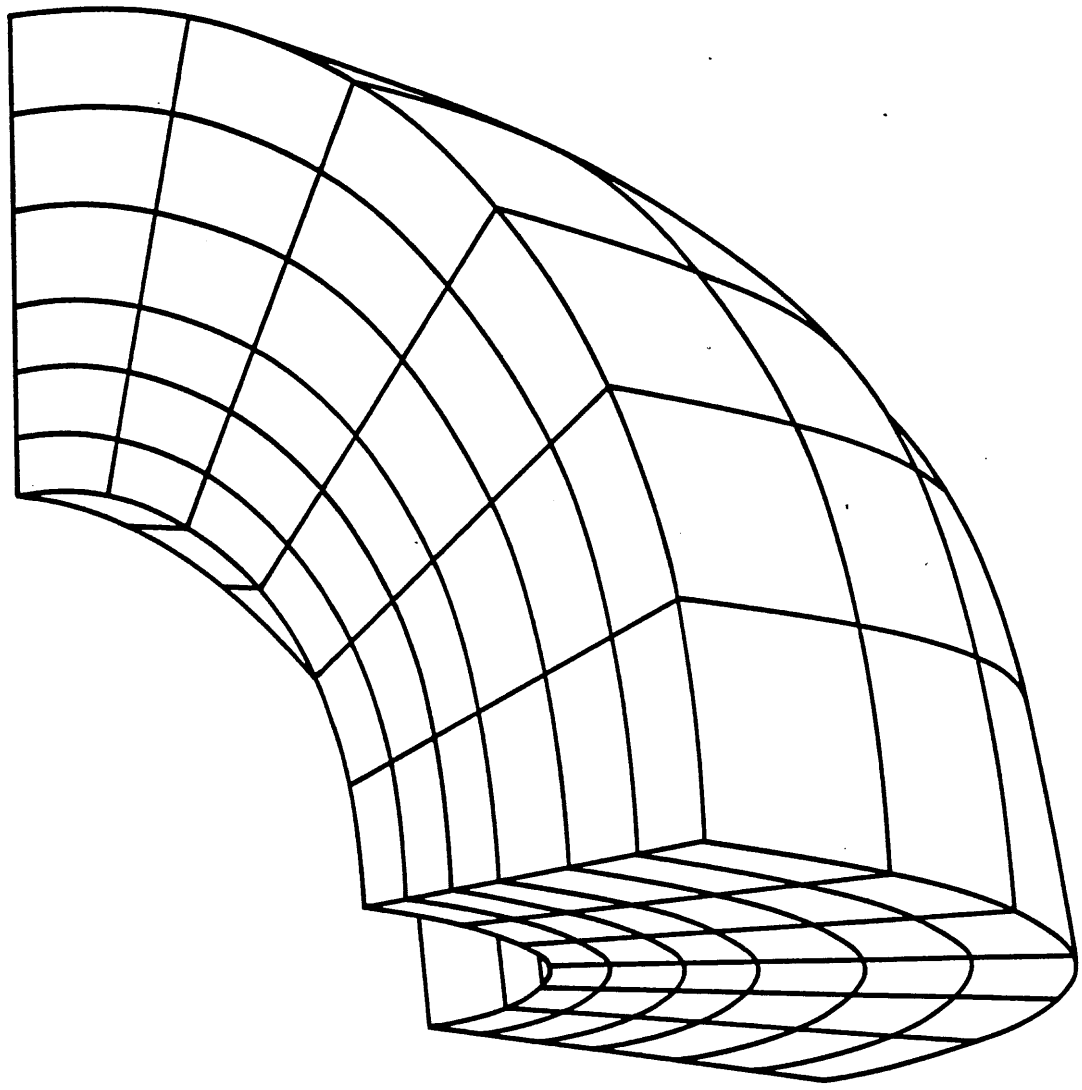


Fig. 17 Finite element mesh of $\frac{1}{8}$ sphere

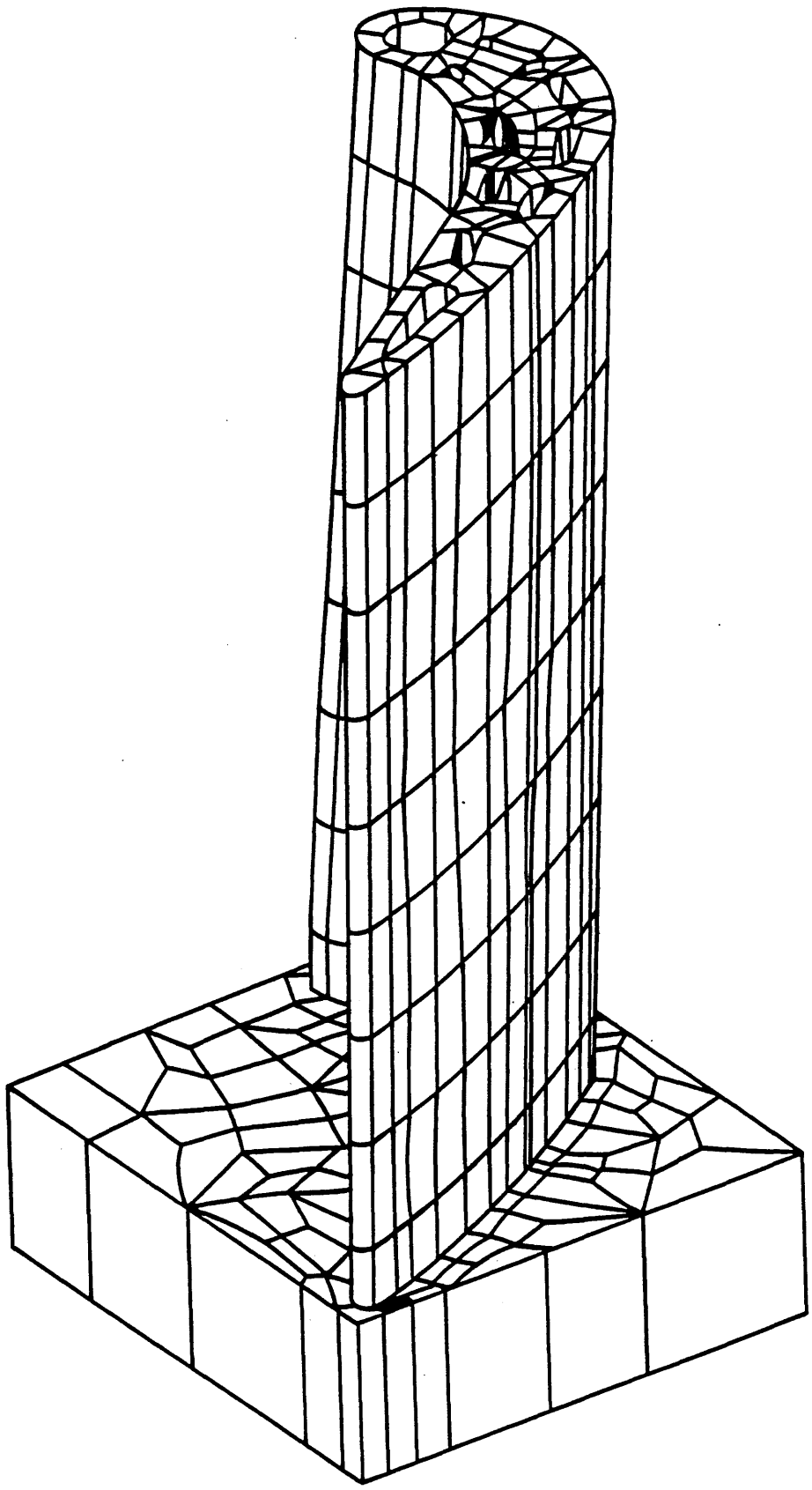


Fig. 18 Finite element mesh of RB211 "Racetrack" blade

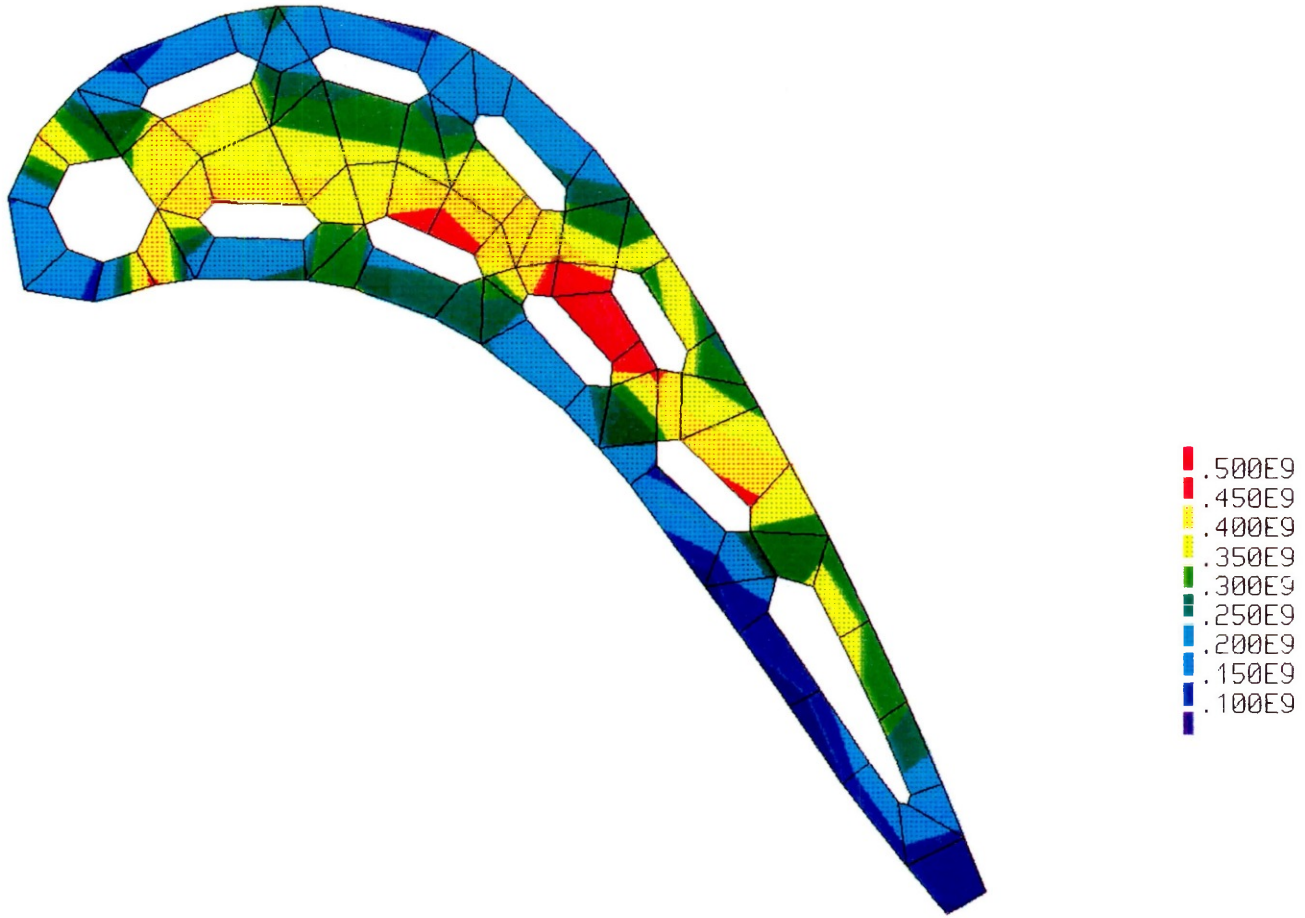


FIG. 19 VON MISES STRESS AFTER LOAD UP (N/m**2)

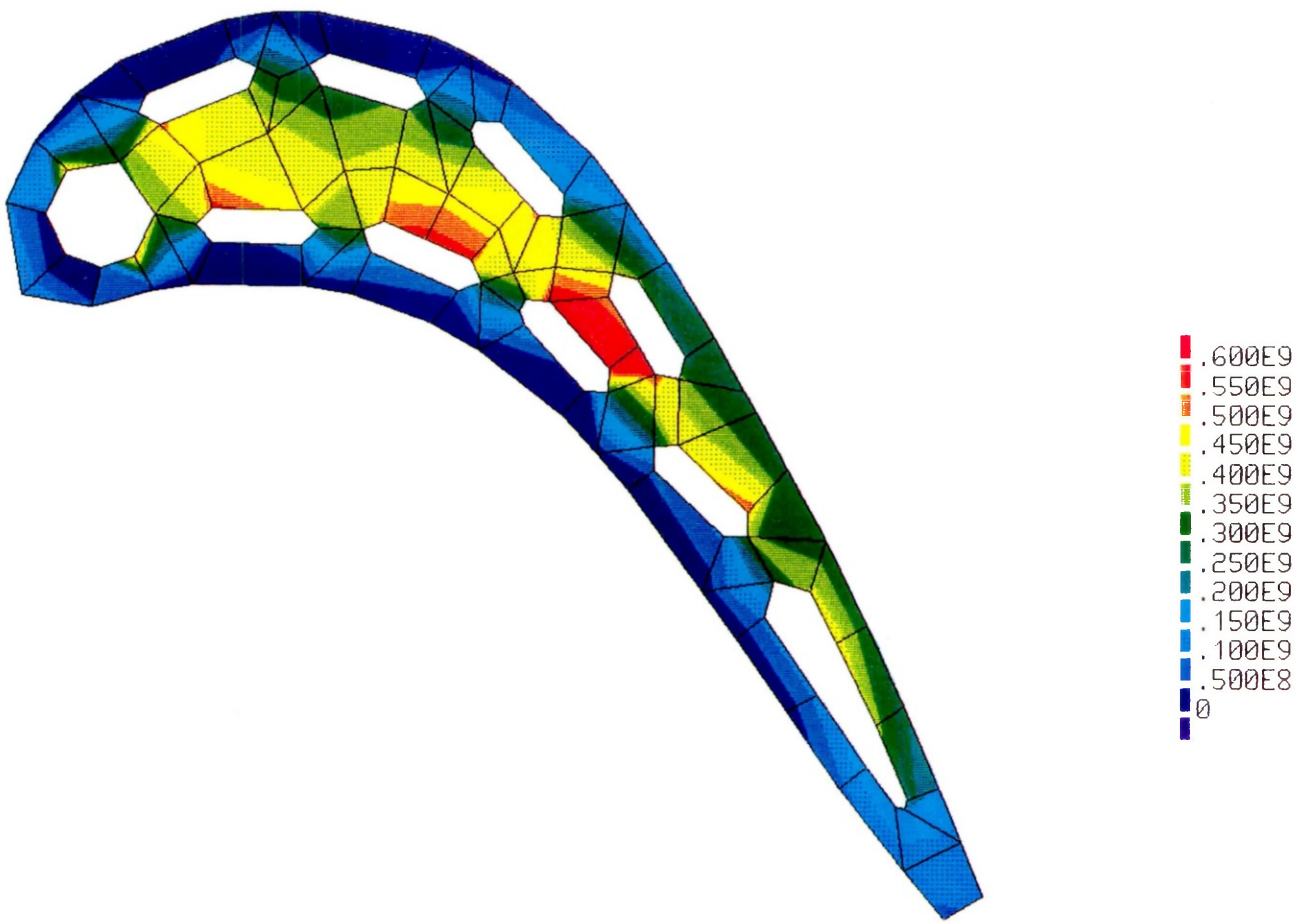


FIG. 20 MAXIMUM PRINCIPAL STRESS AFTER LOAD UP (N/m**2)

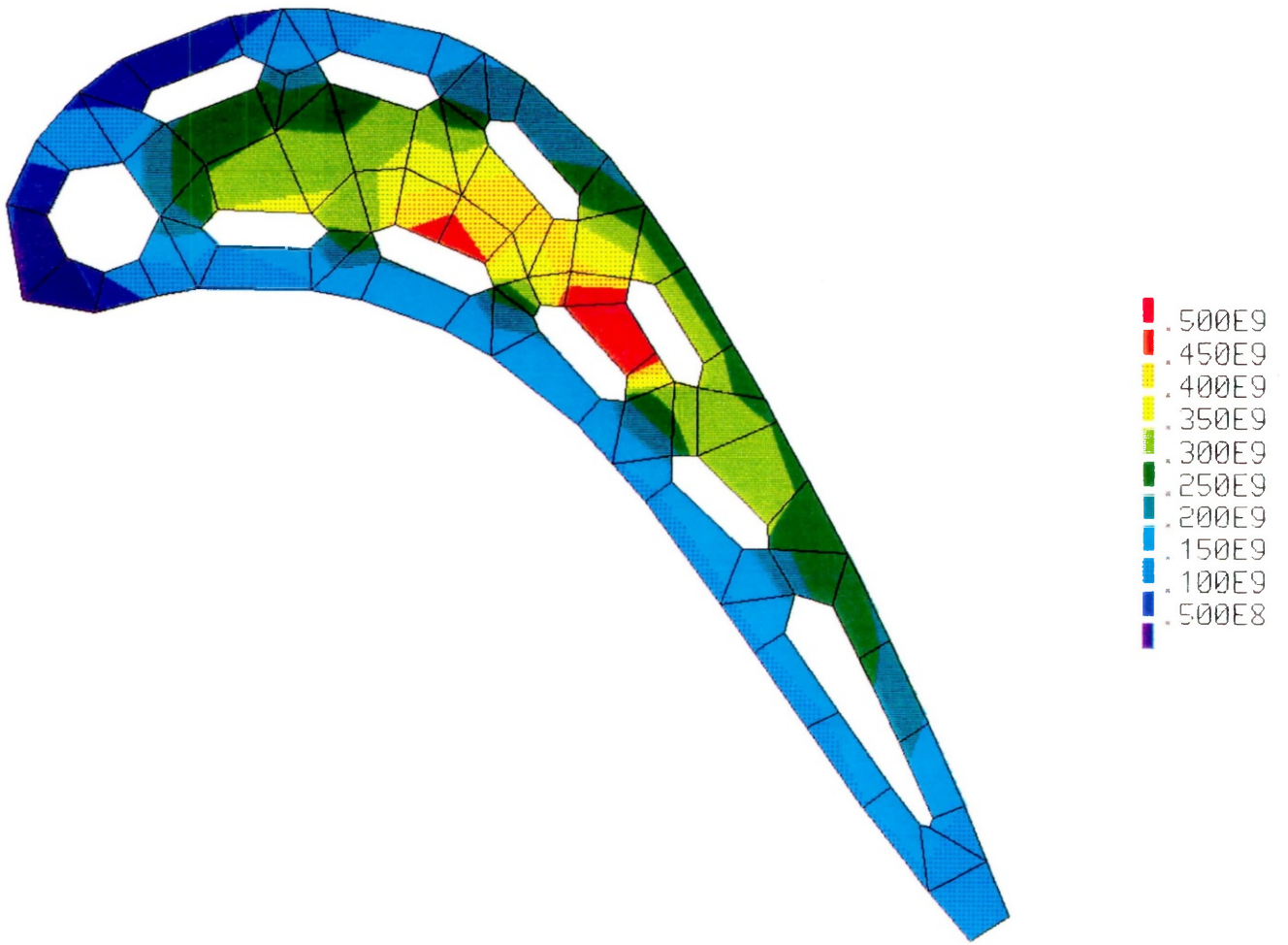


FIG. 21 VON MISES STRESS AFTER CREEP (N/m**2)

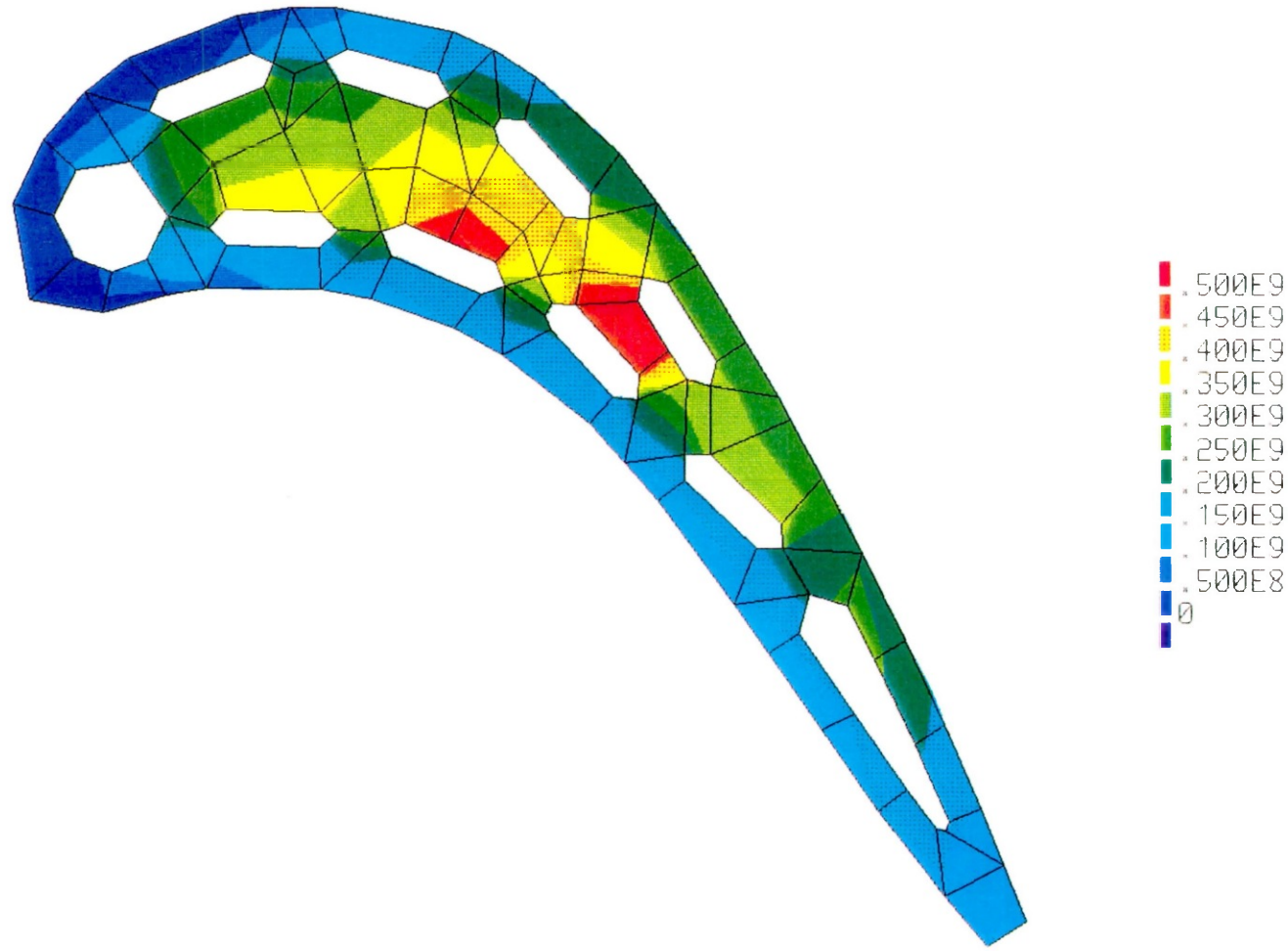


FIG. 22 MAXIMUM PRINCIPAL STRESS AFTER CREEP (N/m**2)

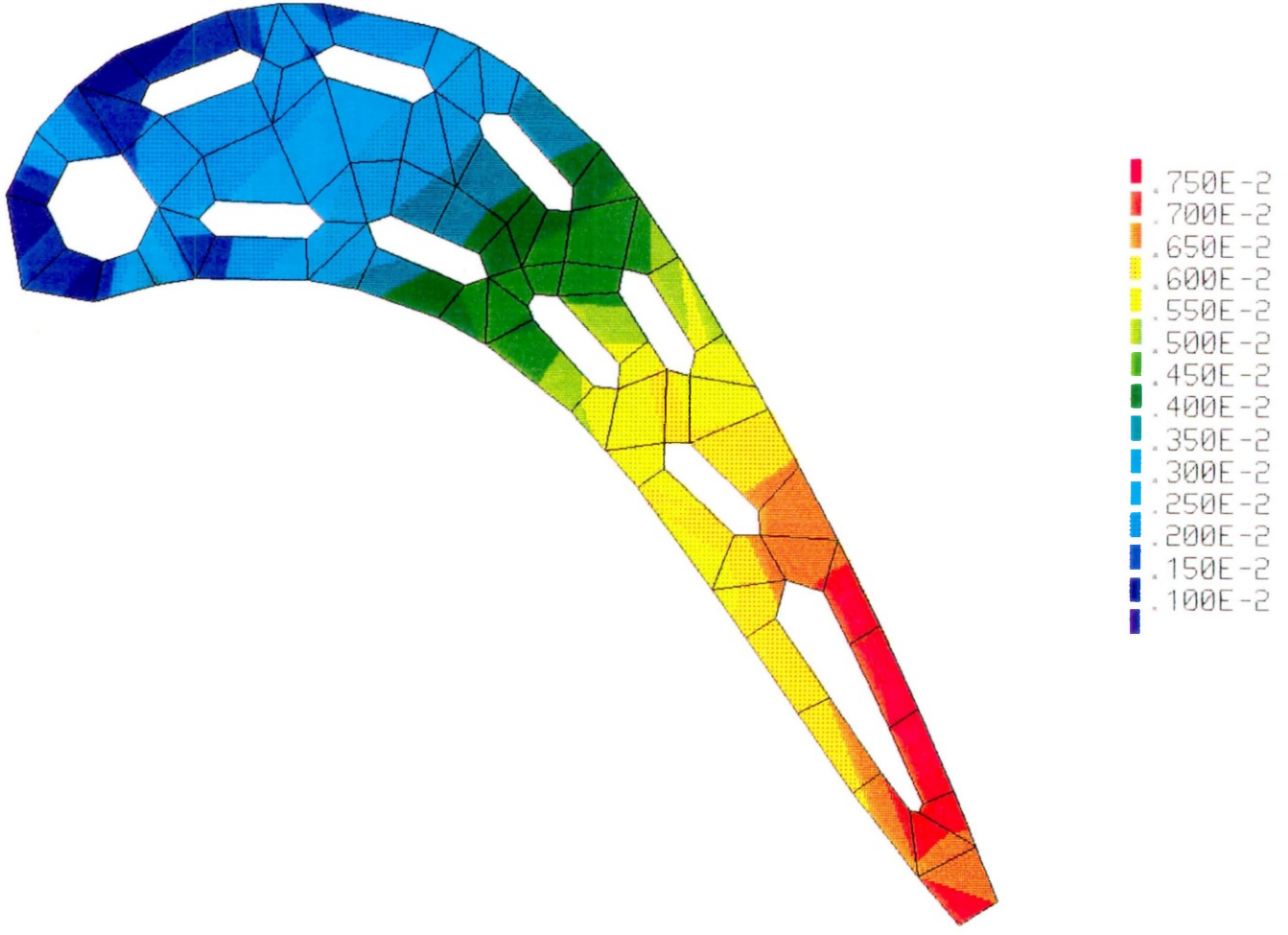


FIG. 23 CREEP HARDENING PARAMETER

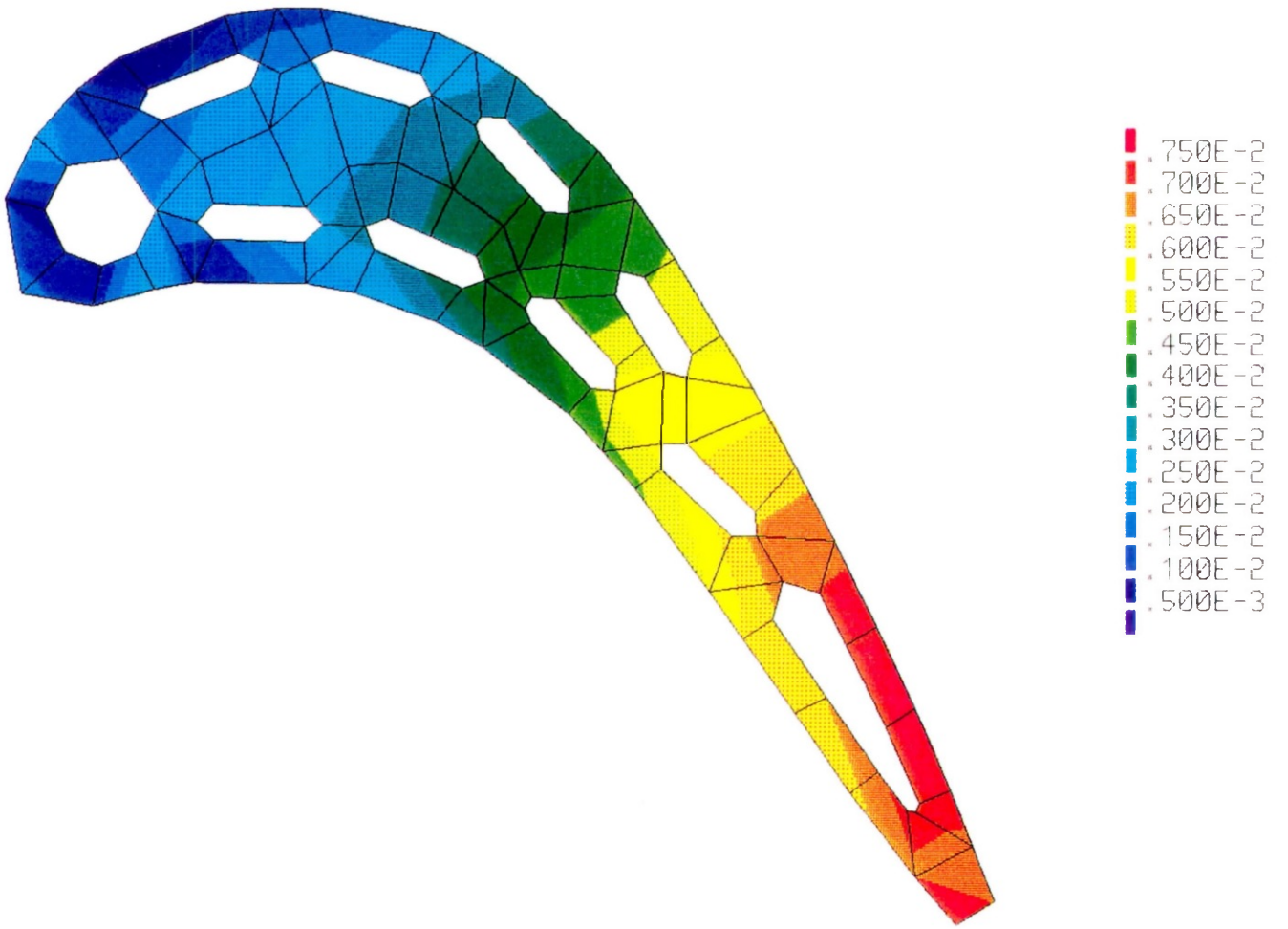


FIG. 24 MAXIMUM PRINCIPAL CREEP STRAIN

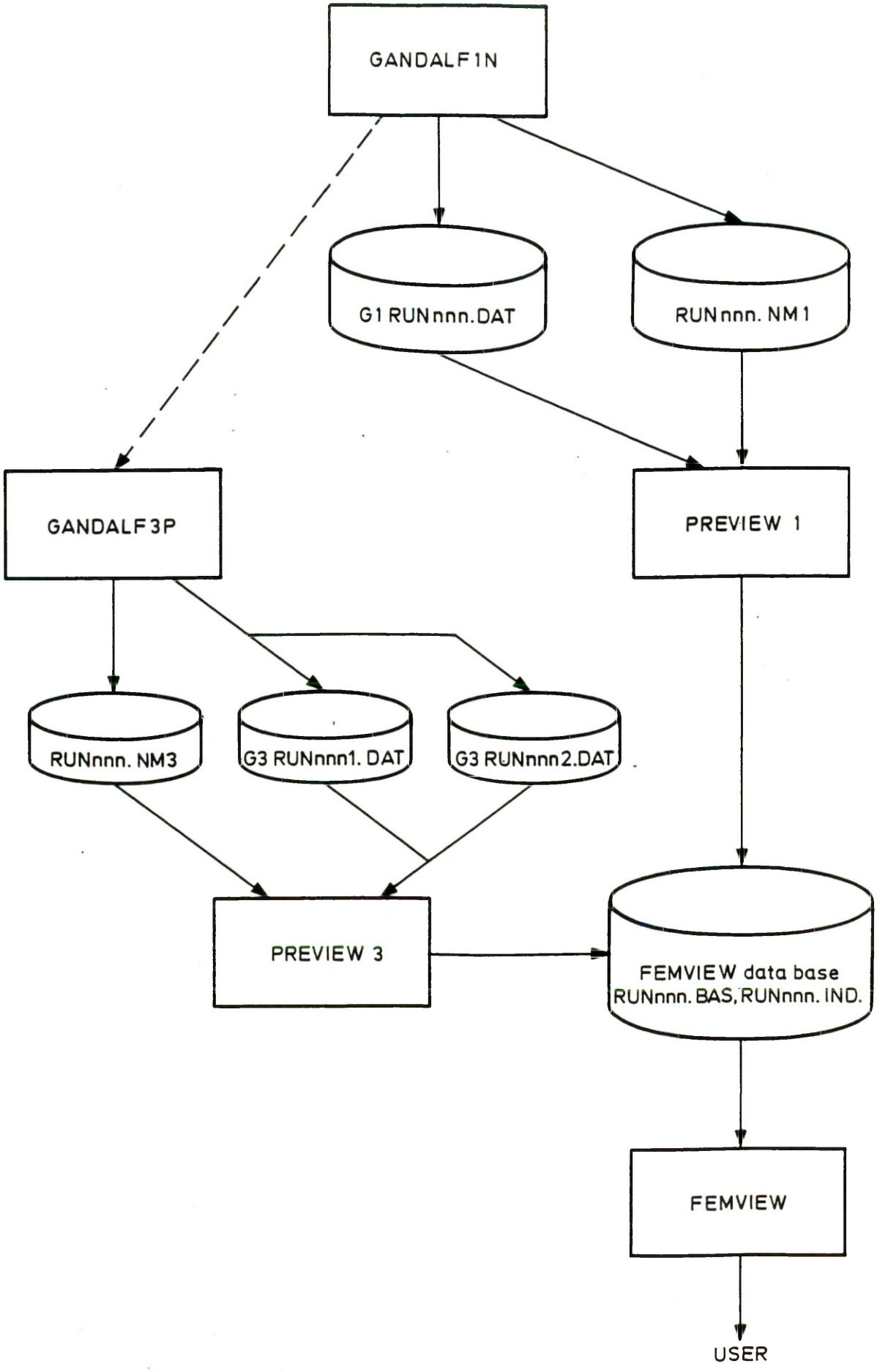


Fig. 25 The GANDALF graphics system

Event flag Number	Set By	Cleared By	Function
64	PREVIEW1 PREVIEW3 FEMVIEW	-	Determines if process is the first to associate with event flag cluster.
65	PREVIEW1 PREVIEW3 FEMVIEW	-	Signal that lock flag has been set.
66	PREVIEW1 PREVIEW3 FEMVIEW	PREVIEW1 PREVIEW3 FEMVIEW	Lock on graphics database.
67	PREVIEW3	GANDALF3P	When set allows GANDALF3P to write to file 1.
68	GANDALF3P	PREVIEW3	When set allows PREVIEW3 to read from file 1.
69	PREVIEW3	GANDALF3P	When set allows GANDALF3P to write to file 2.
70	GANDALF3P	PREVIEW3	When set allows PREVIEW3 to read from file 2.
71	PREVIEW1	-	Signal to GANDALFIN that PREVIEW1 has started.
72	GANDALF3P	-	Signal to PREVIEW3 that analysis is complete.
73	GANDALFIN	-	If set forces GANDALF3P to wait for flag 74 to be set.
74	PREVIEW1	-	Signal to GANDALF3P that database is complete and analysis can proceed.

Fig. 26 GANDALF event flags

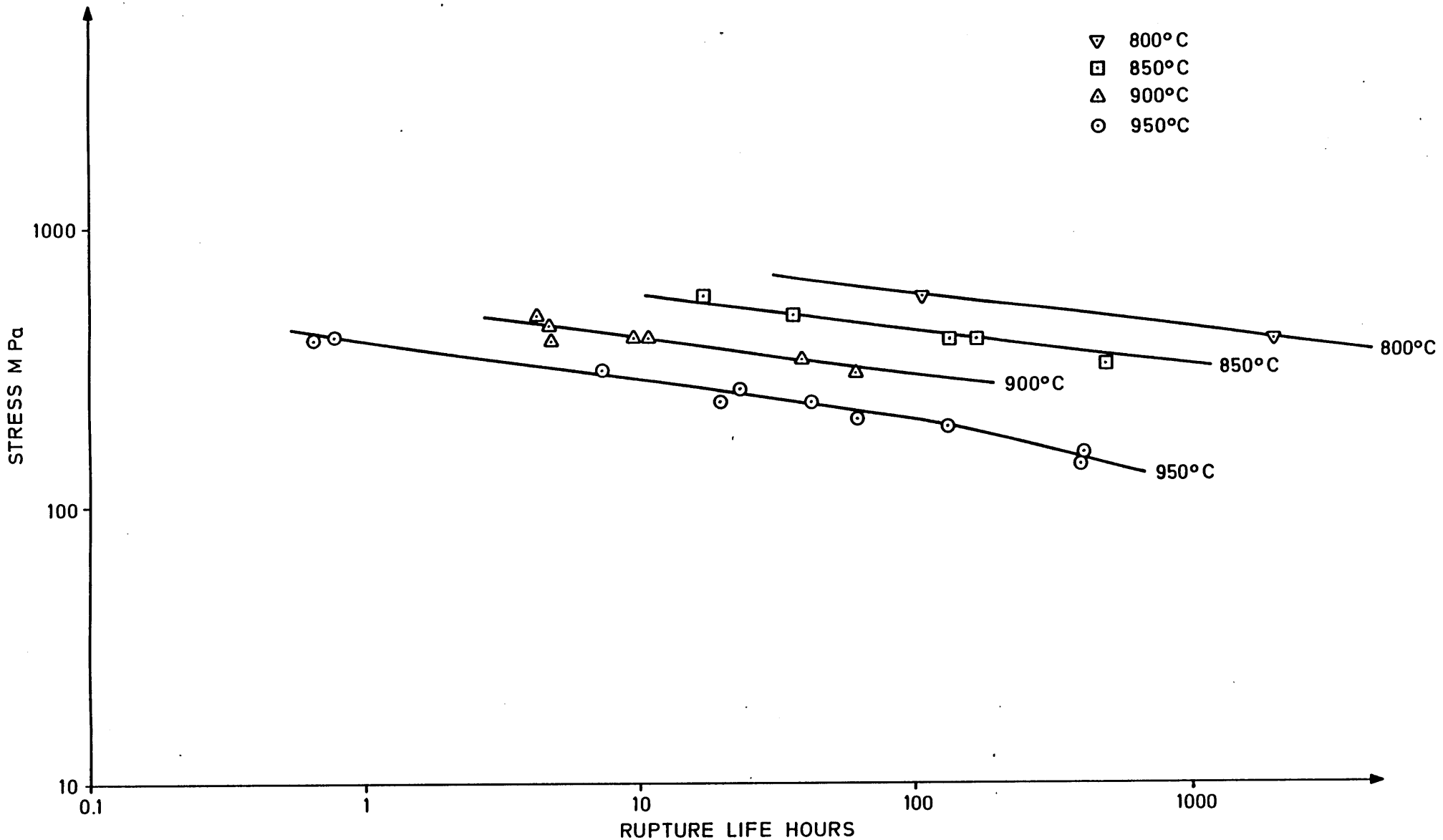


Fig. 27 Conventionally cast Mar-M 002 uniaxial creep rupture data

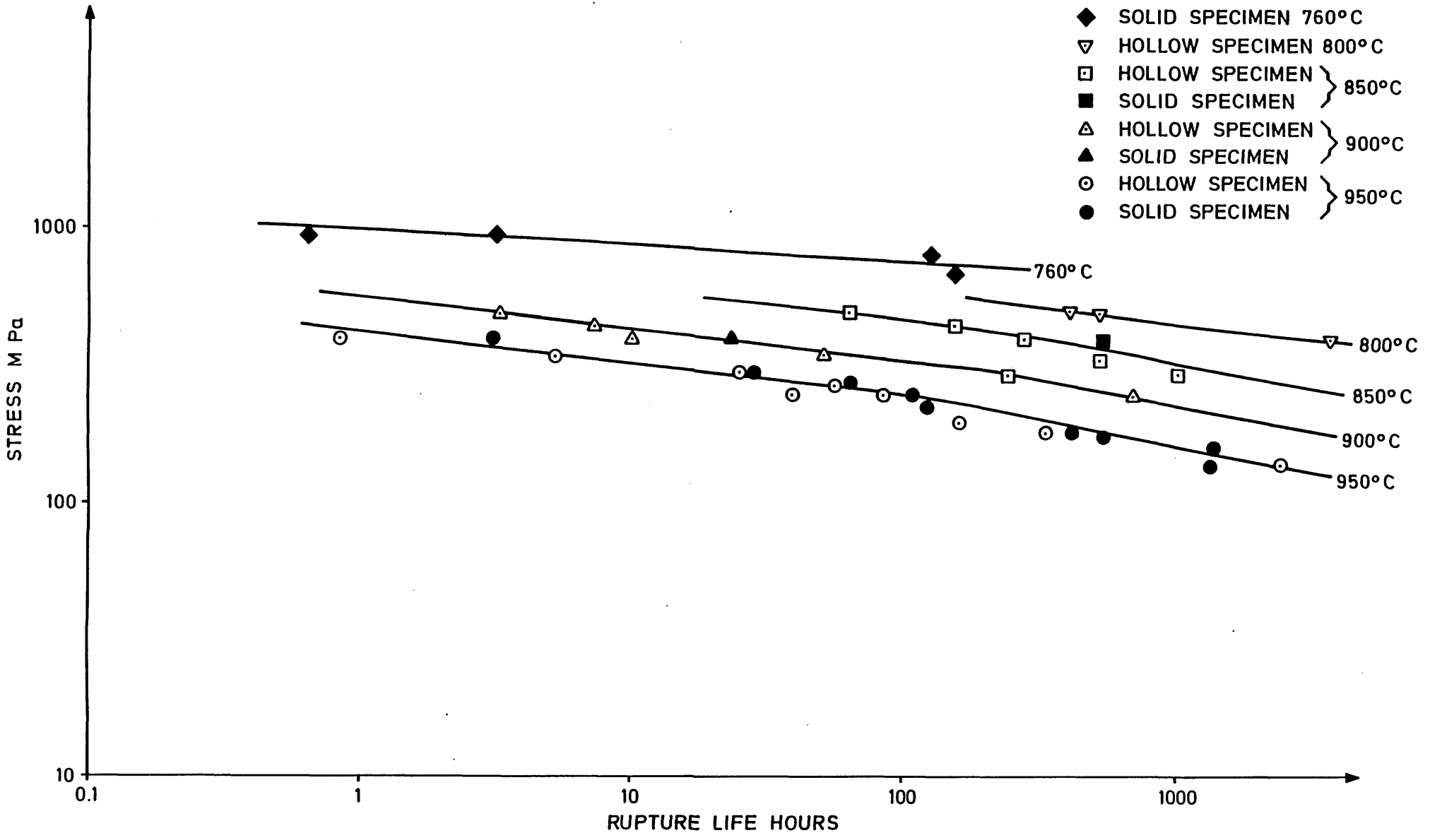


Fig. 28 Directionally solidified Mar-M 002 uniaxial creep rupture data

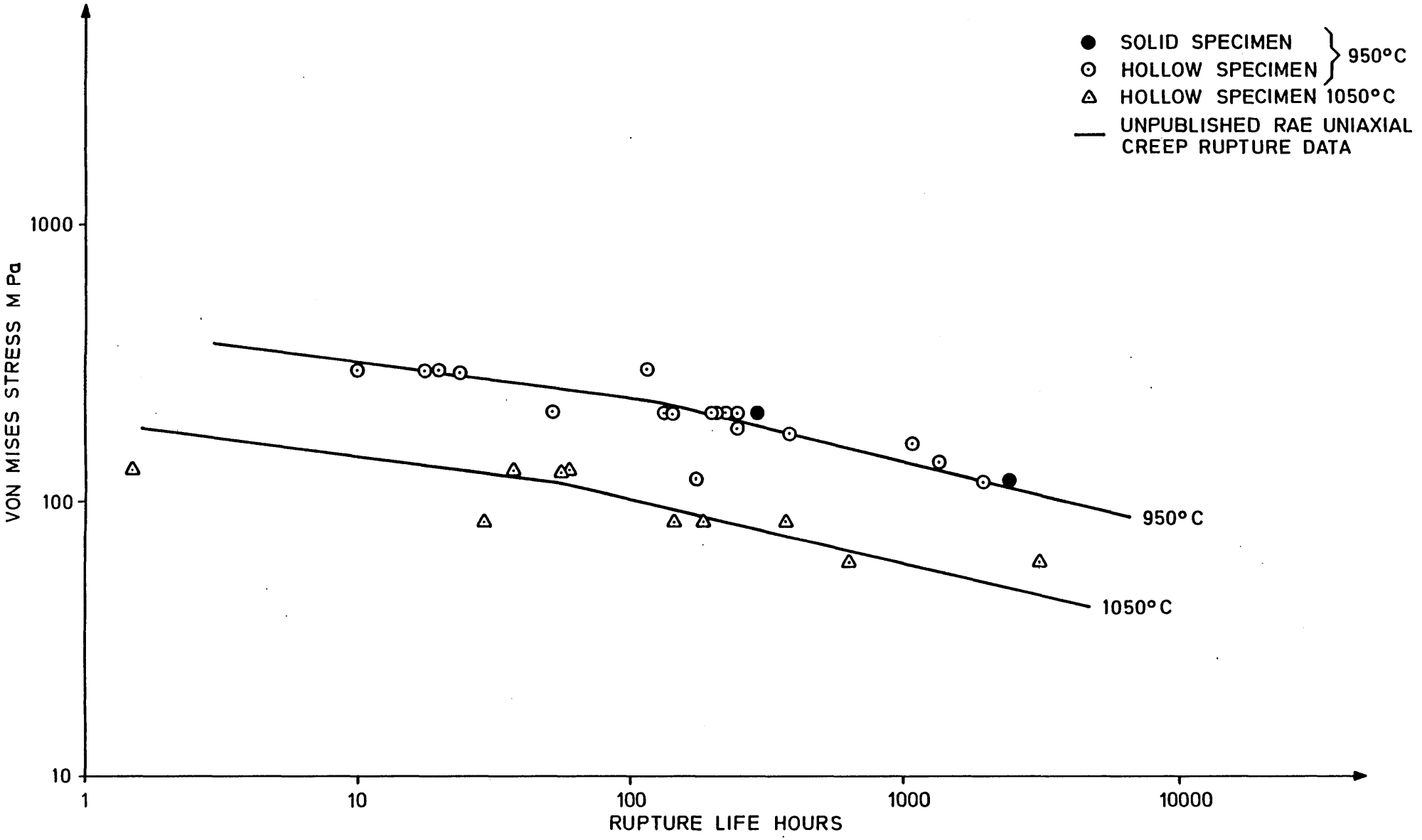


Fig. 29 Mar-M 002 tension-torsion test series

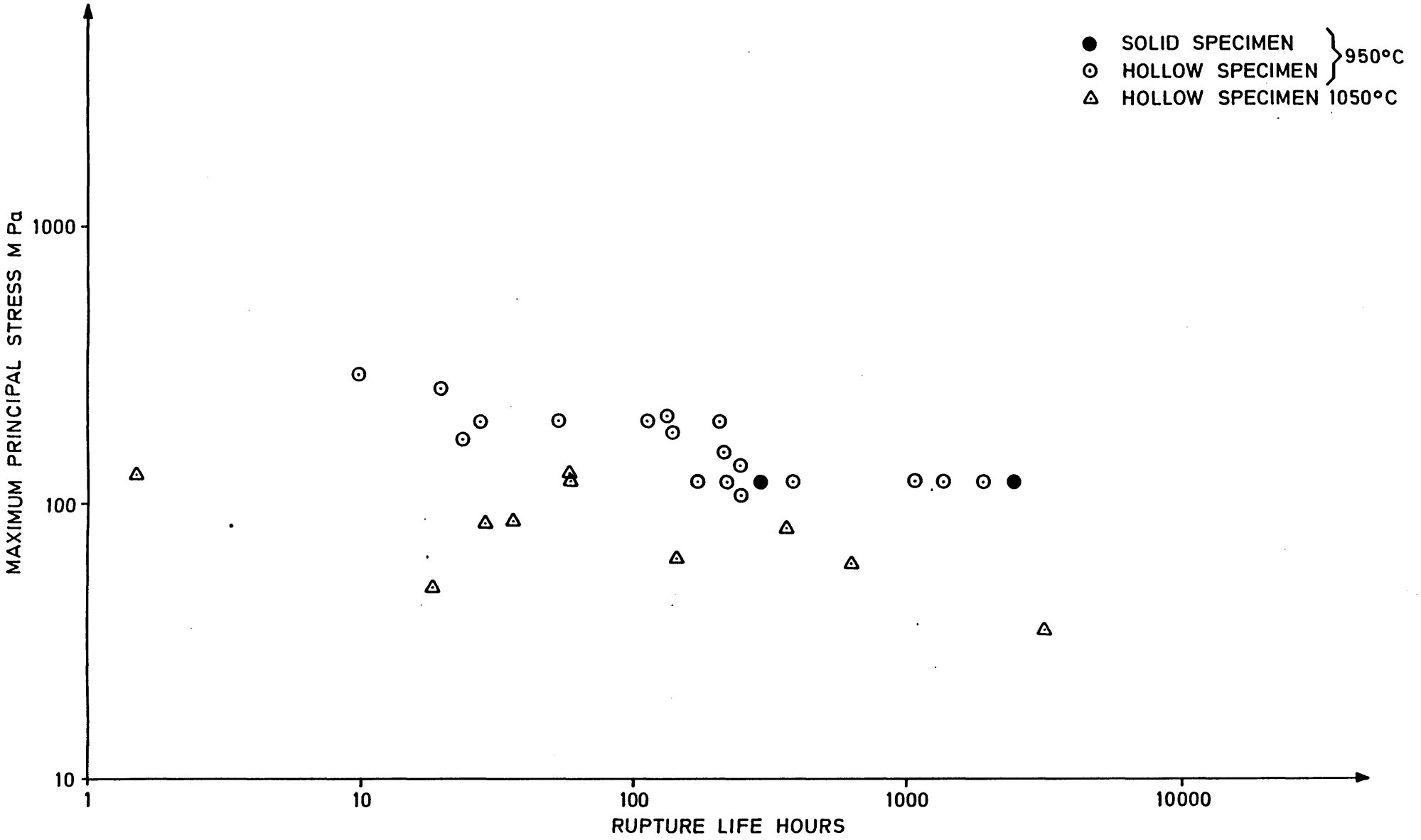


Fig. 30 Mar-M 002 tension-torsion test series

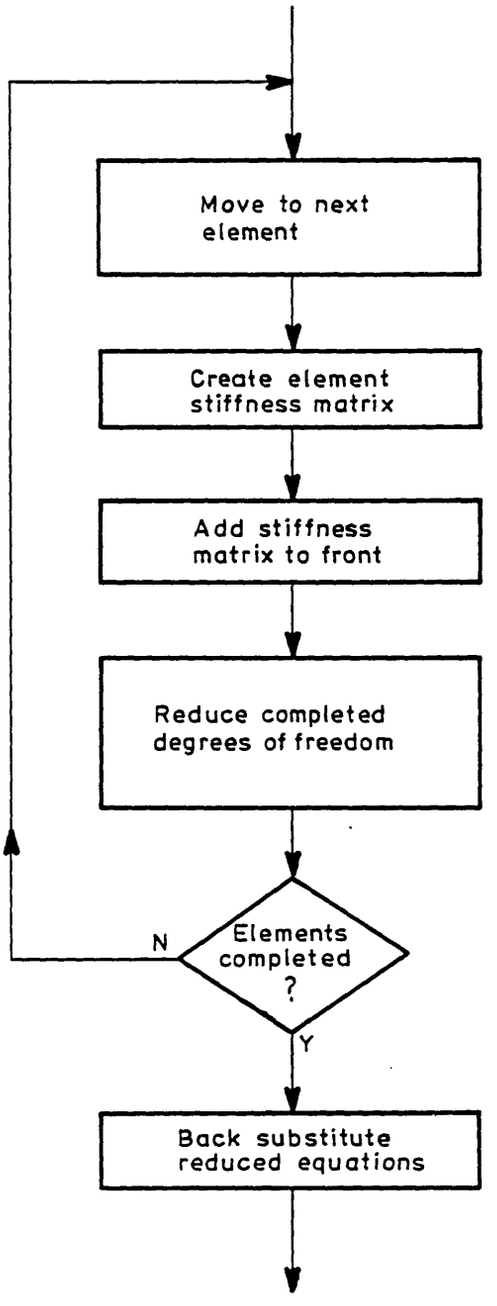


Fig. 31 The frontal solution on a serial machine

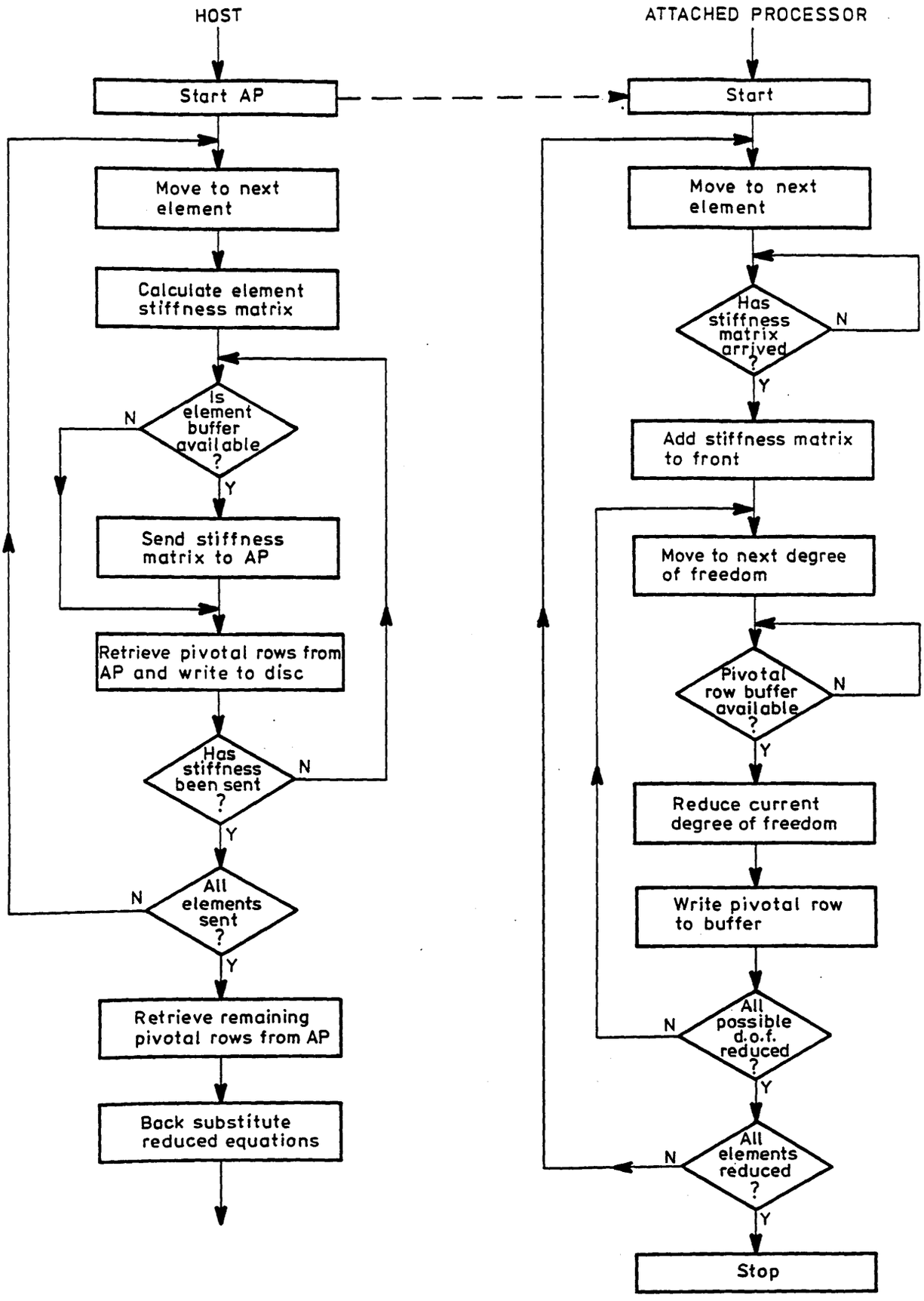


Fig. 32 Implementation of the frontal solution on host / attached processor combination

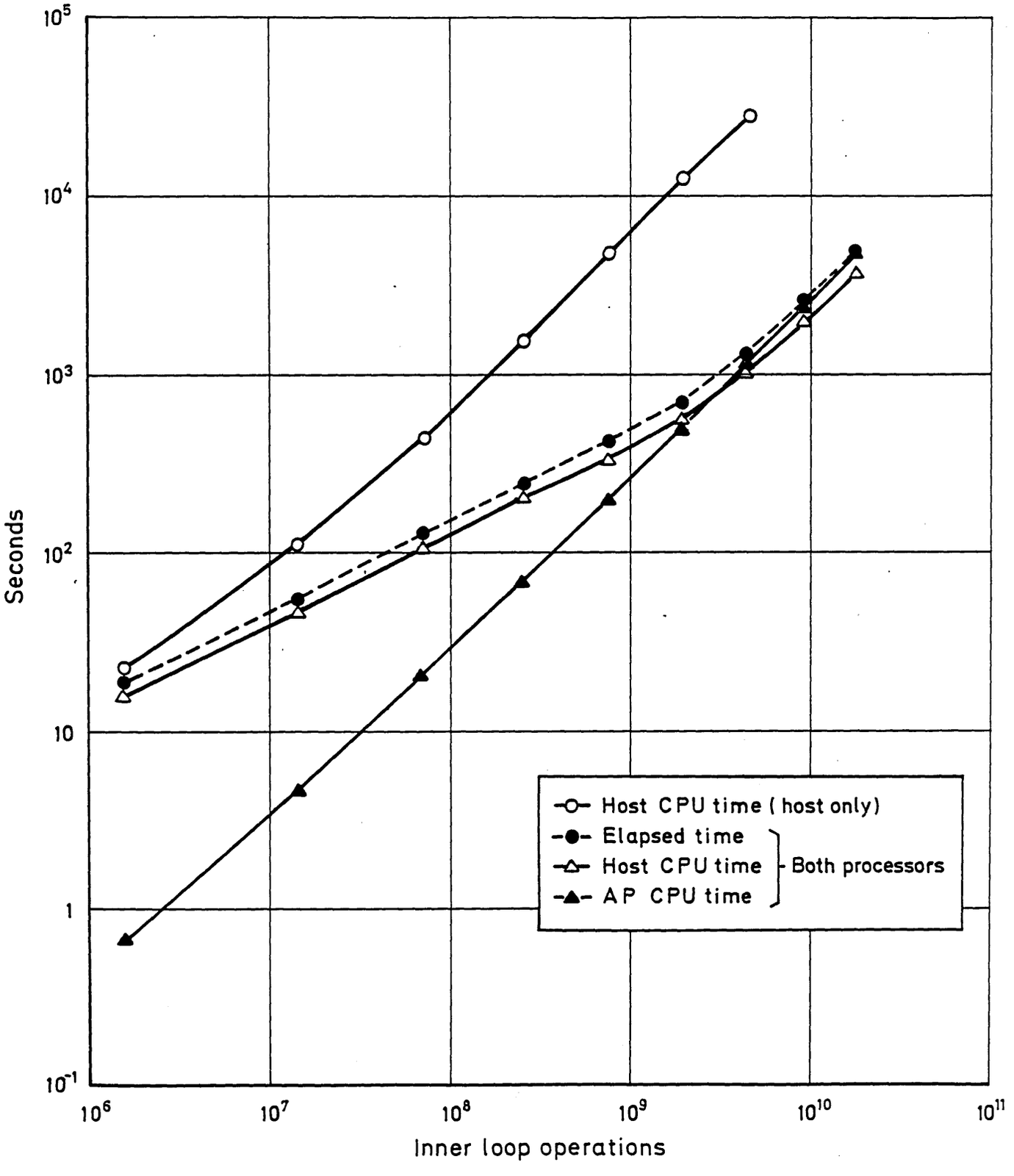


Fig. 33 Results for 'cube of cubes' problem

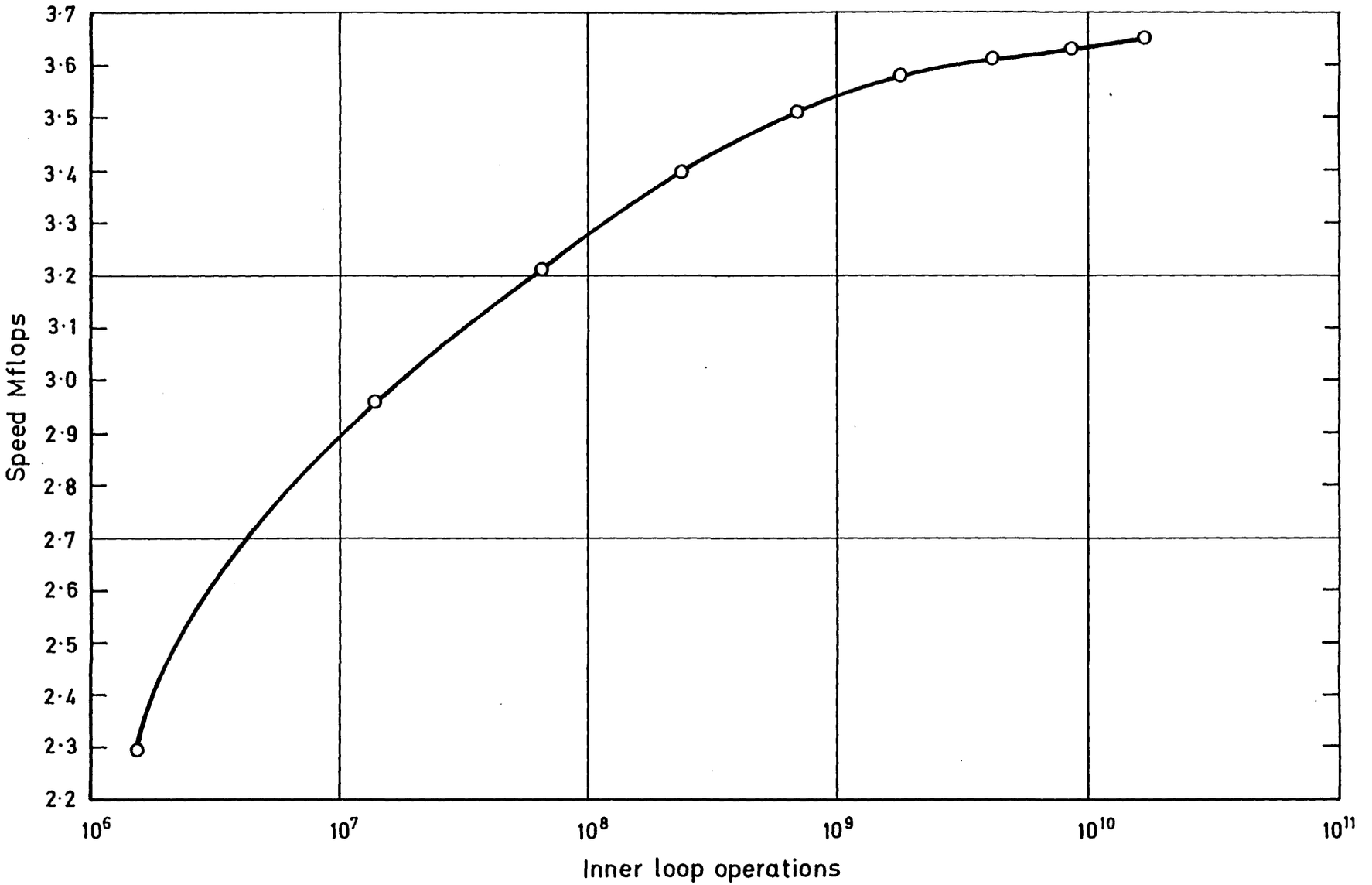


Fig. 34 Variation in attached processor speed with problem size

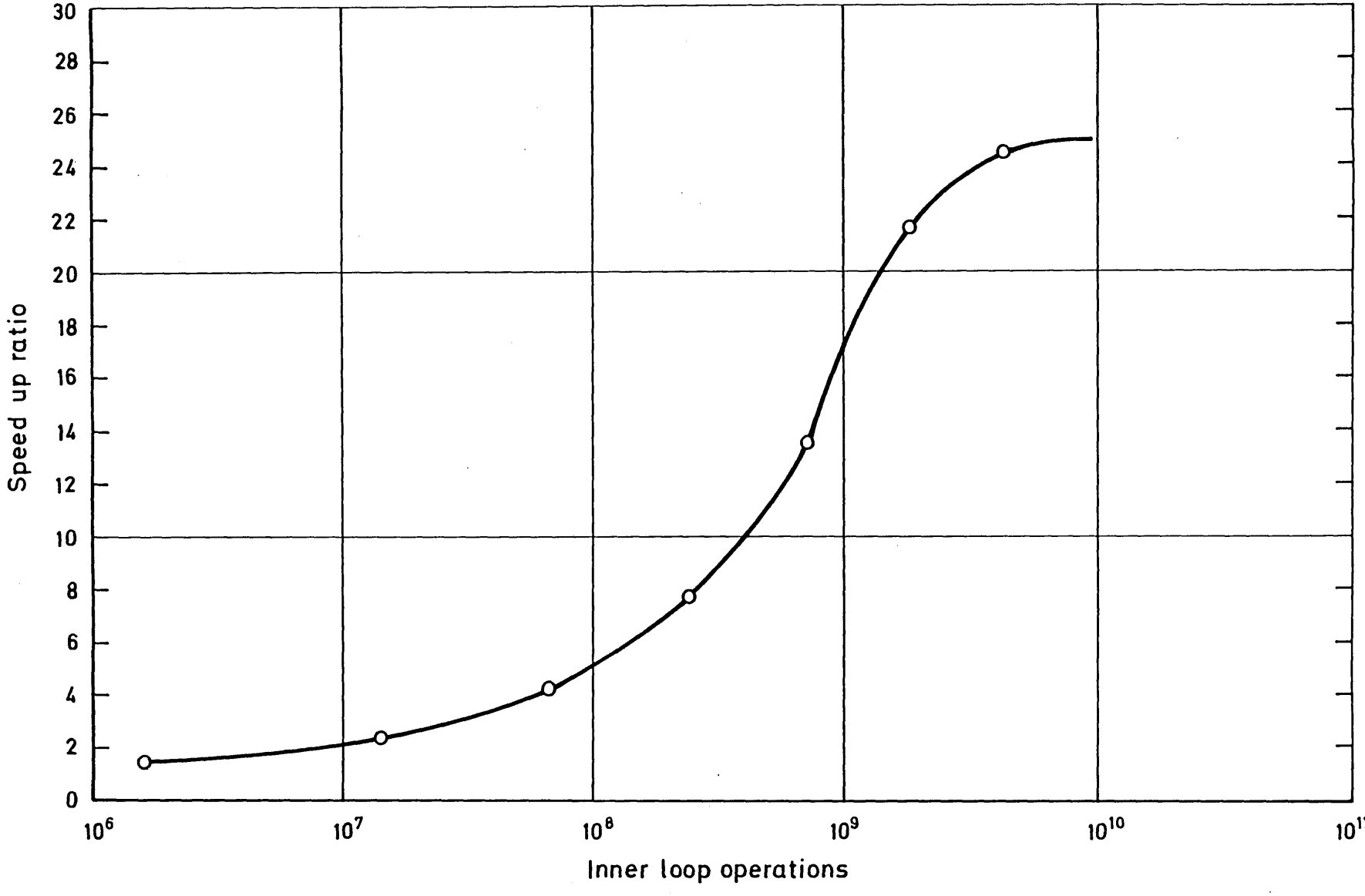


Fig. 35 Variation in speed improvement with problem size