To the Graduate Council:

I am submitting herewith a dissertation written by Robert Michael Pardue entitled "Finite Element Creep Analysis of Axisymmetric Structures Using Linear Path Functions." I recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Engineering Science.

J. E. Akin

J. E. Akin, Major Professor

We have read this dissertation
and recommend its acceptance:

Thomas G. Carley

C. W. Lee

W. T. Becker

Bruce R Dewey

Accepted for the Council

Hilton A. Smith

Vice Chancellor
Graduate Studies and Research

Thesis
76b
.P257
cop. 2

FINITE ELEMENT CREEP ANALYSIS OF AXISYMMETRIC STRUCTURES USING LINEAR PATH FUNCTIONS

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Robert Michael Pardue

December 1976

1303773

Dedicated to

Angie and Douglas

ACKNOWLEDGMENT

The author wishes to express his gratitude to Dr. J. E. Akin for his continued assistance, patience, and encouragement throughout this study. He also wishes to acknowledge his appreciation to the members of his committee and to Mrs. Pat Sculthorpe for her invaluable service in preparing this dissertation.

Special thanks are also expressed to his wife, Angie, for her patience and understanding and to his son, Douglas, for providing many bright spots during a long and sometimes trying process.

ABSTRACT

The purpose of this study was to demonstrate the effectiveness of an improved initial strain finite element method for the creep analysis of axisymmetric structures. The new method utilizes a linear path function to model the material creep behavior in the effective stress-effective creep strain plane. The technique is developed within the framework of the so called power law creep strain equation and either the strain hardening rule or the time hardening rule. An explicit expression for effective creep strain increment as a function of time increment is derived. Instantaneous material properties are assumed linearly elastic but these can be generalized to include plasticity effects.

Examples are given which illustrate the advantages of the new technique over that of the old "constant stress" approach used by numerous investigators. These examples show that the execution time of a typical finite element program to compute the solution of a creep problem to a given accuracy is reduced by a factor of four or more, depending upon the problem. Also it is shown that the stability limit on time increment for the linear path method is probably as large or larger than that for the constant stress method.

An iteration algorithm for use with the linear path function approach is presented. It is shown that using this

iteration scheme, the execution time may be reduced still further over the simple linear path function approach at essentially no loss of solution accuracy.

Methods are described for generalizing the approach to include higher order path functions as well as other parametric creep equations through the use of numerical analysis techniques. A method is suggested for combining the strain hardening rule and the time hardening rule when used with this method. Also a method for generating upper and lower bound solutions for effective creep strain is proposed which is based on the use of both the linear path and constant stress approaches.

As an aside, a method is derived for extending the useful range of the standard finite element formulation for handling nearly incompressible materials. Poisson's ratios in the range of 0.499 to 0.49999 may be used with this modification, depending on the size of the finite element model and the accuracy required.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
	Background	1
	Objective of Present Study	7
II.	THE PATH FUNCTION APPROACH	9
	Separation of Uniaxial Strain Components	10
	The Parametric Creep Equation	14
	The Hardening Rules	15
	Generalization to Three Dimensions	19
	The Path Function Concept	21
	The Linear Path Function Approach	28
	Time Limitation of the Linear Path Function	29
	The Cormeau Stability Criteria	30
	Higher Order Path Functions	31
	The Finite Element Equations	33
	The Near Incompressibility Modification	38
III.	THE LINEAR PATH FUNCTION SOLUTION ALGORITHM	43
	Starting the Solution	44
	Continuing the Solution	46
	The Iteration Algorithm	48
	The Computer Code	48
IV.	EVALUATION OF THE LINEAR PATH METHOD	52
	The Test Problem	52
	The Patch Test	54
	The Relaxation Problem	61
	The Creep Problem	68
V.	DISCUSSION OF RESULTS AND FUTURE AREAS OF ENDEAVOR	83
	Summary of Results and Conclusions	83
	Future Areas of Concern	89
LIST OF REFERENCES		93

PAGE

APPENDIXES 100

A. SHAPE FUNCTIONS FOR THE SIX NODE QUADRILATERAL ELEMENT 101

B. GAUSSIAN QUADRATURES 103

C. A USER GUIDE FOR THE CREEP PROGRAM 105

D. LISTING OF THE CREEP PROGRAM 111

VITA 132

LIST OF FIGURES

FIGURE		PAGE
2.1	Separation of Strain Into Creep and Instantaneous Components	12
2.2	Time Hardening and Strain Hardening Rules for a Two Step Loading History	16
2.3	The Constant Stress Path Approximation	22
2.4	Comparison of the Constant Stress and Linear Path Function Approximations	24
3.1	Starting the Solution	45
3.2	Continuing the Solution	47
3.3	The Iteration Algorithm	49
3.4	A Simplified Flow Chart of the CREEP Program	51
4.1	The Finite Element Demonstration Model	53
4.2	The Solution Path for the Patch Test	60
4.3	Predicted Path Functions for the Relaxation Problem	63
4.4	The Radial Stress Distribution for the Relaxation Case	64
4.5	The Tangential Stress Distribution for the Relaxation Case	65
4.6	The Axial Stress Distribution for the Relaxation Case	66
4.7	The Radial Displacement of the Outer Radius for the Relaxation Case	67
4.8	The Axial Stress in Element Number 1 for the Creep Case	70
4.9	The Tangential Stress in Element Number 1 for the Creep Case	71

FIGURE PAGE

4.10 The Axial Stress Distribution for the Creep Case 72

4.11 The Radial Stress Distribution for the Creep Case 73

4.12 The Tangential Stress Distribution for the Creep Case 74

4.13 The Effective Stress Distribution for the Creep Case 75

4.14 Convergence of the Effective Stress for the Creep Case 77

4.15 The Displacement of the Inner and Outer Cylinder Radii for the Creep Case 78

4.16 Iterated Solutions for Axial Stress 80

4.17 Iterated Solutions for Tangential Stress 81

A.1 The Six Node Quadrilateral Element 101

CHAPTER I

INTRODUCTION

1.1 BACKGROUND

The need for rapid solution techniques for the stress analysis of structures made from elastic-viscoplastic materials has grown rapidly in recent years. In addition to increased use of modern plastics, design requirements increasingly call for using materials at load levels well beyond their elastic limits and at temperatures well into the range at which creep effects occur.

Of the infinite types of structures one encounters, the axisymmetric class (such as pressure vessels, missile shells, pipes, etc.) is of sufficient importance to warrant specific investigation. Despite the reduction of difficulty from general three dimensional problems to the axisymmetric class of problems, the analysis remains of sufficient complexity to require sophisticated numerical methods.

The use of finite element computer programs to predict creep behavior in axisymmetric structures is not new. As early as 1966 Greenbaum (1) used a finite element program based on an initial strain approach to solve for stresses and displacements in axisymmetric problems. Since then several other algorithms have been developed along lines

which are similar to Greenbaum's in their treatment of the creep component of strain (2, 3, 4). The more modern of these programs are designed to handle problems with a very general class of material properties, being able to calculate elastic, plastic, and creep components of strain under general loading histories (5).

The basis of the approach used in these programs is to use the instantaneous material properties of the finite element model to solve for the initial stress-strain state at time t equal to zero. Based on this "elasticity" solution, some parametric creep equation is applied to predict an effective creep strain increment $\Delta\varepsilon^c$. The most common form of this equation appears to be the power law

$$\varepsilon^c = K(\sigma^e)^n \ (t^e)^m \qquad (1.1)$$

where ε^c is the effective creep strain, σ^e is the effective stress and t^e is the parametric time. K, n, and m are material constants chosen to best fit experimental data obtained from uniaxial creep tests. While better curve fits can be obtained with other more complex parametric forms (6, 7, 8) the simplicity of Equation (1.1) is no doubt responsible for the popularity of this equation.

The imposition of an additional restraint called the hardening rule is required to relate the manner in which the

material behaves under a changing loading history to the material behavior under constant uniaxial tension as described by Equation (1.1). The two hardening rules which have been used most frequently are the time hardening rule and the strain hardening rule as discussed in Chapter II of this dissertation.

Theories exist which circumvent the use of a hardening law (8, 9), but application of these theories requires the use of multistress level testing. Since creep tests are time consuming at best, such methods have not gained widespread acceptance despite the claims of improved accuracy for variable loading histories.

Once a choice of hardening law has been selected, Equation (1.1) is then used along with the well known Prandtl-Reuss equations to calculate the creep strain components for three dimension as is described in Chapter II. These creep strain increments are now treated as "initial strains" and are used to appropriately modify the force vector. Another elasticity solution can be obtained using the modified force vector with the original stiffness matrix. This two phase process of dividing up the structural behavior into instantaneous effects and "creep" effects is the initial strain approach to the solution of time dependent material property problems. The two phases of the

solution are alternated until enough time increments have been added to step the solution out to the required elapsed time.

In the basic approach used by investigators (10) up to the present, σ^e is assumed to be held constant for the duration of the time increment Δt, at the end of which a stress redistribution is assumed to take place instantaneously. This approach simplifies the use of Equation (1.1), but at the cost of accuracy of the solution for displacements and stresses at time $t + \Delta t$. The error in the solution from each time step is carried along into the next time increment, with errors accumulating with each new time step.

The need to maintain the numerical accuracy of the solution algorithm within certain limits is one factor which governs the size of the time increment. Typical results in the literature (2, 11, 12, 13) indicate that frequently a hundred or more time increments are required to reach the desired elapsed time while maintaining the necessary accuracy of solution. This means that creep solutions, especially for large problems, are costly and time consuming if accurate results are to be obtained.

It should be pointed out that the term accuracy as is used in this report refers to the nearness of the numerical finite element solution to the exact theoretical solution.

This leaves unanswered, the question of how accurately the assumptions made about structural and material behavior actually fit the true case. While some discussion of this topic is included in Chapter II, in general the subject is left to others to describe more fully (6, 7, 8, 9). It is assumed that the exact theoretical solution as implied by the defining equations of equilibrium, compatibility, etc., along with any assumptions concerning material behavior is the "correct" solution. It is this solution which the numerical methods to be described in this text are aimed at approximating.

A second factor which controls time increment size stems from the inherent numerical instability of all "marching" type solution techniques. This stability factor imposes an upper limit on the time increment which is in general a function of stress level and material properties—both elastic and time dependent. As shown by Cormeau (14), an explicit relation for the maximum time increment can be derived for the case of the constant stress method. Using a time increment greater than this maximum results in error terms which grow without bounds as the solution is "marched" out in time.

Which of the two factors predominates depends on the accuracy of the solution required and the type of problem under consideration. If the solution accuracy required is

low enough that the instability factor predominates, then in general the desired solution can be rapidly obtained using only the constant stress technique. The stability limit on time increment is not highly restrictive and only a few (say ten) solution cycles may be required to reach the steady state creep solution.

In the past, iteration schemes have been proposed to improve the accuracy of the constant stress approach while increasing the time increment size (though not beyond the stability limit). The results of these attempts have generally been less than favorable (10). The improvement in accuracy has not been sufficient to justify the increase in program complexity and computer storage requirements. Thus the usual response to this problem has been, if the analysts needed more accuracy he simply reduced his time step.

The disadvantage of the constant stress approach is that it is too crude a model to accurately represent the actual behavior of the stress level for large time steps. The true path of a point through the effective stress-effective strain (σ^e-ε^c) plane is frequently one in which the magnitude of the slope of this path or curve is rapidly decreasing. Thus the stress level quickly moves away from the initial stress level and then more gradually approaches the final one. The result is that the constant stress level

assumed to be present for the duration of the time increment quickly becomes inaccurate.

1.2 OBJECTIVE OF PRESENT STUDY

What is needed is a better method of approximating the true path in the σ^e-ε^c space. The general method described in the analysis is called the path function approach for obvious reasons. It is an attempt to remedy a long standing weakness in the currently used constant stress approach. The true path in the σ^e-ε^c plane is modeled not by a series of step functions, but in the present work by straight line segments. Other, more complicated path functions are possible, but as will be explained in Chapter II, the linear path function approach leads to a simple, accurate technique, which as of now, is not possible with higher level path functions such as quadratic and cubic polynomials. In Chapter II the linear path function approach as applied to the initial strain finite element method is discussed in detail. As finite element solutions based on the elasticity matrix, [D], deteriorate as Poisson's ratio approaches 0.5, a modification was made to the formulation which greatly improved the stress solution accuracy. A discussion of this new technique is included in Chapter II.

In Chapter III the solution algorithm developed from the linear path function theory is described. The algorithm includes provisions for starting the solution and continuing the solution. An iteration scheme is also described which has proven to be very effective in reducing numerical error in a relatively few iterations per time increment.

To verify the method a finite element model was created for the solution of a thick walled axisymmetric cylinder. A patch test, a relaxation problem, and a creep problem were run with this model as described in Chapter IV. It was felt that the choice of the creep problem and the relaxation problem encompassed a sufficiently broad class of problems to prove the generality of the method. Also the creep problem and the patch test made possible comparison of the numerical solution with known exact solutions.

A discussion of the results of Chapter IV as well as recommendations for future work in the area is included in Chapter V.

CHAPTER II

THE PATH FUNCTION APPROACH

In this chapter the equations necessary for the development of the path function approach are presented. The initial discussion will be aimed at describing uniaxial creep behavior and then at the appropriate point the use of the Prandtl-Reuss equations will be introduced to generalize to the three dimensional state of stress and strain. Also discussed is the application of hardening laws for the extension of a parametric creep equation based on constant tension creep data to cover general loading histories.

The above topics provide the necessary foundation for the path function concept. Within this framework the additional assumption is made that the material behavior, if forced to follow a given path through the effective stress-effective creep strain plane, will approximate actual material behavior to a degree of accuracy which is dependent on the nearness of the assumed path to the actual path. This assumption is the basis of the path function approach.

Though not described as such, this assumption has been made by the various investigators that have used the constant stress approach in the solution of creep problems. The path

that they have chosen is the step function. This is obviously not a particularly good model of a true path which in general is smooth and varying. The last section of this chapter is devoted to deriving the equations necessary to generalize from the above step function approach to a piecewise linear function. The possibility of using still higher order path approximations is also discussed.

The final section of this chapter contains the finite element equations which apply to the solution of axisymmetric problems. The initial strain method is summarized. Since the above equations can be found in any of several excellent texts, little detail is included. A modification to this theory, which was developed during the course of the present investigations to handle the near incompressibility situation in the elasticity part of the solution is presented in detail. It is felt that this modification should be quite useful to analysts working with materials having Poisson's ratios in the range of say 0.499 to 0.49999. Little or no loss of program efficiency resulted from this minor programming modification as discussed in Chapter V.

2.1 SEPARATION OF UNIAXIAL STRAIN COMPONENTS

For the moment, consider only the case of a uniaxial test specimen under an arbitrary loading history. The

premise upon which all subsequent discussion in the work is based, is that the total accumulated strain, ε, which a material experiences can be split up into a time independent component, ε^{i}, and a time dependent component, ε^{c}. The time independent component, referred to as the instantaneous component includes the elastic component and the plastic component. In this report no attempt will be made to extend the analysis to include either nonlinear elastic effects or plastic effects.

This in no way limits the generality of the proposed method. Inclusion of such effects is straight forward and is discussed in numerous papers in the literature (10, 12, 15).

The time dependent component of strain is usually referred to as the creep strain component. It has been observed that in uniaxial constant tension creep tests (called simply creep tests for the rest of this report) three more or less will defined stages of straining occur (see Figure 2.1). Initially the strain rate is high.

This rate monotonically drops off to a nearly constant value which persists for a relatively long period of time. The strain which occurs in the initial transient stage is termed primary strain while strain which occurs in the steady state stage is termed secondary strain. The division between primary and secondary strain is somewhat arbitrary

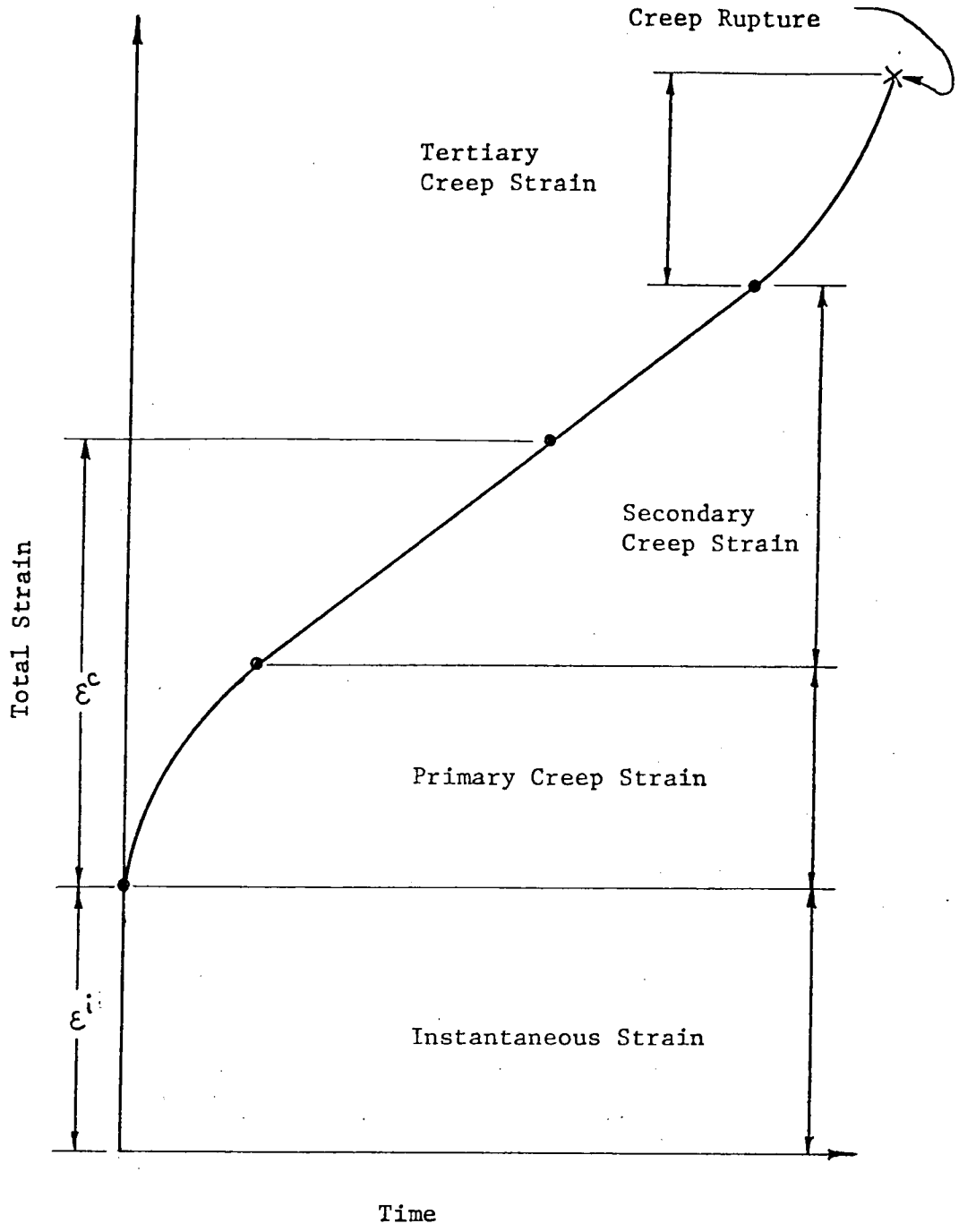


Figure 2.1. Separation of Strain Into Creep and Instantaneous Components

since the strain rate in fact is never truly constant. The third or tertiary stage of strain is characterized by an increase in strain rate and ultimate failure of the specimen called creep rupture. Historically most analysts have limited their work to description of the primary and secondary strain components since the tertiary stage is unstable and difficult to describe. For practical purposes the useful life of a structure is over at the onset of tertiary strain. No attempt will be made to include tertiary strain effects in the present work.

Mathematically the idea of separation of strain into instantaneous and creep strain components is expressed as

$$\varepsilon = \varepsilon^{i} + \varepsilon^{c} \quad . \tag{2.1}$$

While it is obvious that by definition the total strain can always be subdivided as discussed above, it is not obvious that Equation (2.1) represents true material behavior. As discussed by Perzyna (16), it is possible that the concept of an instantaneous strain component is entirely fictitious. It is also questionable whether or not it is logical to separate plastic and creep strain since under certain loading histories the distinction between the two breaks down. Despite these arguments, use of Equation (2.1) greatly simplifies viscoplastoelastic analysis and it is widely used.

2.2 THE PARAMETRIC CREEP EQUATION

Since 1910, when Andrade first introduced his one third law (17) to describe the behavior of metal in a creep test, various forms of parametric equations have been proposed to fit creep test data. Of the forms proposed, the one most frequently encountered in the literature in connection with the finite element method is the so called power law

$$\varepsilon^c = K(\sigma^e)^n \ (t^e)^m \quad . \tag{2.2}$$

The popularity of this equation stems not from its accuracy but from its simplicity. There are in fact several equations which in general fit creep test data better than Equation (2.2). Several investigators (7, 8, 9) have proposed equations involving exponential or hyperbolic functions which given better empirical fit to creep test data than Equation (2.2). Due to their increased complexity they are less frequently chosen for the analysis of large structures.

Equation (2.2) is chosen for the present work for a more basic reason. As will be shown later in this chapter, the use of Equation (2.2) along with a suitable hardening law can be combined with the linear path function approximation to yield an explicit expression for creep strain increment as a function of time.

2.3 THE HARDENING RULES

Since in general a structure undergoes variable loading conditions Equation (2.2) cannot be directly applied to obtain the amount of creep strain accumulated over a given period of time. It is necessary to first make an additional assumption as to how the behavior of a material under a varying load relates to its behavior under a constant load as expressed by Equation (2.2). Several such assumptions, called hardening rules, have been suggested, none of which accurately describes material behavior under all loading conditions. Historically two hardening rules have received most of the attention. One is the time hardening rule and the other is the strain hardening rule. Figure 2.2 illustrates the predicted creep strain obtained from the time hardening law and the strain hardening law for a simple two step loading history.

The time hardening law says that regardless of the previous loading history, the material will behave according to the current stress level and according to the elapsed time that the material has been subjected to a nonzero loading. This rule has a logical fallacy in the definition of elapsed time since there is a discontinuity in time as stress goes to zero. That is, it is assumed that at some arbitrary time t equal to zero a nonzero load is applied

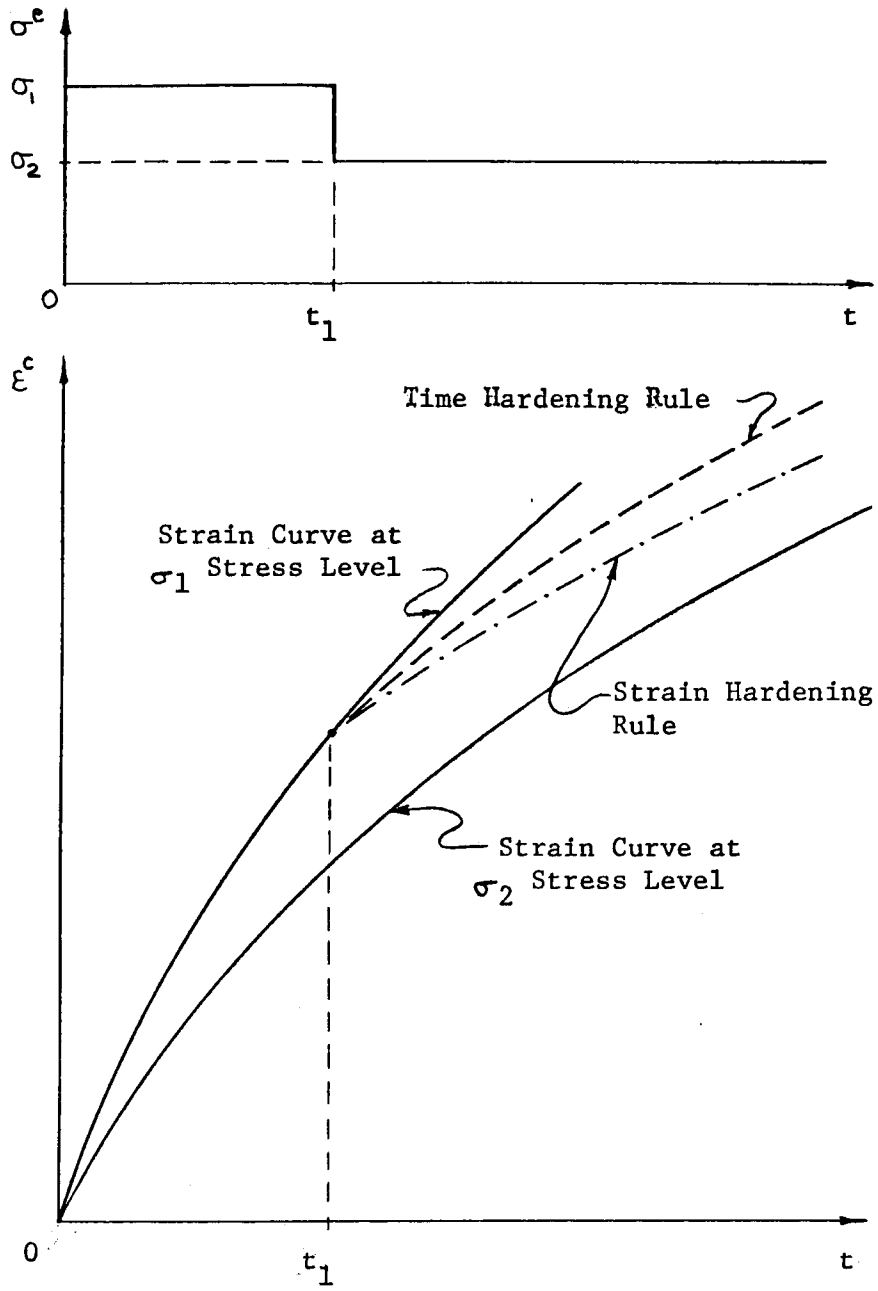


Figure 2.2. Time Hardening and Strain Hardening Rules for a Two Step Loading History

to the material. Up to time t equal to zero no load has been applied. Since no provision is made for the possibility of the load dropping again to zero and being held there for some finite time prior to again reloading with a nonzero load what assumption is to be made on the aging of the material? If the analysts assumes that aging is discontinued while no loading is present, then what does he assume if an infinitesimal load be applied? If he assumes that aging continues always, then in effect he is violating the premise that the "life" of the material began at time equal to zero. Since, however, the rule is very easy to apply, and since for some cases results are reasonable, this rule nonetheless remains popular.

The strain hardening rule is also easily applied and in general gives better results than the time hardening law. This accounts for the greater popularity of the latter in the literature. The strain hardening rule states that the material will behave according to the current stress level and according to the total accumulated creep strain regardless of previous stress history. Total accumulated creep strain is defined as

$$\varepsilon^c = \int_{-\infty}^{t} d\varepsilon^c \tag{2.3}$$

where $d\varepsilon^c$ is greater than or equal to zero for all time t. The effect of a compressive stress is accounted for in the case of a uniaxial stress by defining an effective stress to be

$$\sigma^e = |\sigma| \tag{2.4}$$

where $|\sigma|$ is the magnitude of the axial stress σ. In the following section this equation will be generalized to handle the three dimensional case.

The strain hardening rule may be expressed mathematically as

$$\dot{\varepsilon}^c = f(\sigma^e, \varepsilon^c) \tag{2.5}$$

where the dot indicates rate of change with time. This is in effect an equation of state of two variables, namely accumulated creep strain ε^c and applied effective stress σ^e.

It is well known that Equation (2.5) does not accurately describe material behavior in interrupted creep tests (18, 19, 20, 21) on certain materials. Attempts have been made to improve on the above hardening rules by including both time and strain hardening effects (6). None of these attempts have succeeded in describing material behavior under all loading conditions. In the analysis presented in the current work, the strain hardening rule

will be used extensively. While the creep strain equations are derived for the time hardening rule as a matter of exercise, the generally improved material behavior characterization afforded by the strain hardening rule warrants its use over the time hardening rule. Discussion of a method for combining the two rules can be found in Chapter V.

2.4 GENERALIZATION TO THREE DIMENSIONS

Up to now the discussion has been limited to the case of uniaxial loading. Before the preceding equations can be applied to the case of axisymmetric loading it is necessary to present equations which relate material behavior under uniaxial loading to the more general case of three dimensional loading. It will be assumed that the material behaves under a complex state of stress according to the Prandtl-Reuss incremental creep strain equations. In tensor notation these equations are given as

$$d\varepsilon_{ij}^{c} = \frac{3}{2} \frac{d\varepsilon^{c}}{\sigma^{e}} s_{ij} \tag{2.6}$$

where $d\varepsilon_{ij}^{c}$ are the tensor components of creep strain and s_{ij} are the deviatoric stress components defined in terms of the stress components σ_{ij} to be

$$s_{ij} = \sigma_{ij} - \sigma\delta_{ij} \tag{2.7}$$

$$\sigma = \frac{1}{3}\sigma_{kk} \quad . \tag{2.8}$$

In the above equations the summation convention is assumed to apply to the k subscript and extends from k equal to one to three, δ_{ij} is the well known Kronecker delta symbol, and σ is the mean stress. The effective stress σ^e is defined in terms of the deviatoric stress tensor

$$\sigma^e = \sqrt{\frac{3}{2} s_{ij} \cdot s_{ij}} \quad . \tag{2.9}$$

Other equations can be used in place of Equation (2.6) which allow for variation of $d\varepsilon_{ij}^c$ with second order powers of s_{ij} (6). Such variations are generally small however, and the added complexity of the analysis does not justify the use of such equations. They will not be considered further in this discussion. Also the possibility of using other theories of creep such as flow theory with aging or total deformation theory will not be considered.

To prepare the way for later discussion it will be useful if notation appropriate for axisymmetric problems is now introduced. Thus in terms of radial, tangential, and axial components (r, t, z) Equation (2.6) can be written as

$$d\varepsilon_r = \frac{d\varepsilon^c}{\sigma^e}\left(\sigma_r - \frac{1}{2}\sigma_t - \frac{1}{2}\sigma_z\right)$$

$$d\varepsilon_z = \frac{d\varepsilon^c}{\sigma^e}\left(\sigma_z - \frac{1}{2}\sigma_r - \frac{1}{2}\sigma_t\right)$$

$$d\varepsilon_{rz} = \frac{d\varepsilon^c}{\sigma^e}\left(\frac{3}{2}\sigma_{rz}\right)$$

$$d\varepsilon_t = \frac{d\varepsilon^c}{\sigma^e}\left(\sigma_t - \frac{1}{2}\sigma_z - \frac{1}{2}\sigma_r\right) \quad . \qquad (2.10)$$

Also Equation (2.9) can be written as

$$\sigma^e = \frac{1}{\sqrt{2}}\left(\sigma_r - \sigma_t\right)^2 + \left(\sigma_r - \sigma_z\right)^2 + \left(\sigma_t - \sigma_z\right)^2 + 6\,\sigma_{rz}{}^2\Big)^{1/2} \quad . \qquad (2.11)$$

2.5 THE PATH FUNCTION CONCEPT

To date, the technique used to apply Equation (2.2) (along with one of the hardening rules) to the initial strain finite element method has been the so called constant stress method (see Figure 2.3). In this approach the actual material stress history is approximated by a series of constant stress steps in time. The initial state of stress at point A is solved for "elastically" and is then assumed to persist for a specified time increment during which the material creeps to point B. At the end of this interval the initial state of stress is then relaxed or redistributed

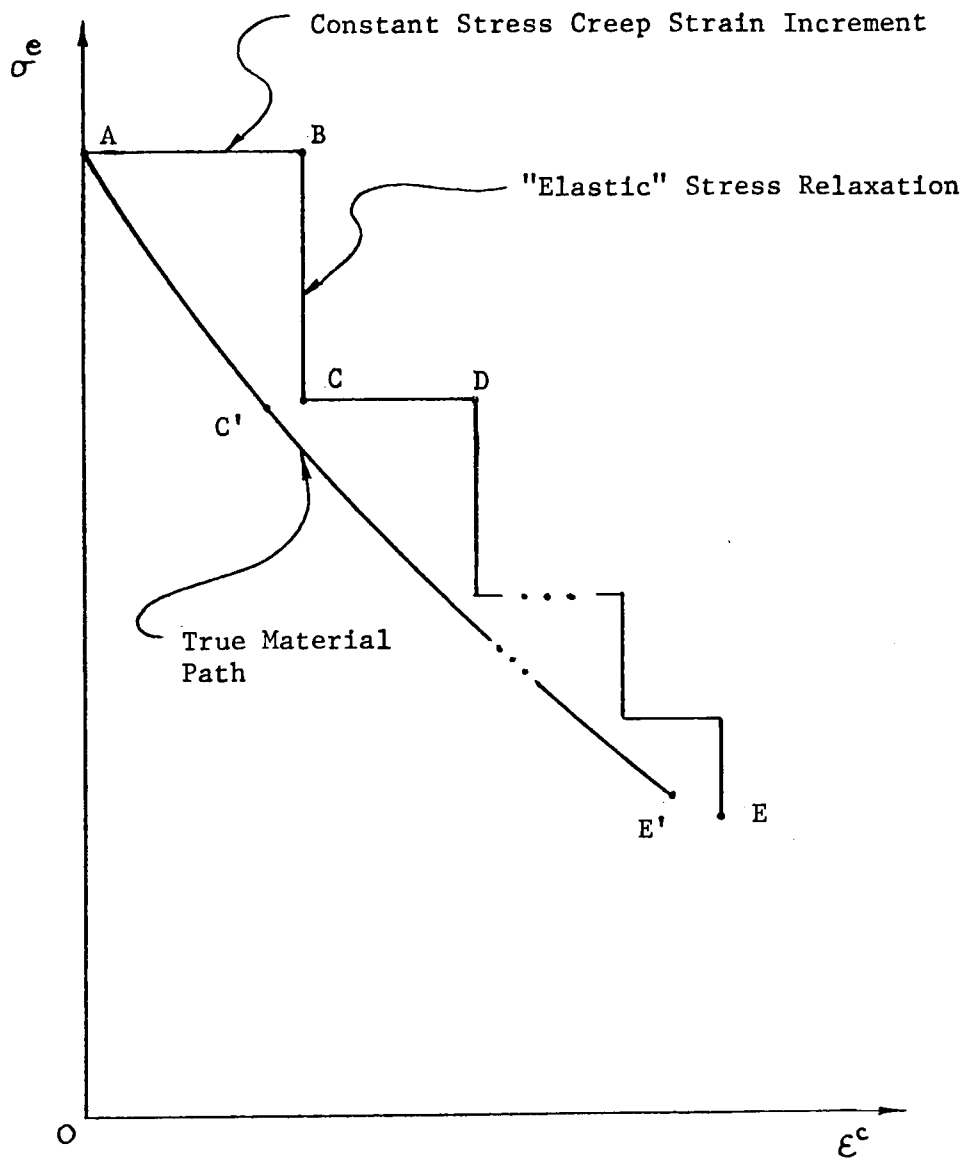


Figure 2.3. The Constant Stress Path Approximation

"elastically" to point C to adjust for the creep strains accumulated over the time interval. Point C is assumed to approximate a point C' of the true material path as shown in Figure 2.3. A series of creep phase-relaxation phase cycles is repeated until the desired total elapsed time is achieved. Thus the true path OAC'E', which the material (as specified by the material property assumptions) actually follows is approximated by the step function path OABCD . . . E.

To improve on this approach consider at time t_A a material at point A in the effective stress-effective creep strain plane as indicated in Figure 2.4. Assuming no sudden changes in applied loads over the time interval from t_A to t_B the material will in general undergo a smooth change in stress-strain to arrive at point B. As can be seen from the figure, unless the material path from A to B is nearly flat and has a small slope, the constant stress approximation (path ACD) to the true path is relatively crude.

As an alternative, let it be assumed that the material follows the straight line from A to E as shown in Figure 2.4. The slope of line AE is that of the true path at point A. At point E, the material can be relaxed elastically to reach point F. It would be logical to assume that in general path AEF would lie closer to the true path AB than does path ACD. That such is the case is amply demonstrated in Chapter IV of this work.

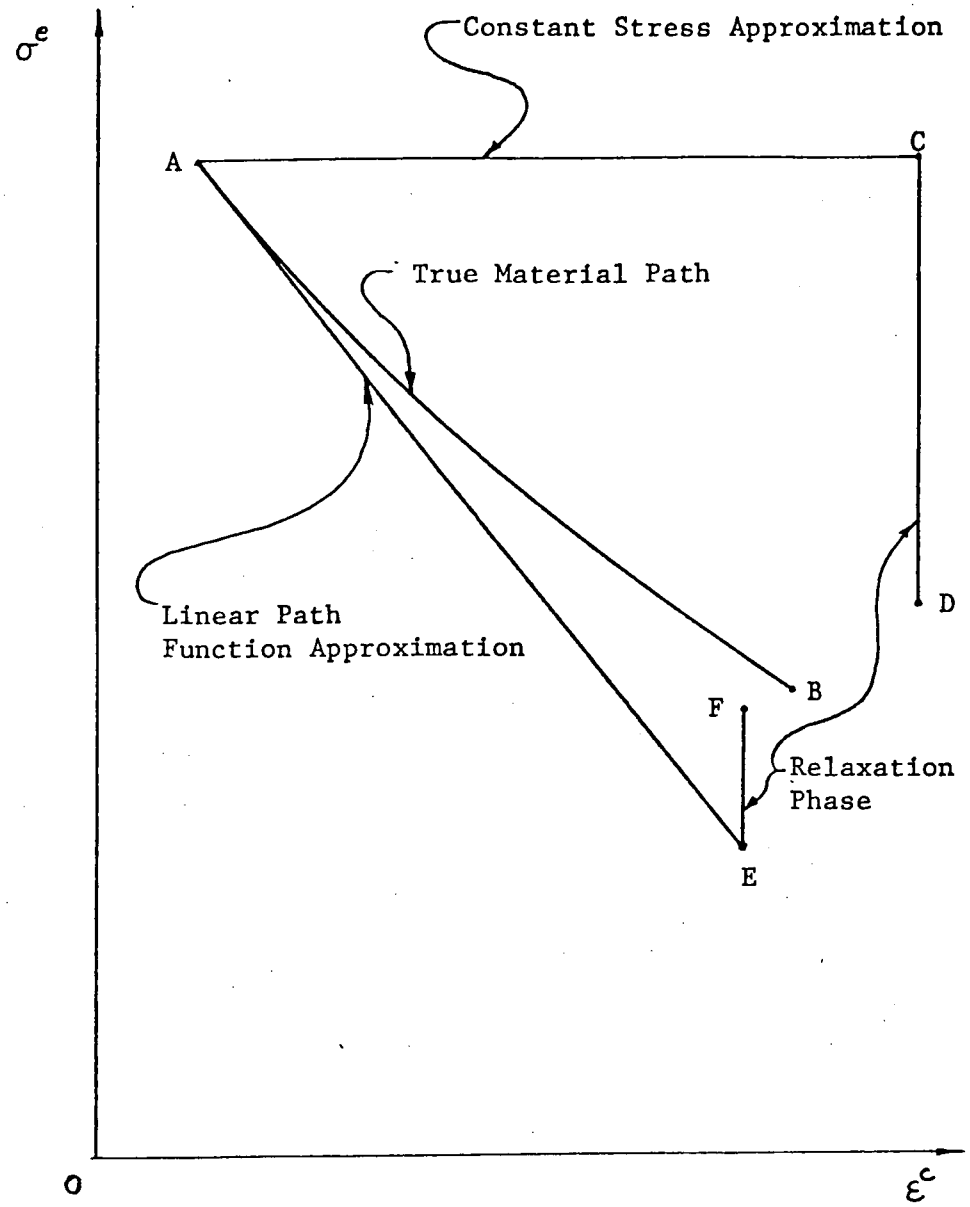


Figure 2.4. Comparison of the Constant Stress and Linear Path Function Approximations

For the present, let it be assumed that a general class of path functions exist

$$\sigma^e = P(\varepsilon^c) \tag{2.12}$$

which can be used to approximate the true material behavior over a specified time increment. Later it will be shown that the linear path function leads to an efficient solution algorithm, but it is not as yet necessary to limit the choice of $P(\varepsilon^c)$.

In the application of Equation (2.12) a complication arises in conjunction with Equation (2.2). Under the assumption of strain hardening, all that is required to apply Equation (2.2) in the constant stress method is to derive an expression for parametric time t^e. Thus solving Equation (2.2) for parametric time t^e leads to

$$t^e = \left(\frac{\varepsilon^c}{K\ (\sigma^e)^n} \right)^{1/m} . \tag{2.13}$$

Since effective stress σ_1^e and effective creep strain $\varepsilon^c{}_1$ at time t_1 are known at the beginning of each time increment Equation (2.13) can be used to evaluate the corresponding parametric time

$$t_1^e = \left(\frac{\varepsilon_1^c}{K\ (\sigma_1^e)^n} \right)^{1/m} . \tag{2.14}$$

Then Equation (2.2) can be used to calculate the effective creep strain ε_2^c at the end of the time increment Δt

$$\varepsilon_2^c = K\ (\sigma_1^e)^n\ (t_2^e)^m \tag{2.15}$$

where $$t_2^e = t_1^e + \Delta t \quad .$$

The effective creep strain increment is then given by

$$\Delta\varepsilon^c = \varepsilon_2^c - \varepsilon_1^c \quad . \tag{2.16}$$

If time hardening is used in the constant stress approach then Equation (2.14) is not required. The effective creep strain increment is simply evaluated directly using Equation (2.2). Thus

$$\Delta\varepsilon^c = K(\sigma_1^e)^n\ (t_2^e)^m - K(\sigma_1^e)^n\ (t_1^e)^m \quad . \tag{2.17}$$

In the case of the linear path function however stress is not a constant over a given time increment so in general it is necessary to form a differential equation relating effective creep strain rate to stress and time. To derive this equation for strain hardening we may begin with Equation (2.13) and take the partial derivative of this equation with respect to ε^c to get

$$\frac{\partial t^e}{\partial \varepsilon^c} = \frac{1}{mK^{1/m}} \frac{(\varepsilon^c)^{(1/m-1)}}{(\sigma^e)^{n/m}} \quad . \tag{2.18}$$

The strain hardening criterium implies that

$$\frac{dt}{d\varepsilon^c} = \frac{\partial t^e}{\partial \varepsilon^c} \tag{2.19}$$

so applying Equations (2.19), (2.12), and (2.18) results in

$$dt = \frac{1}{mK^{1/m}} \frac{(\varepsilon^c)^{(1-m)/m}}{(P(\varepsilon^c))^{n/m}} d\varepsilon^c \quad . \tag{2.20}$$

This equation is the basis for the path function method as applied in the remainder of this analysis. It is an ordinary differential equation relating incremental creep strain to incremental real time. As will be shown in the next section, for suitable choice of the function $P(\varepsilon^c)$ this expression leads to an explicit function for ε^c as a function to time t.

In the case of time hardening a similar equation can be derived from Equation (2.2). The derivation is somewhat more straightforward, however, since the parametric time t^e is now the true time t. Thus differentiating Equation (2.2) with respect to time t there follows

$$\frac{d\varepsilon^c}{dt} = mK\sigma^n \, t^{m-1} \quad .$$

Using Equation (2.12) a form similar to Equation (2.20) is obtained:

$$\frac{1}{mK} \frac{d\varepsilon^c}{(P(\varepsilon^c))^n} = t^{m-1} \, dt \quad . \tag{2.21}$$

This again may be integrated for suitable choice of $P(\varepsilon^c)$ to obtain an expression for ε^c as a function of time t.

2.6 THE LINEAR PATH FUNCTION APPROACH

The explicit integration of Equation (2.20) (or Equation (2.21)) is not possible in general. However if during a typical time step the linear form

$$P(\varepsilon^c) = A + B\varepsilon^c \tag{2.22}$$

is chosen for the path function and if m is set to unity, then not only can Equation (2.20) be integrated in closed form, but the function can be inverted to yield an explicit expression for ε^c as a function of time t. Thus substituting Equation (2.22) into Equation (2.20) and setting m to one yields

$$dt = \frac{1}{K} \frac{1}{(A + B\varepsilon^c)^n} \, d\varepsilon^c \quad ,$$

or integrating

$$t - t_1 = \frac{1}{KB} \frac{(A + B\varepsilon^c)^{1-n}}{1-n} \Bigg|_{\varepsilon_1}^{\varepsilon} \tag{2.23}$$

where (see Figure 2.3) t_1 = time at point A, ε_1 = effective creep strain at point A, t = time at arbitrary point C, and ε = effective creep strain at point C. Equation (2.23) can then be inverted to obtain

$$\varepsilon^c = \frac{1}{B} \left[KB(1-n)\ (t-t_1) + (A + B\varepsilon_1)^{1-n}\right]^{\frac{1}{1-n}} - \frac{A}{B} \quad . \tag{2.24}$$

Thus for a given time step (within certain limits prescribed below) and for arbitrary choice of path constants A and B the creep strain, ε^c, can be directly obtained. The application of this equation is discussed in the derivation of the governing finite element equations.

2.7 TIME LIMITATION OF THE LINEAR PATH FUNCTION

A limitation on time step occurs for B greater than zero and is a property of Equation (2.24) since in general n will be greater than one it can be seen that ε^c goes to infinity as the term in brackets in Equation (2.24) goes to zero. That is, the expression

$$KB\ (1-n)\ (t-t_1) + (A + B\varepsilon_1)^{1-n} = 0 \qquad (2.25)$$

defines a limit on the maximum time step which is consistent with the linear path function. Let Δt be defined as

$$\Delta t = t - t_1 \quad . \qquad (2.26)$$

Then if Δt_{max} is that time step which corresponds to the limit set by Equation (2.25), solving for Δt_{max} in Equation (2.25) yields

$$\Delta t_{max} = - \frac{(A + B\varepsilon_1)^{1-n}}{KB\ (1-n)} \qquad (2.27)$$

as the upper bound on time step which can be taken with the current choice of path function.

2.8 THE CORMEAU STABILITY CRITERIA

A more basic time increment size limit than that discussed in Section 2.7 is the so called Cormeau stability criteria. As shown in reference (14) for an isotropic material an upper limit on the size of time increment which can be chosen in a marching type solution with constant stress is given by

$$\Delta t_c = \frac{4}{3} \frac{(1 + \nu)}{EKt^{m-1}} \frac{1}{n\ (\sigma^e)^{n-1}} \qquad (2.28)$$

where E and ν are the elastic modulus and Poisson's ratio respectively. This same result had previously been derived by Irons and Treharne (22) as a result of a stress perturbation creep analysis.

As later examples will show, in certain cases where the linear path function approach was used, the above limit did not apply. In fact, once a nearly steady path (not necessarily constant) function was achieved it was found that an almost unlimited time increment could be chosen without loss of accuracy, much less instability setting in.

2.9 HIGHER ORDER PATH FUNCTIONS

There are obviously alternatives to Equation (2.18) as a path function. At this time, no other choices for $P(\varepsilon^c)$ are known which allow for a closed form solution of ε^c as a function of t. A direct numerical method can be obtained which calculates ε^c as a function of t. However, a two step numerical procedure would have to be adopted. First Equation (2.20) would have to be integrated numerically for some choice of ε. The obtained value of t would then need to be compared to the actual value of t. An iterative loop could then be established by making an improved guess on ε and repeating the cycle until suitable agreement between predicted time and actual time was achieved.

Moreover, the advantage of improved accuracy which can be obtained from higher order polynomials or other suitable choices for $P(\varepsilon^c)$ require that more information be created and stored concerning the material behavior. For example, if

$$P(\varepsilon^c) = A + B\varepsilon^c + C(\varepsilon^c)^2 \tag{2.29}$$

were chosen as the path function it would be required that additional information be known about the path the material follows. Either the initial slope of the path or an additional value of ε^c at some intermediate time could be used to determine the value of the new unknown coefficient C. This would require that the additional coefficient C be stored for each time step.

If the equation

$$P(\varepsilon^c) = A + B\varepsilon^c + C(\varepsilon^c)^2 + D(\varepsilon^c)^3 \tag{2.30}$$

were chosen, then four coefficients would need to be calculated and stored. In this case the obvious choice for boundary conditions would be to match value and slope of the path function $P(\varepsilon^c)$ to the initial value and slope and to the final value and "small time step" slope at that final value. (See Section 3.1 for discussion of the "reduced time increment" slope). With either of the above choices,

iteration would be required between the creep solution and the elasticity solution to take advantage of the improved approximation properties of those functions. Thus going to higher order path functions would seem to be more expensive both in terms of computer time and in storage requirements. If an efficient algorithm could be developed which would allow rapid evaluation of ε as a function of time however, such path functions might still prove more efficient in the long run than the linear path function. With no such algorithm available at present, the choice of the linear path function seems best.

2.10 THE FINITE ELEMENT EQUATIONS

The derivation of the governing finite element equations for the initial strain method may be found in several finite element texts (23, 24). However for convenience, the following is a brief derivation for the method used here. The only addition to that found in the literature is the modification of the constitutive equation formulation to improve the solution accuracy in the vicinity of $\nu = 0.5$. This change only affects the calculation of stresses from the displacements and in no way affects the stiffness matrix equations or the column vector equations. The discussion of these modifications follows in the next section.

As shown in Zienkiewicz's text (23) we begin by relating the displacement of a material point to the displacements of the nodal points using the equation

$$\{f\} = [N]\ \{\delta\}^e = [N_i,\ N_j,\ N_m\ \cdots] \begin{Bmatrix} \delta_i \\ \delta_j \\ \delta_m \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix} \tag{2.31}$$

where $\{\delta\}^e$ is the vector of nodal displacements of the finite element model and [N] is some assumed shape function. For axisymmetric problems

$$\{f\} = \begin{Bmatrix} u\ (r,z) \\ w\ (r,z) \end{Bmatrix} \tag{2.32}$$

where u and w are the displacements of a natural point in the r and z directions, respectively.

Also

$$\{\delta_i\} = \begin{Bmatrix} u_i \\ w_i \end{Bmatrix} \tag{2.33}$$

where u_i and w_i are the r-z components of displacement at the ith node.

Strain components are then related to displacements through the matrix equation

$$\{\varepsilon\} = [B]\ \{\delta\}^e \tag{2.34}$$

where [B] is defined in terms of the shape functions [N] and follows from the strain-displacement relations

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_r \\ \varepsilon_z \\ \varepsilon_{rz} \\ \varepsilon_t \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{\partial w}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \\ \frac{u}{r} \end{Bmatrix} . \tag{2.35}$$

Define the partitions of [B] and [N] as

$$[B] = [B_i \quad B_j \quad B_k \quad \cdots] \tag{2.36}$$

$$[N] = [N_i \quad N_j \quad N_k \quad \cdots] \tag{2.37}$$

where i, j, k, $\cdots$ extend over the nodes of the element containing the point under consideration.

Combining Equation (2.32) and Equation (2.35) leads to

$$[B_i] = \begin{bmatrix} \frac{\partial N_i}{\partial r} & 0 \\ 0 & \frac{\partial N_i}{\partial z} \\ \frac{\partial N_i}{\partial r} & \frac{\partial N_i}{\partial z} \\ \frac{N_i}{r} & 0 \end{bmatrix} . \tag{2.38}$$

The material properties are assumed separable into long term creep properties and instantaneous "elastic" properties. The equation defining the creep strain components has already been presented in a previous section of this chapter. It is assumed that the elastic properties are given by

$$\{\sigma\} = [D]\ (\{\varepsilon\} - \{\varepsilon_0\}) + \{\sigma_0\} \tag{2.39}$$

where [D] is the elasticity matrix (see next section for modifications), $\{\varepsilon_0\}$ is the "initial" strain, and $\{\sigma_0\}$ is the initial stress present in the material.

For axisymmetric problems

$$[D] = \frac{E\ (1-\nu)}{(1+\nu)\ (1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & 0 & \frac{\nu}{1-\nu} \\ \frac{\nu}{1-\nu} & 1 & 0 & \frac{\nu}{1-\nu} \\ 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 1 \end{bmatrix} . \tag{2.40}$$

In the initial strain method of solution for creep problems, the creep strain components are included in Equation (2.39) in the form of initial strains.

By the method of virtual work, the above expressions can be used to integrate the virtual work performed over the entire structure. Setting this equal to the work performed by the external loads we get

$$[K]\ \{\delta\} + \{F\}_p + \{F\}_b + \{F\}_{\varepsilon_o} + \{F\}_{r_o} - \{R\} = 0 \tag{2.41}$$

where for N elements:

$$[K] = \sum_{i=1}^{N} \int_{v_i} [B_i]^T\ [D_i]\ [B_i]\ d(vol) \tag{2.42}$$

$$\{F\}_p = -\sum_{i=1}^{N} \int_{v_i} [N_i]^T\ \{p\}\ d(vol) \tag{2.43}$$

$$\{F\}_b = -\sum_{i=1}^{N} \int_{s_i} [N_i]^T\ \{g\}\ d(area) \tag{2.44}$$

$$\{F\}_{\varepsilon_0} = -\sum_{i=1}^{N} \int_{v_i} [B_i]^T\ [D_i]\ \{\varepsilon_o\}\ d(vol) \tag{2.45}$$

$$\{F\}_{\sigma_0} = \sum_{i=1}^{N} \int_{v_i} [B_i]^T\ \{\sigma_o\}\ d(vol) \tag{2.46}$$

$\{R\}$ = vector of applied external loads.

and where the subscripts p, b, ε_o, and σ_o denote loads due to body forces, surface tractions, initial strains, and initial stresses, respectively.

Equation (2.41) is the "elasticity" equation describing the instantaneous behavior of the structure. Starting with the assumption that no creep strain has been accumulated up until time t equal to zero (though initial strains and

stresses might be present) an initial elasticity solution for the stresses in the structure is obtained. These stresses are then used in the previously defined creep strain equations to determine creep strain components. Treating these creep strain components as initial strains, the system load vector (the sum of Equations (2.43) through (2.46) plus {R}) is then augmented by the contribution of $\{F\}_{\varepsilon_o}$. A new elasticity solution can now be obtained, and so on.

2.11 THE NEAR INCOMPRESSIBILITY MODIFICATION

A modification in the usual formulation for stress calculations was made which considerably improves the accuracy of stress calculations in the case of nearly incompressible material properties. Experience has shown that for values of $\nu < 0.49$ the standard equations described above yield accurate values for displacements and stresses. As ν approaches 0.5, however, the elasticity matrix Equation (2.40) becomes ill conditioned.

As shown by Herrman (25) an alternative formulation can be arrived at using the calculus of variations. In addition to treating the set of nodal displacements of the finite element model as the dependent variables, the values of a function of stress, H, at the centroids of each element

are also included in the set of unknowns. This function is defined in terms of the mean pressure σ to be

$$H = \frac{3\sigma}{2\mu\ (1+\nu)} \tag{2.47}$$

where ν is Poisson's ratio, and μ is the shear modulus of the material.

Herrman, using the calculus of variations, showed that upon including the above function the following equation (neglecting thermal strains) results:

$$\delta \sum_{n=1}^{N} \int_{B_n} \iint \{\mu\ (\theta_1^2 - 2\theta_2 + 2\nu H\theta_1 - \nu(1-2\nu)H^2 - F_i u_i\}\ dv$$
$$- \int_{S1_n} \int P_i u_i dS \quad = 0 \tag{2.48}$$

where θ_1 and θ_2 are the first two invariants of the strain tensor.

B_n is the volume of the nth element, $S1_n$ is that portion of the boundary of the nth element subjected to traction type boundary conditions, P_i are the components of the tractions at the boundary, F_i are the components of distributed body forces, and u_i are components of displacements. The summation is carried out over the N finite elements comprising the modelled structure.

Assuming H is constant over a given finite element and taking the variation of Equation (2.48) with respect to H_n leads directly to

$$\int_{B_n} \mu \{2\nu \theta_1 - 2\nu (1-2\nu) H_n\} dv = 0 \qquad (2.49)$$

for each element.

If it is further assumed that ν and μ are constant throughout that element it can be concluded that H_n should be expressed as

$$H_n = \frac{\int_{B_n} \theta_1 \, dv}{(1-2\nu_n) \int_{B_n} dv} \qquad (2.50)$$

provided ν_n not equal to 0.5. Thus the variational theorem leads to the statement that the function H_n, under the assumptions made above, is calculated from the volumetric average of θ_1 over each given element, so long as $\nu \neq 0.5$.

Herrman proceeded further. Using the linear strain-displacement relations, namely,

$$\varepsilon_{ij} = (u_{i,j} + u_{j,i})/2 \qquad (2.51)$$

it is possible to replace the strain terms in Equation (2.48) with equivalent expressions using displacements.

Variations with respect to the displacement components at the nodes of the discretized structure then yields a second set of equations which in conjunction with Equation (2.50) form a complete set of equations for the solution of the stress-displacement state of the discretized structure.

This set of equations is larger than the standard set as described in a previous section of this chapter. It now includes N additional equations to handle the extra set of unknown H_n. As was done with the standard set of equations, a stiffness matrix can be developed which would include coefficients for the H_n terms. Similarly the column vector would be expanded to include terms corresponding to the added equation.

Experience has shown that the standard displacement solution remains much more accurate then the stress solution as Poisson's ratio approaches 0.5. A quick look at the elasticity matrix (Equation 2.40) explains why. Even with a relatively good displacement solution the nearly singular elasticity matrix, which results from a Poisson's ratio near 0.5, causes the stress solution to degrade much faster than the displacement solution.

The above variational principle predicts that given the optimum solution for displacements (as predicted by that principle) the optimum solution for H_n will be given by

Equation (2.50). Therefore if we already have a good solution for displacements using the standard set of equations it is reasonable to presume that Equation (2.50) should still be applied to calculate H_n. Once H_n has been obtained the constitutive equations can be used in the form

$$\sigma_{ij} = 2\mu\ \varepsilon_{ij} + \frac{E\nu}{1+\nu} H \quad . \tag{2.52}$$

That this presumption leads to an improved formulation is shown in Chapter V.

Equations (2.50) and (2.52) constitute the near incompressibility modification used in this study. Since the displacements (strains) were already available at the points of integration, no additional coding was required other than to perform the averaging indicated in Equation (2.50) using Gaussian qradratures. The above constitutive equation replaced the elasticity matrix in the routine that calculated stresses. The only additional storage requirement was that of the H_n terms at the element centroids.

CHAPTER III

THE LINEAR PATH FUNCTION SOLUTION ALGORITHM

In this chapter, the method used for implementing the effective creep strain equation (Equation (2.24)) in the solution of creep problems is presented. The method involves three separate phases of creep strain increment prediction. The first section of this chapter deals with the starting solution algorithm. The first time increment differs from all subsequent time increments since no information is available initially about the slope of the linear path functions. Once the solution is started however, the path function slope of a given time step can be used as a starting slope for the subsequent time step. This technique is demonstrated in Section 3.2. In many cases the starting slope is close enough that no iteration is required to improve the solution. In those cases for which iteration is required the linear path function is easily adapted for an iterative solution described in Section 3.3. Section 3.4 describes the computer code developed to implement the algorithm.

3.1 STARTING THE SOLUTION

It is assumed that initially the material is unstrained and has experienced no loading prior to time t equal to zero. At t equal to zero, a load is applied and the material deforms instantaneously to point A in Figure 3.1. This point is solved for using the finite element method assuming zero creep strain. To establish the initial slope of the linear path function a very small time increment is assumed to occur at constant stress level taking the material out to point B of Figure 3.1. The effective creep strain increment predicted during this time increment is used along with the current stress components in the Prandtl-Reuss equations to predict creep strain component increments. As described in Section 2.10, these creep strain increments are used to modify the system column matrix and a second "elasticity" solution is carried out to allow stress redistribution. This takes the material to point C. The above three operations have now established the initial slope of the path function. It should be pointed out that the size of the effective creep strain increment in Figure 3.1 for this time increment is greatly exaggerated for clarity. In practice this first time step should be very small since the accuracy of the predicted initial slope improves with decrease in time increment size (within reason of course).

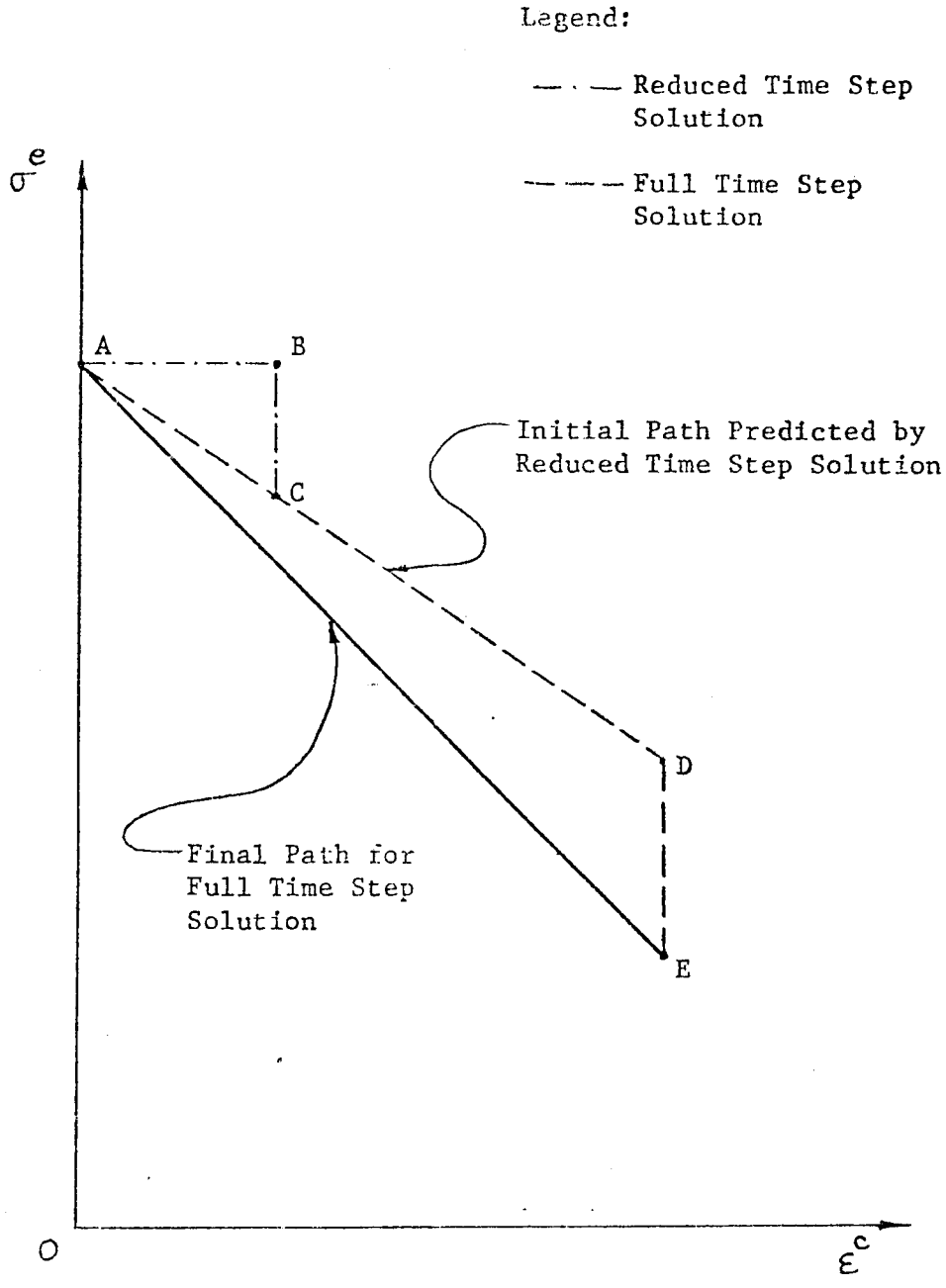


Figure 3.1. Starting the Solution

The second part of the starting solution is then to apply the slope of the path function just obtained to Equation (2.24) using the full time increment size to predict the effective creep strain increment (point D, Figure 3.1). A stress redistribution is now allowed (elasticity solution) based on this creep strain increment to complete the first solution cycle arriving at point E. Point E represents the final effective stress-effective creep strain point for the first time step unless iteration as described in Section 3.3 is required.

3.2 CONTINUING THE SOLUTION

Continuing the solution for the second and all subsequent time increments is the same. The solution at the start of the second time increment is at point E. Assume that the final path function slope of the first time increment is a good approximation to the slope of the linear path function for the second time increment. Applying the effective creep strain equation takes the solution to point F of Figure 3.2. A stress redistribution phase then completes the solution cycle and takes the material to point G. Unless iteration is required, point G is the final solution for the second time increment.

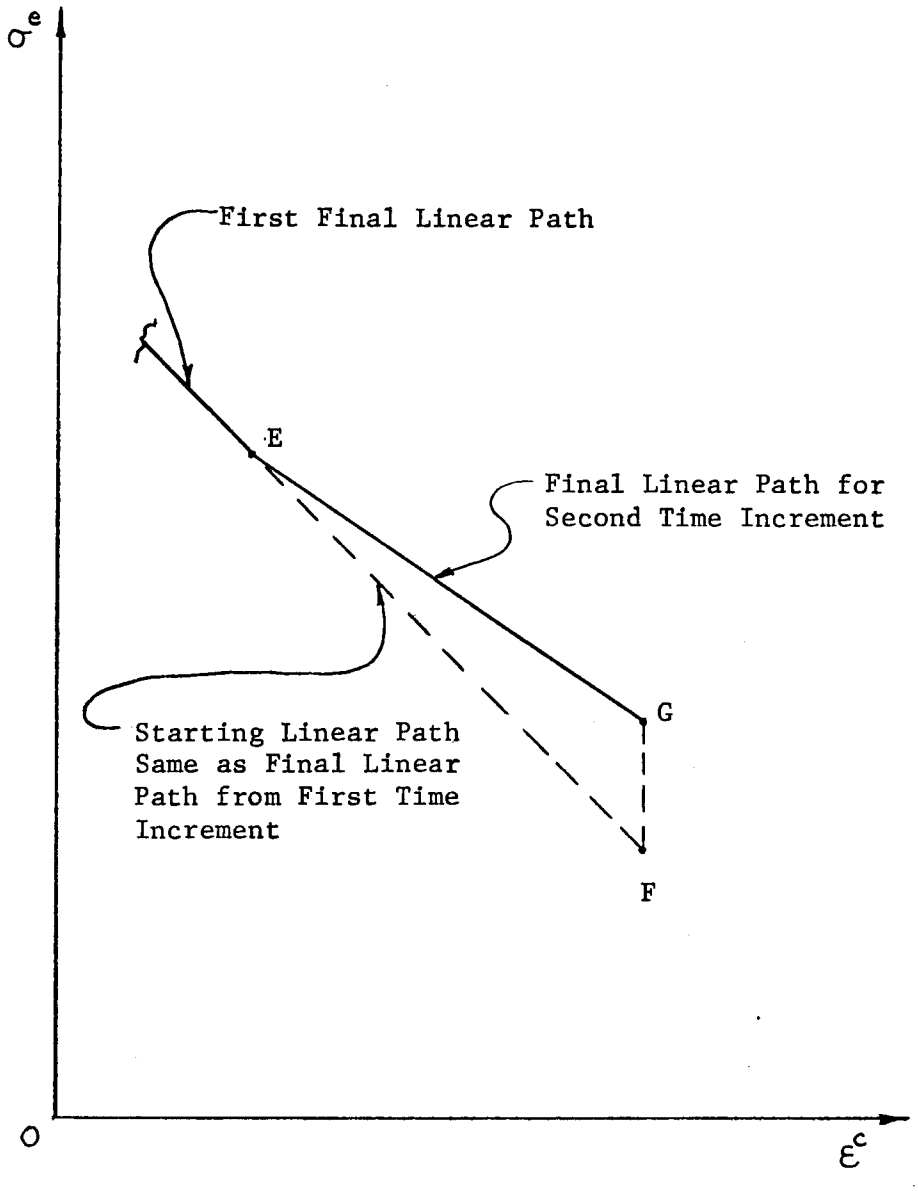


Figure 3.2. Continuing the Solution

3.3 THE ITERATION ALGORITHM

If desired the solution obtained at the completion of any solution cycle (creep strain increment followed by stress redistribution) may be improved by iteration. Referring to Figure 3.3, if EG is the solution path predicted after stress redistribution, the slope of path EG can be used in Equation (2.24) to improve the estimate of creep strain increment obtained using path EF. This predicts point H for the creep increment phase and point I for the stress redistribution phase. If point I is deemed acceptable then path EI is considered to be the final solution and the next time increment taken. If point I is not acceptable the iterative steps can be repeated until suitable convergence is achieved. A simple test on convergence can be established if desired, which allows iteration to continue until the change in effective stress on redistribution is less than some prescribed value. Alternatively or in conjunction with the above test, a set maximum number of iterations could be allowed per time step.

3.4 THE COMPUTER CODE

To implement the above concepts a computer code called CREEP was generated based on the MODEL (26) finite element program. The MODEL program is basically a building block

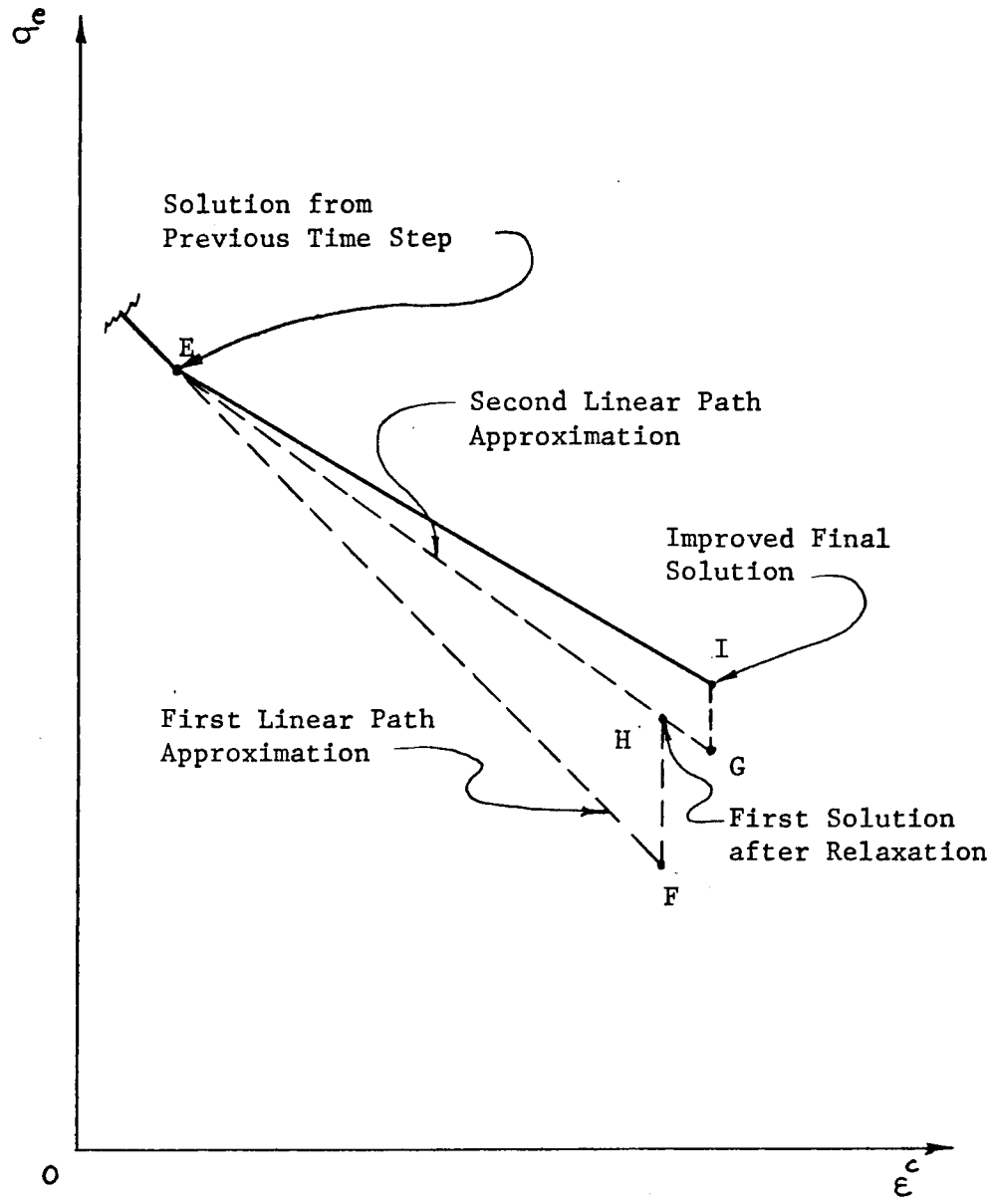


Figure 3.3. The Iteration Algorithm

program designed to allow the user to tailor the program to create a finite element program to suit his needs. For the case in point it was necessary to create four basic routines, one of which was simply a modification of the MODEL supervisor routine. In addition to the supervisor or control routine, routines were added to calculate the system stiffness matrix, the system column matrix, the element creep strain components, and the element stress components. A smaller routine was added to modify the boundary conditions consistent with the initial strain method. Listings and user guide for these routines can be found in Appendixes C and D.

A simplified flow chart of the computer code is shown in Figure 3.4. A full description of the MODEL code can be found in Reference (26).

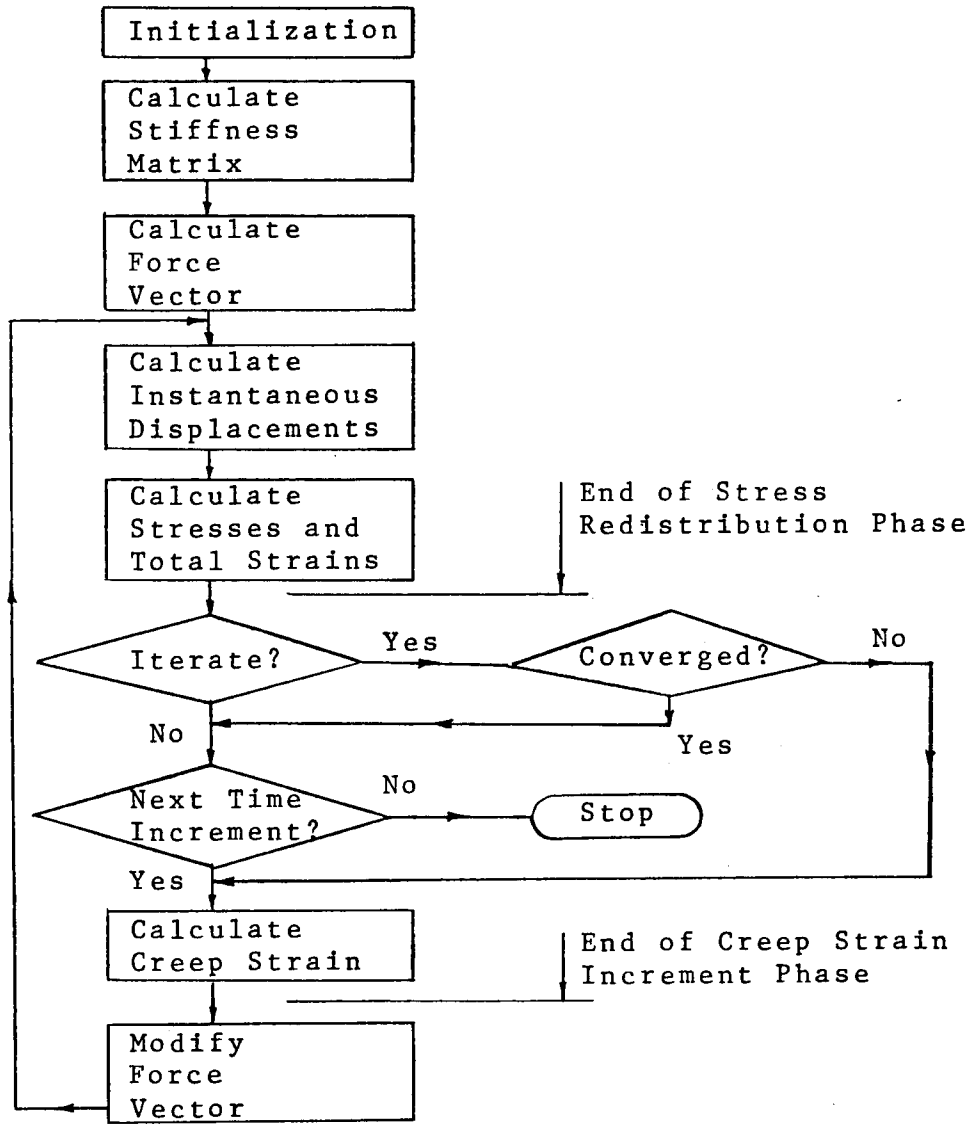


Figure 3.4. A Simplified Flow Chart of the CREEP Program

CHAPTER IV

EVALUATION OF THE LINEAR PATH METHOD

4.1 THE TEST PROBLEM

To evaluate the linear path function method the test problem of analyzing an infinitely long internally pressurized thick wall axisymmetric cylinder was chosen. A model consisting of a single layer of twelve elements was developed as shown in Figure 4.1. A zero displacement condition in the axial z direction was imposed as indicated by the roller supports in the figure. Each element was a six node isoparametric quadrilateral with the three nodes in the radial r direction to allow more accurate modeling of the expected 1/r variation of displacement in that direction. The shape functions utilized to fit the six node configuration can be found in Appendix A. Numerical integration using nine point (a three by three grid) Gaussian quadratures was used (included in Appendix B). For this simple test problem, three point quadratures in each direction is certainly excessive, at least in the axial direction, but since it is expected that later use of the computer code will require more accurate integration, it was decided to include the more accurate integration routine.

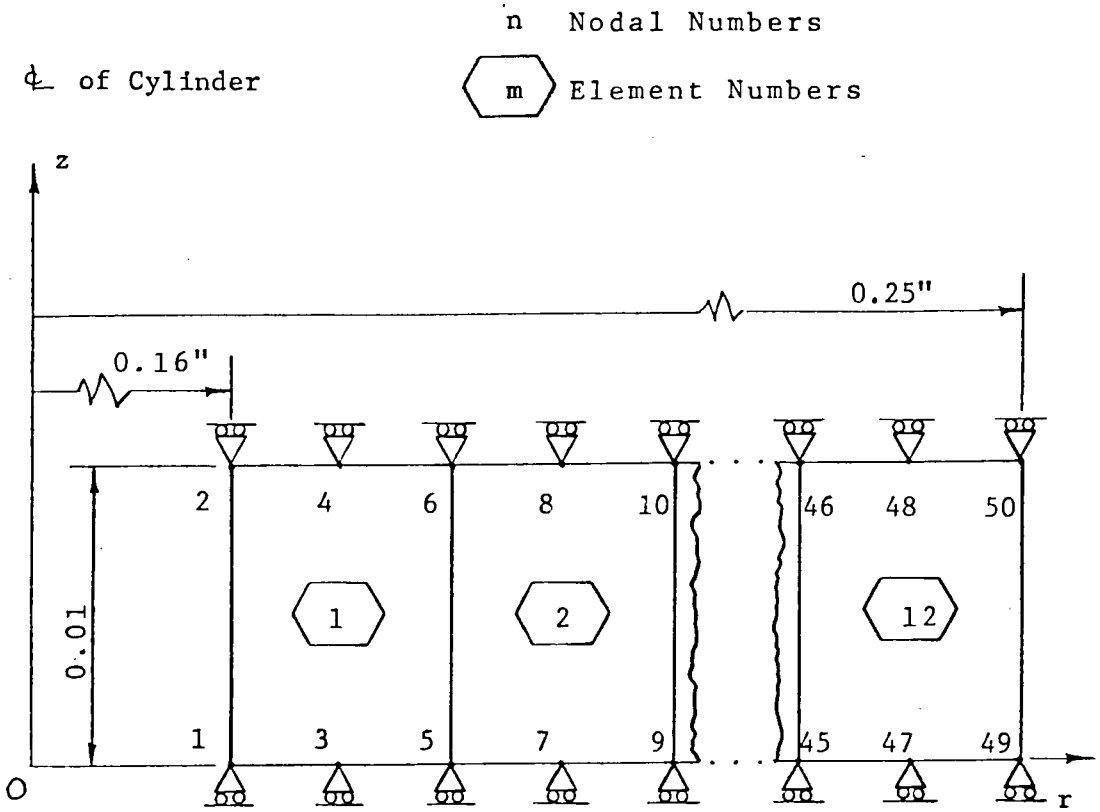


Figure 4.1. The Finite Element Demonstration Model

The material properties were chosen as indicated in Table 1.

TABLE 1

MATERIAL PROPERTIES FOR EXAMPLE PROBLEMS

Elastic Properties	
Modulus of Elasticity	$E - 20 \times 10^6$ PSI
Poisson's Ratio	$\nu - 0.499$ or 0.25
Creep Properties	
Parametric effective creep strain equation	
$\varepsilon^c = 6.4 \times 10^{-13} (\sigma^e)^{4.4} t^e$	

4.2 THE PATCH TEST

To check the correctness of the model and the computer code a patch test was performed. As discussed in references (27) and (28) such a test is adequate proof of the proper functioning of the formulation.

The basis of such a test for the initial elasticity solution is to impose on the finite element model a displacement field known to produce a constant state of strain within each element of the model. Then if the model

and finite element program are working correctly the calculated values of the strain components predicted by the model can be compared with the known values of these components. In the present case the following displacement field was imposed on the model:

$$u = C_1 \, r \tag{4.1}$$

$$w = C_2 \, r + C_3 \, z \tag{4.2}$$

so the displacement field was linear in r and z.

The strain components are readily determined from Equations (4.1) and (4.2) to be

$$\varepsilon_r = \frac{\partial u}{\partial r} = C_1$$

$$\varepsilon_\theta = \frac{u}{r} = C_1$$

$$\varepsilon_z = \frac{\partial w}{\partial z} = C_3$$

$$\varepsilon_{rz} = \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial r}\right) = C_2/2 \quad . \tag{4.3}$$

Thus the total strain tensor is constant throughout each element and from element to element. To include any "nearly incompressible" effects the Poisson's ratio of 0.499 was chosen.

Values for C_1, C_2 and C_3 were set at 0.001, 0.002, and 0.003, respectively for the test. Results indicated an agreement of the elastic finite element solution with that of Equation (4.3) to six significant digits (the number of digits printed by the code).

In addition to serving as a check of the elastic solution, the patch test also served as a check on the creep formulation when no plastic effects are present. If a set of displacements such as those prescribed by the above patch test is imposed on the nodes of a finite element, a state of relaxation results in which the true path through the $\sigma^e - \varepsilon^c$ plane is linear with constant slope equal to

$$\frac{d\sigma^e}{d\varepsilon^c} = - \frac{3E}{2(1+\nu)} \quad . \tag{4.4}$$

To show that Equation (4.4) is correct assume

$$\varepsilon_{kl} = \varepsilon^i_{kl} + \varepsilon^c_{kl} = \text{constant tensor} \tag{4.5}$$

then

$$d\varepsilon_{kl} = -d\varepsilon^c_{kl} \quad . \tag{4.6}$$

Equation (2.7) defines the deviatoric stress tensor as

$$s_{kl} = \sigma_{kl} - \sigma\delta_{kl} \quad .$$

Similarly the following three strain deviators, ε_{kl}, can be introduced for total strain, instantaneous strain, and creep strain, respectively,

$$e_{kl} = \varepsilon_{kl} - e\ \delta_{kl} \tag{4.7}$$

$$e = \frac{1}{3}\ \varepsilon_{kk} \tag{4.8}$$

$$e^{i}_{kl} = \varepsilon^{i}_{kl} - e\ \delta_{kl} \tag{4.9}$$

$$e^{i} = \frac{1}{3}\ \varepsilon^{i}_{kk} \tag{4.10}$$

$$e^{c}_{kl} = \varepsilon^{c}_{kl} \tag{4.11}$$

since $$e^{c} = \frac{1}{3}\ \varepsilon^{c}_{kk} \equiv 0 \quad . \tag{4.12}$$

In terms of Equations (4.7), (4.9), and (4.10) if it is assumed that the instantaneous response is linearly elastic then Hooke's law may be written as

$$d\varepsilon^{i}_{kl} = \frac{1+\nu}{E}\ ds_{kl} + de^{i}\ \delta_{kl} \quad . \tag{4.13}$$

Recalling the Prandtl-Reuss equations (Equation 2.6) it follows that

$$\frac{3}{2}\ \frac{d\varepsilon^{c}}{\sigma^{e}} = -\ \left(\frac{1+\nu}{E}\ ds_{kl} + de^{i}\ \delta_{kl}\right) \quad . \tag{4.14}$$

Multiplying both sides of Equation (4.14) by s_{kl} gives

$$\frac{3}{2}\frac{d\varepsilon^c}{\sigma^e} s_{kl}\, s_{kl} = -\frac{1+\nu}{E} s_{kl}\, ds_{kl} - de^i\, s_{kl}\, \delta_{kl} \quad . \tag{4.15}$$

From Equation (2.9)

$$(\sigma^e)^2 = \frac{3}{2} s_{kl}\, s_{kl} \quad .$$

Taking the derivative of this equation produces

$$s_{kl}\, ds_{kl} = \frac{2}{3}\sigma^e\, d\sigma^e \quad . \tag{4.16}$$

Since by the definition of s_{kl}

$$s_{kl}\, \delta_{kl} \equiv 0 \tag{4.17}$$

it follows that Equation (4.15) becomes

$$\frac{3}{2}\frac{d\varepsilon^c}{\sigma^e}\frac{2}{3}(\sigma^e)^2 = -\frac{1+\nu}{E}\frac{2}{3}\sigma^e\, d\sigma^e \quad . \tag{4.18}$$

Cancelling terms and solving for $\frac{d\sigma^e}{d\varepsilon^c}$ there finally results

$$\frac{d\sigma^e}{d\varepsilon^c} = -\frac{3E}{2\,(1+\nu)}$$

which is the same as Equation (4.4).

Figure 4.2 shows the results of three runs made with the test model using the linear path function formulation with time increments of 0.1 hour, 2.0 hours, and 10,000 hours respectively. The first run included twenty time increments for a total of 2.0 hours total elapsed time while runs two and three used only one time increment each. The effective stress-effective creep strain values obtained at the centroid of element number one for these runs is shown in Figure 4.2. As predicted the finite element solutions fall on a linear path with slope given by Equation (4.4). The value for the slope calculated using Equation (4.4) is 2.001E7 which agreed with the value calculated using the finite element solution. The stress-strain values at two hours for runs one and two agreed exactly.

This patch test is one clear example in which the linear path function method is superior to the constant stress method. As described in Chapter III, to get a stress solution for the patch test using linear path functions for any time t, three solution cycles would be required. One to calculate the initial stress field, a second to calculate the initial path function slope, and a third to calculate the final stress state. Since the Cormeau stability criteria limits the time increment size that can be used with the constant stress approach, if the final time t is large many

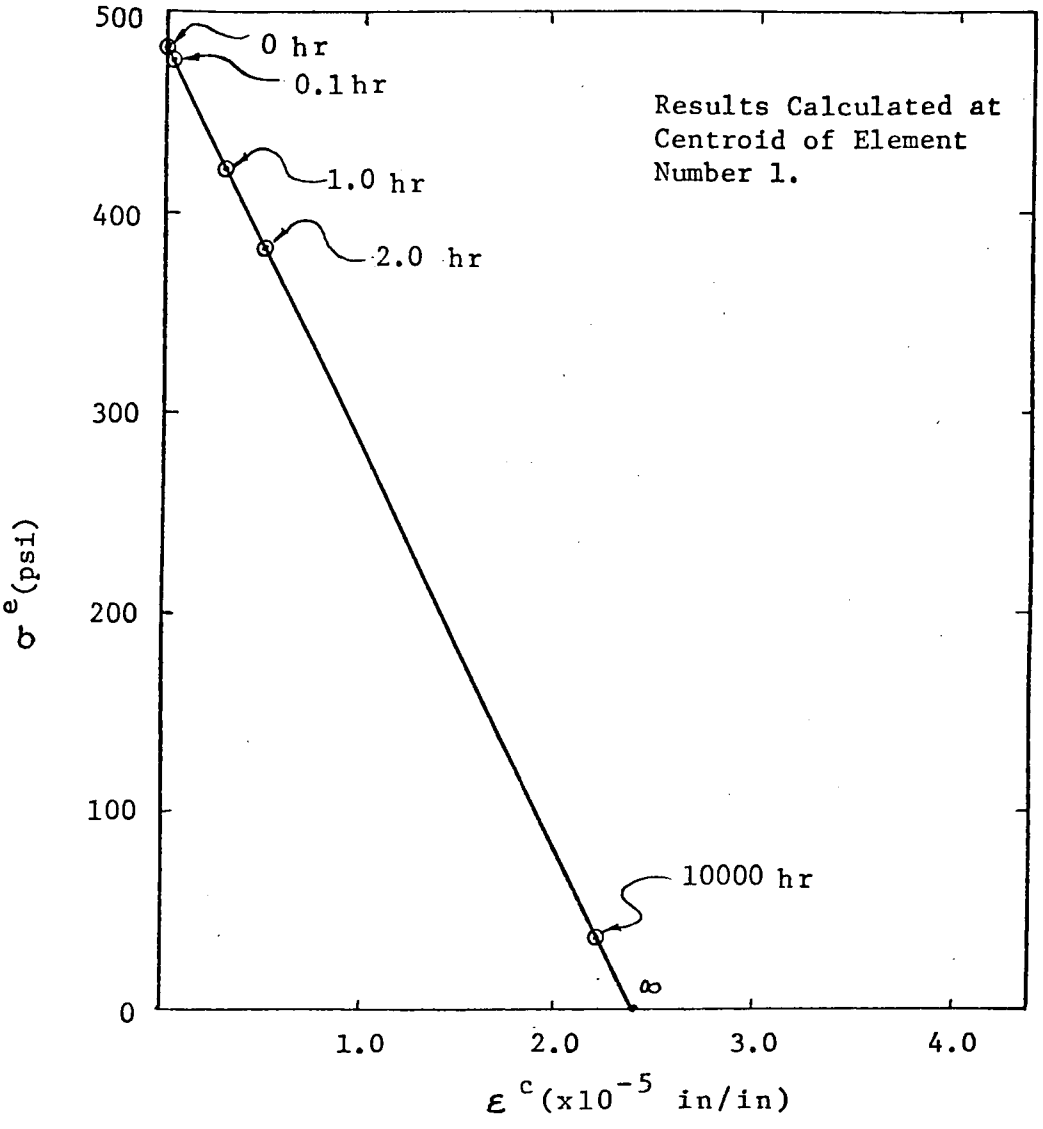


Figure 4.2. The Solution Path for the Patch Test

solution cycles would be required to obtain a solution, and this solution would still not be as accurate as the linear path function solution.

4.3 THE RELAXATION PROBLEM

While the patch test is an artificially contrived problem which is not likely to come up in a practical situation, it is nonetheless a member of a class of problems for which superior results should be anticipated using linear path functions—namely relaxation problems. Relaxation problems are defined to be that set of problems for which a displacement field is imposed on part or all of the boundary of a structure at some time t_o and then maintained at that level throughout some interval of time following t_o.

As an example of this class of problem a relaxation case was run using the thick wall cylinder model. The following displacement function was applied to the inner radius of the cylinder:

$$u\Big|_{r = 0.16} = \left\{ \begin{array}{l} 0, \; t < 0 \\ 0.74197E\text{-}5, \; t \geq 0 \end{array} \right\} \quad . \qquad (4.19)$$

Since little relaxation will occur if a Poisson's ratio is chosen near to one half, the Poisson ratio was set to 0.25.

Three runs were made at time increments of 0.1 hour, 0.2 hours, and 0.3 hours, respectively, using the linear path formulation. For purposes of comparison a run was also made with a 0.1 hour time increment using the constant stress formulation. Results of these runs are shown in Figures 4.3 through 4.7.

Figure 4.3 shows the material path in the $\sigma^e - \varepsilon^c$ plane for the centroids of elements one and twelve as predicted by the linear path and constant path formulations. The 0.1 hour and 0.3 hour linear path solutions and the 0.1 hour constant path solution are shown for comparison purposes. Qualitatively the area between the predicted material path (after the elastic stress relaxation) and the assumed linear or constant path is a measure of the accuracy of the predicted effective creep strain increment. As can be seen from the figure the 0.1 hour linear path virtually coincides with the predicted material path. The 0.3 hour linear path initially has a small discrepancy, but this rapidly dies out so that by the time five or six time increments have been taken little difference is visible in the figure between the assumed linear path and the predicted material path.

Such is not the case with the constant path approach since there is a finite difference between the limit slope of the predicted material path and the constant path, error

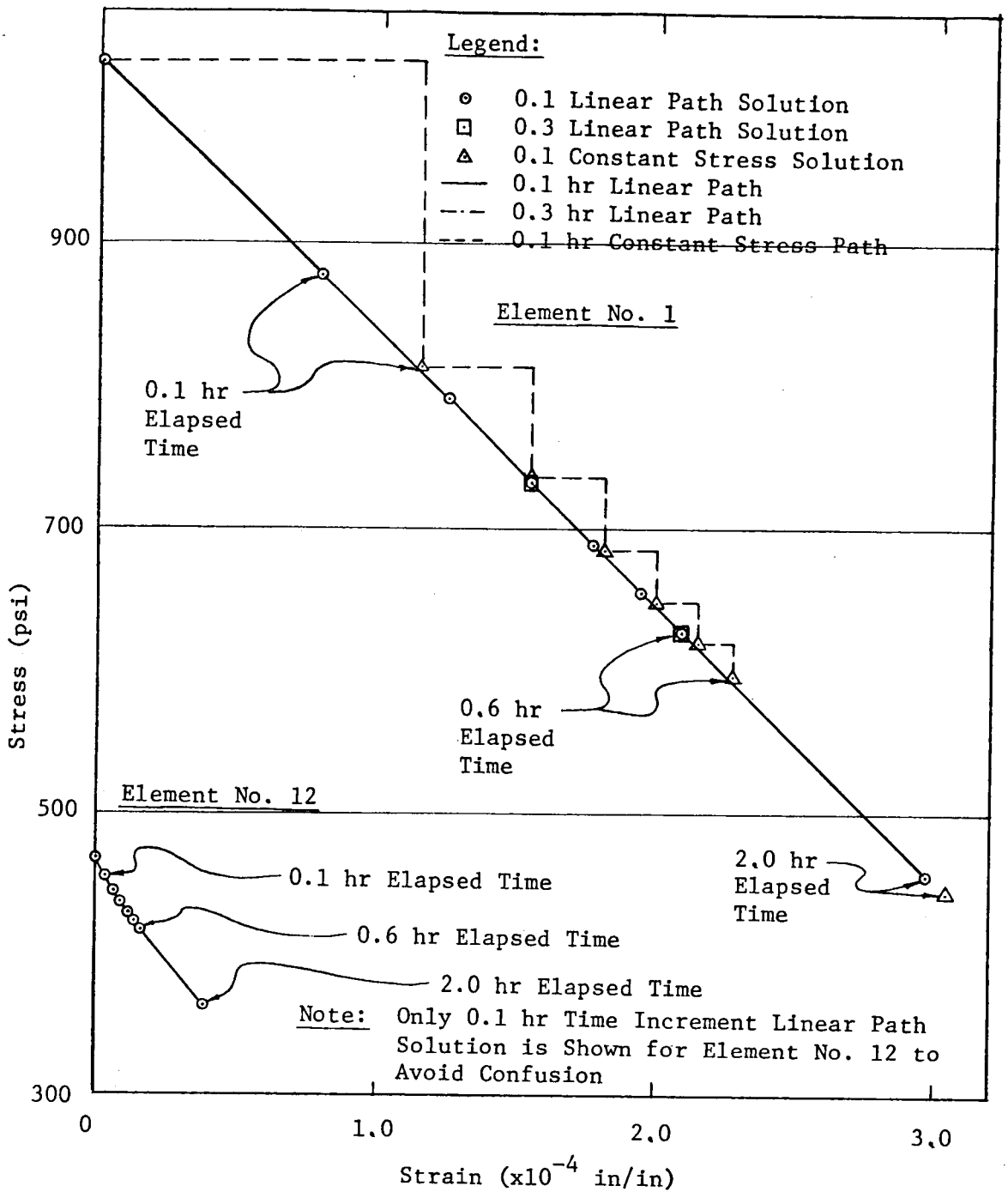


Figure 4.3. Predicted Path Functions for the Relaxation Problem

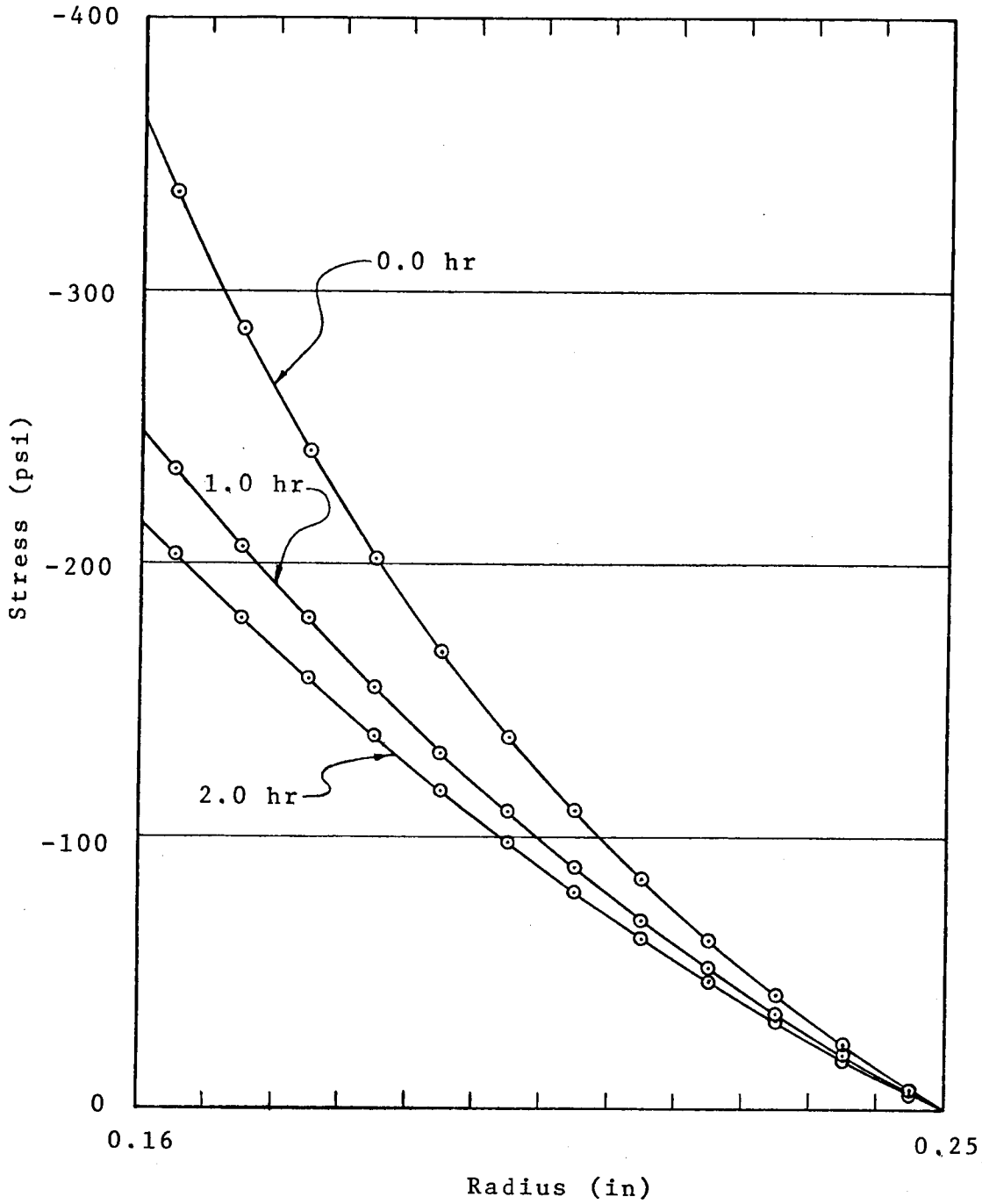


Figure 4.4 The Radial Stress Distribution for the Relaxation Case

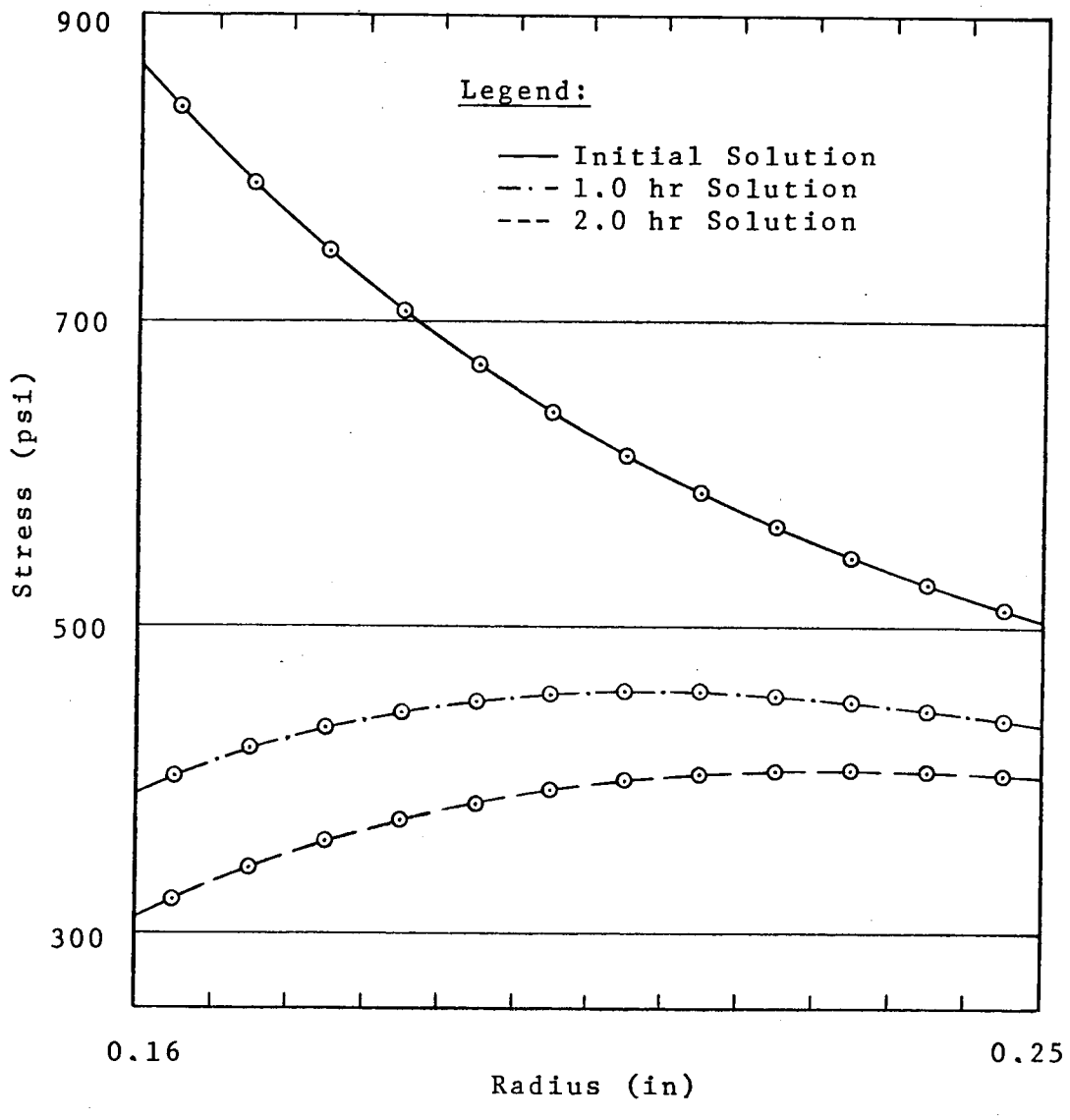


Figure 4.5. The Tangential Stress Distribution for the Relaxation Case

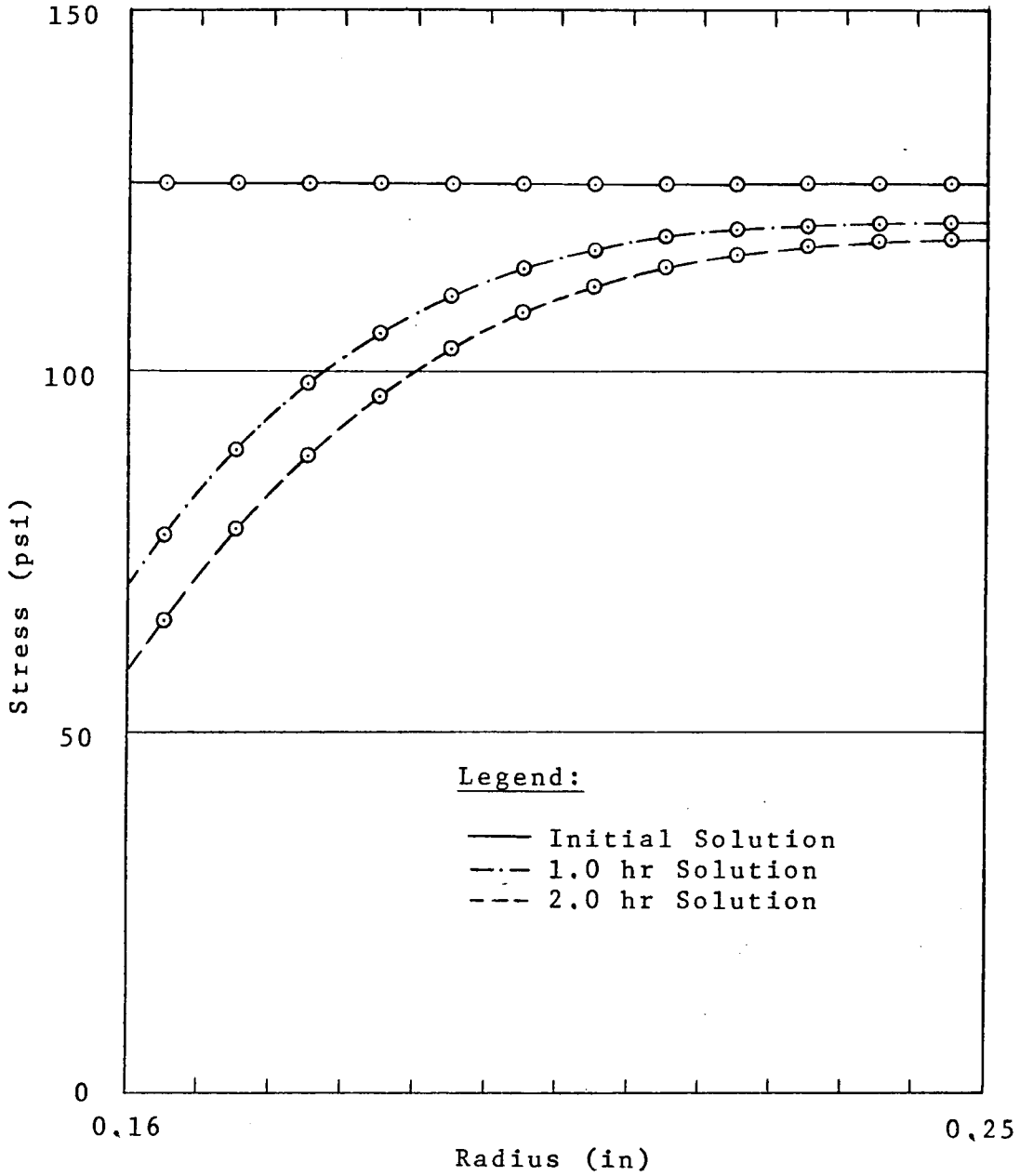


Figure 4.6. The Axial Stress Distribution for the Relaxation Case

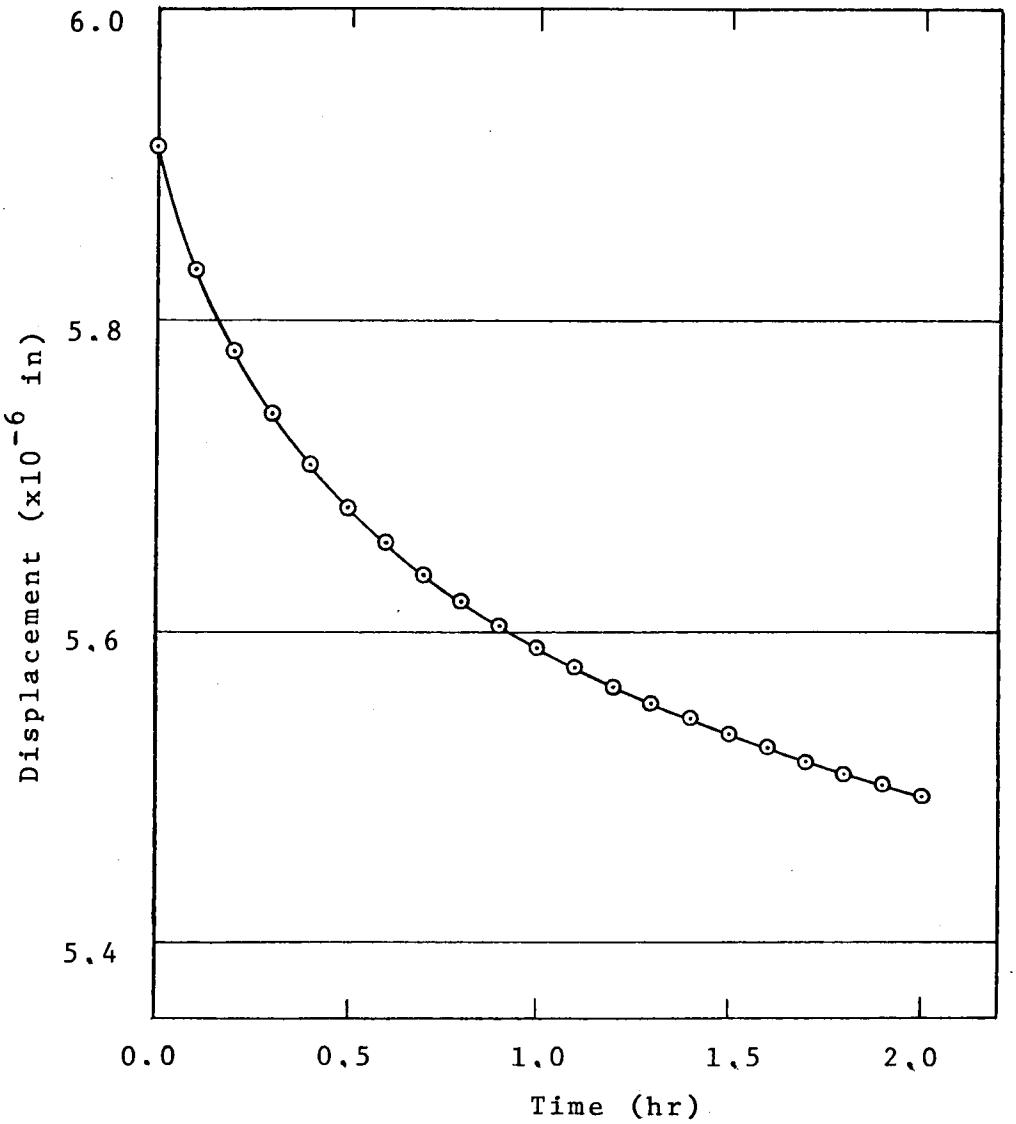


Figure 4.7. The Radial Displacement of the Outer Radius for the Relaxation Case

continually accumulates with each new time increment. As can be seen from Figure 4.3, by the time 2.0 hours have elapsed, the values for effective stress and effective creep strain predicted by the constant path method significantly differ from the "correct" value. Here it is assumed from the closenss of the 0.3 hour and 0.1 linear path solutions that the method has essentially converged to within the accuracy of the figure. The linear path 0.1 hour solution is thus considered to be the "correct" solution. The linear path solution at 2.0 hours for the 0.2 hour increment run differed very little from the 0.1 hour solution.

Figures 4.4, 4.5, and 4.6 show the radial tangential, and axial stress distributions respectively across the radius of the cylinder. Results are taken from the element centroidal values of the 0.1 hour linear path function solution at time equal to zero, one hour, and two hours. Figure 4.7 shows the displacement of the outer radius of the cylinder as a function of time as predicted by this same run.

4.4 THE CREEP PROBLEM

As a final application, the thick wall cylinder model was used with an internal pressure boundary condition

$$p\Big|_{R = 0.16 \text{ in}} = \left\{ \begin{array}{l} 0, \; t < 0 \\ 365 \text{ psi}, \; t \geq 0 \end{array} \right\} .$$

Since this is a creep problem in the classical sense of the word, i.e., the driving force is held constant, this application is the most favorable one for the constant stress method. This follows from the fact that the true path quickly settles down to a constant stress level. In order to compare results with the theoretical solution from reference (30) and a previous finite element solution from reference (1) a Poisson's ratio of 0.499 was used for this problem. Solutions were obtained using time increments of 0.3 hour, 0.2 hour, 0.1 hour, 0.05 hour, and 0.01 hour for both the constant stress method and the linear path methods. As shown in Figure 4.8, the solution for the axial stresses varies considerably as the time increments are reduced from 0.3 hour to 0.01 hour but both methods approach the same solution as the time increments are refined. Similar results are shown in Figure 4.9 for the tangential stresses. Since radial stresses do not vary much in changing from the elastic to the steady state creep stress distribution similar plots to those shown in Figure 4.8 and 4.9 do not add any meaningful information to that already shown and are not included.

A comparison of the finite element solution using the model of Figure 4.1 with theoretical solutions for the initial elastic stress distribution and the limiting steady state creep stress solution is shown in Figure 4.10-4.13.

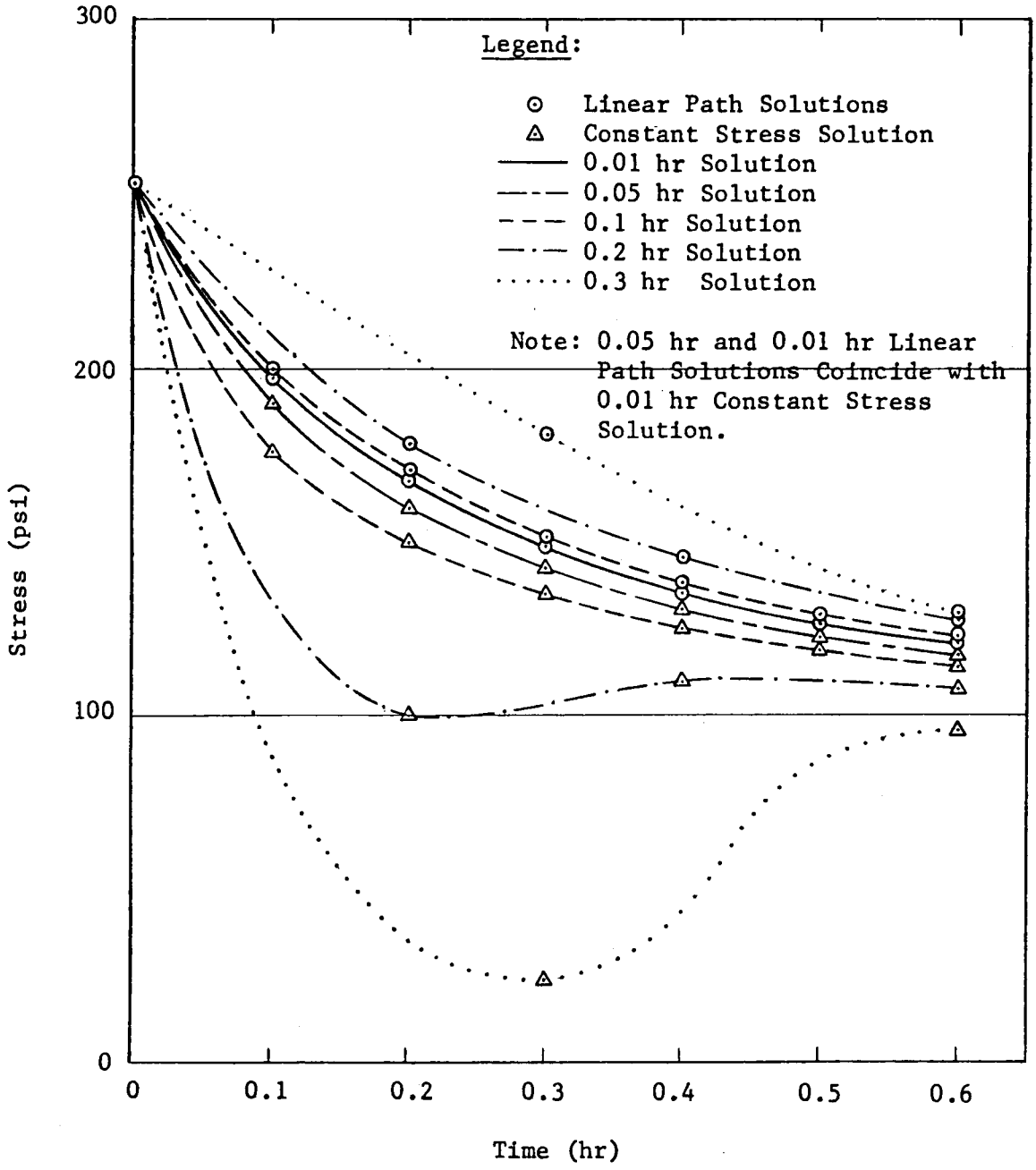


Figure 4.8. The Axial Stress in Element Number 1 for the Creep Case

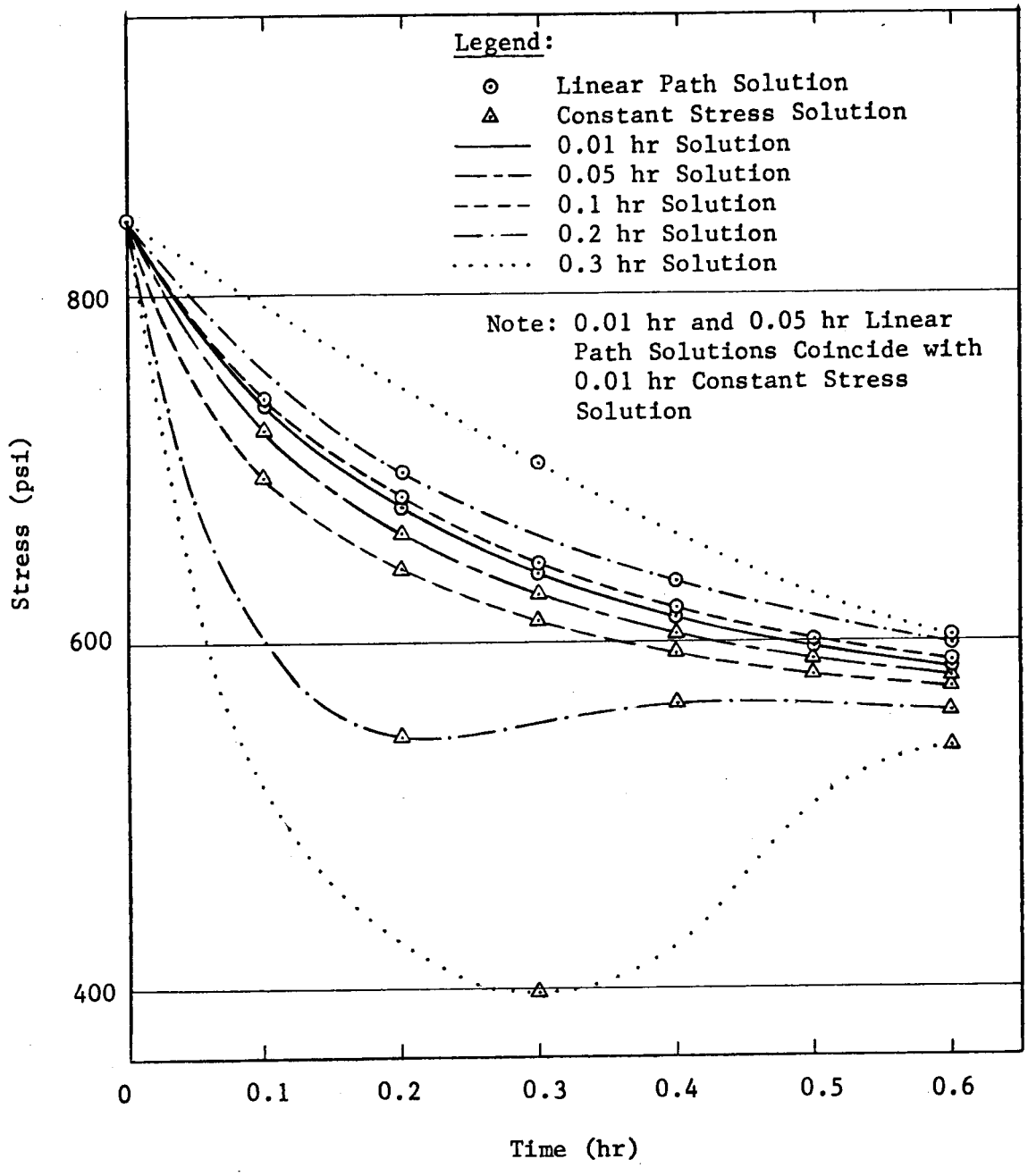


Figure 4.9. The Tangential Stress in Element Number 1 for the Creep Case

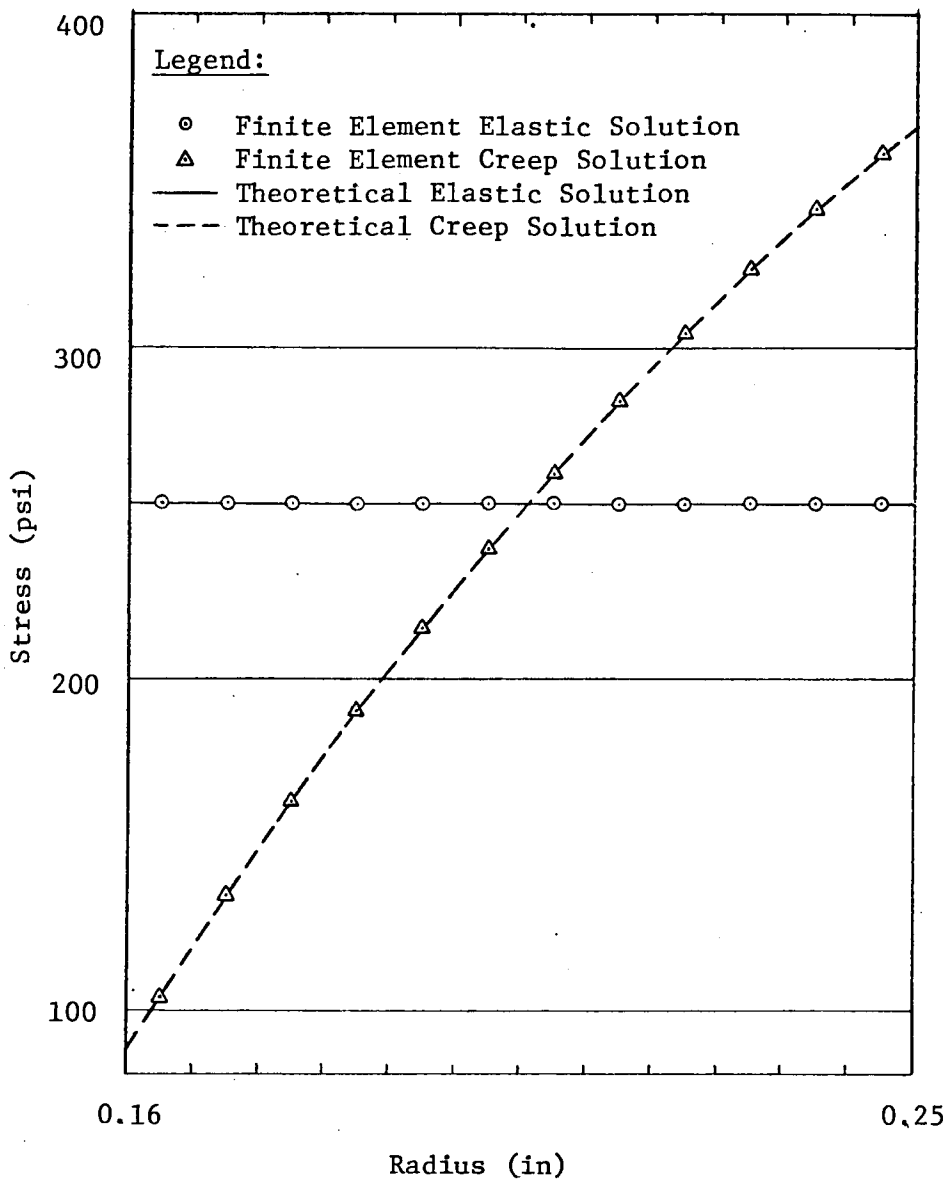


Figure 4.10. The Axial Stress Distribution for the Creep Case

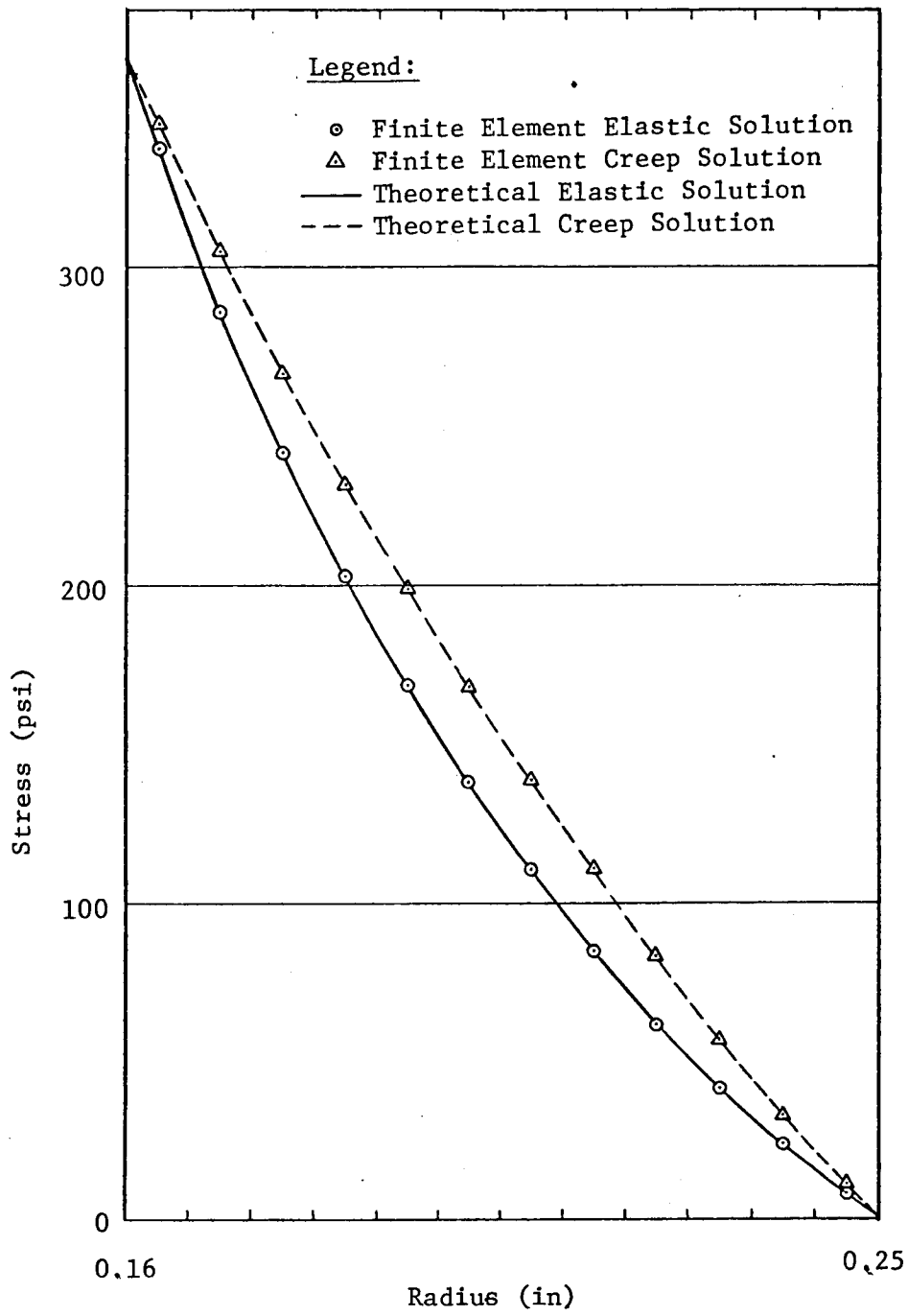


Figure 4.11. The Radial Stress Distribution for the Creep Case

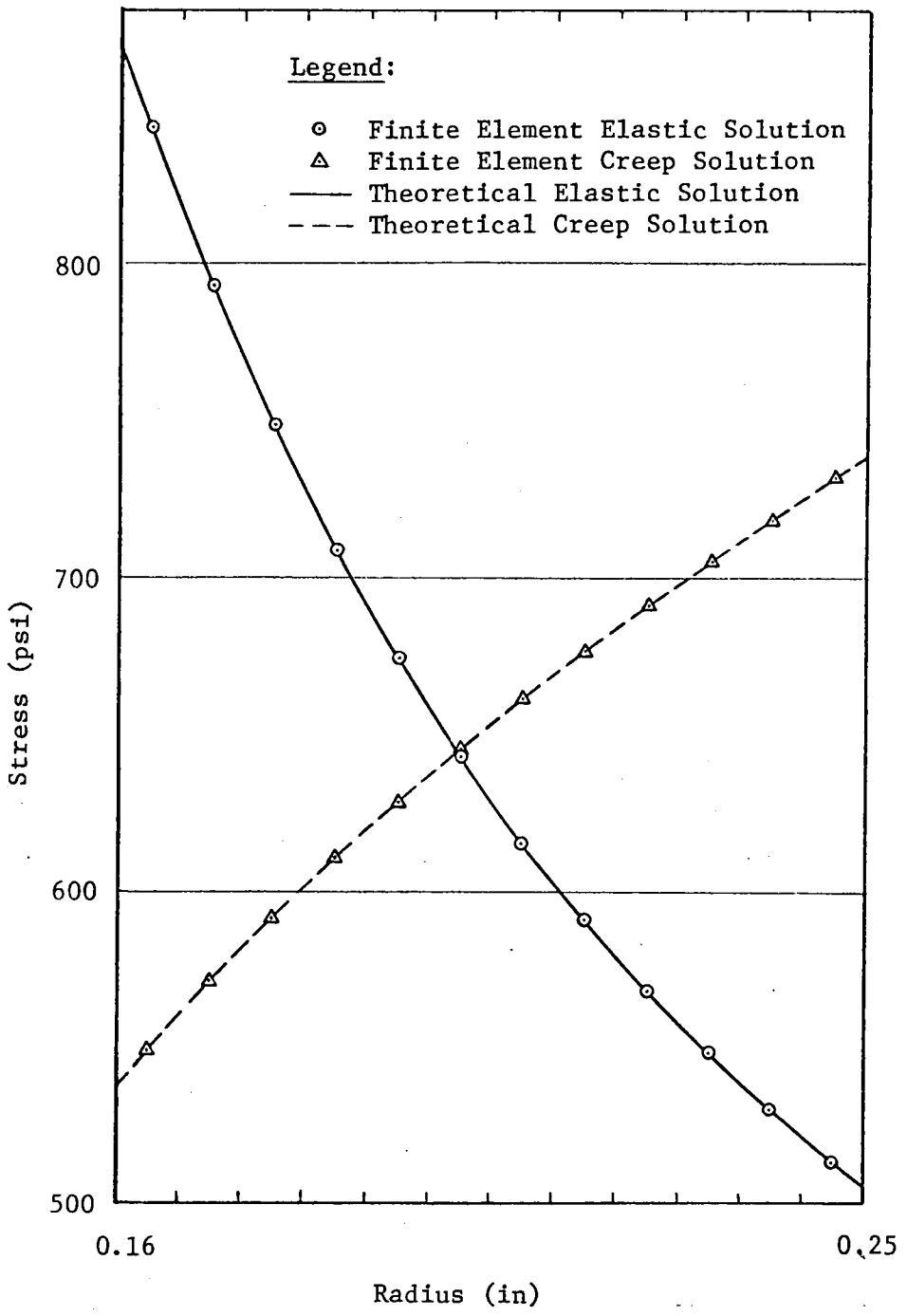


Figure 4.12. The Tangential Stress Distribution for the Creep Case

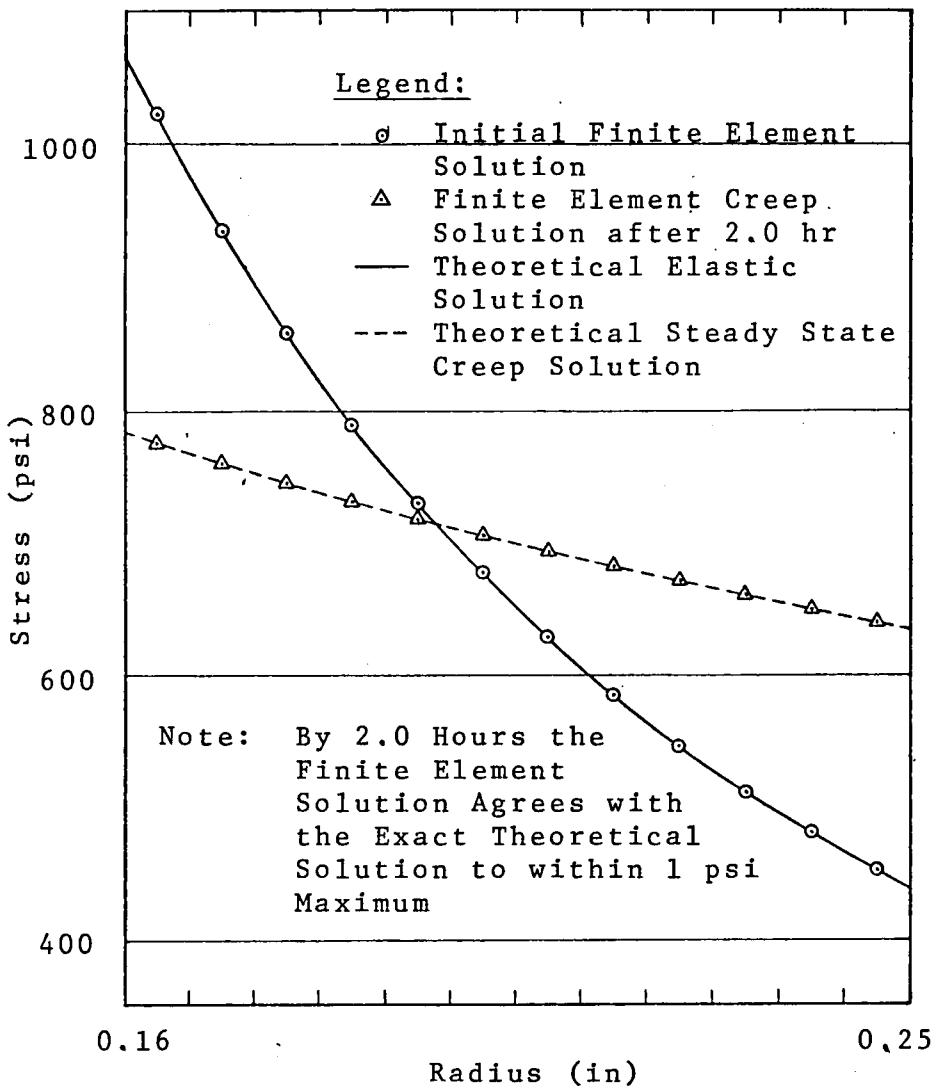


Figure 4.13. The Effective Stress Distribution for the Creep Case

The elastic solution was obtained using the standard thick wall elasticity formula (29). The steady state creep solution came from reference (30). Disagreement between the numerical solution and the theoretical solution for the initial stress state is 0.09 percent maximum. After two hours elapsed time, the creep stress solution from the finite element program differs from the theoretical steady state creep solution by a maximum of 0.6 percent. By four hours the difference had dropped to within 0.1 percent. Figure 4.14 shows the convergence of the effective stress to the steady state solution at the centroids of elements number one and twelve.

Figure 4.15 shows the displacement plots for the inner and outer radii of the cylinder for the first 0.6 hours elapsed time. The plots are essentially linear thereafter. A final slope of 4.78×10^{-6} in/in/hr was calculated for the inner radius which agrees with the theoretical steady state creep rate from reference (30).

To determine the effectiveness of the iteration routine, three runs were made at time increments of 0.3 hour, 0.1 hour and 0.01 hour, respectively. In each case iteration was allowed to continue for each new time increment until the maximum difference in effective stress from one iteration to the next was less than 1.0 psi at each and every integration

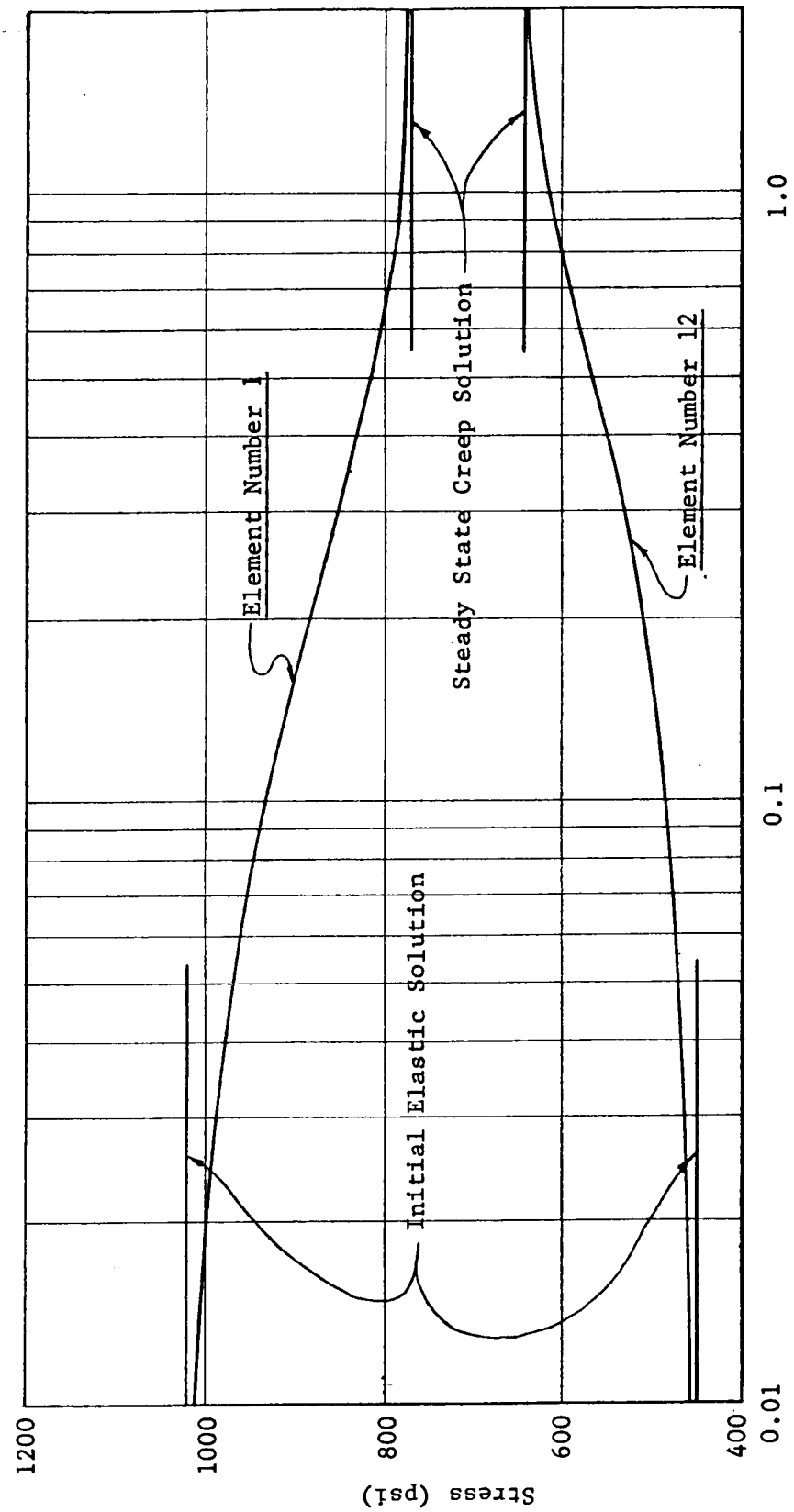


Figure 4.14. Convergence of the Effective Stress for the Creep Case

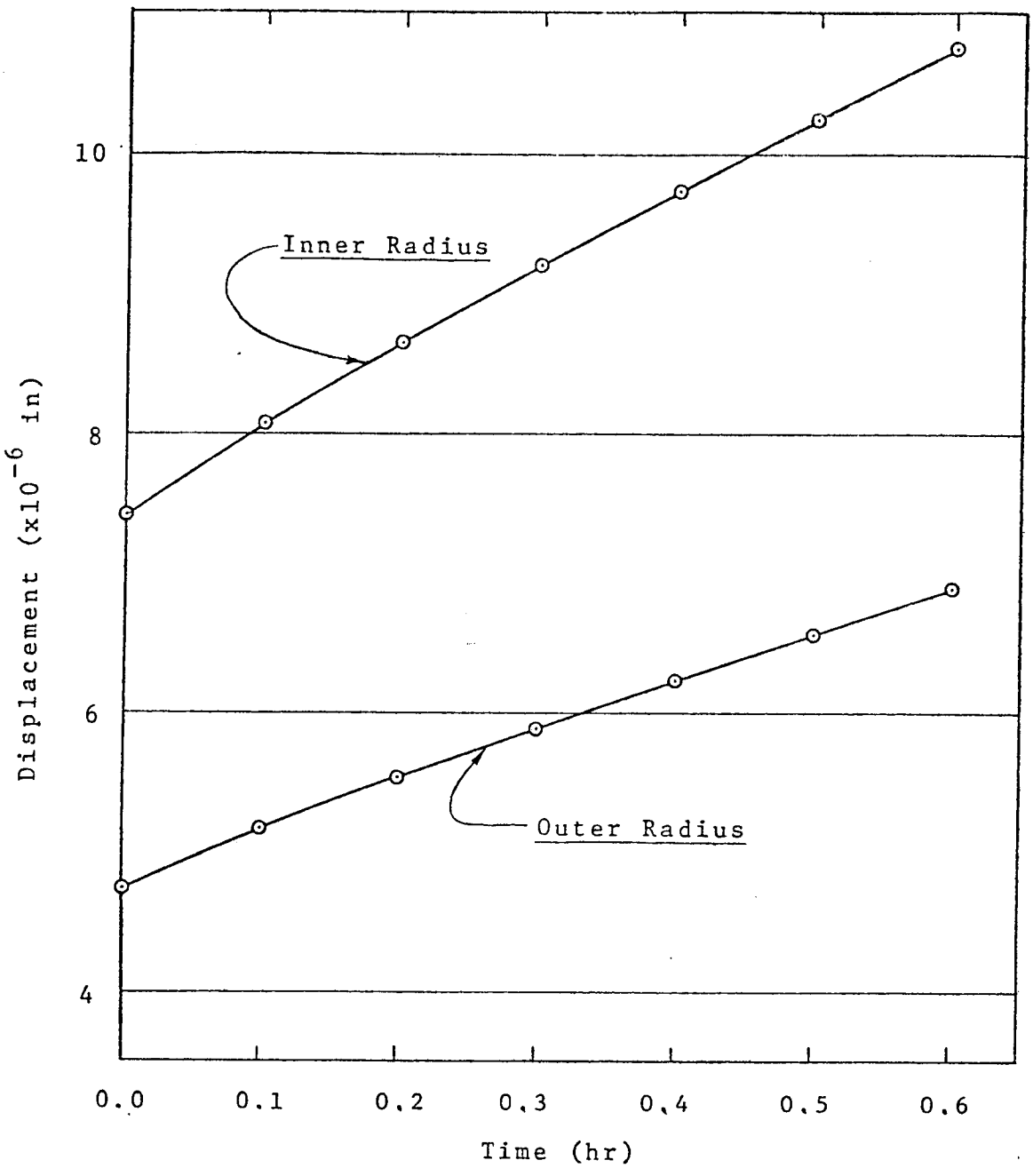


Figure 4.15. The Displacement of the Inner and Outer Cylinder Radii for the Creep Case

point. This convergence test was made after each relaxation solution step.

For the 0.3 hour time increment run, the number of iterations required to satisfy the above convergence criteria for each time increment was: 11, 10, 7, 5, 4, 3, 2, 2, 2, and 1 iteration per time increment thereafter. Thus for the total elapsed time of 2.7 hours the program completed a total of 48 iterations including the initial elasticity solution and the reduced time increment solution required to obtain the initial path function slopes. Likewise, for the 0.1 hour time increment run the number of iterations per time increment was: 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, and 1 thereafter. The total number of iterations necessary to reach the total elapsed time of 2.7 hours was 47 iterations. For the 0.01 hour run only 1 iteration per time increment was necessary, but a total of 272 iterations were required to reach the total elapsed time of 2.7 hours.

A comparison of results from the 0.1 hour and 0.3 hour time increment runs is shown in Figure 4.16 and 4.17. Not shown in these figures are the results from the 0.01 hour time increment run. The results from the latter run agreed with that of 0.1 hour run within 1.0 psi maximum on all values of calculated stresses. Thus to within the plotting accuracy of these two figures the results of the two runs

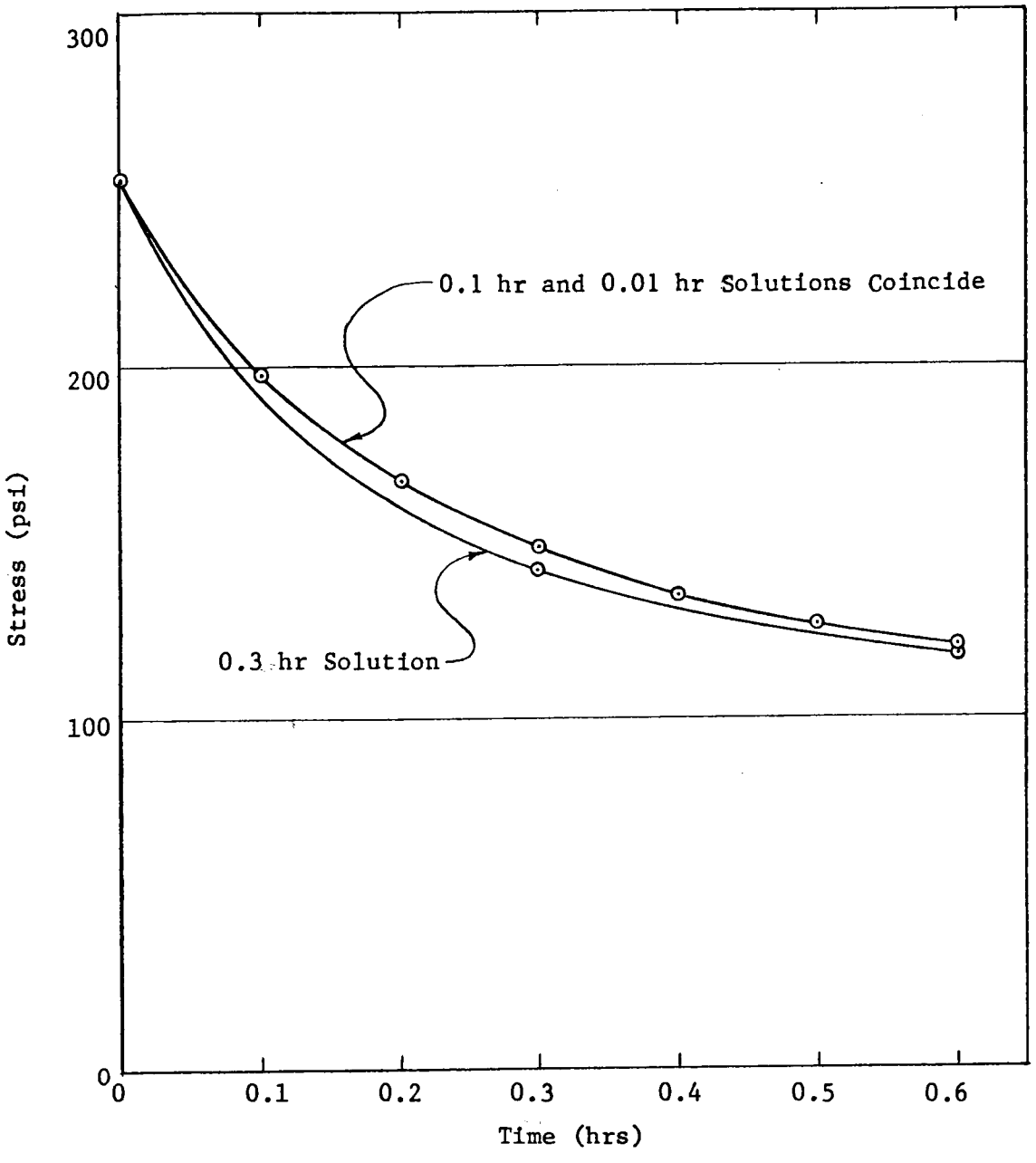


Figure 4.16. Iterated Solutions for Axial Stress

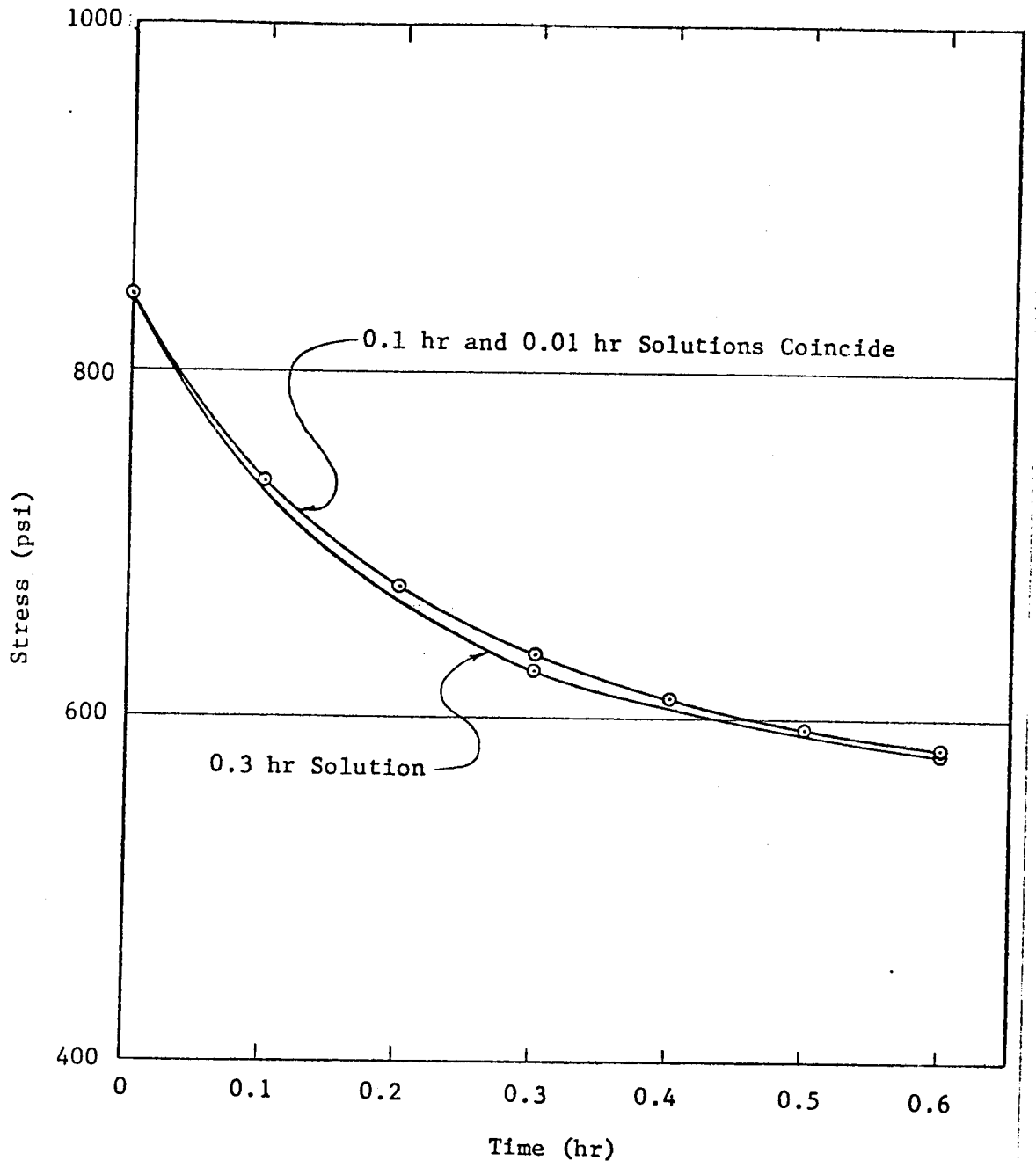


Figure 4.17. Iterated Solutions for Tangential Stress

coincided. Even for the 0.3 hour run the results were remarkably accurate.

Tentatively the above results seem to indicate that there exists an optimum time increment which minimizes the total number of iterations required to reach a given total elapsed time. Moreover if the results above are typical, this optimum time increment will yield this solution at little or no expense to the accuracy of the solution. In fact even the 0.3 hour iterated solution appears to be more accurate than the 0.05 hour constant stress solution (compare Figures 4.16, 4.17, 4.8 and 4.9).

CHAPTER V

DISCUSSION OF RESULTS AND FUTURE AREAS OF ENDEAVOR

5.1 SUMMARY OF RESULTS AND CONCLUSIONS

The results of Chapter IV are aimed at demonstrating the utility and efficiency of the linear path function approach for a few simple problems. It was shown that the patch test, used usually to demonstrate proper functioning of an elastic finite element model and program, could be extended to give meaningful information concerning the creep portion of the finite element program. In particular the correct slope of the material path through the $\sigma^e - \varepsilon^c$ plane can be predicted by Equation (4.4). Comparison can be made between this slope and that calculated using the finite element code.

The more practical relaxation problem chosen for the second example clearly shows (see Figure 4.3, page 63) the superior performance of the linear path function approach over that of the constant stress approach. As has been previously mentioned, the 0.3 hour time increment linear path function solution clearly gives a more accurate solution to this problem than does the 0.1 time increment constant stress solution. The fact that the linear path function

rapidly approaches the true material path is worth repeating. The linear path solution introduces virtually no additional error in the solution once convergence to the true path has occurred. Convergence of the constant stress method is greatly retarded by the crudeness of the path approximation.

The pure creep problem was chosen as the third problem as an additional check on the formulation since the theoretical steady state solution (29) and the previous finite element solution (30) were available. Since the previous finite element solution used constant strain triangular elements it turned out that the present constant stress solution based on the six node quadrilateral element was closer to the theoretical solution. To reduce clutter in the figures of Chapter IV, the previous finite element solutions were not shown but were reasonably close. The creep problem also was chosen since it made an excellent companion problem to the relaxation problem. If for example the boundary condition at the inner radius of the example cylinder were elastic, the resulting time dependent creep problem should be successfully bracketed by these two examples.

The results of the creep example were very revealing. As Figures 4.8 and 4.9 indicate, solutions by both approaches improve uniformly as the time increment is refined. Two things stand out in these plots. One is that the two

methods both converge toward the same solution but from opposite directions. The second is that the linear path function solution converges much more rapidly than the constant stress solution.

Since the plots are of stress data for the first element where the stresses are uniformly dropping with time, but at an ever decreasing rate, the first observation is not surprising. It simply says that the constant stress method which uses the initial high stress level to predict the effective creep strain increment overshoots the correct value. And the linear path function, since it assumes that the rate of decrease of stress is constant, predicts too rapid a drop over the time increment and thus under shoots the mark. Had attention been centered on the stress behavior of element number twelve at the outer edge of the cylinder where the effective stress is uniformly rising the reverse would be true. The linear path function method would overshoot the mark and the constant stress method would undershoot the mark. In applications where the above path conditions on stress-strain behavior are satisfied, an algorithm could be devised which would use this idea to advantage. For each time increment, the effective creep strain increment would be calculated using both the linear path function and the constant stress approaches. According to the sign of the slope of the linear path function, the

strain increment calculated by that function would be added to the upper or lower bound on effective creep strain. Whichever bound was augmented by the linear path predicted strain increment, the increment calculated from the constant stress approach would be added to the other bound.

The second observation is more pertinent to the topic at hand. Results from this example indicate that comparable accuracy can be achieved with the linear path function approach using time increments of four to five times that of the constant stress approach. Also the additional program complexity does not add substantially to the CPU time required for a typical solution cycle. Two runs, for example, were made of 200 time steps each, using the two solution approaches. The total execution time for the linear path approach was only five percent higher than that for the constant stress approach. So it is realistic to conclude that the uses of the linear path method should realize substantial CPU time savings in obtaining a solution to a given accuracy.

It is important to note that the pure creep problem presents the constant stress approach at its best. Yet even in this case, the results of the present work indicate a reduction in CPU time by a factor of four or five. In general, CPU time required for the solution of problems involving larger finite element models does not increase at a

rate which is linear with model size. The rate is substantially greater. Percentage differences, predicted by the model used here, for computation time differences between the linear and constant path approaches should be larger than that which would be obtained for larger problems. This follows from the fact that the creep strain prediction algorithm is used once per integration point (using numerical integration, see Appendix B) so CPU time required for the creep strain algorithm is roughly linear with model size. It may be concluded that for larger problems involving general loading histories that substantially better performance will be obtained using the linear path function approach.

A significant additional outcome of the present work was the technique developed for handling nearly incompressible materials using the finite element method. As discussed in references (25), (31) and (32) it has previously been necessary to make considerable modifications to the finite element code to handle such problems. In the old approach an additional unknown, the mean normal stress or its equivalent, had to be added to the set of unknowns to be solved for. In the proposed approach, only the way in which the mean normal stress is calculated is changed as discussed in Section 2.11. Results from the example problems indicate a great improvement in solution accuracy using this technique

with no increase in CPU time. Without the modification, the elastic solution using a Poisson's ratio of 0.499 was found to be in error by over 14 percent. With the modification this error was reduced to less than 0.1 percent.

The results of Chapter IV clearly indicated the potential advantages of using the iteration scheme. It is important to note that this iteration algorithm requires essentially no additional computer storage, since all the parameters required by the iteration procedure are already available in the linear path function formulation. The solutions for axial and tangential stresses (Figures 4.16 and 4.17) indicate that remarkably accurate solutions can be obtained using iteration even for relatively crude time steps. The axial stress at 0.3 hour predicted by the 0.3 hour time increment solution differed from the assumed correct 0.01 hour time increment solution by only 7.0 psi or 4.7 percent. The 0.1 hour time increment solution was still better with an error of only 0.7 percent. That this solution was attained at a reduction of the total number of iterations from that required by the 0.3 hour run is quite unusual. With the constant stress approach and for that matter the noniterated linear path function approach improved accuracy could only be had at additional expense (smaller time increment). This is an area which certainly merits further attention.

5.2 FUTURE AREAS OF CONCERN

The present work has opened up several areas of interest which require further investigation. As has already been mentioned, a method for determining upper and lower bounds for effective creep strain is available using the techniques outlined in this report. This method should be investigated further. In addition the area of higher order path functions is far from closed. Application of numerical analysis techniques to integrate and invert the creep strain equation might make the use of higher order path functions practical. Series methods or continued fraction techniques might be used to both integrate and invert that equation simultaneously.

Of more direct concern is the current restriction of the material constant m to unity in the parametric creep strain equation. The same numerical analysis techniques that are required to incorporate higher order path functions into Equation (2.20) would also allow lifting the restriction on m. It is anticipated that the development of such techniques would lead to the widespread acceptance of the path function approach.

Other techniques may be possible for lifting the restriction on material constant m. For example, it might be possible to reevaluate constants K and n in the strain

equation while holding m to unity for each time increment. A lookup table could be constructed or other numerical method developed using best fit values for K and n as a function of parametric time. This would in effect linearize the material behavior in time over a relatively small time increment. How such a technique will work remains to be seen.

Methods might also be tried to adapt the method to general hardening rules. The two hardening rules mentioned in this report, namely the time hardening and strain hardening rules, are only two of many possible rules. While in general it is true that for each new hardening rule an equation equivalent to Equation (2.20) must be derived, one simple scheme suggests itself for the combination of hardening rules. And that is to form a simple weighting function which weighs the effective creep strain increments predicted by the separate hardening rules. For example if α is a numerical parameter between zero and one, the effective creep strain increment could be defined as

$$d\varepsilon^{o} = \alpha\, d\varepsilon_{s}^{c} + (1-\alpha)\, d\varepsilon_{T}^{c}$$

where $d\varepsilon_{s}^{c}$ is the effective creep strain increment for strain hardening materials and $d\varepsilon_{T}^{c}$ is the increment for time

hardening materials. That way a material which exhibits some degree of both hardening conditions could be better approximated. Other schemes are obviously possible.

As was mentioned in the previous section, in the iteration approach there appears to exist an optimum time increment. This optimum value is probably a function of the desired solution accuracy. Certainly changing the convergence criteria would change the number of iterations required per time step. Further efforts in the area of deriving adequate guidelines for the efficient use of iteration techniques with the linear path approach should prove very rewarding. Great economies in the creep analysis of structures may possibly be had if this subject is further defined.

It remains to be shown just what stability limits are imposed by the Cormeau criteria when working with linear path functions. As was shown in Chapter II for the patch test example, essentially no time limit exists since once the initial effective stress value and path slope are calculated, a one shot solution is available. In the creep example problem, attempts to use time increments beyond 0.4 hour led to divergence of the solution using the linear path approach. As was shown in Chapter II, a stability limit of 0.2 hours exists for the constant stress approach at time equal to

zero. Thus it is probable that stability limits for the linear path function approach are slightly higher than those for the constant stress approach. Figure 4.8 and 4.9 seem to substantiate this claim.

LIST OF REFERENCES

LIST OF REFERENCES

1. Greenbaum, G. A. "Creep analysis of axisymmetric bodies." Ph.D. in Engineering, Univ. of California, Los Angeles, 1966.

2. Sutherland, W. H. "AXICRP - Finite element computer code for creep analysis of plane stress, plane strain, and axisymmetric bodies." Nuclear Engineering Design, 11, 269-85, 1970.

3. Ayres, D. J. "Elastic-plastic and creep analysis via the MARC finite element program." Proceedings of the Symposium on Numerical and Computational Methods in Structural Mechanics. Univ. of Illinois, Chicago, Ill., Sept., 1971.

4. Rashid, Y. R. "Theory report for CREEP-PLAST computer program: Analysis of two dimension problems under simultaneous creep and plasticity." GEAP - 10546, Jan., 1972.

5. Zienkiewicz, O. C. "An overview and categorization of modern computational methods in engineering." Course on Recent Advances in Finite Element Technology. Palo Alto, California, July 15-19, 1974.

6. Rabotnov, Yu. N., Creep Problems in Structural Members. John Wiley & Son, Inc., New York, 1969.

7. Conway, J. B., Numerical Methods for Creep and Rupture Analyses, Gordon and Breach, New York, 1967.

8. Smith, J. O. and Sidebottom, O. M. Inelastic Behavior of Load-Carrying Members. John Wiley & Sons, 1965.

9. Onat, E. and Fardshisheh, F. "Representation of creep of metals." ORNL - 4783, Aug., 1972.

10. Gallagher, R. H. "Computational methods in nuclear reactor structural design for high-temperatures applications: An interpretive report." ORNL - 4756, Feb., 1973.

11. Greenbaum, G. A. and Rubenstein, M. F. "Creep analysis of axisymmetric bodies using finite elements." Nuclear Engineering Design, 7, 379-97, 1968.

12. Schultz, C. and Van Fossen, D. B. "Combined elastic-plastic-creep analysis of two-dimensional bodies." ASME paper 71-PVP-30, Pressure Vessel and Piping Conference, May, 1971.

13. Chan, S. K., M. J. Manjoine, C. Visser, "Analysis of Stresses in Pressurized Welded Pipes in the Creep Range," ASME Paper 71-PVP-66, Pressure Vessel and Piping Conference, May, 1971.

14. Cormeau, I. C. "Numerical stability in quasi static elasto-viscoplasticity." Int. J. Numer. Meth. Eng. 9, 109-127, 1975.

15. Zienkiewicz, O. C., et al., "Visco-plasticity solution by finite element process." Archives of Mechanics (Poland) 24, 873-89, 1972.

16. Perzyna, P. "Fundamental problems in viscoplasticity." Advances in Applied Mechanics, 9, 243-377, 1966.

17. Andrade, E. N., da C., "The validity of the $t^{1/3}$ law of flow of metals," Phil. Mag. 7, No. 84, 1962.

18. Wang, T. T. and E. T. Onat, "Nonlinear mechanical behavior of 1100 Aluminum at 300° F," Acta. Mech. 5, 54-70, 1968.

19. Lubahn, J. D., "Deformation Phenomena," pp. 319-92 in Mechanical Behavior of Materials at Elevated Temperatures, J. E. Dorn (ed.), McGraw-Hill, 1961.

20. Taira, S., "Lifetime of structures subjected to varying load and temperature," pp. 96-124 in Creep in Structure IUTAIN Colloquium, Stanford University, Calif., 1960, Springer-Verlag, 1962.

21. Onat, E. T. and T. T. Wang, "The effect of incremental loading on creep behavior of metals," pp. 125-36 in Creep in Structures, IUTAIN Colloquium, Stanford Univ., Calif., 1960, Springer-Verlag, 1962.

22. Irons, B. and G. Treharne, "A bound theorem in eigen-values and its practical application," 3rd Conference Matrix Meth. Structural Mechanics, Wright-Patterson A. F. B., Ohio, 1971.

23. Zienkiewicz, O. C. The Finite Element Method in Engineering Science, McGraw-Hill, 1971.

24. Gallagher, R. H., Finite Element Analysis Fundamentals, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1975.

25. Herrman, L. R., "Elasticity equations for incompressible and nearly incompressible materials by a variational theorem," AIAA Journal, Vol. 3, No. 10, pp. 1896-1900, 1965.

26. Akin, J. E. and W. C. Stoddart, "Computer programs for implementing the finite element method," Report ESM 75-1, University of Tennessee, Department of Engineering Science and Mechanics, 1975.

27. Strange, W. G. and G. J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs, N.J., 1973.

28. Bazeley, G. P., Y. K. Cheung, B. M. Irons and O. C. Zienkiewicz, "Triangular elements in plate bending—conforming and noncomforming solutions," 1st Conference Matrix Meth. Structural Mechanics, Wright-Patterson A. F. B., Ohio, 1965.

29. Roark, R. J. and W. C. Young, Formulas for Stress and Strain, McGraw-Hill, New York, 1975.

30. Odqvist, Folke K. G., Mathematical Theory of Creep and Creep Rupture, Oxford University Press, Ely House, London, W. 1, 1966.

31. Becker, E. B. and J. J. Brisbane, "Application of the finite element method of stress analysis of solid propellant rocket grains," Vol. I, Report No. 5-76, Rohm and Haas Co., 1965.

32. Thompson, Erik G., "Average and complete incompressibility in the finite element method," Int. J. for Numerical Methods in Engineering, Vol. 9, 925-932, 1975.

33. Arutyunyan, N., Some Problems in the Theory of Creep, Pergamon, Oxford, 1966.

34. Argyris, J., P. C. Dunne, T. Angolflopoulos, "Finite element difficulties in nonlinear and incompressible material," Proc. Conf. Comp. Meth. in Nonlinear Mech., Austin, Texas, 1974.

35. Aziz, A. K. (Editor), The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations, Academic Press, New York and London, 1972.

36. Balmer, H., J. St. Doltsinis and M. Konig, "Elasto-plastic and creep analysis with the ASKA program system," Comp. Meth. Appl. Mech. Eng. 3, 87-104, 1974.

37. Bazant, Z. P., "Matrix differential equation and higher order numerical methods for problems of nonlinear creep, viscoelasticity and elasto-plasticity," Int. J. Numer. Meth. Eng. 4, 11-15, 1972.

38. Christensen, R. M., Theory of Viscoelasticity, Academic Press, 1971.

39. Cormeau, I. C., "Viscoplasticity and plasticity in the finite element method," Ph.D. Thesis, University College of Swansea, U.K., 1974.

40. Cozzerelli, F. A. and R. P. Shaw, "A combined strain and time hardening nonlinear creep law," Third Canadian Congress of Applied Mechanics, Calgary, Alberta, May, 1971.

41. Cyr, N. A. and R. D. Teter, "Finite element elastic plastic creep analysis of two dimensional continuum with temperature dependent material properties," National Symposium Computerized Material Properties, Computerized Structural Analysis and Design, George Washington Univ., 1972.

42. Desai, C. S., "Nonlinear analysis using spline functions," J. Soil Mechancis, Foundations Div., ASCE 97, SM 10, 1461-1480, 1971.

43. DeVeubeke, G. Fraejis, "Variational principles and the patch test," Int. J. for Numerical Meth. in Engineering, 8, 763-801, 1974.

44. Finnie, I., "Stress analysis in the presence of creep," App. Mech. Rev., 12(10), 705-12, 1960.

45. Finnie, I. and W. Heller, Creep of Engineering Materials, McGraw-Hill, 1959.

46. Gallagher, R. J., J. Padlog, P. P. Bijlaard, "Stress analysis of heated complex shapes," A.R.S.J. 32(5), 700-7, May, 1962.

47. Hoff, N. J. (ed.), Creep in Structures, Academic, New York, 1962.

48. Zefzernik, A., A. Aliyevic and J. L. Head, "Finite element matrix displacement method for time dependent stress analysis of reactor materials in two- and three-dimensions," Paper M5/8, Proc. 1st Structural Mech. in Reactor Tech. Conf., Berlin, 1971.

49. Jezernik, A. and A. Leech, "The comparison of iterative and direct solution techniques in the analysis of time-dependent stress problems, including creep, by the finite element method," pp. 449-462 of J. R. Whiteman (ed.), The Mathematics of Finite Elements and Applications, Academic Press, London, 1973.

50. Kawahara, M. and K. Kamemura, "Elastoviscoplastic finite element analysis by perturbation method," Proc. Conf. on Comp. Meth. in Nonlinear Mech., Austin, Texas, 1974.

51. Kelsey, S. J., "The application of quasilinearization to nonlinear systems," Int. J. Numer. Meth. Eng. 8, 589-611, 1974.

52. Lansing, W., W. Jensen and W. Falby, "Matrix methods for inelastic structures," Proc. of 1st Air Force Conf. on Matrix Methods in Struc. Mechanics, AFFDL-TR-66-80, pp. 605-34, 1966.

53. Lubahn, J. and R. P. Filger, Plasticity and Creep of Metals, Wiley, New York, 1961.

54. Mendelson, A., Plasticity: Theory and Application, The Macmillan Co., 1968.

55. Mendelson, A., M. H. Hischberg and S. S. Manson, "A general approach to the practical solution of creep problems," J. of Basic Engineering, Trans. ASME, Series D, 81, 585-98, 1959.

56. Marcal, P. V. and D. Hebbett, User's Manual, MARC-II Program, available from Control Data Corp., 1971.

57. Norton, F. H., Creep of Steel at High Temperatures, McGraw-Hill, New York, 1927.

58. Oden, J. T., "A generalization of the finite element concept and its applications to a class of problems in

nonlinear viscoelasticity," in D. Fredrick (ed.), Development in Theoretical and Applied Mechanics, Vol. 8, Pergamon Press, Oxford, 1970.

59. Soderberg, C. R., "Interpretation of creep tests for machine design," Trans. ASME 58, 733-43, 1936.

60. Sutherland, W. H., "Numerical instabilities associated with finite element incremental procedures for solving creep problems," ASME 1st National Congress on Pressure Vessels and Piping, May, 1971.

61. Swanson, J. A. and J. F. Patterson, "Application of finite element methods for the analysis of thermal creep, irradiation induced creep and swelling for LMFBR design," Proc. of 1st Int. Conf. on Struc. Mechanics in Reactor Technology, Berlin, 1971.

62. Treharne, G., "Applications of the finite element method to the stress analysis of materials subject to cree," Ph.D. Thesis, Univ. of Wales, Swansea, 1971.

63. Wilson, E. A., "A design oriented approach to creep and plasticity in finite element programs," ASME Paper No. 70-WA/DE-4, 1970; also in J. Eng. Ind., Vol. 93, No. 3, 1971, pp. 793-798.

64. Yagawa, G. and Y. Ando, "Three-dimensional finite element method of thermoelastoplasticity with creep effect," Paper L4/3, Proc. 2nd Structural Mech. in Reactor Tech. Conf., Berlin, 1973.

65. Zienkiewicz, O. C., M. Watson and I. P. King, "A numerical method of viscoelastic stress analysis," Int. J. of Mech. Sci. 10, 807-27, 1968.

66. Zienkiewicz, O. C. and S. Valliappan, "Analysis of real structure of creep plasticity and other complex constitutive laws," Conf. on Materials in Civ. Eng., Univ. of South Hampton, 1969, Wiley, London, 1970.

67. Zienkiewicz, O. C., "An analysis of various time-stepping schemes for initial value problems," Int. J. Earthquake Eng. Struct. Dyn. 1, 407-408, 1973.

68. Zienkiewicz, O. C. and I. C. Cormeau, "Viscoplasticity, plasticity and creep in elastic solids. A unified numerical solution approach," Int. J. Numer. Meth. Eng. 8, 821-845, 1974.

APPENDIXES

APPENDIX A

SHAPE FUNCTIONS FOR THE SIX NODE QUADRILATERAL ELEMENT

Shown in Figure A.1 is a six node quadrilateral element. The three nodes in the r-direction allow the fitting of a unique quadratic polynomial, while the two points in the z-direction require a linear polynomial for a unique fit.

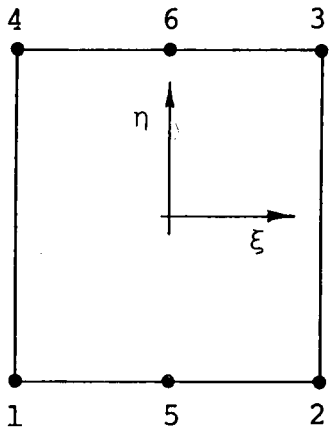


Figure A.1. The Six Node Quadrilateral Element

The appropriate shape functions for this element are as follows:

For the corner nodes

$$N_1 = \frac{1}{4} (1-\eta) (-\xi + \xi^2)$$

$$N_2 = \frac{1}{4} (1-\eta) (\xi + \xi^2)$$

$$N_3 = \frac{1}{4} (1+\eta) (\xi + \xi^2)$$

$$N_4 = \frac{1}{4} (1+\eta) (-\xi + \xi^2)$$

For the midside nodes

$$N_5 = \frac{1}{2} (1-\eta) (1 - \xi^2)$$

$$N_6 = \frac{1}{2} (1+\eta) (1 - \xi^2)$$

APPENDIX B

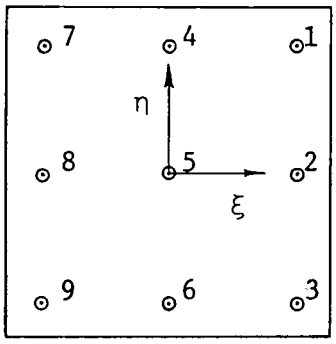
GAUSSIAN QUADRATURES

The integration routine incorporated into the finite element code used in the present study was a nine point (three by three grid) Gaussian quadrature method. The following table summarizes the coordinates and the corresponding weighting factors specified by the Gauss method.

The data assumes that the domain of integration has been normalized to ($-1 \leq \xi \leq 1$, $-1 \leq \eta \leq 1$). If such is not the case, then the integral must be scaled accordingly.

TABLE B.1

GAUSSIAN QUADRATURE COORDINATES AND WEIGHTING FACTORS



i	ξ_i	η_i	w_i
1	0.774596669241	0.774596669241	0.308644197525
2	0.774596669241	0.0	0.493827160395
3	0.774596669241	-0.774596669241	0.308644197525
4	0.0	0.774596669241	0.493827160395
5	0.0	0.0	0.790123456632
6	0.0	-0.774596669241	0.493827160395
7	-0.774596669241	0.774596669241	0.308644197525
8	-0.774596669241	0.0	0.493827160395
9	-0.774596669241	-0.774596669241	0.308644197525

APPENDIX C

A USER GUIDE FOR THE CREEP PROGRAM

The following is a summary of the input data and associated input formats required to execute the CREEP program:

Title Card (20A4) - General title information to be used as heading on output data (see next section)

Job Control Card (I5, 5X, A4)

JTYP - Sets job type Options available are:

1 - Axisymmetric
2 - Plane Strain
3 - Plane Stress

PTYP - Sets solution type. Two options are:

LINE - Solution by linear path functions
CONS - Solution by constant stress path

Creep Parameter Card (2F10.0,I5)

ERRMAX - Sets maximum stress change for convergence on iteration

DELR - Reduced time step (See section 3.1)

ITLIM - Sets maximum number of iterations per time step

Model Parameter Card* (9I5,2F10.0)

M	-	Number of nodal points
NE	-	Number of elements
NG	-	Number of unknowns per node
N	-	Number of nodes per element
NSPACE	-	Number of dimensions of space (2 for the above problem types)
NITER	-	0 (Iteration controlled by ITLIM as described above)
NCURVE	-	0 (Not used)
ISOLVT	-	0 (Not used)
NTS	-	Number of time steps to be run
DELT	-	Time increment
TIMEOUT	-	Elapsed time interval at which output data is printed

Input/Output Control Card** (10I5)

NNPFIX	-	0	The input and output for the CREEP program are currently set to override the MODEL program input/ output options. To ensure correct operation the I/O parameters on this card should be set as indicated.
NNPFLO	-	0	
NLPFIX	-	0	
NLPFLO	-	0	
MISCFX	-	0	
MISCFL	-	2	

*This card is modified from the MODEL code. See reference (26).

**This card is used as is in the MODEL code. See reference (26).

NHOMO - 1

LHOMO - 1

NPTWRT - 1

LEMWRT - 1

Nodal Point Cards** (2I5,2F10.0)

J - Node number

IBC(J) - Boundary Constraint Parameter
See reference (26)

Currently allowed options:

01 - Horizontal constraint only
10 - Vertical constraint only
11 - Vertical and horizontal constraint
00 - No constraint

X(J,1) - r coordinate of nodal point

X(J,2) - z coordinate of nodal point

Element Card(s)** (I5,(15I5))

J - Element number

NODES(J,N) - List of N nodes for the Jth element

Nodal Point Constraint Card(s)** (2I5,F10.0)

NODE - Node where constraint is applied

IPAR - Sets which direction is to be constrained

1 - r direction
2 - z direction

**This card is used as is in the MODEL code. See reference (26).

A - Enforced displacement in the specified direction

Elastic Material Property Card (2F10.0)

E - Modulus of elasticity

ν - Poisson's ratio

External Loads Card(s) (I5,2F10.0)

J - Node number

FR - Radial load component

FZ - Axial load component

This set of cards must be followed by one blank card. If no loads are applied, the blank card must still be present.

Included below is a sample input deck for running the CREEP program.

```
 29     1    0.21250    0.0
 31     1    0.21625    0.0
 33     1    0.22000    0.0
 35     1    0.22375    0.0
 37     1    0.22750    0.0
 39     1    0.23125    0.0
 41     1    0.23500    0.0
 43     1    0.23875    0.0
 45     1    0.24250    0.0
 47     1    0.24625    0.0
 49     1    0.25000    0.0
  2    11    0.16000    0.01
  4     1    0.16375    0.01
  6     1    0.16750    0.01
  8     1    0.17125    0.01
 10     1    0.17500    0.01
 12     1    0.17875    0.01
 14     1    0.18250    0.01
 16     1    0.18625    0.01
 18     1    0.19000    0.01
 20     1    0.19375    0.01
 22     1    0.19750    0.01
 24     1    0.20125    0.01
 26     1    0.20500    0.01
 28     1    0.20875    0.01
 30     1    0.21250    0.01
 32     1    0.21625    0.01
 34     1    0.22000    0.01
 36     1    0.22375    0.01
 38     1    0.22750    0.01
 40     1    0.23125    0.01
 42     1    0.23500    0.01
 44     1    0.23875    0.01
 46     1    0.24250    0.01
 48     1    0.24625    0.01
 50     1    0.25000    0.01
  1     1     5     6     2     3     4
  2     5     9    10     6     7     8
  3     9    13    14    10    11    12
  4    13    17    18    14    15    16
  5    17    21    22    18    19    20
  6    21    25    26    22    23    24
  7    25    29    30    26    27    28
  8    29    33    34    30    31    32
  9    33    37    38    34    35    36
 10    37    41    42    38    39    40
 11    41    45    46    42    43    44
 12    45    49    50    46    47    48
  1     1 .74197E-5
  2     1 .74197E-5
  1     2  0.0
  2     2  0.0
  3     2  0.0
  4     2  0.0
  5     2  0.0
  6     2  0.0
  7     2  0.0
  8     2  0.0
  9     2  0.0
```

```
   10     2  0.0
   11     2  0.0
   12     2  0.0
   13     2  0.0
   14     2  0.0
   15     2  0.0
   16     2  0.0
   17     2  0.0
   18     2  0.0
   19     2  0.0
   20     2  0.0
   21     2  0.0
   22     2  0.0
   23     2  0.0
   24     2  0.0
   25     2  0.0
   26     2  0.0
   27     2  0.0
   28     2  0.0
   29     2  0.0
   30     2  0.0
   31     2  0.0
   32     2  0.0
   33     2  0.0
   34     2  0.0
   35     2  0.0
   36     2  0.0
   37     2  0.0
   38     2  0.0
   39     2  0.0
   40     2  0.0
   41     2  0.0
   42     2  0.0
   43     2  0.0
   44     2  0.0
   45     2  0.0
   46     2  0.0
   47     2  0.0
   48     2  0.0
   49     2  0.0
   50     2  0.0
 20000.0  0.25
*** LAST CARD OF LOAD SET MUST BE BLANK EVEN IF NO LOADS ARE INPUT ***
```

APPENDIX D

LISTING OF THE CREEP PROGRAM

This appendix contains a partial listing of the CREEP program. It includes those routines supplementary to, or modified from, the MODEL code. For a listing and an explanation of those routines contained in the MODEL code the user should consult reference (26).

```
C     ******************************************************************
C     *                                                                *
C     *                        -M-O-D-E-L-                             *
C     *          MODULAR PROGRAMS FOR FINITE ELEMENT ANALYSES          *
C     *                                                                *
C     ******************************************************************
C
C                  A SET OF BUILDING BLOCK PROGRAMS
C                                BY
C                         DR. J. E. AKIN, P.E.
C           DEPARTMENT OF ENGINEERING SCIENCE AND MECHANICS
C                     THE UNIVERSITY OF TENNESSEE
C                            VERSION  8.2
C                           COPYRIGHT 1972
C
C     ******************************************************************
C
C     THE STORAGE REQUIREMENTS OF THE MODEL CODE ARE ALLOWED
C     FOR BY SETTING THE VALUES OF KFLOAT AND KFIXED AND THE
C     CORRESPONDING VALUES IN THE FOLLOWING DIMENSION STATEMENT:
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AD(5000),ID(1000)
C     ADDITIONAL VECTOR STORAGE IS ALLOWED FOR IN SUBROUTINE RDWRT
      KFLOAT=5000
      KFIXED=1000
      CALL CONTRL(AD,ID,KFLOAT,KFIXED)
      END
```

```
      SUBROUTINE ASYMBL (M,N,NE,NG,NSPACE,NELFRE,MAXBAN,NDFREE,NITER,    ASYM0005
     1        LPTEST,LHOMO,NNPFLO,IP1,IP3,IP4,IP5,JP2,KP1,KP2,NTAPE1,    ASYM0010
     2        NODES,LNODE,INDEX,LPFIX,LPROP,X,FLTNP,FLTEL,FLTMIS,        ASYM0015
     3        ELPROP,COORD,S,C,SS,CC,D,DDOLD,PRTLPT,KP3)                 ASYM0020
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASYM0025
C        ASSEMBLE ( IN CORE ) THE ELEMENT EQUATIONS                      ASYM0030
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ASYM0035
C        GENERATE ELEMENT EQUATIONS & POST SOLTUION MATRICES,            ASYM0040
      IMPLICIT REAL*8 (A-H,O-Z)                                          ASYM0045
      DIMENSION X(M,NSPACE), FLTNP(IP1,KP1), FLTEL(IP3,KP2), FLTMIS(KP3)ASYM0050
     1          ,ELPROP(KP2), COORD(N,NSPACE), S(NELFRE,NELFRE),        ASYM0055
     2           C(NELFRE), SS(NDFREE,MAXBAN), CC(NDFREE), D(NELFRE),   ASYM0060
     3           DDOLD(NDFREE), PRTLPT(IP5,KP1), NODES(NE,N), LNODE(N),  ASYM0065
     4           INDEX(NELFRE), LPFIX(IP4,JP2), LPROP(JP2)              ASYM0070
           COMMON /CONVRG/ERRMAX,FLAGE,FLAGS,IC,IT                       ASYM0075
      DATA ICRP/1/                                                       ASYM0080
C     M = NO. OF SYSTEM NODES, NG = NO. OF PARAMETERS PER NODE           ASYM0085
C     N = NUMBER OF NODES PER ELEMENT                                    ASYM0090
C     NE = NUMBER OF ELEMENTS IN SYSTEM                                  ASYM0095
C     NSPACE = DIMENSION OF SPACE                                        ASYM0100
C     NELFRE = NUMBER OF DEGREES OF FREEDOM PER ELEMENT                  ASYM0105
C     MAXBAN = MAX. HALF BANDWIDTH OF SYSTEM EQUATIONS                   ASYM0110
C     NDFREE = TOTAL NUMBER OF SYSTEM DEGREES OF FREEDOM                 ASYM0115
C     NITER = NO. OF ITERATIONS TO BE RUN (USUALLY NITER=1 )             ASYM0120
C     IF LHOMO=1 ELEMENT PROPERTIES ARE HOMOGENEOUS                      ASYM0125
C     NNPFLO = NO.  FLOATING POINT NODAL PROPERTIES                      ASYM0130
C     NTAPE1 = UNIT ON WHICH POST SOLUTION MATRICES ARE TO BE STORED     ASYM0135
C     NODES = ELEMENT INCIDENCES OF ALL ELEMENTS IN THE SYSTEM           ASYM0140
C     LNODE = THE N ELEMENT INCIDENCES OF THE ELEMENT                    ASYM0145
C     INDEX = SYSTEM D.O.F. NUMBERS ASSOCIATED WITH THE ELEMENT          ASYM0150
C     LPFIX = SYS. ARRAY OF FIXED PT. ELEM. PROP.,LPROP = ELEM. ARRAY    ASYM0155
C     X = COORDINATES OF SYSTEM NODES                                    ASYM0160
C     FLTNP, FLTEL, FLTMIS = FLOATING POINT PROP. OF SYSTEM NODES,       ASYM0165
C        ELEMENTS, AND MISC.                                             ASYM0170
C     COORD = COORDINATES OF ELEMENT NODES                               ASYM0175
C     SS = SYS. EQ. SQ. MATRIX,  CC = SYS. EQ. COL. MATRIX               ASYM0180
C     D = ARRAY OF ELEMENT DEGREES OF FREEDOM                            ASYM0185
C     PRTLPT= FLOATING POINT PROP. ARRAY OF ELEMENT NODES                ASYM0190
      DO 20  IE = 1,NE                                                   ASYM0195
      CALL  LNODES (IE,NE,N,NODES,LNODE)                                 ASYM0200
      CALL  ELCORD (IE,M,N,NSPACE,X,COORD,LNODE)                         ASYM0205
      IF ( NNPFLO.GT.0 )  CALL  LPTPRT (N,IP1,IP5,KP1,FLTNP,
     1                                  PRTLPT,LNODE)                    ASYM0215
      CALL  INDXEL (N,NELFRE,NG,LNODE,INDEX)                             ASYM0220
      IF ( LPTEST.GT.0 )  CALL ELPRTY (IE,LHOMO,IP3,IP4,JP2,KP2,
     1                                 LPFIX,FLTEL,LPROP,ELPROP)         ASYM0230
      IF(IT.GT.1) GO TO 10                                               ASYM0235
      CALL  ELSQ (N,NELFRE,NSPACE,IP5,JP2,KP1,KP2,KP3,COORD,D,ELPROP,    ASYM0240
     1            LPROP,PRTLPT,FLTMIS,S)                                 ASYM0245
      CALL  STORSQ (NDFREE,MAXBAN,NELFRE,INDEX,S,SS)                     ASYM0250
      GO TO 20                                                           ASYM0255
   10 CONTINUE                                                           ASYM0260
      CALL ELCOL(C)                                                      ASYM0265
      CALL  STORCL (NDFREE,NELFRE,INDEX,C,CC)                            ASYM0270
   20 CONTINUE                                                           ASYM0275
      RETURN                                                             ASYM0280
      END                                                                ASYM0285
```

```
      SUBROUTINE BC2(NDFREE,CC,NRES1,NDX1,CEQ1  )                       BC2 0005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *BC2 0010
C                                                                       BC2 0015
C     THIS ROUTINE APPLIES THE CONSTANT DISPLACEMENT BOUNDARY CONDITIONS BC2 0020
C      TO THE SYSTEM EQUATIONS                                          BC2 0025
C                                                                       BC2 0030
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *BC2 0035
      IMPLICIT REAL*8(A-H,O-Z)                                          BC2 0040
      DIMENSION CC(NDFREE),NDX1(NRES1),CEQ1(NRES1)                      BC2 0045
      DO 10 I=1,NRES1                                                  BC2 0050
   10 CC(NDX1(I))=CEQ1(I)                                              BC2 0055
      RETURN                                                           BC2 0060
      END                                                              BC2 0065



      SUBROUTINE CONTRL(AD,ID,KFLOAT,KFIXED)                           CONT0005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CONT0010
C                                                                      CONT0015
C     THIS ROUTINE IS THE MAIN CONTROL ROUTINE THE CREEP PROGRAM       CONT0020
C     JOB SETUP, TIME INCREMENTING, ITERATION CONTROL, AND OUTPUT      CONT0025
C     CONTROL ARE HANDLED BY THIS ROUTINE.                             CONT0030
C     POINTERS ARE DEFINED TO ALLOCATE STORAGE OF MOST OF THE MAIN     CONT0035
C     ARRAYS USED BY THE MODEL CODE                                    CONT0040
C                                                                      CONT0045
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CONT0050
C     AD IS A VECTOR CONTAINING ALL FLOATING POINT SUBSCRIPTED VARIABLESCONT0055
C     ID IS A VECTOR CONTAINING ALL FIXED POINT SUBSCRIPTED VARIABLES  CONT0060
      IMPLICIT REAL*8(A-H,O-Z)                                         CONT0065
      DIMENSION AD(KFLOAT),ID(KFIXED)                                  CONT0070
      LOGICAL INOUT,IOKODE                                             CONT0075
      DATA IOKODE/.FALSE./                                             CONT0080
      DATA IP1,IP2,IP3,IP4,IP5/5*1/                                    CONT0085
      DIMENSION  TITLE(20), TIME(7)                                    CONT0090
      MAXTIM = 7                                                       CONT0095
      DIMENSION DD(4,4)                                                CONT0100
      COMMON /MATERL/E,V,DD                                            CONT0105
      COMMON /CONVRG/ERRMAX,FLAGE,FLAGS,IC,IT                          CONT0110
      COMMON/PLOT/NTS                                                  CONT0115
      DATA ICRP/1/                                                     CONT0120
      DATA CONS,ALIN/'CONS','LINE'/                                    CONT0125
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CONT0130
C     IF PTYPE EQUAL CONS PATH IS CONSTANT STRESS PATH                 CONT0135
C     OTHERWISE PATH IS LINEAR                                         CONT0140
C     IF  JTYP = 1, AXISYMMETRIC                                       CONT0145
C     IF  JTYP = 2, PLAIN  STRAIN                                      CONT0150
C     IF  JTYP = 3, PLANE  STRESS                                      CONT0155
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CONT0160
      CALL ZERO (KFLOAT,KFIXED,AD,ID)                                  CONT0165
      CALL  ZERO (MAXTIM,1,TIME,IDUMMY)                                CONT0170
      READ (5,5100)  TITLE                                             CONT0175
      WRITE (6,5120)  TITLE                                            CONT0180
      WRITE(1,5120) TITLE                                              CONT0185
      COMMON /JOBTYP/JTYP                                              CONT0190
      COMMON /PATH/PTYPE                                               CONT0195
      READ(5,5000) JTYP,PTYPE                                          CONT0200
 5000 FORMAT(I5,5X,A4)                                                 CONT0205
      IF(JTYP.EQ.1) WRITE(1,5010)                                      CONT0210
      IF(JTYP.EQ.2) WRITE(1,5020)                                      CONT0215
```

```
      IF(JTYP.EQ.3) WRITE(1,5030)                                       CONT0220
 5010 FORMAT('0JOB TYPE IS AXISYMMETRIC')                               CONT0225
 5020 FORMAT('0JOB TYPE IS PLANE STRAIN')                               CONT0230
 5030 FORMAT('0JOB TYPE IS PLANE STRESS')                               CONT0235
      IF(PTYPE.NE.ALIN.AND.PTYPE.NE.CONS) WRITE(1,5040)                 CONT0240
      IF(PTYPE.NE.ALIN.AND.PTYPE.NE.CONS) CALL EXIT                     CONT0245
      IF(PTYPE.EQ.ALIN) WRITE(1,5050)                                   CONT0250
      IF(PTYPE.EQ.CONS) WRITE(1,5060)                                   CONT0255
 5040 FORMAT('0PATH TYPE UNKNOWN - JOB TERMINATING')                    CONT0260
 5050 FORMAT('0PATH TYPE IS LINEAR')                                    CONT0265
 5060 FORMAT('0PATH TYPE IS CONSTANT STRESS')                           CONT0270
C     IF ITLIM = 0 , ITERATION WILL CONTINUE UNTIL ERRMAX > FLAGS.      CONT0275
      READ(5,5070) ERRMAX,DELR,ITLIM                                    CONT0280
 5070 FORMAT(2F10.0,I5)                                                 CONT0285
C     M=#NODAL PTS, NE=#ELEMENTS, NG=#PARAMETERS/NODE,                  CONT0290
C     N=#NODES/ELEMENT, NSPACE=DIMENSION OF SPACE                       CONT0295
C     NITER = NO. OF ITERATIONS ( IF ANY )                              CONT0300
C     NCURVE = NO. OF CONTOUR CURVES CALCULATED PER PARAMETER           CONT0305
C     IF ISOLVT.NE.0 INITIAL VALUES OF SOLUTION VECTOR, CC , ARE TO     CONT0310
C        BE INPUT                                                       CONT0315
      READ(5,5110) M,NE,NG,N,NSPACE,NITER,NCURVE,ISOLVT,NTS,DELT,TIMOUT CONT0320
      IF(NTS.EQ.0) GO TO 10                                             CONT0325
      IF(DELR.EQ.0.0) DELR=DELT                                         CONT0330
      WRITE(1,5260) NTS,DELT,ERRMAX,DELR                                CONT0335
      IF(ITLIM.EQ.0) WRITE(1,5270)                                      CONT0340
      IF(ITLIM.NE.0) WRITE(1,5280) ITLIM                                CONT0345
   10 CONTINUE                                                          CONT0350
C     NTS - NUMBER OF TIME STEPS TO BE RUN                              CONT0355
C    TIMOUT = TIME INTERVALS BETWEEN OUTPUT OF RESULTS                  CONT0360
C    DELT = CREEP TIME INCREMENTS                                       CONT0365
      IF(TIMOUT.EQ.0.0) TIMOUT=DELT                                     CONT0370
      TIMOUT=TIMOUT+1.0E-6                                              CONT0375
      IOT=TIMOUT/DELT                                                   CONT0380
      ACUTIM=0.0                                                        CONT0385
      NELFRE = N*NG                                                     CONT0390
      NDFREE = NG*M                                                     CONT0395
      WRITE (6,5130) M,NE,N,NG,NSPACE                                   CONT0400
      IF ( NITER.LT.1 )  NITER = 1                                      CONT0405
      IF ( NITER.GT.1 )  WRITE (6,5140)  NITER                          CONT0410
      IF ( NCURVE.GT.0 )  WRITE (6,5240)  NCURVE                        CONT0415
C     NNPFIX = NO. FIXED POINT NODAL PROPERTIES                         CONT0420
C     NNPFLO = NO.  FLOATING POINT NODAL PROPERTIES                     CONT0425
C     NLPFIX = NO. FIXED POINT ELEMENT PROPERTIES                       CONT0430
C     NLPFLO = NO. FLOATING POINT ELEMENT PROPERTIES                    CONT0435
C     MISCFX = NO. OF MISC. FIXED POINT SYSTEM PROPERTIES               CONT0440
C     MISCFL = NO. MISC. FLOATING POINT SYSTEM PROPERTIES               CONT0445
C     IF LHOMO=1 ELEMENT PROPERTIES ARE HOMOGENEOUS                     CONT0450
C     IF NHOMO=1   NODAL PROPERTIES ARE HOMOGENEOUS                     CONT0455
C     IF NPTWRT(LEMWRT)=0 OUTPUT RESULTS BY NODES(ELEMENTS)             CONT0460
      READ (5,5160) NNPFIX, NNPFLO, NLPFIX, NLPFLO,MISCFX, MISCFL,      CONT0465
     1  NHOMO, LHOMO, NPTWRT, LEMWRT                                    CONT0470
      WRITE (6,5170) NNPFIX,NNPFLO,NLPFIX,NLPFLO,MISCFX,MISCFL          CONT0475
      IF ( NHOMO.EQ.1 )  WRITE (6,5180)                                 CONT0480
      IF ( LHOMO.EQ.1 )  WRITE (6,5190)                                 CONT0485
C       SET INITIAL DIMENSIONS OF PROPERTIES ARRAYS                     CONT0490
      IPTEST = NNPFIX + NNPFLO + NLPFIX + NLPFLO + MISCFX + MISCFL      CONT0495
      LPTEST = NLPFIX + NLPFLO                                          CONT0500
      IF ( NNPFLO.GT.0 )  IP1 = M                                       CONT0505
      IF ( NNPFIX.GT.0 )  IP2 = M                                       CONT0510
      IF ( NLPFLO.GT.0 )  IP3 = NE                                      CONT0515
      IF ( NLPFIX.GT.0 )  IP4 = NE                                      CONT0520
      IF ( NNPFLO.GT.0 )  IP5 = N                                       CONT0525
```

```
      IF ( NHOMO.EQ.1 )  IP1 = 1                                        CONT0530
      IF ( NHOMO.EQ.1 )  IP2 = 1                                        CONT0535
      IF ( LHOMO.EQ.1 )  IP3 = 1                                        CONT0540
      IF ( LHOMO.EQ.1 )  IP4 = 1                                        CONT0545
      JP1 = MAXO (NNPFIX,1)                                             CONT0550
      JP2 = MAXO (NLPFIX,1)                                             CONT0555
      JP3 = MAXO (MISCFX,1)                                             CONT0560
      KP1 = MAXO (NNPFLO,1)                                             CONT0565
      KP2 = MAXO (NLPFLO,1)                                             CONT0570
      KP3 = MAXO (MISCFL,1)                                             CONT0575
C                                                                       CONT0580
C            *** CALCULATE ARRAY POINTERS ***                           CONT0585
C                                                                       CONT0590
C***  ORDER OF FIXED POINT ARRAY STORAGE (NI): 1-IBC 2-NODES 3-LNODE    CONT0595
C           4-KODES 5-NREQ 6-INDEX 7-LIST 8-NDX1 9-NDX2 10-NDX3 11-NDX4 CONT0600
C           12-NDX5 13-NDX6 14-NDX7 15-NDX8 16-NDX9 17-NPFIX 18-LPFIX   CONT0605
C           19-MISFIX 20-LPROP 21-NPROP 22-NRANGE                       CONT0610
C                                                                       CONT0615
C***  ORDER OF FLOATING POINT ARRAY STORAGE (MI): 1-X 2-CEQ1 3-CEQ2     CONT0620
C           4-CEQ3 5-CEQ4 6-CEQ5 7-CEQ6 8-CEQ7 9-CEQ8 10-CEQ9 11-FLTNP  CONT0625
C           12-FLTEL 13-FLTMIS 14-ELPROP 15-PTPROP 16-COORD 17-S 18-C   CONT0630
C           19-SS 20-CC 21-DD 22-D 23-DDOLD 24-RANGE 25-XPT             CONT0635
C                 26-PRTLPT                                             CONT0640
C                                                                       CONT0645
      N1 = 1                                                            CONT0650
      N2 = N1 + M                                                       CONT0655
      N3 = N2 + NE*N                                                    CONT0660
      N4 = N3 + N                                                       CONT0665
      N5 = N4 + NG                                                      CONT0670
      N6 = N5 + 9                                                       CONT0675
      N7 = N6 + NELFRE                                                  CONT0680
C     N8 = N7 + NDFREE      ( IF USED )                                 CONT0685
      N8 = N7                                                           CONT0690
C                                                                       CONT0695
      M1 = 1                                                            CONT0700
      M2 = M1 + M*NSPACE                                                CONT0705
C                                                                       CONT0710
      MTOTAL = M2                                                       CONT0715
      NTOTAL = N8                                                       CONT0720
      IF ( MTOTAL.GT.KFLOAT .OR. NTOTAL.GT.KFIXED )  GO TO 30           CONT0725
C                     *** READ MESH DATA ***                            CONT0730
      CALL  INPUT (M,N,NE,NG,NSPACE,AD(M1),ID(N1),ID(N2))               CONT0735
C                *** COUNT NO. OF CONSTRAINTS ***                       CONT0740
      CALL  CCOUNT (1,M,NG,ID(N5),ID(N1),ID(N4),NOTHER)                 CONT0745
      CALL  CEQDIM (1,I1,I2,I3,I4,I5,I6,I7,I8,I9,K1,K2,K3,K4,K5,        CONT0750
     1              K6,K7,K8,K9,J1,J2,J3,J4,J5,J6,J7,J8,J9,ID(N5))      CONT0755
C                                                                       CONT0760
      N9 = N8 + I1*J1                                                   CONT0765
      N10 = N9 + I2*J2                                                  CONT0770
      N11 = N10 + I3*J3                                                 CONT0775
      N12 = N11 + I4*J4                                                 CONT0780
      N13 = N12 + I5*J5                                                 CONT0785
      N14 = N13 + I6*J6                                                 CONT0790
      N15 = N14 + I7*J7                                                 CONT0795
      N16 = N15 + I8*J8                                                 CONT0800
      N17 = N16 + I9*J9                                                 CONT0805
      N18 = N17 + IP2*JP1                                               CONT0810
      N19 = N18 + IP4*JP2                                               CONT0815
      N20 = N19 + JP3                                                   CONT0820
      N21 = N20 + JP2                                                   CONT0825
      N22 = N21 + JP1                                                   CONT0830
      N23 = N22 + NG*2                                                  CONT0835
```

```
C                                                                       CONT0840
      M3 = M2 + I1*K1                                                   CONT0845
      M4 = M3 + I2*K2                                                   CONT0850
      M5 = M4 + I3*K3                                                   CONT0855
      M6 = M5 + I4*K4                                                   CONT0860
      M7 = M6 + I5*K5                                                   CONT0865
      M8 = M7 + I6*K6                                                   CONT0870
      M9 = M8 + I7*K7                                                   CONT0875
      M10 = M9 + I8*K8                                                  CONT0880
      M11 = M10 + I9*K9                                                 CONT0885
      M12 = M11 + IP1*KP1                                               CONT0890
      M13 = M12 + IP3*KP2                                               CONT0895
      M14 = M13 + KP3                                                   CONT0900
      M15 = M14 + KP2                                                   CONT0905
      M16 = M15 + KP1                                                   CONT0910
      M17 = M16 + N*NSPACE                                              CONT0915
      M18 = M17 + NELFRE*NELFRE                                         CONT0920
      M19 = M18 + NELFRE                                                CONT0925
      MTOTAL = M19                                                      CONT0930
      NTOTAL = N23                                                      CONT0935
      IF ( MTOTAL.GT.KFLOAT .OR. NTOTAL.GT.KFIXED )  GO TO 30           CONT0940
C             *** READ CONSTRAINT EQUATIONS ***                         CONT0945
      CALL   INCEQ    (NG,I1,I2,I3,I4,I5,I6,I7,I8,I9,J1,J2,J3,J4,J5,    CONT0950
     1                  J6,J7,J8,J9,K1,K2,K3,K4,K5,K6,K7,K8,K9,ID(N5),   CONT0955
     2                  AD(M2),AD(M3),AD(M4),AD(M5),AD(M6),AD(M7),       CONT0960
     3                  AD(M8),AD(M9),AD(M10),ID(N8),ID(N9),ID(N10),     CONT0965
     4                  ID(N11),ID(N12),ID(N13),ID(N14),ID(N15),ID(N16))CONT0970
C               *** READ ELASTIC MATERIAL PROPERTIES ***                CONT0975
C     FORM ELEMENT ELASTICITY MATRIX (ASSUMED CONSTANT THROUGHOUT       CONT0980
C     STRUCTURE).                                                       CONT0985
      READ(5,5080) E,V                                                  CONT0990
 5080 FORMAT((8F10.4))                                                  CONT0995
      WRITE(6,5090) E,V                                                 CONT1000
 5090 FORMAT('0 MOD. OF ELAST. =',2PE12.2/'0 POIS. RATIO =',0PF10.3)    CONT1005
C     FORM  ELASTICITY  MATRIX                                          CONT1010
      CALL ZERO(16,1,DD,IDUMMY)                                         CONT1015
      IF(JTYP.LT.3) F=E/(1.+V)*(1.-V)/(1.-2.*V)                         CONT1020
      IF(JTYP.LT.3) D=V/(1.-V)                                          CONT1025
      IF(JTYP.EQ.3) F=E/(1.-V**2)                                       CONT1030
      IF(JTYP.EQ.3) D=V                                                 CONT1035
      DD(1,1)=F                                                         CONT1040
      DD(2,2)=F                                                         CONT1045
      D(3,3)=E/(1.+V)                                                   CONT1050
      IF(JTYP.EQ.1) DD(4,4)=F                                           CONT1055
      DD(1,2)=F*D                                                       CONT1060
      DD(2,1)=F*D                                                       CONT1065
      IF(JTYP.EQ.3) GO TO 20                                            CONT1070
      DD(1,4)=F*D                                                       CONT1075
      DD(4,1)=F*D                                                       CONT1080
      DD(2,4)=F*D                                                       CONT1085
      DD(4,2)=F*D                                                       CONT1090
   20 CONTINUE                                                          CONT1095
C        DETERMINE SYSTEM HALF-BANDWIDTH                                CONT1100
      IBW = 1                                                           CONT1105
      CALL SYSBAN (NE,N,NG,IBW,ID(N2),ID(N3),LID)                       CONT1110
      JBW = 1                                                           CONT1115
      MAXBAN = MAXO ( JBW,IBW )                                         CONT1120
      WRITE (6,5200)  IBW, LID, JBW, MAXBAN, NDFREE, NELFRE             CONT1125
      NBANCO = NDFREE*MAXBAN                                            CONT1130
      M20 = M19 + NBANCO                                                CONT1135
      M21 = M20 + NDFREE                                                CONT1140
      M22 = M21 + NDFREE                                                CONT1145
```

```
      M23 = M22 + NELFRE                                                CONT1150
      M24 = M23 + 1                                                     CONT1155
      IF ( NITER.GT.1 )  M24 = M23 + NDFREE                             CONT1160
      M25 = M24 + NG*2                                                  CONT1165
      M26 = M25 + NSPACE                                                CONT1170
      IP5=1                                                             CONT1175
      M27 = M26 + IP5*KP1                                               CONT1180
C                                                                       CONT1185
      NNEXT = N23                                                       CONT1190
      MNEXT = M27                                                       CONT1195
      MTOTAL = MNEXT - 1                                                CONT1200
      NTOTAL = NNEXT - 1                                                CONT1205
C     FOR CREEP ANALYSIS THE FOLLOWING ADDITIONAL POINTERS ARE ADDED:   CONT1210
C     M27 - AUXILIARY STORAGE OF EXTERNAL LOADS                         CONT1215
C     M28 - BOUNDARY CONDITION CONTRIBUTIONS TO CC                      CONT1220
C     M29 - EFFECTIVE STRESS AT P0                                      CONT1225
C     M30 - EFFECTIVE STRESS AT P1                                      CONT1230
C     M31 - EFFECTIVE CREEP STRAIN AT P0                                CONT1235
C     M32 - INCREMENT OF EFFECTIVE CREEP STRAIN                         CONT1240
      NGP=NE*9                                                          CONT1245
      M28=M27+NDFREE                                                    CONT1250
      M29=M28+NDFREE                                                    CONT1255
      M30=M29+NGP                                                       CONT1260
      M31=M30+NGP                                                       CONT1265
      M32=M31+NGP                                                       CONT1270
      MTOTAL=M32+NGP                                                    CONT1275
      NTOTAL=N23                                                        CONT1280
C                                                                       CONT1285
C           *** POINTERS COMPLETED ***                                  CONT1290
C        CHECK DATA AGAINST DIMENSION STATEMENTS                        CONT1295
C                                                                       CONT1300
   30 WRITE (6,5210) MTOTAL,KFLOAT,NTOTAL,KFIXED                        CONT1305
      IF ( MTOTAL.LE.KFLOAT.AND.NTOTAL.LE.KFIXED )  GO TO 40            CONT1310
      WRITE (6,5220)                                                    CONT1315
      CALL EXIT                                                         CONT1320
   40 CONTINUE                                                          CONT1325
C     INPUT EXTERNAL FORCE VECTOR                                       CONT1330
      CALL ZERO(NDFREE,1,AD(M27),IDUM)                                  CONT1335
      CALL EXLOAD(NG,NDFREE,AD(M27))                                    CONT1340
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CONT1345
C     PROBLEM DESCRIPTION COMPLETE                                      CONT1350
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * CONT1355
      CALL RDWRT(NE)                                                    CONT1360
      CALL INITL(NE)                                                    CONT1365
C     THE SEQUENCE OF EVENTS FOR THE CREEP PROGRAM IS AS FOLLOWS:       CONT1370
C     PASS #1 ASSEMBLES THE STIFFNESS MATRIX AND STORES VALUES          CONT1375
C     REQUIRED IN PASSES #2 ON USING SUBROUTINE RDWRT                   CONT1380
C     PASS #2 ASSEMBLES THE FORCE VECTOR AND CALCULATES THE FIRST       CONT1385
C     SET OF VALUES FOR ELASTIC STRESSES AND STRAINS.                   CONT1390
C     SUBSEQUENT PASSES REPEAT THE SOLUTION FOR EACH NEW TIME STEP.     CONT1395
      ACUTIM=ACUTIM-DELT-DELT                                           CONT1400
      IFACT=0                                                           CONT1405
      NTSP2=NTS+2                                                       CONT1410
      DO 100 IT = 1,NTSP2                                               CONT1415
      ACUTIM=ACUTIM+DELT                                                CONT1420
      IC=0                                                              CONT1425
   50 CONTINUE                                                          CONT1430
      FLAGS=0.                                                          CONT1435
      CALL RDWRT(NE)                                                    CONT1440
      IF(ACUTIM.GT.1.E-6)                                               CONT1445
     2  CALL CREEP(AD(M29),AD(M30),AD(M31),AD(M32),DELT,NGP  )          CONT1450
      IC=IC+1                                                           CONT1455
```

```
      IF(IT.GT.1) CALL VEQUAT ( NDFREE,AD(M20),AD(M27)  )                CONT1460
C                                                                        CONT1465
C        *** CALCULATE AND ASSEMBLE ELEMENT MATRICES ***                 CONT1470
C                                                                        CONT1475
      CALL RDWRT(NE)                                                     CONT1480
      CALL     ASYMBL   (M,N,NE,NG,NSPACE,NELFRE,MAXBAN,NDFREE,NITER,    CONT1485
     1        LPTEST,LHOMO,NNPFLO,IP1,IP3,IP4,IP5,JP2,KP1,KP2,NTAPE1,    CONT1490
     2        ID(N2),ID(N3),ID(N6),ID(N18),ID(N20),AD(M1),AD(M11),       CONT1495
     3        AD(M12),AD(M13),AD(M14),AD(M16),AD(M17),AD(M18),AD(M19),   CONT1500
     4        AD(M20),AD(M22),AD(M23),AD(M26),KP3)                       CONT1505
      IF(IT.EQ.1) GO TO 100                                              CONT1510
C                                                                        CONT1515
C        *** ASSEMBLY COMPLETED ***                                      CONT1520
C                                                                        CONT1525
C        *** APPLY BOUNDARY CONSTRAINTS TO NODAL PARAMETERS ***          CONT1530
C                                                                        CONT1535
      IF(IT.NE.2) GO TO 70                                               CONT1540
      IDN5=ID(N5)                                                        CONT1545
      DO 60 NEQ=1,IDN5                                                   CONT1550
   60  CALL MODFY1(NDFREE,MAXBAN,ID(N8+NEQ-1),AD(M2+NEQ-1),              CONT1555
     2  AD(M19),AD(M28)  )                                               CONT1560
   70 CONTINUE                                                           CONT1565
      CALL MADD(AD(M20),AD(M28),AD(M20),NDFREE,1)                        CONT1570
      CALL BC2(NDFREE,AD(M20),ID(N5),ID(N8),AD(M2)  )                    CONT1575
C                                                                        CONT1580
C        *** SOLVE FOR UNKNOWN NODAL PARAMETERS ***                      CONT1585
C                                                                        CONT1590
      IF(IFACT.EQ.0)   CALL FACTOR(NDFREE,MAXBAN,AD(M19))                CONT1595
      IFACT=1                                                            CONT1600
      CALL  SOLVE (NDFREE,MAXBAN,AD(M19),AD(M20),AD(M21))                CONT1605
C                                                                        CONT1610
C        *** DISPLACEMENT SOLUTION  COMPLETE ***                         CONT1615
C                                                                        CONT1620
      IF ( NPTWRT.EQ.0 )                                                 CONT1625
     1CALL  WRTPT (M,NG,NDFREE,NSPACE,AD(M1),AD(M21),ID(N6))             CONT1630
      IF ( LEMWRT.EQ.0 )                                                 CONT1635
     1CALL  WRTELM (NE,N,NG,NDFREE,NELFRE,AD(M21),ID(N6),ID(N2),ID(N3))  CONT1640
C                                                                        CONT1645
C        *** COMPUTE STRESSES AND STRAINS AND PRINT RESULTS ***          CONT1650
C                                                                        CONT1655
      INOUT=MOD(IT-2,IOT).EQ.0                                           CONT1660
      CALL RDWRT(NE)                                                     CONT1665
      CALL STRESS(NDFREE,NE,N,NELFRE,ID(N2),ID(N3),ID(N6),AD(M21),       CONT1670
     2   AD(M22),AD(M30),NGP,INOUT  )                                    CONT1675
C                                                                        CONT1680
      CALL RDWRT(NE)                                                     CONT1685
      IF(IT.EQ.2) GO TO 90                                               CONT1690
      IF(IC.EQ.ITLIM) GO TO 80                                           CONT1695
      IF(ERRMAX.LT.FLAGS) GO TO 50                                       CONT1700
   80 CONTINUE                                                           CONT1705
      IF(INOUT) WRITE(1,5250) ACUTIM,IC                                  CONT1710
      IF(INOUT) CALL OUT2(AD(M29),AD(M30),AD(M31),AD(M32),NE)            CONT1715
   90 CONTINUE                                                           CONT1720
      IF(INOUT) CALL OUTPUT                                              CONT1725
C     UPDATE VECTORS FOR NEXT TIME INCREMENT                             CONT1730
      CALL UPDATE                                                        CONT1735
  100 CONTINUE                                                           CONT1740
C     ***********************************************************************CONT1745
C                                F O R M A T S                           CONT1750
C     ***********************************************************************CONT1755
C                                                                        CONT1760
 5100 FORMAT ( 20A4 )                                                    CONT1765
```

```
 5110 FORMAT(9I5,3F10.0)                                                    CONT1770
 5120 FORMAT ('1**********************************************************CONT1775
     1**********',/,1X,20A4,/,' **********************************CONT1780
     2**************************************************',///)            CONT1785
 5130 FORMAT('0*****  PROBLEM PARAMETERS  *****',/,                        CONT1790
     1 ' NUMBER OF NODAL POINTS IN SYSTEM = ',I5,/,                         CONT1795
     2 ' NUMBER OF ELEMENTS IN SYSTEM = ',I5,/,                             CONT1800
     3 ' NUMBER OF NODES PER ELEMENT = ',I2,/,                              CONT1805
     4 ' NUMBER OF PARAMETERS PER NODE = ',I2,/,                            CONT1810
     5 ' DIMENSION OF SPACE = ',I2, /. )                                    CONT1815
 5140 FORMAT (' NUMBER OF ITERATIONS TO BE RUN = ',I5)                      CONT1820
 5150 FORMAT(' INCREMENT OF TIME = ',F12.4/)                                CONT1825
 5160 FORMAT ( 10I5 )                                                       CONT1830
 5170 FORMAT (' NUMBER OF FIXED POINT PROPERTIES PER NODE = ',I5,/,         CONT1835
     1 ' NUMBER OF FLOATING POINT PROPERTIES PER NODE = ',I5,/,             CONT1840
     2 ' NUMBER OF FIXED POINT PROPERTIES PER ELEMENT =',I5,/,              CONT1845
     3 ' NUMBER OF FLOATING POINT PROPERTIES PER ELEMENT = ', I5,/,         CONT1850
     4 ' NUMBER OF FIXED POINT MISCELLANEOUS SYSTEM PROPERTIES =',I5,/,     CONT1855
     5 ' NUMBER OF FLOATING POINT MISCELLANEOUS SYSTEM PROPERTIES =',       CONT1860
     6 I5,// )                                                              CONT1865
 5180 FORMAT (' NODAL POINT PROPERTIES ARE HOMOGENEOUS THROUGHOUT SYSTEM.') CONT1870
     1                                                                      CONT1875
 5190 FORMAT (' ELEMENT PROPERTIES ARE HOMOGENEOUS THROUGHOUT SYSTEM.')     CONT1880
 5200 FORMAT ('1EQUATION HALF BANDWIDTH =',I5,/,                            CONT1885
     1 ' AND OCCURS IN ELEMENT NUMBER',I5,'.',/,                            CONT1890
     2 ' MODIFICATION HALF BANDWIDTH =',I5,/,                               CONT1895
     3 ' MAXIMUM HALF BANDWIDTH OF SYSTEM =',I5,/,                          CONT1900
     4 ' TOTAL NUMBER OF SYSTEM EQUATIONS =',I5,/,                          CONT1905
     5 ' NUMBER OF ELEMENT DEGREES OF FREEDOM =',I5,//)                     CONT1910
 5210 FORMAT ('0*** ARRAY STORAGE ***',/,                                   CONT1915
     1 '       TYPE          REQUIRED      AVAILABLE',/,                  CONT1920
     2 ' FLOATING POINT',6X,I5,9X,I5,/,                                     CONT1925
     3 ' FIXED POINT',9X,I5,9X,I5,//)                                       CONT1930
 5220 FORMAT ('0AVAILABLE STORAGE EXCEEDED, ABNORMAL PROGRAM ENDING.')      CONT1935
 5230 FORMAT ('0ITERATION NUMBER = ',I5)                                    CONT1940
 5240 FORMAT (' NUMBER OF CONTOUR CURVES BETWEEN',/,                        CONT1945
     1 '   5TH & 95TH PERCENTILE OF EACH PARAMETER = ',I5)                  CONT1950
 5250 FORMAT(  '1 ACCUMULATED CREEP TIME =',F11.3/                          CONT1955
     2 '0 NUMBER OF ITERATIONS =',I5)                                       CONT1960
 5260 FORMAT('0*****  CREEP PARAMETERS  *****'/                            CONT1965
     2       ' NUMBER OF TIME STEPS TO BE RUN =',I5/                        CONT1970
     2       ' INCREMENT OF TIME =',E12.4/                                  CONT1975
     2       ' MAXIMUM CHANGE IN STRESS BETWEEN ITERATIONS =',E12.4/        CONT1980
     2       ' REDUCED TIME INCREMENT =',E12.4)                             CONT1985
 5270 FORMAT(' ITERATION TO CONTINUE UNTIL ERRMAX > FLAGS')                 CONT1990
 5280 FORMAT(' MAXIMUM NUMBER OF ITERATIONS PER TIME STEP =',I5)            CONT1995
      CALL EXIT                                                             CONT2000
      RETURN                                                                CONT2005
      END                                                                   CONT2010
```

```
      SUBROUTINE CREEP(SEFF0,SEFF1,EPEFF0,DEPEFF,DELT,NGP)                  CREE0005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CREE0010
C                                                                           CREE0015
C     THIS IS THE CREEP STRAIN SUBROUTINE                                   CREE0020
C     THE EFFECTIVE CREEP STRAIN INCREMENT AND CREEP STRAIN COMPONENTS      CREE0025
C     ARE CALCULATED HERE                                                   CREE0030
C                                                                           CREE0035
```

```
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CREE0040
C     THE FOLLOWING NOTATION APPLIES:                                      CPEE0045
C        SIG0,SIG1 - INITIAL AND FINAL STRESS COMPONENTS                   CREE0050
C        EP0,EP1   - INITIAL AND FINAL CREEP STRAIN COMPONENTS             CREE0055
C        EPS0 EPS1 - INITIAL AND FINAL TOTAL STRAIN COMPONENTS             CREE0060
C        EPEFF0    - INITIAL ACCUMULATED EFFECTIVE CREEP STRAIN            CREE0065
C        DEPEFF    - EFFECTIVE CREEP STRAIN INCREMENT                      CREE0070
C        SEFF0,SEFF1 - INITIAL AND FINAL EFFECTIVE STRESSES                CREE0075
C        DELT IS THE SPECIFIED TIME INCREMENT                              CREE0080
C     B IS THE LINEAR PATH FUNCTION SLOPE                                  CREE0085
C     A IS THE STRESS INTERCEPT OF THE LINEAR PATH FUNCTION                CREE0090
      IMPLICIT REAL*8 (A-H,O-Z)                                            CREE0095
      DIMENSION SEFF1(NGP),DEPEFF(NGP),EPEFF0(NGP),SEFF0(NGP),EP1(4),      CREE0100
     2  SIG0(4),SIG1(4),EP0(4)                                             CREE0105
      DIMENSION ZERO(4)                                                    CREE0110
      COMMON /CONVRG/ERRMAX,FLAGE,FLAGS,IC,IT                              CREE0115
      COMMON /PATH/PTYPE                                                   CREE0120
      DATA CONS/'CONS'/                                                    CREE0125
      DATA ZERO/4*0.D0/                                                    CREE0130
      DATA TFAC/0.9/                                                       CREE0135
      DATA IP,B,DTS/0,0.,0.0001/                                           CREE0140
C     DTS IS THE REDUCED TIME STEP USED TO ESTABLISH INITIAL PATH SLOPE.   CREE0145
      IF(DTS.EQ.0.) DTS=DELT                                               CREE0150
C     THE FOLLOWING DATA IS EQUIVALENT TO                                  CREE0155
C          EPS=6.4E-18*SIG**4.4*T                                          CREE0160
C     WHERE    EPS = % STRAIN                                              CREE0165
C              SIG = PSI                                                   CREE0170
C              T = HRS                                                     CREE0175
C     THE DATA HAS BEEN MODIFIED TO CALCULATE                              CREE0180
C              EPS = IN/IN                                                 CREE0185
C     WITH     SIG = KSI                                                   CREE0190
C              T = HRS                                                     CREE0195
C     LET  C = 1. - ETA     **************                                 CREE0200
      DATA G,ETA,C/1.01433E-4,4.4,-3.4/                                    CREE0205
C     FOR FIRST PASS OF EACH TIME INCREMENT IC EQUALS ZERO                 CREE0210
C     IF PATH TYPE IS CONSTANT GO TO 390 FOR EACH NEW TIME INCREMENT       CREE0215
      IF(PTYPE.EQ.CONS) GO TO 90                                           CREE0220
      DKP=DELT                                                             CREE0225
C     IF PATH TYPE IS LINEAR GO TO 290 FOR FIRST PASS OF THE FIRST TIME    CREE0230
C     INCREMENT TO SET VALUES OF SLOPE B USING STARTING TIME INCREMENT DC  REE0235
C     THE SECOND PASS OF THE FIRST TIME INCREMENT GOES TO 340 AND USES DC  REE0240
C     IF PATH TYPE IS LINEAR GO TO 190 FOR FIRST PASS OF ALL SUBSEQUENT    CREE0245
C     TIME INCREMENTS.  VALUE OF B FROM LAST STEP IS USED FOR THIS STEP    CREE0250
C     ONCE FIRST ESTIMATE OF EFF. CREEP STRAIN IS OBTAINED SUBSEQUENT PA   CREE0255
C     GO TO 20 TO USE ITERATIVE ALGORITHM                                  CREE0260
      IF(IP.EQ.0) GO TO 50                                                 CREE0265
      IF(IP.EQ.1) GO TO 70                                                 CREE0270
      IF(IC.EQ.0) GO TO 30                                                 CREE0275
   10 CONTINUE                                                             CREE0280
C    SECTION 1 --ITERATIVE CREEP ROUTINE                                   CREE0285
      DO 20 IG=1,NGP                                                       CREE0290
      CALL RSIG0(SIG0)                                                     CREE0295
C     CALCULATE IMPROVED ESTIMATE FOR B                                    CREE0300
      CALL RSIG1(SIG1)                                                     CREE0305
      SEF0=SEFF0(IG)                                                       CREE0310
      EPEF0=EPEFF0(IG)                                                     CREE0315
      DEP=DEPEFF(IG)                                                       CREE0320
      SEF1=SEFF1(IG)                                                       CREE0325
      B=(SEF1-SEF0)/DEP                                                    CREE0330
      SEFAV=(SEF0+SEF1)/2.D0                                               CREE0335
C     IF B IS TOO SMALL USE CONSTANT STRESS FORMULA.                       CREE0340
      IF(DABS(B).LT. 1.0) DIF=G*SEFAV**ETA*DELT                            CREE0345
```

```
C     FOR POSITIVE B THERE IS A LIMITATION ON DELT                      CREE0350
C     SUCH THAT DELT < -SEF1**C/(G*B*C) = DTMAX                         CREE0355
      DTMAX=-SEFO**C/(G*B*C)                                            CREE0360
      IF(B.LE.0.) DTMAX=1.E20                                           CREE0365
C     IF DELT > DTMAX REDUCE TIME STEP                                  CREE0370
      IF(DELT.GE.DTMAX) DELT=TFAC*DTMAX                                 CREE0375
      IF(DABS(B).GE.1.0)                                                CREE0380
     2  DIF=((G*B*C*DELT+SEFO**C)**(1./C)-SEFO)/B                       CREE0385
      DELT=DKP                                                          CREE0390
      CALL REPO(EP1)                                                    CREE0395
C     CALCULATE CREEP STRAIN COMPONENTS USING MEAN STRESSES             CREE0400
      CALL CRPCOM(SIGO,EP1,EP1,DIF,SEFO,2.DO)                           CREE0405
      CALL CRPCOM(SIG1,EP1,EP1,DIF,SEF1,2.DO)                           CREE0410
C     STORE CREEP STRAIN COMPONENTS                                     CREE0415
      CALL WEP1(EP1)                                                    CREE0420
C     STORE EFFECTIVE CREEP STRAIN INCREMENT                            CREE0425
      DEPEFF(IG)=DIF                                                    CREE0430
C     UPDATE FINAL EFFECTIVE STRESS VALUE.                              CREE0435
      SEFF1(IG)=SEFO+B*DIF                                              CREE0440
      IF(IG.NE.5) GO TO 20                                              CREE0445
   20 CONTINUE                                                          CREE0450
      RETURN                                                            CREE0455
   30 CONTINUE                                                          CREE0460
C    SECTION 2 -- STARTING CREEP ROUTINE FOR SECOND TIME STEP ON        CREE0465
      DO 40 IG=1,NGP                                                    CREE0470
C     CALCULATE PATH SLOPE OF PREVIOUS TIME STEP TO USE AS INITIAL      CREE0475
C     ESTIMATE THIS STEP                                                CREE0480
      SEF1=SEFF1(IG)                                                    CREE0485
      DEP=DEPEFF(IG)                                                    CREE0490
      B=(SEF1-SEFFO(IG))/DEP                                            CREE0495
      EPEF1=EPEFFO(IG)+DEP                                              CREE0500
C     UPDATE ACCUMULATED CREEP STRAIN AND INITIAL EFFECTIVE STRESS VALUECREE0505
      EPEFFO(IG)=EPEF1                                                  CREE0510
      SEFFO(IG)=SEF1                                                    CREE0515
      A=SEF1-EPEF1*B                                                    CREE0520
      IF(DABS(B).LT. 1.0) DIF=G*SEF1 **ETA*DELT                         CREE0525
      DTMAX=-SEF1**C/(G*B*C)                                            CREE0530
      IF(B.LE.0.) DTMAX=1.E20                                           CREE0535
      IF(DELT.GE.DTMAX) WRITE(1,5000) DTMAX,IG                          CREE0540
      IF(DELT.GE.DTMAX) DELT=TFAC*DTMAX                                 CREE0545
      IF(DABS(B).GE.1.0)                                                CREE0550
     2  DIF=((G*B*C*DELT+SEF1**C)**(1./C)-SEF1)/B                       CREE0555
      DELT=DKP                                                          CREE0560
C     STORE EFFECTIVE CREEP STRAIN INCREMENT                            CREE0565
C     STORE FINAL EFFECTIVE STRESS VALUE                                CREE0570
      DEPEFF(IG)=DIF                                                    CREE0575
      SEFF1(IG)=SEF1+B*DIF                                              CREE0580
C     RECALL LAST VALUES OF CREEP STRAIN AND STRESS                     CREE0585
      CALL REPO(EPO)                                                    CREE0590
      CALL RSIG1(SIG1)                                                  CREE0595
C     CALCULATE CREEP STRAIN COMPONENTS                                 CREE0600
      CALL CRPCOM(SIG1,EPO,EP1,DIF,SEF1,1.DO)                           CREE0605
C     STORE NEW CREEP STRAINS                                           CREE0610
      CALL WEP1(EP1)                                                    CREE0615
   40 CONTINUE                                                          CREE0620
      RETURN                                                            CREE0625
   50 CONTINUE                                                          CREE0630
C    SECTION 3 -- STARTING CREEP ROUTINE FOR FIRST TIME STEP            CREE0635
C     THIS STEP IS A DUMMY STEP TO SET INITIAL SLOPE OF STRESS-STRAIN PACREE0640
C     NO CREEP IS ACCUMULATED ON THIS STEP                              CREE0645
      DO 60 IG=1,NGP                                                    CREE0650
C     SET STRESS INTERCEPT, B IS ASSUMED ZERO                           CREE0655
```

```
      A=SEFF1(IG)                                                       CREE0660
      SEFF0(IG)=A                                                       CREE0665
C     DTS IS TIME STEP USED TO ESTABLISH INITIAL SLOPE                  CREE0670
      DIF=G*A**ETA*DTS                                                  CREE0675
C     STORE EFF. CREEP STRAIN INCREMENT                                 CREE0680
      DEPEFF(IG)=DIF                                                    CREE0685
C     CALCULATE AND STORE CREEP STRAINS                                 CREE0690
      CALL PSIG1(SIG1)                                                  CREE0695
      CALL CRPCOM(SIG1,ZERO,EP1,DIF,A,1.D0)                             CREE0700
      CALL WEP1(EP1)                                                    CREE0705
   60 CONTINUE                                                          CREE0710
C     SINCE THIS IS A DUMMY RUN, SET IC TO -1                           CREE0715
C     TO AVOID TRIPPING CONVERGENCE FLAG, TEMPORARILY SET STRESS FLAG   CREE0720
C     TO 1.KSI                                                          CREE0725
C     SET PASS POINTER TO 1 TO ROUTE NEXT PASS TO PART 2 OF THIS SECTIONCREE0730
      IP=1                                                              CREE0735
      IC=-1                                                             CREE0740
      FLAGS=1.                                                          CREE0745
      RETURN                                                            CREE0750
   70 CONTINUE                                                          CREE0755
C     THIS IS THE SECOND HALF OF THE FIRST TIME STEP CREEP SOLUTION.    CREE0760
C     THE B VALUE ESTABLISHED IN THE FIRST HALF IS USED TO CALCULATE    CREE0765
C     THE EFFECTIVE CREEP STRAIN INCREMENT FOR THE FIRST TIME STEP.     CREE0770
      DO 80 IG=1,NGP                                                    CREE0775
      SEF0=SEFF0(IG)                                                    CREE0780
      SEF1=SEFF1(IG)                                                    CREE0785
      DEP=DEPEFF(IG)                                                    CREE0790
      B=(SEF1-SEF0)/DEP                                                 CREE0795
      DTMAX=-SEF0**C/(G*B*C)                                            CREE0800
      IF(B.LE.0.) DTMAX=1.E20                                           CREE0805
      IF(DELT.GE.DTMAX) WRITE(1,5000) DTMAX,IG                          CREE0810
      IF(DABS(B).LT.1.0) DIF=G*SEF0**ETA*DELT                           CREE0815
      IF(DABS(B).GE.1.0) DIF=((G*B*C*DELT+SEF0**C)**(1./C)-SEF0)/B      CREE0820
C     STORE THE EFFECTIVE CREEP STRAIN INCREMENT FOR THIS TIME STEP.    CREE0825
      DEPEFF(IG)=DIF                                                    CREE0830
C     ADD NEW INCREMENTS OF CREEP STRAIN COMPONENTS AND STORE.          CREE0835
      CALL RSIG0(SIG0)                                                  CREE0840
      CALL CRPCOM(SIG0,ZERO,EP1,DIF,SEF0,1.D0)                          CREE0845
      CALL WEP1(EP1)                                                    CREE0850
   80 CONTINUE                                                          CREE0855
C     SET PASS POINTER TO 2 TO ROUTE ALL SUBSEQUENT PASSES TO           CREE0860
C     SECTIONS 1 AND 2 OF THIS ROUTINE                                  CREE0865
      IP=2                                                              CREE0870
      RETURN                                                            CREE0875
   90 CONTINUE                                                          CREE0880
C  SECTION 4 -- CONSTANT STRESS PATH CREEP ROUTINE                      CREE0885
C     THIS ROUTINE CALCULATES THE CONSTANT STRESS CREEP STRAIN INCREMENCREE0890
      DO 100 IG=1,NGP                                                   CREE0895
      SEF0=SEFF1(IG)                                                    CREE0900
      SEFF0(IG)=SEF0                                                    CREE0905
      EPEFF0(IG)=EPEFF0(IG)+DEPEFF(IG)                                  CREE0910
      DEP=G*SEF0**ETA*DELT                                              CREE0915
      DEPEFF(IG)=DEP                                                    CREE0920
      CALL REP0(EP0)                                                    CREE0925
      CALL RSIG1(SIG1)                                                  CREE0930
      CALL CRPCOM(SIG1,EP0,EP1,DEP,SEF0,1.D0)                           CREE0935
      CALL WEP1(EP1)                                                    CREE0940
  100 CONTINUE                                                          CREE0945
      RETURN                                                            CREE0950
 5000 FORMAT(' ', 6('*'),'WARNING--- MAXIMUM TIME STEP OF ',F12.4,2X,   CREE0955
     2  'EXCEEDED AT SYSTEM GAUSS POINT NO. ',I4)                      CREE0960
 5010 FORMAT(' ',9E14.5)                                                CREE0965
      END                                                               CREE0970
```

```
      SUBROUTINE CRPCOM(S,E0,E1,DE,SEF,PTS)                              CRPC0005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CRPC0010
C                                                                        CRPC0015
C     THIS ROUTINE CALCULATES THE CREEP STRAIN COMPONENTS                CRPC0020
C                                                                        CRPC0025
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *CRPC0030
C     TO ALLOW FOR MULTIPLE POINT AVERAGING SET PTS TO THE NUMBER        CRPC0035
C     OF POINTS TO BE CONSIDERED.  ONE CALL IS MADE TO THIS ROUTINE      CRPC0040
C     PER POINT.                                                         CRPC0045
C     THE FOLLOWING NOTATION APPLIES:                                    CRPC0050
C        E0   - INITIAL ACCUMULATED CREEP STRAIN COMPONENTS              CRPC0055
C        E1   - FINAL ACCUMULATED CREEP STRAIN COMPONENTS                CRPC0060
C        S    - STRESS COMPONENTS                                        CRPC0065
C        DE   - EFFECTIVE CREEP STRAIN INCREMENT                         CRPC0070
C        SEF  - EFFECTIVE STRESS                                         CRPC0075
      IMPLICIT REAL*8(A-H,O-Z)                                           CRPC0080
      DIMENSION S(1),E0(1),E1(1)                                         CRPC0085
      FAC=DE/(SEF*PTS)                                                   CRPC0090
      E1(1)=E0(1)+(S(1)-0.5D0*S(2)-0.5D0*S(4))*FAC                       CRPC0095
      E1(2)=E0(2)+(S(2)-0.5D0*S(4)-0.5D0*S(1))*FAC                       CRPC0100
      E1(4)=E0(4)+(S(4)-0.5D0*S(1)-0.5D0*S(2))*FAC                       CRPC0105
      E1(3)=E0(3)+1.5D0*S(3)*FAC                                         CRPC0110
      RETURN                                                             CRPC0115
      END                                                                CRPC0120
```

```
      SUBROUTINE EXLOAD(NG,NDFREE,XLOAD)                                 EXL00005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *EXL00010
C                                                                        EXL00015
C     THIS ROUTINE INPUTS THE EXTERNAL FORCES                            EXL00020
C                                                                        EXL00025
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *EXL00030
      IMPLICIT REAL*8(A-H,O-Z)                                           EXL00035
      DIMENSION XLOAD(NDFREE)                                            EXL00040
      WRITE(6,5000)                                                      EXL00045
 5000 FORMAT('1    ***EXTERNAL LOADS***'/                               EXL00050
     2  '0   NODE',6X,'XCOMP',10X,'YCOMP'//)                            EXL00055
      M=NDFREE/NG                                                        EXL00060
      DO 10 I=1,M                                                        EXL00065
      READ(5,5010) J,FX,FY                                               EXL00070
 5010 FORMAT(I5,2F10.0)                                                  EXL00075
      IF(FX.EQ.0.0.AND.FY.EQ.0.0) GO TO 20                               EXL00080
      WRITE(6,5020) J,FX,FY                                              EXL00085
      K=J*NG-1                                                           EXL00090
 5020 FORMAT('0',I5,2E15.5)                                              EXL00095
      L=J*NG                                                             EXL00100
      XLOAD(K)=FX                                                        EXL00105
      XLOAD(L)=FY                                                        EXL00110
   10 CONTINUE                                                           EXL00115
   20 CONTINUE                                                           EXL00120
      RETURN                                                             EXL00125
      END                                                                EXL00130
```

```
      SUBROUTINE  ELSQ  (N,NELFRE,NSPACE,IP5,JP2,KP1,KP2,KP3,COORD,     ELSQ0005
     1                   D,ELPROP,LPROP,PRTLPT,FLTMIS,S)                ELSQ0010
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0015
C                    SQUARE ELEMENT MATRIX                             ELSQ0020
C       A PROBLEM DEPENDENT ROUTINE                                    ELSQ0025
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0030
      IMPLICIT REAL*8(A-H,O-Z)                                         ELSQ0035
      DIMENSION COORD(N,NSPACE),D(NELFRE),ELPROP(KP2),LPROP(JP2),      ELSQ0040
     1          S(NELFRE,NELFRE), PRTLPT(IP5,KP1), FLTMIS(KP3)         ELSQ0045
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0050
      DIMENSION BDB(12,12),B(4,12),BD(12,4),DD(4,4)                    ELSQ0055
      COMMON /MATERL/E,V,DD                                            ELSQ0060
      COMMON /JOBTYP/JTYP                                              ELSQ0065
C     . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ELSQ0070
C     ISOPARAMETRIC QUADRILATERAL WITH SIX NODES                       ELSQ0075
C     NELFRE = 12, N = 6, NSPACE = 2, NG = 2                           ELSQ0080
C     APPLICATION:  AXISYMMETRIC, PLANE STRAIN, OR PLANE STRESS        ELSQ0085
C          CREEP ANALYSIS ---  INITIAL STRAIN APPROACH                 ELSQ0090
C     VERSION 1                                                        ELSQ0095
      DIMENSION AJ(2,2),DELTA(2,6),GLOBAL(2,6),DUM1(2),DUM2(2)         ELSQ0100
      DIMENSION WT(9),ETAPT(9),ZETAPT(9),H(4),GWT(3),GPT(3)            ELSQ0105
      DIMENSION  IDUM1(2), IDUM2(2)                                    ELSQ0110
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0115
C     IF  JTYP = 1, AXISYMMETRIC                                       ELSQ0120
C     IF  JTYP = 2, PLAIN  STRAIN                                      ELSQ0125
C     IF  JTYP = 3, PLANE  STRESS                                      ELSQ0130
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0135
      COMMON /RADIUS/RGP(36)                                           ELSQ0140
      DATA IGP/0/                                                      ELSQ0145
      LFRE2 = NELFRE*NELFRE                                            ELSQ0150
      NB=4*NELFRE                                                      ELSQ0155
      CALL ZERO(NB,1,B,IDUM)                                           ELSQ0160
      CALL ZERO(LFRE2,1,BDB,IDUM)                                      ELSQ0165
C     SET UP INTEGRATION DATA FOR NITG POINTS IN EACH DIMENSION        ELSQ0170
      NITG = 3                                                         ELSQ0175
      CALL  GAUSCO (NITG,GPT,GWT)                                      ELSQ0180
      NUMPTS = NITG*NITG                                               ELSQ0185
      K = 0                                                            ELSQ0190
      DO 20  I = 1,NITG                                                ELSQ0195
      DO 10  J = 1,NITG                                                ELSQ0200
      K = K + 1                                                        ELSQ0205
      WT(K) = GWT(I)*GWT(J)                                            ELSQ0210
      ZETAPT(K) = GPT(I)                                               ELSQ0215
      ETAPT(K) = GPT(J)                                                ELSQ0220
   10 CONTINUE                                                         ELSQ0225
   20 CONTINUE                                                         ELSQ0230
C     ZERO WORKING SPACE MATRICES                                      ELSQ0235
      CALL  ZERO ( LFRE2,1,S,IDUM)                                     ELSQ0240
C     NUMERICAL INTEGRATION LOOP                                       ELSQ0245
      DO 60 IP = 1,NUMPTS                                              ELSQ0250
C     FORM JACOBIAN MATRIX,   AJ                                       ELSQ0255
      CALL DER6Q(ZETAPT(IP),ETAPT(IP),DELTA)                           ELSQ0260
      CALL  MMULT(DELTA,COORD,AJ,NSPACE,N,NSPACE)                      ELSQ0265
C     CALCULATE INVERSE & DETERMINATE OF JACOBIAN                      ELSQ0270
      DET=AJ(1,1)*AJ(2,2)-AJ(1,2)*AJ(2,1)                              ELSQ0275
```

```
      A11=AJ(1,1)                                                       ELSQ0280
      AJ(1,1)=AJ(2,2)/DET                                               ELSQ0285
      AJ(1,2)=-AJ(1,2 )/DET                                             ELSQ0290
      AJ(2,1)=-AJ(2,1 )/DET                                             ELSQ0295
      AJ(2,2)=A11/DET                                                   ELSQ0300
C     ESTABLISH GLOBAL DERIVATIVES RELATIONS                            ELSQ0305
      CALL  MMULT     (AJ,DELTA,GLOBAL,NSPACE,NSPACE,N)                 ELSQ0310
      IF(JTYP.GT.1) GO TO 40                                            ELSQ0315
C     CALCULATE RADIUS AT GAUSS POINT & COMPUTE SHAPE FUNCTIONS         ELSQ0320
      CALL SHP6Q(ZETAPT(IP),ETAPT(IP),H)                                ELSQ0325
      CALL MMULT(H,COORD,R,1,6,1)                                       ELSQ0330
      IF(IP.NE.8.AND.IP.NE.5.AND.IP.NE.2) GO TO 30                      ELSQ0335
      IGP=IGP+1                                                         ELSQ0340
      RGP(IGP)=R                                                        ELSQ0345
   30 CONTINUE                                                          ELSQ0350
   40 CONTINUE                                                          ELSQ0355
      IF(JTYP.NE.1) R=1.0                                               ELSQ0360
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0365
C                                                                       ELSQ0370
C         FORM B MATRIX                                                 ELSQ0375
C                                                                       ELSQ0380
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0385
      DO 50 L=1,6                                                       ELSQ0390
      J=2*L-1                                                           ELSQ0395
      K=2*L                                                             ELSQ0400
      B(3,K)=GLOBAL(1,L)                                                ELSQ0405
      B(2,K)=GLOBAL(2,L)                                                ELSQ0410
      IF(JTYP.EQ.1) B(4,J)=H(L)/R                                       ELSQ0415
      DO 50 I=1,2                                                       ELSQ0420
      II=I*2-1                                                          ELSQ0425
   50 B(II,J)=GLOBAL(I,L)                                               ELSQ0430
C     STORE VALUE OF B AT GAUSS POINT                                   ELSQ0435
      CALL WB(B)                                                        ELSQ0440
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0445
C                                                                       ELSQ0450
C     FORM BTRANS*DD AND BTRANS*DD*B                                    ELSQ0455
C                                                                       ELSQ0460
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0465
      CALL MTMULT(B,DD,BD,4,NELFRE,4)                                   ELSQ0470
      CALL MMULT(BD,B,BDB,NELFRE,4,NELFRE)                              ELSQ0475
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0480
C     FORM PARTIAL CONTRIBUTIONS TO INTEGRALS,STORE B,BTRANS*DD,        ELSQ0485
C                                                                       ELSQ0490
C     S = WT*BDB*DET + S                                                ELSQ0495
C                                                                       ELSQ0500
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELSQ0505
      F=WT(IP)*DET*R                                                    ELSQ0510
      CALL MSMULT(F,BD,NELFRE,4)                                        ELSQ0515
      CALL MSMULT(F,BDB,12,12)                                          ELSQ0520
      CALL MADD(S,BDB,S,12,12)                                          ELSQ0525
C     STORE VALUE OF BD AT GAUSS POINT                                  ELSQ0530
      CALL WBD(BD)                                                      ELSQ0535
   60 CONTINUE                                                          ELSQ0540
C     INTEGRATION COMPLETED                                             ELSQ0545
      RETURN                                                            ELSQ0550
      END                                                               ELSQ0555
```

```
      SUBROUTINE ELCOL(C)                                                ELC00005
      IMPLICIT REAL*8 (A-H,O-Z)                                          ELC00010
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELC00015
C      ELEMENT COLUMN MATRIX                                             ELC00020
C       A PROBLEM DEPENDENT ROUTINE                                      ELC00025
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELC00030
      DIMENSION B(4,12),BD(12,4),FE(12),C(12),EP1(4)                     ELC00035
C     . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ELC00040
C     ISOPARAMETRIC QUADRILATERAL WITH SIX NODES                         ELC00045
C     APPLICATION: PLANE STRESS SOLUTION OF CREEP PROBLEM                ELC00050
C     NELEME = 12, N = 6, NSPACE = 2, NG = 2                             ELC00055
C     NUMERICAL INTEGRATION LOOP                                         ELC00060
      DO 10 I=1,12                                                       ELC00065
   10 C(I)=0.0                                                           ELC00070
      NUMPTS=9                                                           ELC00075
      DO 20 IP = 1,NUMPTS                                                ELC00080
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELC00085
C                                                                        ELC00090
C      GENERATE INITIAL STRESS & STRAIN  LOAD CONTRIBUTIONS              ELC00095
C                                                                        ELC00100
C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ELC00105
C     RECALL VALUE OF BD AND EP1 AT GAUSS POINT                          ELC00110
      CALL REP1(EP1)                                                     ELC00115
      CALL RBD(BD)                                                       ELC00120
      CALL MMULT(BD,EP1,FE,12,4,1)                                       ELC00125
      CALL MADD(FE,C,C,12,1)                                             ELC00130
   20 CONTINUE                                                           ELC00135
C     INTEGRATION COMPLETED                                              ELC00140
      RETURN                                                             ELC00145
      END                                                                ELC00150
```

```
      SUBROUTINE OUT2(SEFF0,SEFF1,EPEFF0,DEPEFF,NE)                      OUT20005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *OUT20010
C                                                                        OUT20015
C     THIS ROUTINE OUTPUTS THE EFFECTIVE CREEP PARAMETERS AFTER          OUT20020
C     THE SPECIFIED ACCUMULATED ELAPSED TIME.                            OUT20025
C                                                                        OUT20030
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *OUT20035
      IMPLICIT REAL*8 (A-H,O-Z)                                          OUT20040
      DIMENSION SEFF0(1),SEFF1(1),EPEFF0(1),DEPEFF(1)                    OUT20045
      WRITE(1,5000)                                                      OUT20050
 5000 FORMAT(//'0EFFECTIVE CREEP PARAMETERS AT ELEMENT CENTROIDS'//      OUT20055
     2' ELEMENT',3X,'EPEFF',10X,'DEPEFF',9X,'SEFF',11X,'DSEFF',10X,'B')  OUT20060
      DO 10 IE=1,NE                                                      OUT20065
      I=(IE-1)*9+5                                                       OUT20070
      DS=SEFF1(I)-SEFF0(I)                                               OUT20075
      B=DS/DEPEFF(I)                                                     OUT20080
      EPEFF1=EPEFF0(I)+DEPEFF(I)                                         OUT20085
      WRITE(1,5010) IE,EPEFF1,DEPEFF(I),SEFF1(I),DS,B                    OUT20090
 5010 FORMAT('0',I5,6E15.4)                                              OUT20095
   10 CONTINUE                                                           OUT20100
      RETURN                                                             OUT20105
      END                                                                OUT20110
```

```
      SUBROUTINE RDWRT(NE)                                                  RDWR0005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *RDWR0010
C                                                                           RDWR0015
C     THIS ROUTINE STORES ALL ADDITIONAL VECTORS AND MATRICES REQUIRED      RDWR0020
C     BY THE CREEP PROGRAM AND NOT SPECIFICALLY ALLOWED FOR IN SUBROUTINRDWR0025
C     CONTRL.  DATA IS STORED AND RETRIEVED SEQUENTIALLY IN A MANNER        RDWR0030
C     ANALOGOUS TO WRITING ON AND READING FROM A SEQUENTIAL DATA SET.       RDWR0035
C     A CALL TO ENTRY POINT RDWRT INITIALIZES ALL INDEDES.  SUBSEQUENT      RDWR0040
C     CALLS TO A SPECIFIC "W" ENTRY POINT STORES THE SOECIFIED VARIABLE     RDWR0045
C     AT THE CURRENT CORRESPONDING INDEX LOCATION AND INCREMENTS THAT       RDWR0050
C     INDEX FOR THE NEXT CALL TO THAT ENTRY POINT.  A CALL TO A SPECIFICRDWR0055
C     "R" ENTRY POINT RETURNS THE CURRENT VALUE OF THE VARIABLE AT THE      RDWR0060
C     CURRENT CORRESPONDING INDEX LOCATION AND INCREMENTS THAT INDEX        RDWR0065
C     FOR THE NEXT CALL TO THAT ENTRY POINT.  A CALL TO ENTRY POINT INITRDWR0070
C     SETS THE ENTIRE C VECTOR TO ZERO.  A CALL TO ENTRY POINT OUTPUT       RDWR0075
C     CAUSES THE PRINTING OF THE CURRENT VALUES OF THE INDICATED            RDWR0080
C     VARIABLES.                                                            RDWR0085
C     THE FOLLOWING PARAMETERS ARE STORED AT EACH GAUSS POINT OF THE        RDWR0090
C     MODEL:                                                                RDWR0095
C        B    - STRAIN DISPLACEMENT MATRIX                                  RDWR0100
C        BD   - PRODUCT OF B AND ELEMENT ELASTICITY MATRIX D                RDWR0105
C        BDB  - MATRIX PRODUCT (BTRANS*(D*B))                               RDWR0110
C        EPO  - INITIAL CREEP STRAIN COMPONENTS                             RDWR0115
C        SIGO - INITIAL STRESS COMPONENTS                                   RDWR0120
C        EP1  - FINAL CREEP STRAIN COMPONENTS                               RDWR0125
C        SIG1 - FINAL STRESS COMPONENTS                                     RDWR0130
C        EPS1 - FINAL TOTAL STRAIN COMPONENTS                               RDWR0135
C     THE DIMENSION OF C MUST SATISFY:                                      RDWR0140
C         N >= 116*NE*NGP                                                   RDWR0145
C     WHERE NE IS NUMBER OF ELEMENTS AND NGP IS NUMBER OF GAUSS POINTS      RDWR0150
C     PER ELEMENT                                                           RDWR0155
C                                                                           RDWR0160
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *RDWR0165
      IMPLICIT REAL*8 (A-H,O-Z)                                             RDWR0170
      COMMON /CONVRG/ERRMAX,FLAGE,FLAGS,IC,ITER                             RDWR0175
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * RDWR0180
C    . IF NO. OF ELEMENTS EXCEEDS 12, INCREASE DIMENSION OF C ACCORDINGLYRDWR0185
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * RDWR0190
      DIMENSION C(12528),S(1)                                               RDWR0195
      MB=0                                                                  RDWR0200
      MBD=432*NE                                                            RDWR0205
      MEPO=864*NE                                                           RDWR0210
      MSIGO=900*NE                                                          RDWR0215
      MEP1=936*NE                                                           RDWR0220
      MSIG1=972*NE                                                          RDWR0225
      MEPS1=1008*NE                                                         RDWR0230
      RETURN                                                                RDWR0235
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *RDWR0240
C                                                                           RDWR0245
C     "W" ENTRY POINTS                                                      RDWR0250
C                                                                           RDWR0255
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *RDWR0260
      ENTRY WB    (S)                                                       RDWR0265
      DO 10 I=1,48                                                          RDWR0270
   10 C(MB+I)=S(I)                                                          RDWR0275
      MB=MB+48                                                              RDWR0280
      RETURN                                                                RDWR0285
      ENTRY WBD   (S)                                                       RDWR0290
      DO 20 I=1,48                                                          RDWR0295
   20 C(MBD+I)=S(I)                                                         RDWR0300
      MBD=MBD+48                                                            RDWR0305
      RETURN                                                                RDWR0310
```

```
      ENTRY WBDB  (S)                                                   RDWR0315
      DO 30 I=1,64                                                      RDWR0320
   30 C(MBDB+I)=S(I)                                                    RDWR0325
      MBDB=MBDB+64                                                      RDWR0330
      RETURN                                                            RDWR0335
      ENTRY WEPO  (S)                                                   RDWP0340
      DO 40 I=1,4                                                       RDWR0345
   40 C(MEPO+I)=S(I)                                                    RDWR0350
      MEPO=MEPO+4                                                       RDWR0355
      RETURN                                                            RDWP0360
      ENTRY WEP1  (S)                                                   RDWR0365
      DO 50 I=1,4                                                       RDWR0370
   50 C(MEP1+I)=S(I)                                                    RDWR0375
      MEP1=MEP1+4                                                       RDWR0380
      RETURN                                                            RDWP0385
      ENTRY WEPS1 (S)                                                   RDWR0390
      DO 60 I=1,4                                                       RDWR0395
   60 C(MEPS1+I)=S(I)                                                   RDWR0400
      MEPS1=MEPS1+4                                                     RDWR0405
      RETURN                                                            RDWR0410
      ENTRY WSIGO (S)                                                   RDWR0415
      DO 70 I=1,4                                                       RDWR0420
   70 C(MSIGO+I)=S(I)                                                   RDWR0425
      MSIGO=MSIGO+4                                                     RDWR0430
      RETURN                                                            RDWR0435
      ENTRY WSIG1 (S)                                                   RDWR0440
      DO 80 I=1,4                                                       RDWR0445
   80 C(MSIG1+I)=S(I)                                                   RDWR0450
      MSIG1=MSIG1+4                                                     RDWR0455
      RETURN                                                            RDWR0460
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *RDWR0465
C                                                                       RDWR0470
C     "R" ENTRY POINTS                                                  RDWR0475
C                                                                       RDWR0480
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *RDWR0485
      ENTRY RB    (S)                                                   RDWR0490
      DO 90 I=1,48                                                      RDWR0495
   90 S(I)=C(MB+I)                                                      RDWR0500
      MB=MB+48                                                          RDWR0505
      RETURN                                                            RDWR0510
      ENTRY RBD   (S)                                                   RDWR0515
      DO 100 I=1,48                                                     RDWP0520
  100  S(I)=C(MBD+I)                                                    RDWR0525
      MBD=MBD+48                                                        RDWR0530
      RETURN                                                            RDWR0535
      ENTRY RBDB  (S)                                                   RDWR0540
      DO 110 I=1,64                                                     RDWR0545
  110  S(I)=C(MBDB+I)                                                   RDWR0550
      MBDB=MBDB+64                                                      RDWR0555
      RETURN                                                            RDWR0560
      ENTRY REPO  (S)                                                   RDWR0565
      DO 120 I=1,4                                                      RDWP0570
  120 S(I)=C(MEPO  +I)                                                  RDWR0575
      MEPO=MEPO+4                                                       RDWR0580
      RETURN                                                            RDWR0585
      ENTRY REP1  (S)                                                   RDWR0590
      DO 130 I=1,4                                                      RDWP0595
  130 S(I)=C(MEP1  +I)                                                  RDWR0600
      MEP1=MEP1+4                                                       RDWP0605
      RETURN                                                            PDWR0610
      ENTRY RSIGO (S)                                                   RDWP0615
      DO 140 I=1,4                                                      RDWP0620
```

```
  140 S(I)=C(MSIGO +I)                                                  RDWR0625
      MSIGO=MSIGO+4                                                     RDWR0630
      RETURN                                                            RDWR0635
      ENTRY RSIG1 (S)                                                   RDWR0640
      DO 150 I=1,4                                                      RDWR0645
  150 S(I)=C(MSIG1 +I)                                                  RDWR0650
      MSIG1=MSIG1+4                                                     RDWR0655
      RETURN                                                            RDWR0660
      ENTRY UPDATE                                                      RDWR0665
      IT=72*NE                                                          RDWR0670
      MEPO=864*NE                                                       RDWR0675
      MEP1=MEPO+IT                                                      RDWR0680
      DO 160 I=1,IT                                                     RDWR0685
  160 C(I+MEPO)=C(I+MEP1)                                               RDWR0690
      RETURN                                                            RDWR0695
      ENTRY INITL                                                       RDWR0700
      MEND=1044*NE                                                      RDWR0705
      DO 170 I=1,MEND                                                   RDWR0710
  170 C(I)=0.                                                           RDWR0715
      RETURN                                                            RDWR0720
      ENTRY OUTPUT                                                      RDWR0725
      WRITE(1,5000)                                                     RDWR0730
      WRITE(1,5010)                                                     RDWR0735
      DO 180 I=1,NE                                                     RDWR0740
      L1=(I-1)*36 + 21                                                  RDWR0745
      L2=L1+3                                                           RDWR0750
      WRITE(1,5020) I,(C(MEPS1+J),J=L1,L2),(C(MSIG1+J),J=L1,L2)         RDWR0755
  180 CONTINUE                                                          RDWR0760
 5000 FORMAT(///'0 TOTAL  STRAINS  AND  STRESSES  AT  ELEMENT  CENTROIDSRDWR0765
     2 ')                                                               RDWR0770
 5010 FORMAT('0ELEMENT', 5X,'EXX',13X,'EYY',13X,'EXY',13X,'EZZ',        RDWR0775
     2  13X,'SXX',13X,'SYY',13X,'SXY',13X,'SZZ')                       RDWR0780
 5020 FORMAT('0',I4,8E16.6)                                             RDWR0785
 5030 FORMAT(///'0 TOTAL STRAINS AND STRESSES AT LEFTMOST GAUSS POINTS')RDWR0790
 5040 FORMAT(///'0TOTAL STRAINS AND STRESSES AT RIGHTMOST GAUSS POINTS')RDWR0795
      RETURN                                                            RDWR0800
      END                                                               RDWR0805
```

```
      SUBROUTINE SHP6Q (ZETA,ETA,H)                                     SHP60005
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *SHP60010
C                                                                       SHP60015
C     THIS ROUTINE RETURNS THE VALUE OF THE SHAPE FUNCTIONS             SHP60020
C     FOR A SPECIFIED VALUE OF THE LOCAL ELEMENT COORDINATES.           SHP60025
C     ENTRY POINT DER6Q RETURNS THE CORRESPONDING DERIVATIVES.          SHP60030
C                                                                       SHP60035
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *SHP60040
      IMPLICIT REAL*8(A-H,O-Z)                                          SHP60045
      DIMENSION H(6),D(2,6)                                             SHP60050
      ZP=1.+ZETA                                                        SHP60055
      ZM=1.-ZETA                                                        SHP60060
      EP=1.+ETA                                                         SHP60065
      EM=1.-ETA                                                         SHP60070
      H(5)=0.5*ZP*ZM*EM                                                 SHP60075
      H(6)=0.5*ZP*ZM*EP                                                 SHP60080
      H(1)=0.25*ZM*EM-0.5*H(5)                                          SHP60085
      H(2)=0.25*ZP*EM-0.5*H(5)                                          SHP60090
      H(3)=0.25*ZP*EP-0.5*H(6)                                          SHP60095
```

```
      H(4)=0.25*ZM*EP-0.5*H(6)                                          SHP60100
      RETURN                                                            SHP60105
      ENTRY DER6Q (ZETA,ETA,D)                                          SHP60110
      ZP=1.+ZETA                                                        SHP60115
      ZM=1.-ZETA                                                        SHP60120
      EP=1.+ETA                                                         SHP60125
      EM=1.-ETA                                                         SHP60130
      D(1,5)=-ZETA*EM                                                   SHP60135
      D(2,5)=-0.5*ZP*ZM                                                 SHP60140
      D(1,6)=-ZETA*EP                                                   SHP60145
      D(2,6)=0.5*ZP*ZM                                                  SHP60150
      D(1,1)=-0.25*EM-0.5*D(1,5)                                        SHP60155
      D(2,1)=-0.25*ZM-0.5*D(2,5)                                        SHP60160
      D(1,2)=0.25*EM-0.5*D(1,5)                                         SHP60165
      D(2,2)=-0.25*ZP-0.5*D(2,5)                                        SHP60170
      D(1,3)=0.25*EP-0.5*D(1,6)                                         SHP60175
      D(2,3)=0.25*ZP-0.5*D(2,6)                                         SHP60180
      D(1,4)=-0.25*EP-0.5*D(1,6)                                        SHP60185
      D(2,4)=0.25*ZM-0.5*D(2,6)                                         SHP60190
      RETURN                                                            SHP60195
      END                                                               SHP60200
```

```
      SUBROUTINE STRESS(NDFREE,NE,N,NDF,NODES,LNODE,INDEX,D,DEL,SEFF1   STRE0005
     2  ,NGP,INOUT  )                                                   STRE0010
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * STRE0015
C                                                                       STRE0020
C    THIS ROUTINE CALCULATES THE STRESSES AND STRAINS FROM THE SYSTEM D. STRE0025
C                                                                       STRE0030
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * STRE0035
      IMPLICIT REAL*8(A-H,O-Z)                                          STRE0040
      LOGICAL INOUT                                                     STRE0045
      DIMENSION B(4,12),DD(4,4),EP1(4),EPS1(4),SIG1(4),SEFF1(NGP)       STRE0050
     2  ,D(NDFREE),NODES(NE,N),LNODE (N),INDEX(NDF),DEL(NDF)            STRE0055
      COMMON /MATERL/E,V,DD                                             STRE0060
      COMMON /CONVRG/ERRMAX,FLAGE,FLAGS,IC,IT                           STRE0065
      COMMON /JOBTYP/JTYP                                               STRE0070
      DIMENSION X(3),GWT(3),WT(9)                                       STRE0075
      DIMENSION HP(12)                                                  STRE0080
      DIMENSION SD(4,9)                                                 STRE0085
      COMMON /RADIUS/RGP(36)                                            STRE0090
      CALL GAUSCO(3,X,GWT)                                              STRE0095
      IR=0                                                              STRE0100
      K=0                                                               STRE0105
      DO 10 I=1,3                                                       STRE0110
      DO 10 J=1,3                                                       STRE0115
      K=K+1                                                             STRE0120
   10 WT(K)=GWT(I)*GWT(J)                                               STRE0125
      NG=2                                                              STRE0130
      DO 60 IG=1,NGP                                                    STRE0135
      IE=(IG-1)/9+1                                                     STRE0140
C     RECALL B AND EP1 AT GAUSS POINT                                   STRE0145
      CALL RB(8)                                                        STRE0150
      CALL REP1(EP1)                                                    STRE0155
C     USE NEAR INCOMPRESSIBILITY MODIFICATION TO COMPUTE STRESSES.      STRE0160
C     USING GAUSSIAN QUADRATURES CALCULATE THE VOLUMETRIC AVERAGE OF THESTRE0165
C     DILATATION.  THE STRESS DEVIATOR SD IS CALCULATED USING THE       STRE0170
C     LAME FORM OF THE ELASTIC CONSTITUTIVE EQUATIONS.                  STRE0175
```

```
C      THE FORMULA USED FOR THE VOLUMETRIC AVERAGE IS:                  STRE0180
C         DILATATION = SUM / DIV                                        STRE0185
C      WHERE SUM IS THE INTEGRAL OF THE DILATATION OVER THE ELEMENT     STRE0190
C      VOLUME AND DIV IS THE ELEMENT VOLUME.                            STRE0195
       ESUM=0.                                                          STRE0200
       IP=MOD(IG,9)                                                     STRE0205
       IF(IP.EQ.0) IP=9                                                 STRE0210
       IF(MOD(IG,9).NE.1) GO TO 30                                      STRE0215
       SUM=0.                                                           STRE0220
       CALL LNODES(IE,NE,N,NODES,LNODE)                                 STRE0225
       CALL INDXEL(N,NDF,NG,LNODE,INDEX)                                STRE0230
       DO 20 I=1,NDF                                                    STRE0235
       DEL(I)=D(INDEX(I))                                               STRE0240
    20 CONTINUE                                                         STRE0245
    30 CONTINUE                                                         STRE0250
C                                                                       STRE0255
C     CALCULATE NEW STRAINS                                             STRE0260
C                                                                       STRE0265
       CALL MMULT(B,DEL,EPS1,4,12,1)                                    STRE0270
C                                                                       STRE0275
C                                                                       STRE0280
       CALL ZERO(4,1,SIG1,I)                                            STRE0285
       DO 40 I=1,4                                                      STRE0290
       DE=EPS1(I)-EP1(I)                                                STRE0295
       IF(I.NE.3) ESUM=ESUM+DE                                          STRE0300
    40 SD(I,IP)=E/(1.+V)*DE                                             STRE0305
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * STRE0310
C                                                                       STRE0315
C       STRESSES AND STRAINS FOR RELAXATION PHASE COMPLETED             STRE0320
C                                                                       STRE0325
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * STRE0330
       CALL WEPS1(EPS1)                                                 STRE0335
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * STRE0340
C    COMPUTE EFFECTIVE STRESS                                           STRE0345
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * STRE0350
       SEF1=0.7071067812*DSQRT((SD(1,IP)-SD(2,IP))**2+                  STRE0355
      2  (SD(2,IP)-SD(4,IP))**2+(SD(4,IP)-SD(1,IP))**2+                 STRE0360
      3  6.*SD(3,IP)**2)                                               STRE0365
       FLAGS=DMAX1(DABS(SEF1-SEFF1(IG)),FLAGS)                          STRE0370
       SEFF1(IG)=SEF1                                                   STRE0375
       IF(MOD(IG,3).EQ.1) IR=IR+1                                       STRE0380
       SUM=SUM+ESUM*WT(IP)*RGP(IR)                                      STRE0385
       IF(IP.NE.9) GO TO 60                                             STRE0390
       DIV=2.*GWT(1)*RGP(IR-2)+2.*GWT(2)*RGP(IR-1)+2.*GWT(3)*RGP(IR)    STRE0395
       H=SUM/(DIV*(1.-2.*V))                                            STPE0400
       S=H*E/3.                                                         STRE0405
       HP(IE)=V*E/(1.+V)*H                                              STRE0410
       DO 50 I=1,9                                                      STRE0415
C      CALCULATE STRESSES FROM STRESS DEVIATORS AND FROM THE AVERAGE    STRE0420
C      DILATATION.                                                      STRE0425
       SIG1(1)=SD(1,I)+HP(IE)                                           STRE0430
       SIG1(2)=SD(2,I)+HP(IE)                                           STRE0435
       SIG1(3)=SD(3,I)                                                  STRE0440
       SIG1(4)=SD(4,I)+HP(IE)                                           STRE0445
C      STORE THE STRESSES AT THE GAUSS POINTS.                          STRE0450
       CALL WSIG1(SIG1)                                                 STRE0455
    50 CONTINUE                                                         STRE0460
    60 CONTINUE                                                         STRE0465
       RETURN                                                           STRE0470
        END                                                             STRE0475
```

VITA

Robert Michael Pardue was born in New Orleans, Louisiana on May 17, 1944. He attended school in that city and in June 1962, he was graduated from John McDonogh Senior High School. In September, 1962, he entered the School of Engineering at Louisiana State University, receiving a B.S. degree in Engineering Science in August 1966, and an M.S. in Engineering Mechanics in December 1968.

Upon completion of his M.S. degree, the author accepted a position with the Union Carbide Corporation Y-12 Plant in Oak Ridge, Tennessee. After three years, he returned to graduate school at the University of Tennessee to pursue a doctoral degree. While working on his dissertation, he was also employed full time at Mechanics Research, Incorporated from May, 1974 until April, 1975.

He is married to the former Angela Masini and has one son, Douglas Michael.