

AD-A237 416



DEVELOPMENT AND IMPLEMENTATION OF TIME-DEPENDENT CRACKING MATERIAL MODEL FOR CONCRETE

by

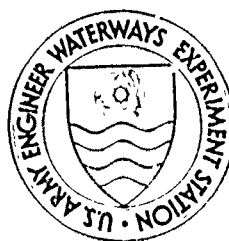
Sharon B. Garner, Michael I. Hammons

Structures Laboratory

DEPARTMENT OF THE ARMY

Waterways Experiment Station, Corps of Engineers
3909 Halls Ferry Road, Vicksburg, Mississippi 39180-6199

DTIC
ELECTE
JUL 03 1991
S c D



April 1991

Final Report

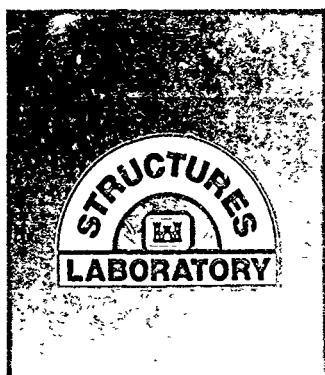
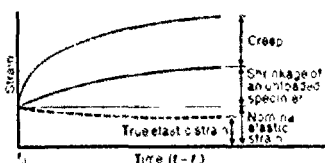
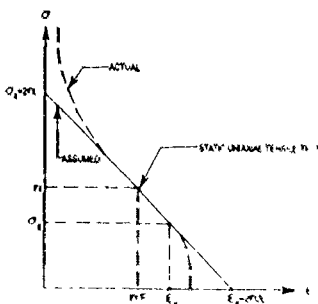
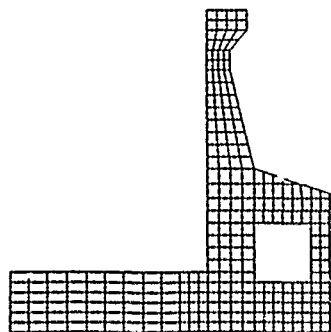
Approved For Public Release; Distribution Unlimited

91-03930



Prepared for DEPARTMENT OF THE ARMY
US Army Corps of Engineers
Washington, DC 20314-1000

91 7 02 016



Destroy this report when no longer needed. Do not return
it to the originator.

The findings in this report are not to be construed as an official
Department of the Army position unless so designated
by other authorized documents.

The contents of this report are not to be used for
advertising, publication, or promotional purposes.
Citation of trade names does not constitute an
official endorsement or approval of the use of
such commercial products.

REPORT DOCUMENTATION PAGE			Form Approved GMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE April 1991		3. REPORT TYPE AND DATES COVERED Final Report
4. TITLE AND SUBTITLE Development and Implementation of Time-Dependent Cracking Material Model for Concrete			5. FUNDING NUMBERS WU 32260	
6. AUTHOR(S) Sharon B. Garner, Michael I. Hammons				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) USAE Waterways Experiment Station Structures Laboratory 3909 Halls Ferry Road Vicksburg, MS 39180-6199			8. PERFORMING ORGANIZATION REPORT NUMBER Technical Report SL-91-7	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Army Corps of Engineers Washington, DC 20314-1000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Available from National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Minimizing thermal cracking in mass concrete continues to be a concern for the US Army Corps of Engineers. Thermal cracking is due to the restraint of volume change due to hydration, shrinkage, and creep. The prediction of stresses, strains, and cracking at early times presents special problems because many of the properties of concrete depend on the degree of hydration of the cementitious materials. To better model the time-dependent properties and response of mass concrete, a time-dependent cracking material model was developed for use in a general-purpose heat-transfer and structural analysis finite element code. The model includes the effects of time and temperature on compressive strength, elastic modulus, and creep. An interactive strain-driven, stress-modified cracking criterion based on the smeared-crack approach is included. Examples of calibration and verification of the model are included.				
14. SUBJECT TERMS Concrete research Mass concrete Thermal Cracking of concrete Strength of materials stress Creep of concrete Stresses and strains			15. NUMBER OF PAGES 120	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED			16. PRICE CODE	
18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED		20. LIMITATION OF ABSTRACT

PREFACE

The research described in this report was conducted for Headquarters, US Army Corps of Engineers (HQUSACE), under the Concrete Research Program, Work Unit 32260, Cracking of Concrete. Dr. Tony Liu, HQUSACE, was the Technical Monitor.

The research was performed by the US Army Engineer Waterways Experiment Station (WES) by personnel of the Structures Laboratory (SL), Concrete Technology Division (CTD), under the general supervision of Messrs. Bryant Mather, Chief, SL; J. T. Ballard, Assistant Chief, SL; and Kenneth L. Saucier, Chief, CTD. Direct supervision was provided by Mr. Steve Ragan, Chief, Engineering Mechanics Branch. Principal Investigators were Dr. C. Dean Norman and Mr. Michael I. Hammons, Applied Mechanics Group (AMG), Engineering Mechanics Branch, CTD. This report was prepared by Mrs. Sharon Garner, AMG, and Mr. Hammons. The authors acknowledge Messrs. Anthony A. Bombich, Donald M. Smith, Dan E. Wilson, Brent Lamb, and Mrs. Linda Mayfield, AMG, for their help during this investigation.

Commander and Director of WES during preparation of the report was COL Larry B. Fulton, EN. Technical Director was Dr. Robert W. Whalin.



Accession For	
DTIC GRA&I	☒
DTIC Tab	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

CONTENTS

PREFACE	1
CONVERSION FACTORS, NON-SI TO SI (METRIC) UNITS OF MEASUREMENT	3
PART I: INTRODUCTION	4
Background	4
Objective	5
Scope	5
PART II: SELECTION OF THE FINITE ELEMENT PROGRAM	6
Selection Criteria	6
Selection of ABAQUS	6
PART III: BASIC CONCEPTS	8
Parameters Affecting Cracking in Mass Concrete	8
Definitions	9
PART IV: MODEL DESCRIPTION	13
The Prediction of Cracking in Mass Concrete	13
UMAT Subroutine	15
PART V: CALIBRATION AND VERIFICATION OF THE MATERIAL MODEL	21
Calibration	21
Verification	26
PART VI: USING THE UMAT SUBROUTINE IN AN INCREMENTAL CONSTRUCTION ANALYSIS	30
Conducting an Incremental Construction Analysis	30
Heat-Transfer Analysis	31
Stress Analysis	34
Factor Affecting the Accuracy of the Calculations	35
PART VII: CONCLUSIONS AND RECOMMENDATIONS	41
Conclusions	41
Recommendations	42
REFERENCES	44
APPENDIX A: EXAMPLE HEAT-TRANSFER ANALYSIS DECK	A-1
APPENDIX B: EXAMPLE STRESS ANALYSIS DECK	B-1
APPENDIX C: UMAT SUBROUTINE	C-1

CONVERSION FACTORS, NON-SI TO SI (METRIC) UNITS OF MEASUREMENT

Non-SI units of measurement used in this report can be converted to SI (metric) units as follows:

<u>Multiply</u>	<u>By</u>	<u>To Obtain</u>
Fahrenheit degrees	5/9	Celsius degrees or kelvins*
feet	0.3048	metres
inches	25.4	millimetres
pounds (force) per square inch	0.006894757	megapascals

* To obtain Celsius (C) temperature readings from Fahrenheit (F) readings, use the following formula: $C = (5/9)(F - 32)$. To obtain Kelvin (K) readings, use $K = (5/9)(F - 32) + 273.15$.

DEVELOPMENT AND IMPLEMENTATION OF
TIME-DEPENDENT CRACKING MATERIAL MODEL FOR CONCRETE

PART I: INTRODUCTION

Background

1. Mass concrete structures are built in incremental layers commonly called lifts. This procedure, called incremental construction, is necessary to limit heat rise in the concrete and is further dictated by concrete batch plant capacity and by the cost of formwork. As the cement in the concrete in each lift hydrates, heat is liberated. This heat causes a temperature rise leading to a corresponding increase in volume of the concrete. However, the concrete is usually restrained by boundaries such as the lift of concrete directly beneath and by thermal gradients which exist across the lift. In addition, other mechanisms interact in a complicated fashion to cause additional volume changes. The most important of these are creep (or alternatively, stress relaxation) and shrinkage (both drying and autogenous).

2. The restraint of these volume changes leads to construction-related cracking. Although this cracking has not yet caused a catastrophic failure of a massive structure, it has led to increased maintenance and repair costs over the service life of locks, dams, bridge piers, bridge abutments, and other mass concrete structures. It appears that some investment in measures intended to provide a reduction of construction-related cracking can lead to considerable cost savings over the expected life of the structure by reducing costs associated with remedial repairs to crack-damaged structures. In addition, many of the steps which can be taken to reduce construction-related cracking can lead to substantial savings in the cost of construction. For Locks and Dams 4 & 5 on the Red River Waterway, the use of a high percentage of fly ash in the concrete mixtures resulted in a cost savings of at least \$738,000 in the cost of cementitious materials alone. Not only did the use of fly ash in higher percentages than would otherwise have been employed lead to lower temperature rises in the structures, but it also gave mechanical

properties (modulus of elasticity, creep, and shrinkage) which were beneficial in reducing cracking.

3. In 1985, Headquarters, US Army Corps of Engineers (HQUSACE), recognized that a significant research effort was required to modernize the tools available to Corps field offices to analyze and reduce construction-related cracking. As a result, Work Unit Number 32260 entitled "Cracking of Concrete" was established as a part of the Corps' Concrete Civil Works Research Program. This report is a comprehensive review of the development and implementation of the time-dependent cracking model for concrete that was developed under this work unit.

Objective

4. The objective of the research was to develop a computationally efficient, state-of-the-art material model and to implement that model in a general-purpose heat transfer and structural analysis finite element (FE) code. The model was to be capable of predicting the time-dependent changes in material properties which occur during the critical first few days after placement of concrete prior to the time it has developed stable material properties. For the model to be generally applicable and to take fullest advantage of modern supercomputing capabilities, the model was to be generalized to three dimensions.

Scope

5. This report contains a discussion of the theoretical basis of the model as well as the selection of the FE code for implementation in the model. Instructions for the calibration of the model are given. The use of the model is demonstrated in an incremental construction analysis. Examples are included.

PART II: SELECTION OF THE FINITE ELEMENT PROGRAM

Selection Criteria

6. The selection of the FE program for implementation of the time-dependent cracking model was previously discussed by Bombich, Norman, and Jones (1987). The rationale leading to the selection of the program is described in the following paragraphs.

7. Several specific criteria were established in advance for the selection of an FE code for implementation of the model. These criteria were as follows:

- a. The FE code must be capable of simulating the incremental construction process. This includes the capability to easily include lifts of concrete and to have flexibility in the selection of solution time-steps.
- b. The FE program should have a large element library from which to choose element types (both two-dimensional (2-D) and three-dimensional (3-D) elements).
- c. The FE code must allow the implementation of user-defined material models with relative ease.
- d. The program should have the capability to model significant numbers of reinforcing bars with relative ease.
- e. Because of the computational difficulty of a large, 3-D incremental construction analysis, the program must contain computationally efficient numerical solution procedures to reduce run time on the computer.
- f. Finally, the program should be user oriented and receive a high caliber of technical and scientific support from the developer and have a high potential for staying at the state-of-the-art level.

Selection of ABAQUS

8. Based upon the criteria set forth in paragraph 7, a review of FE programs was conducted. The review consisted of discussing the experiences of other analysts with various programs, reviewing technical journal articles, and meeting with representatives of both private and governmental entities directly involved with FE applications.

9. After reviewing the available FE software, the program ABAQUS was selected. ABAQUS, developed by Hibbitt, Karlsson and Sorensen, Inc. (1988), is a general-purpose heat transfer and structural analysis FE program that allows either user-selected or automated solution time-step sizing. Input is in free format, has key words, and makes use of set definitions for easy cross reference. A broad element library of both 2-D and 3-D elements is available. User-defined material models can be incorporated through the UMAT subroutine. The incremental construction problem can be simulated through the use of the MODEL CHANGE option in the code. This allows the entire structure to be modeled and then element sets corresponding to lifts to be removed prior to the first solution step. Then the element sets can be added in the appropriate time-step to model the placement of lifts in the field.

10. The current version of ABAQUS is the Version 4.7 Release. For more information on ABAQUS, the reader is referred to the ABAQUS User's Manual Version 4.7 (Hibbitt, Karlsson and Sorensen, Inc. 1988).

PART III: BASIC CONCEPTS

Parameters Affecting Cracking in Mass Concrete

11. All concrete elements and structures are subject to volume change. Cracking in mass concrete is caused by restraint of volume change. These volume changes may be due to heat generation and subsequent cooling, shrinkage, creep/stress relaxation, or other mechanisms. Restraint limits the changes in dimensions and causes corresponding tensile, compressive, or flexural stresses in concrete. Of primary concern in mass concrete structures is restraint which causes tensile stresses, particularly in the first few days after the placement of the concrete when the tensile capacity of the concrete can be quite low.

12. Restraint of volume change may be either external or internal. External restraint is caused by bond or frictional forces between the concrete and the foundation or underlying lifts. The degree of external restraint depends upon the stiffness and strength of the concrete and restraining material and upon the geometry of the section. Internal restraint is caused by temperature gradients within the concrete. The warmer concrete in the interior of the lift provides restraint as the concrete in the periphery of the lift cools due to heat transfer to its surroundings. The degree of internal restraint depends upon the quantity of heat generated, the thermal properties of the concrete, and thermal boundary conditions.

13. A number of parameters may be controlled to limit cracking related to the restraint of volume change. These parameters fall into two categories: material parameters and construction parameters. Among the material parameters are the following:

- a. Heat generation of the concrete.
- b. Mechanical properties of the concrete including strength, modulus of elasticity, and creep/stress relaxation.
- c. Shrinkage of the concrete.
- d. Thermal properties of the concrete including coefficient of thermal expansion, specific heat, and thermal conductivity.

The construction parameters are as follows:

- a. Lift height.
- b. Time between placement of lifts.
- c. Placement temperature.
- d. Ambient temperature.
- e. Use of insulation.
- f. Use of cooling coils.
- g. Monolith geometry including section thickness, monolith length, and location and size of inclusions such as galleries, culverts, etc.

14. To be effective, the method used to analyze thermal-related cracking in mass concrete structures must accurately model these complex phenomena. The heat-transfer model must be capable of handling the internal generation of heat and the complex thermal boundary conditions in the incremental construction problem. Similarly, the stress analysis model must be capable of capturing the mechanical properties of the concrete as they change with time. It must also have the ability to predict cracking in a computationally efficient manner.

Definitions

15. Some of the terms used in this report may be unfamiliar to some readers. Therefore, the following definitions have been included.

Adiabatic temperature rise curve

16. The adjective adiabatic refers to a condition in which heat neither leaves nor enters a system. The adiabatic temperature rise curve describes the rise in temperature with time that occurs during hydration of the cement in a specimen in which no heat loss is allowed to occur. This serves as the loading in the heat-transfer analysis.

Creep

17. Creep is defined by American Concrete Institute (ACI) Committee 209 (1990) as "time-dependent increase in strain in hardened concrete subjected to sustained stress" (ACI 1990). Creep strain is obtained in the laboratory by subtracting from the total measured strain in a loaded specimen the sum of:
(a) initial instantaneous (usually considered elastic) strain due to the

sustained stress, (b) shrinkage, and (c) and thermal strain in an identical load-free specimen which is subjected to the same history of relative humidity and temperature conditions.

18. The above definition assumes that strain in a loaded specimen consists of an initial elastic strain, creep strain, shrinkage and thermal expansion or contraction. In a mass concrete structure, however, stresses and moduli are varying with time throughout the structure and construction period, and initial elastic strain has little meaning. Calibration of the material model must be based on time-dependent modulus and creep. The relationship between elastic strain (ϵ^e) and creep strain (ϵ^c) is shown in Figure 1.

Creep compliance

19. Creep compliance is determined from a plot of specific strain (strain per unit stress) versus time from a 3-day creep test and is the difference between the total specific strain and the elastic specific strain. The relationship between total specific strain $J(t)$, creep compliance ($C(t)$), and elastic specific strain ($1/E(t)$) is shown in Figure 2.

DFLUX subroutine

20. DFLUX is a user-supplied FORTRAN subroutine used to specify non-uniform distributed fluxes in an ABAQUS heat-transfer analysis. DFLUX is used to define adiabatic curves for one or more concrete mixtures in the heat-transfer analysis.

Incremental construction

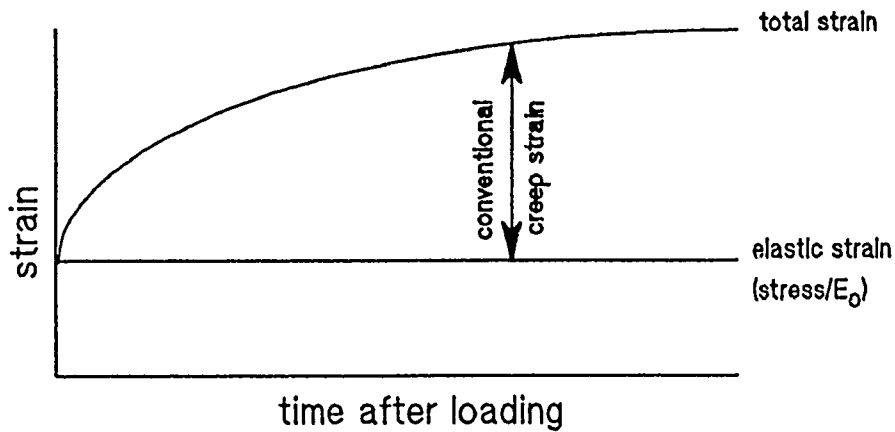
21. Incremental construction is the practice of placing concrete in lifts (or layers). Most mass concrete structures are constructed in lifts (usually 5 to 10 ft in depth) placed at time intervals of several days.

Shrinkage

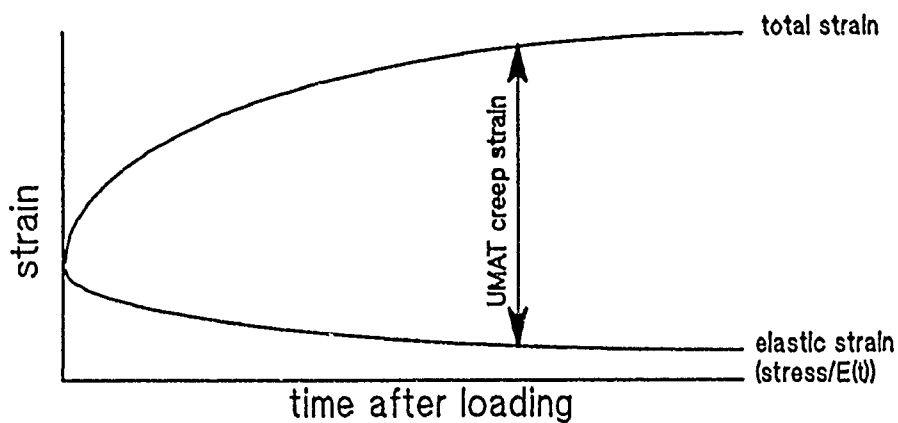
22. ACI defines shrinkage as "decrease in either length or volume" (ACI 1990). The decrease is due to changes in the moisture content of the concrete and physico-chemical changes which occur without stress attributable to actions external to the concrete. Shrinkage due to moisture loss or drying shrinkage occurs only at the surface of mass concrete structures and is not simulated in the material model. However, additional volumetric changes occur during hydration of the cement that are not directly attributable to changes in temperature. In this report shrinkage refers to these volumetric changes.

UMAT subroutine

23. This refers to a user-supplied material model, usually in the form of a FORTRAN subroutine, which can be linked to ABAQUS. External parameters required by UMAT are input using the USER SUBROUTINE key word in the ABAQUS input file.



a. Elastic strain not varying with time



b. Elastic strain varying with time

Figure 1. Relationship between elastic and creep strains

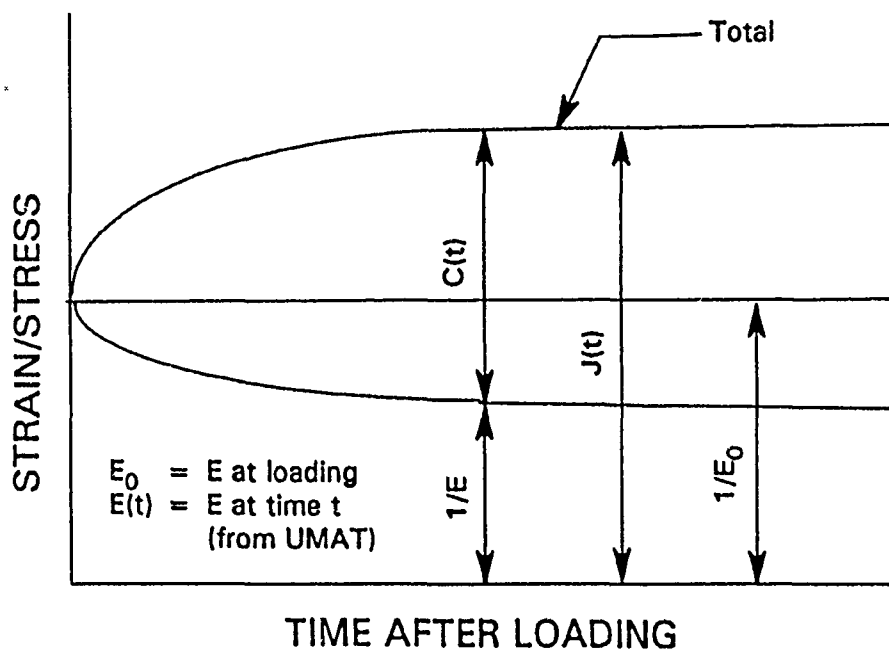


Figure 2. Specific strain relationships

PART IV: MODEL DESCRIPTION

The Prediction of Cracking in Mass Concrete

24. The prediction of stresses, strains, and cracking at early times presents special problems, because many of the properties of concrete depend on the degree of hydration of the cementitious materials. The rate of hydration of the cementitious materials is affected by the type of materials used and by the temperature and moisture history during the period of hydration. At the same time, the internal environment of mass concrete is affected by the hydration of the cementitious materials. Elevated temperatures generated by hydration are maintained for long periods of time in the center of mass concrete structures and affect mechanical properties essential in determining the stress/strain condition of the concrete such as elastic modulus, compressive strength, creep, and volumetric changes associated with hydration.

25. Requirements for the accurate prediction of stresses, strains, and cracking in mass concrete include the following:

- a. An FE grid that accurately defines the structure. The grid may consist of 2-D plane strain, plane stress, axisymmetric elements, or 3-D elements. The choice of elements and geometry must be based on an understanding of the problems to be studied.
- b. Accurate information about thermal boundary conditions. This includes climatic data such as expected temperatures and wind velocities during the construction period. Also, accurate information about the thermal properties of foundation material is needed to establish heat flow from the structure into the foundation.
- c. Accurate thermal and mechanical properties of the concrete.
- d. An FE code that incorporates an accurate, reliable heat-transfer capability allows relatively easy incorporation of a concrete constitutive model and is capable of modeling the incremental construction procedures characteristic of mass concrete construction.
- e. A material model capable of handling the time- and temperature-dependent properties of concrete and capable of predicting and monitoring cracking in a time- and cost-efficient way.

26. Development of an adequate FE grid is the responsibility of the analyst. It should be undertaken only with a thorough knowledge of the problems to be studied and the tools available for this study.

27. Adequate thermal and mechanical properties data are essential. Often an analyst will try to base an FE study on general concrete properties. However, mass concrete mixtures are usually developed to fulfill specific requirements and may employ diverse chemical admixtures, unusual cements, replacement of cement with fly ash or other pozzolans, and diverse aggregate types. Any one of these can affect the thermal and mechanical properties of the concrete. Also, it is necessary to know how the mechanical properties change with time, beginning immediately after time of final set.

28. Necessary thermal properties include the adiabatic temperature rise thermal conductivity, and specific heat of the concrete. These properties can vary with changes in environmental temperature. Significant variations with temperature should be considered prior to selecting final properties for the analysis.

29. Necessary mechanical properties include time- and temperature-dependent properties such as creep, elastic modulus, compressive strength, and shrinkage. Additional properties commonly required are tensile strain capacity and coefficient of thermal expansion.

30. Many FE programs include a heat-transfer capability and a means for including an adiabatic heat-rise curve as the driving function for a heat-transfer analysis. However, few of these codes are designed to allow easy modeling of incremental construction. Failure to correctly simulate incremental construction in the analysis will result in incorrect predictions.

31. A user-defined, time-dependent material model with cracking capabilities (UMAT) has been developed by the US Army Engineer Waterways Experiment Station (WES) for implementation in ABAQUS through the ABAQUS-UMAT format. The model includes the effects of time and temperature dependency on elastic modulus, compressive strength, and creep. Cracking is included using a smeared-crack approach. Although this approach to cracking does not allow the study of specific cracks, it gives a general indication of when and where cracking is likely to occur without causing the calculations to become too expensive and time consuming. Important features of the model are discussed in the following paragraphs.

UMAT Subroutine

32. The mathematical representation of the time- and temperature-dependent properties of concrete must address three fundamental properties of the material: elastic modulus $E(\tau, T)$, ultimate strength $\sigma_u(\tau, T)$, and creep compliance $C(t, \tau; T)$, where t is current time measured from some reference time, t_0 , τ is the time since placement of the concrete, and T is temperature. For an arbitrary stress history $\sigma(\tau)$ and temperature history $T(\tau)$, the stress-strain relationship for an isotropic time-dependent concrete can be written using tensor notation as

$$\Delta \underline{\epsilon}(t) = \underline{B} \left\{ \frac{\Delta \underline{\sigma}}{E(t)} - \frac{\underline{\sigma} \Delta E(t)}{E^2(t)} + \Delta t \int_{-\infty}^t C'[t, \tau; T(\tau)] \frac{\partial \underline{\sigma}(\tau)}{\partial \tau} d\tau \right\} \quad (1)$$

where $\underline{B}$ is a material tensor function of Poisson's ratio. Creep compliance is given by

$$C'[t, \tau; T(\tau)] = \frac{\partial C[t, \tau; T(\tau)]}{\partial t} \quad (2)$$

33. The time difference form of Equation 1 is

$$\Delta \underline{\epsilon}_n = \underline{B} \left\{ \frac{\Delta \underline{\sigma}_n}{E_n} - \frac{\underline{\sigma}_n \Delta E_n}{E_n^2} + \Delta t \sum_{i=0}^{n-1} \int_{\tau_i}^{\tau_{i+1}} C'[t_n, \tau; T(\tau)] \frac{\partial \underline{\sigma}(\tau)}{\partial \tau} d\tau \right\} \quad (3)$$

In this form, the integral at each time-step t_n must be totally reevaluated from t_0 to t_n . This is due to the time-dependent nature of creep and results in unnecessarily expensive calculations.

34. For a nontime-dependent material and neglecting temperature, Equation 3 becomes

$$\Delta \underline{\epsilon}_n = \underline{B} \left\{ \frac{\Delta \underline{\sigma}_n}{E_0} + \Delta t \int_{-\infty}^t C'(t-\tau) \frac{\partial \underline{\sigma}(\tau)}{\partial \tau} d\tau \right\} \quad (4)$$

where E_0 is the elastic modulus at time of loading.

35. Using exponential functions, the nontime-dependent creep compliance can be written in a form that allows separation of the variables t and τ . For

$$C'(t-\tau) = \sum_{i=1}^m a_i (1 - e^{b_i(t-\tau)}) \quad (5)$$

Equation 4 becomes

$$\Delta \epsilon(t) = B \left[\frac{\Delta \sigma}{E_0} + \Delta t \sum_{i=1}^m a_i b_i e^{b_i t} \int_{-\infty}^t e^{b_i \tau} \frac{\partial \sigma(\tau)}{\partial \tau} d\tau \right] \quad (6)$$

The integration of this equation requires only the summation of time-steps.

36. A more generalized form of Equation 5 may be written for a time-dependent material as follows:

$$C'(t-\tau) = \sum_{j=1}^n \sum_{i=1}^m a_i (1 - e^{b_i(t-\tau)}) c_j (1 - e^{d_j \tau}) \quad (7)$$

The time-dependency of creep can then be easily evaluated using the elastic modulus as it is not under the integral sign.

37. Creep properties in the UMAT model are defined by a 3-day creep compliance curve mapped in the time domain by an "aging factor." This aging factor is the ratio of the elastic modulus at the current age to the 3-day elastic modulus. The curve is based on a 70 °F* temperature and modified for current temperature by a temperature factor. The creep equations are given in Equations 8a, 8b, and 8c.

$$C(t, \tau; T) = \sum_{i=1}^n A_i(\tau, T) (1 - e^{\tau_i t}) + D(\tau, T) t \quad (8a)$$

$$A_i(\tau, T) = A_{oi} \left(\frac{e^{-\frac{Q}{RT}}}{e^{-\frac{Q}{RT_0}}} \right) \left[\frac{E(3)}{E(\tau)} \right]^2 \quad (8b)$$

* A table of factors for converting non-SI units of measurement to SI (metric) units is presented on page 3.

$$D(\tau, T) = D_o \left(\frac{e^{-\frac{Q}{RT}}}{e^{-\frac{Q}{RT_o}}} \right) \left[\frac{E(3)}{E(\tau)} \right]^2 \quad (8c)$$

where

- C = creep compliance (strain per unit stress)
- t = time since loading, days
- τ = age of the concrete, days
- T = temperature, °K
- R = gas constant, 1.98
- $E(\tau)$ = modulus of elasticity at age τ
- r_1, A_1, D, Q = constants for a given material
- Q = the activation energy for creep, 4,345.
- T_o = 294 °K (70 °F)

38. The form of the equation for elastic modulus as a function of time is similar to that of Equation 8 and is given in Equation 9.

$$E(\tau) = E(1) + \sum_{i=1}^2 B_i [1 - e^{-m_i(\tau-1)}] + B_3(\tau-1) \quad (9)$$

where

- τ = total age of the concrete in days
- B_i, m_i = constants
- $E(1)$ = 1-day modulus, psi
- $E(\tau)$ = modulus calculated at 70 °F, psi

The elastic modulus from time of placement to 1 day is assumed to be linear from $E = 0$ at $t = 0$ to $E = E(1)$ at $t = 1$ day. Little data exist to verify this assumption. However, since stresses due to temperature changes during hydration are generally low at very early times, early-time errors in modulus may not produce significant errors in the calculations.

39. The effect of temperature on elastic modulus is accounted for by the temperature factor, $H(T)$ as follows:

$$E(\tau, T) = E(\tau) H(T) \quad (10a)$$

where

$$H(T) = \frac{E(T, 28\text{days})}{E(70^\circ F, 28\text{days})} \quad (10b)$$

Ultimate strength $\sigma_u(\tau)$ is calculated using a 3-day reference value and the age factor:

$$\sigma_u(\tau) = \sigma_u(3) \left[\frac{E(\tau)}{E(3)} \right] \quad (11)$$

Shrinkage as a function of age of the concrete is given by the following equation.

$$\epsilon^s(\tau) = C_1(1 - e^{s_1\tau}) + C_2(1 - e^{s_2\tau}) \quad (12)$$

where C_1 , C_2 , s_1 , and s_2 are constants. The tensile strain capacity (ϵ_t), if not defined by the user as a constant, is assumed to be 10 percent of the absolute value of the compressive strain at ultimate strength.

40. Cracking is assumed to occur when a cracking criterion is satisfied. This criterion is strain-driven but is modified by stress. The crack surface normally is in the direction of the principal strain and the cracking criterion is interactive. Figures 3 and 4 have been included to illustrate the cracking criterion. For an isotropic material, such as concrete prior to cracking, the principal strain and stress directions coincide, and the cracking criterion can also be expressed in terms of principal strain.

41. If a cube of concrete is loaded with σ_2 , the cube will split in the direction of the load under the effect of the ϵ_1 strain and σ_1 will be zero. If cracking is based on stress only, the ϵ_1 strain is $-\nu\epsilon_2 = -\nu(\sigma_2/E)$. For $\sigma_2 = f'_c$ and $\nu = 0.2$, the cracking strain is 20 percent of the uniaxial ultimate compressive strain, or twice the value usually assumed. Obviously a strain-dependent criterion is more appropriate.

42. If a small σ_2 is applied and sustained over a long period of time so that creep occurs, cracking could eventually occur under a strain of $\epsilon_1 = -\nu(\sigma_2/E) + \epsilon_1^c$. This indicates a gain in tensile strain capacity under creep. Little creep-cracking data are available, but cracking strain for a specimen undergoing creep appears to be approximately twice that obtained from a uniaxial tensile test (Rashid and Dunham in preparation). A strain-

dependent cracking criterion will not predict this gain, so an interactive criterion was adopted to accommodate creep and stress relaxation as well as elastic effects. This criterion is illustrated in Figure 3 by the diagonal line that crosses the stress axis at $2f'_t$ (where f'_t is the uniaxial tensile strength) and the strain axis at $2f'_t/E$. Some common test conditions are indicated on the figure, with a uniaxial tension test located at the midpoint. The actual curve, indicated by the broken line, is asymptotic to the stress axis, since cracking under zero strain is impossible for compressible materials. The criterion is implemented in the model as follows.

- a. Calculate maximum principal strain ϵ_1 in UMAT.
- b. Enter Figure 3 with ϵ_1 and calculate σ_f .
- c. Adjust the failure surface amplitude, using σ_f as the intercept instead of f'_t .
- d. Enter Figure 4 using the principal stresses σ_1 and σ_2 calculated in UMAT, and determine whether the (σ_1, σ_2) point penetrates the failure surface. If so, introduce a crack normal to the principal strain direction and formulate the constitutive matrix in the principal coordinate system.
- e. Rotate the precracking stresses to the principal coordinate system and adjust these stresses to reflect the new cracking state.
- f. Rotate the constitutive matrix and the stresses back to the coordinate system of the structure.
- g. The new constitutive matrix and stresses are then used by ABAQUS to calculate the nodal forces and the tangent stiffness matrix in the next step. A smeared crack approach is used to model the cracked regions of the structure. The cracked region is modeled as an anisotropic continuum effectively "smearing" the cracks in a continuous manner throughout the element (Norman and Anderson 1985). When cracking occurs, stress in the tensile direction is allowed to drop to zero while shear transfer due to aggregate interlock is maintained. Cracks are allowed to open or close as conditions in the model vary. Thus, the overall structural response can be modeled adequately without regard to completely realistic crack patterns and local stresses (Chen 1982).

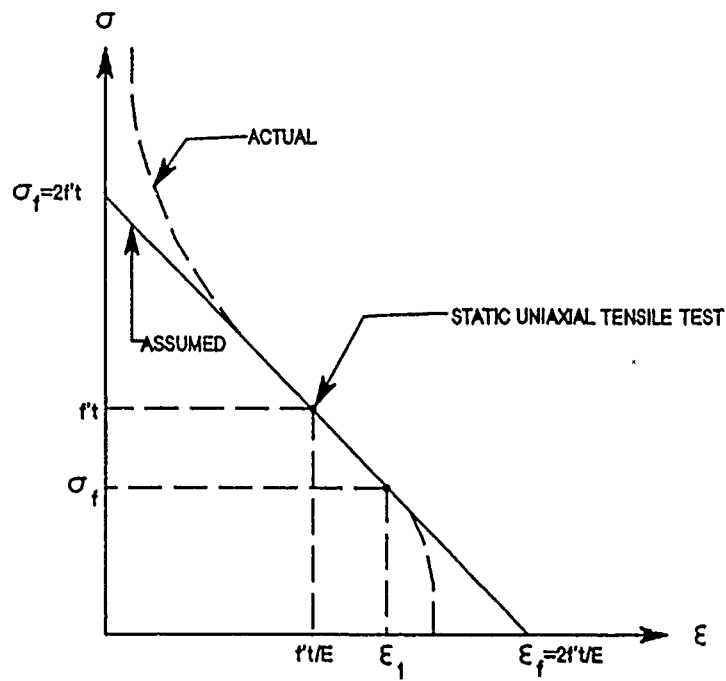


Figure 3. UMAT interactive cracking criterion

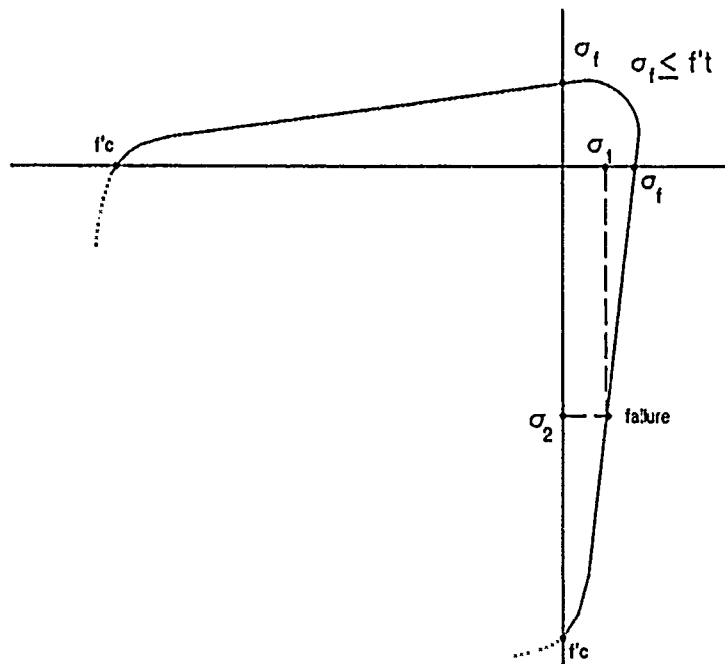


Figure 4. Typical biaxial tensile failure surface for concrete

PART V: CALIBRATION AND VERIFICATION OF THE MATERIAL MODEL

Calibration

43. The UMAT model used in the analysis must be calibrated for each concrete mixture to be simulated in an analysis. Information required for calibration includes 3-day creep compliance, shrinkage, and elastic modulus as a function of time. Each of these are discussed in the following paragraphs.

Creep Compliance

44. Creep compliance is determined from a plot of specific strain (strain per unit stress) versus time from a 3-day creep test and is the difference between the total specific strain and the elastic specific strain (Figure 2). Creep compliance as is given by an equation of the form

$$C(t) = A_1(1 - e^{-r_1 t}) + A_2(1 - e^{-r_2 t}) + A_3(1 - e^{-r_3 t}) + A_4 t \quad (13)$$

where t is time since loading in days, $C(t)$ is in units of inches per inch per pound per square inch. One or more of the exponential terms in Equation 13 may be eliminated as required to improve the fit to test data. The parameters A_1 , A_2 , A_3 , A_4 , r_1 , r_2 , and r_3 are determined by trial and error fit to test data.

45. An example curve fit is given below. Although an adequate curve fit can usually be obtained using only two or three terms, all four terms have been used in the example. Test specific creep values for days given are listed in Table 1.

Table 1
Example of Test Specific Creep

<u>Time Since Loading</u> <u>days</u>	<u>Specific Creep</u> <u>10^{-6} in./in./psi</u>
1	0.10
3	0.15
7	0.20
28	0.25
90	0.26

46. Parameters r_1 , r_2 , and r_3 are determined so that the terms $e^{r_i t}$ saturate (i.e. are set equal to a very small number) at three different times. In this example 1, 3, and 28 days were chosen. If the terms $e^{r_i t}$ are set equal to 0.005 at those days, then

$$\begin{aligned} r_1 &= (\ln 0.005)/28. = -0.189226 \\ r_2 &= (\ln 0.005)/3. = -1.76611 \\ r_3 &= (\ln 0.005)/1. = -5.29832 \end{aligned}$$

47. Substituting these values into Equation 13 for ages of 1, 3, 28, and 90 days and setting each equation equal to the specific creep at that age yields four equations with four unknowns. Solving for A_1 , A_2 , A_3 , and D

$$C(t) = [0.17116(1 - e^{-0.18923t}) + 0.029166(1 - e^{-1.76611t}) + 0.04640(1 - e^{-5.29832t}) + 0.00015t] \times 10^{-6} \quad (14)$$

This equation is plotted against test values in Figure 5. It should be noted that UMAT requires that the number of terms in the creep equation be specified in the subroutine STRN3D. This is done by setting the integer variable $JCREEP = N$ in the subroutine STRN3D, where N is the number of terms in the expression for $C(t)$.

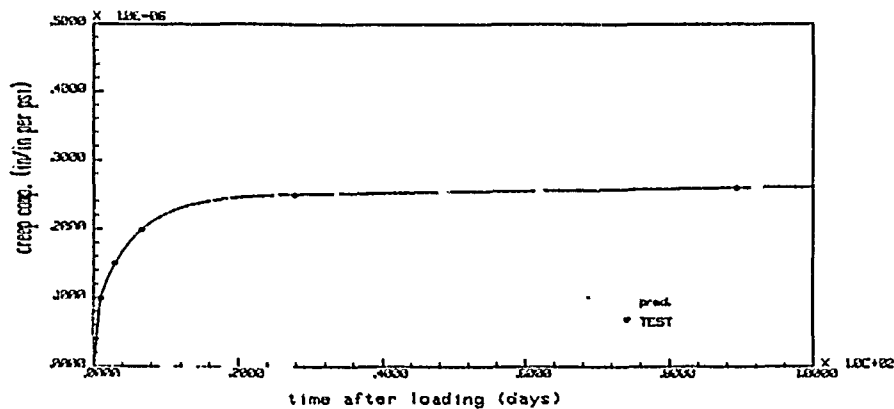


Figure 5. Test specific creep and creep compliance predicted by Equation 14

48. For concretes with extremely low moduli at early times, some modification to the creep aging factor may be required at times when $E(3)/E(\tau)$ is greater than 1. The effects of varying the creep aging factor on early-time creep predictions can be seen in Figure 6. In this figure the results of UMAT creep predictions are plotted against 1-day test results for a high-fly-ash mass concrete mixture. In Run 1, a creep aging factor of $[E(3)/E(\tau)]^2$ was used throughout, and predicted strains were roughly three times as high as test strains. In Run 2, a creep aging factor of $E(3)/E(\tau)$ was used prior to $\tau = 3$ days, producing reasonable results.

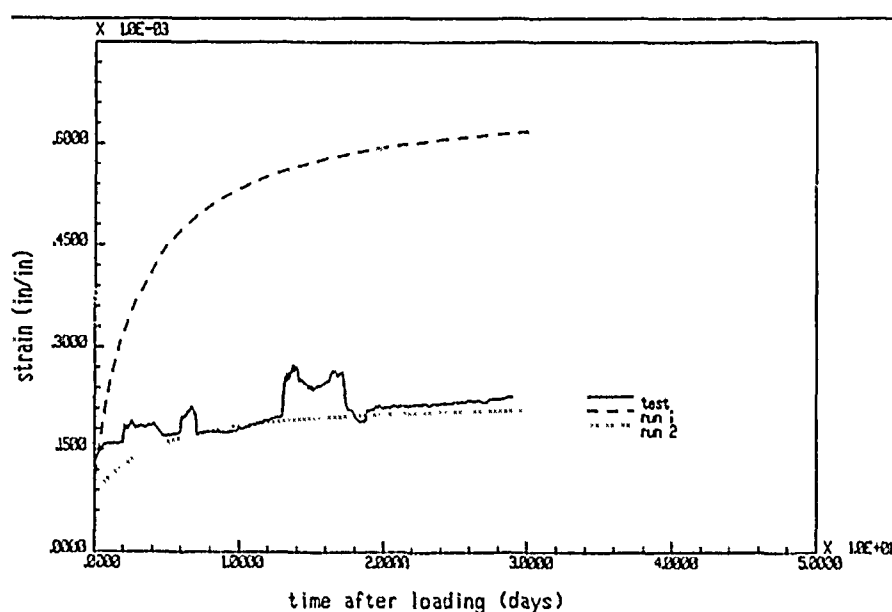


Figure 6. Two different aging factors used in 1-day creep prediction

Elastic Modulus

49. The form of the time-dependent elastic modulus equation is

$$B_1[1 - e^{x_1(t-1)}] + B_2[1 - e^{x_2(t-1)}] + B_3[1 - e^{x_3(t-1)}] + B_4(t-1) + E(1) \quad \text{for } t \geq 1 \quad (15)$$

The constants B_1 and x_1 are determined using a procedure similar to that used in Equation 14. Data for calibration of the elastic modulus curve are obtained from unconfined compression tests on specimens stored at 73 °F.

Although the form of the specific creep equation must be maintained, the form of Equation 15 may be changed by the user if desired.

Shrinkage

50. To account for the volumetric changes that occur during the curing of concrete, the UMAT material model includes a shrinkage equation of the form

$$\epsilon^s = [204.9(1 - e^{-0.15r}) + 145.1(1 - e^{-0.02263r})] \times 10^{-6} \quad (16)$$

where ϵ^s has units of inches per inch and r is time since placement in days. This relationship was developed from test data on silica fume concrete and will predict excessive shrinkage for most mass concrete mixtures.

51. Shrinkage data obtained from sealed specimens for a period of time extending from time of setting until change in strain with time become negligible. These data may be used to develop a new curve or to determine a factor for Equation 16.

52. Additional parameters required by the model are given in the USER MATERIAL statement and are listed in Table 2.

53. Tensile strain capacity, ϵ_t , can be entered as a constant in the USER MATERIAL statement or calculated by the program as 10 percent of the absolute value of compressive strain at ultimate strength. A report by Holland, Liu, and Bombich (1982) on the properties of concretes for Lock and Dam No. 2, Red River Waterway, gives some insight into the appropriate choice. Ultimate strain capacity tests using 12- by 12- by 66-in. beams were run for two mixtures, one with a design compressive strength of 3,000 psi at 28 days and the other with a design compressive strength of 3,000 psi at 90 days. Loading rates of 40 psi/min and 25 psi/week were used. For the higher-strength mixture under rapid loading, average tensile strain capacity varied little after 3 days, but average test capacity at 1 day was only 50 millionths as opposed to 80 millionths at 3 days. For the lower-strength mixture, tensile strain capacity under rapid loading varied from an average of 41 millionths at 1 day to an average of 91 millionths at 90 days. Tensile strain capacities for all specimens loaded at the slower rate were well over 100 millionths regardless of age at loading.

Table 2
Other Concrete Parameters Required For Material Model

<u>Parameter</u>	<u>Notes</u>
$E(3)$	3-day elastic modulus in pounds per square inch.
ν	Poisson's ratio. This is assumed to be a constant.
$\sigma_u(3)$	Ultimate strength at 3 days in pounds per square inch.
ϵ_f	Tensile strain capacity.
α	Coefficient of linear thermal expansion.
T_0	Reference temperature in °F. This is the temperature at zero stress. For incremental construction problems, this is the placement temperature.
age	Concrete age at the start of the calculation in days. For incremental construction problems, this is the age at time of setting.
F_s	Shrinkage factor. This used to factor the shrinkage curve in UMAT.
F_c	Creep factor. This used to factor the creep curve in UMAT.
ϵ_0	Initial strain (inches/inch). This is strain existing at the start of the analysis (usually 0).
t_{ref}	The reference time (in days) is used in incremental construction problems and corresponds to the day of placement.
IPRUM	This should be set equal to 2.
IHANOP	This should be set equal to 0.

54. Mass concrete mixtures generally have low strengths at early times and gain stiffness slowly. This means that tensile stresses due to the restraint of strains are induced fairly slowly. Also, cooling, the method by which loads are applied, is a slow process. Because of these factors the constant tensile strain capacity used in the analyses (100 millionths) should be adequate for most mass concrete mixtures. However, some construction procedures and concrete mixtures may result in rapid gains in tensile stress and may require the use of a tensile strain capacity linked to concrete strength and stiffness.

55. The equations necessary to calibrate the model are found in the subroutines listed in Table 3.

Table 3
Location of Equations

<u>Term to be Calculated or Applied</u>	<u>Subroutine Name</u>	<u>Variable Name</u>
$E(\tau)$	COEF	ETATAU, ETA3
$C(t)$	SHIFT1	A,R,D
ϵ^s	USHRNK	SHRNK
$[E(3)/E(\tau)]^2$	CRPROP	-

Verification

56. The UMAT model, incorporating the above algebraic expressions calibrated with test data, is then used with ABAQUS to simulate the entire suite of creep tests for each mixture. Tests normally used for verification of the model are 1-, 7-, 14-, and possibly 28-day creep tests. Each creep cylinder is modeled using a single axisymmetric element supported on rollers at boundaries and uniformly loaded across the top surface (Figure 7). Loads are varied to simulate test loadings. Axial strains from these runs are then plotted against actual test data for comparison. Results of the verification

analyses for a typical concrete mixture are shown in Figure 8. The equations used in UMAT to model the modulus and creep compliance are given in Equations 17 and 18.

$$E(t) = 1.8012 \times 10^6(1 - e^{-0.031351(t-1)}) + 2.1453 \times 10^6(1 - e^{-0.407563(t-1)}) - 0.437477(1 - e^{-2.649(t-1)}) + 2.25 \times 10^6 \quad (17)$$

$$C(t) = 0.10576 \times 10^{-6}(1 - e^{-0.05887t}) + 0.1589 \times 10^{-6}(1 - e^{-0.189226t}) + 0.13887 \times 10^{-6}(1 - e^{-1.7661t}) \quad (18)$$

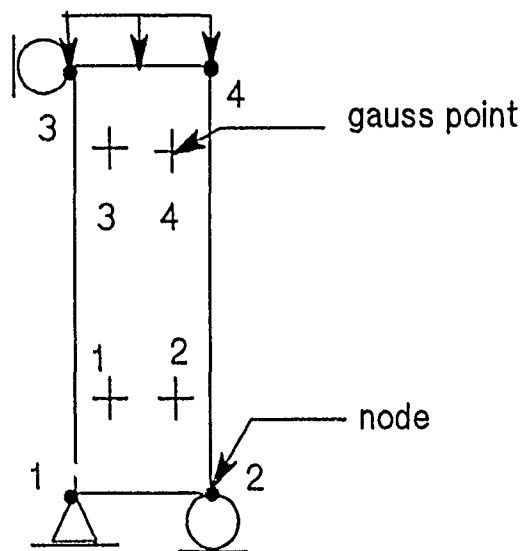
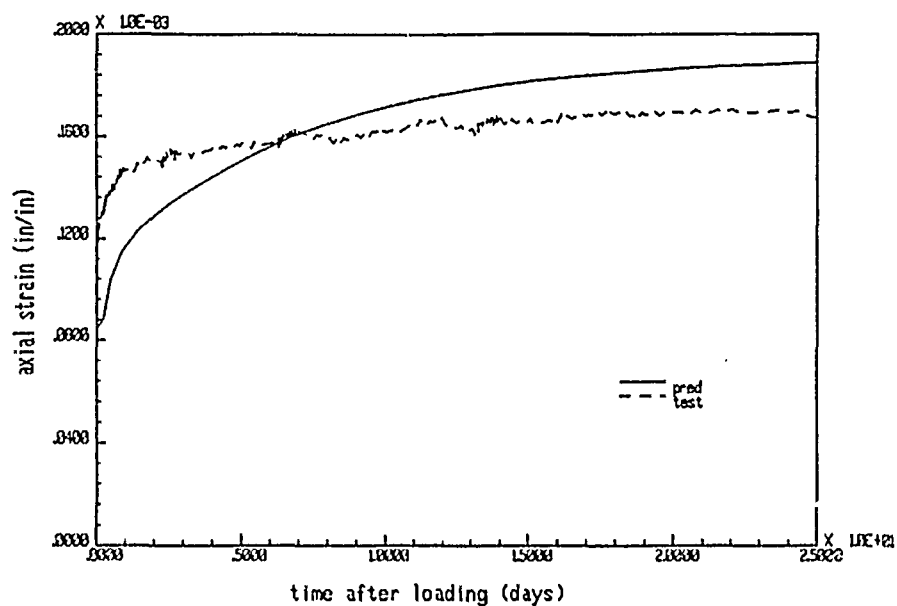
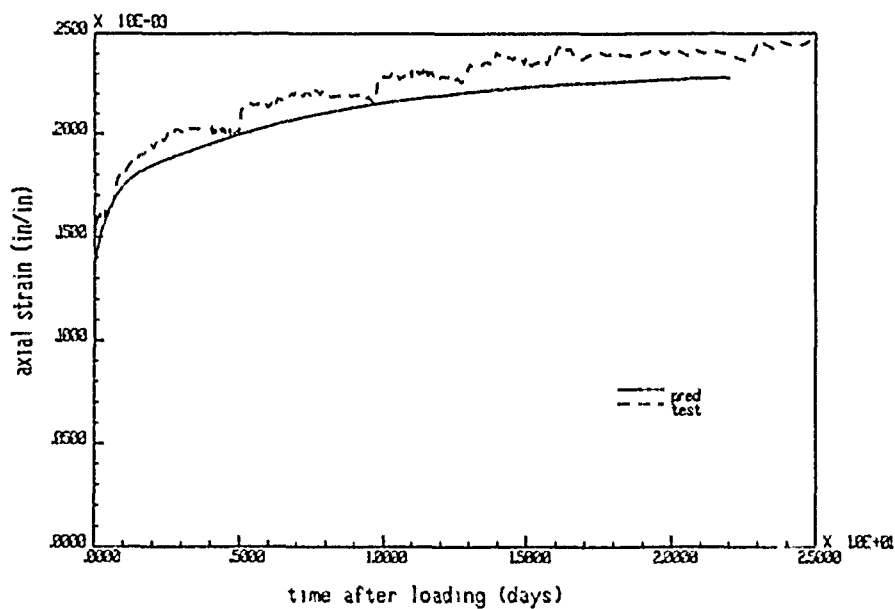


Figure 7. Creep cylinder simulation for FE analysis

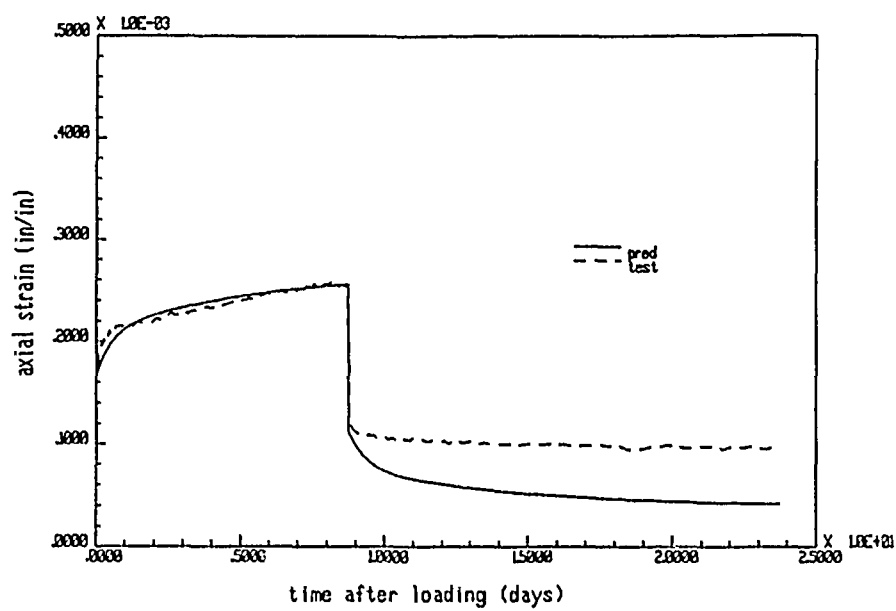


a. Creep test--18 hr

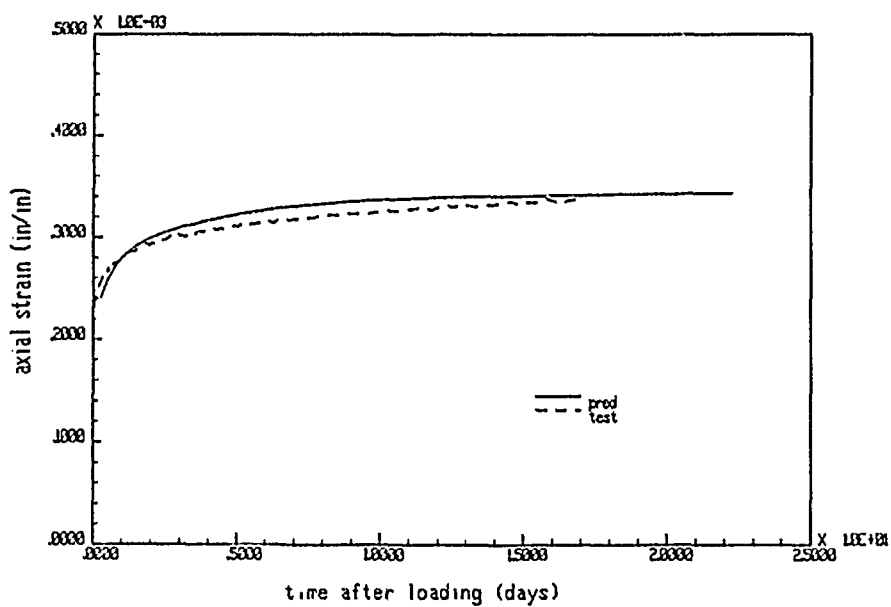


b. Creep test--3 days

Figure 8. Results of verification analyses (Continued)



c. Creep test--7 days



d. Creep test--14 days

Figure 8. (Concluded)

PART VI: USING THE UMAT SUBROUTINE IN AN INCREMENTAL CONSTRUCTION ANALYSIS

Conducting an Incremental Construction Analysis

57. The incremental construction analysis of a mass concrete structure is a two-part procedure. First, a heat-transfer analysis must be performed to determine the temperatures throughout the structure during the construction period. The output from the heat-transfer run in the form of temperatures at nodes is then used as the loading in a stress run. In this section, the mechanics of performing heat transfer and stress analyses pertaining to mass concrete structures are briefly discussed, example input files are developed, factors affecting the accuracy of predictions are discussed, and comparisons are made between 2- and 3-D results.

58. The incremental construction analysis procedure developed at WES uses ABAQUS in conjunction with user-defined, time-dependent routines to define applied heat flux and mechanical behavior of the material. The procedure has been used in several previous projects (Bombich, Norman, and Jones 1987; Norman, Campbell, and Garner 1988; Hammons, Garner, and Smith 1989; Garner, Hammons, and Bombich in preparation). Some of the features of the finite element code are discussed in the following paragraphs.

59. To model the incremental construction, calculations are carried out in time-steps. Using the REMOVE/INCLUDE element options in ABAQUS, new elements are added to the model at regular intervals of time (5, 10, or 15 days) to simulate the placement of additional lifts.

60. The 2- or 3-D transient heat-transfer analysis is performed using heat-transfer elements from the ABAQUS library of elements. The adiabatic temperature rise of the concrete mixture is the loading for the analysis and is supplied by the user in an external subroutine (DFLUX) linked to ABAQUS. Boundary conditions for the heat-transfer analysis are easily varied. External conditions (wind speed, forms, insulation) are modeled using film coefficients applied to external element faces, and ambient and placement temperatures are specified in the input file. The results of the heat-transfer analysis are temperatures at each node for each time increment. The temperature-time history obtained in the heat-transfer analysis is used as the

loading in a stress analysis. This analysis can be conducted using plane stress, plane strain, or 3-D elements from the ABAQUS element library. Time-dependent material characteristics (strength, elastic modulus, creep, and shrinkage), as well as cracking, are incorporated into the calculations using the user-supplied material model, UMAT. The output from the stress analyses includes nodal displacements and stresses and strains at user-selected locations throughout the structure as well as user-selected displacement plots and stress or strain contour plots.

Heat-Transfer Analysis

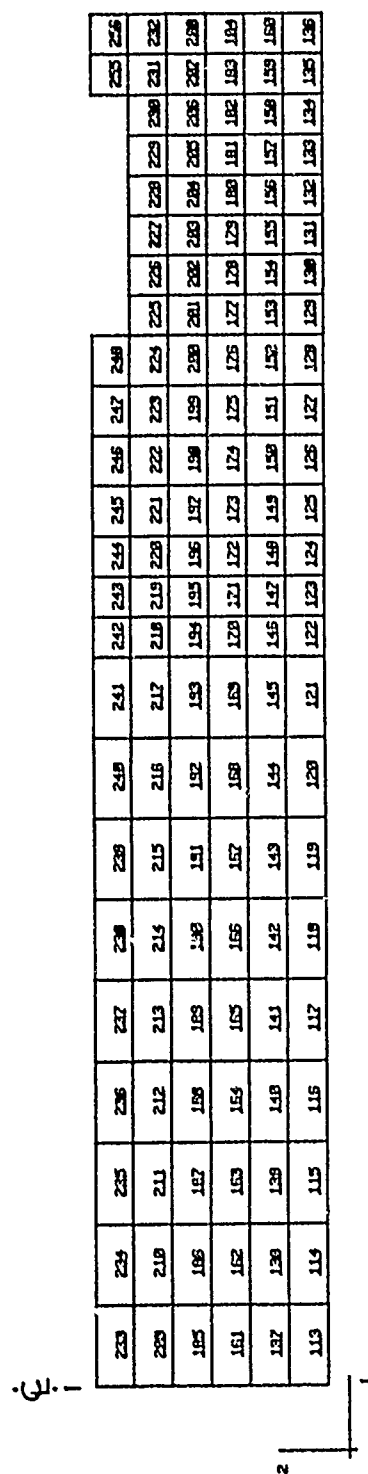
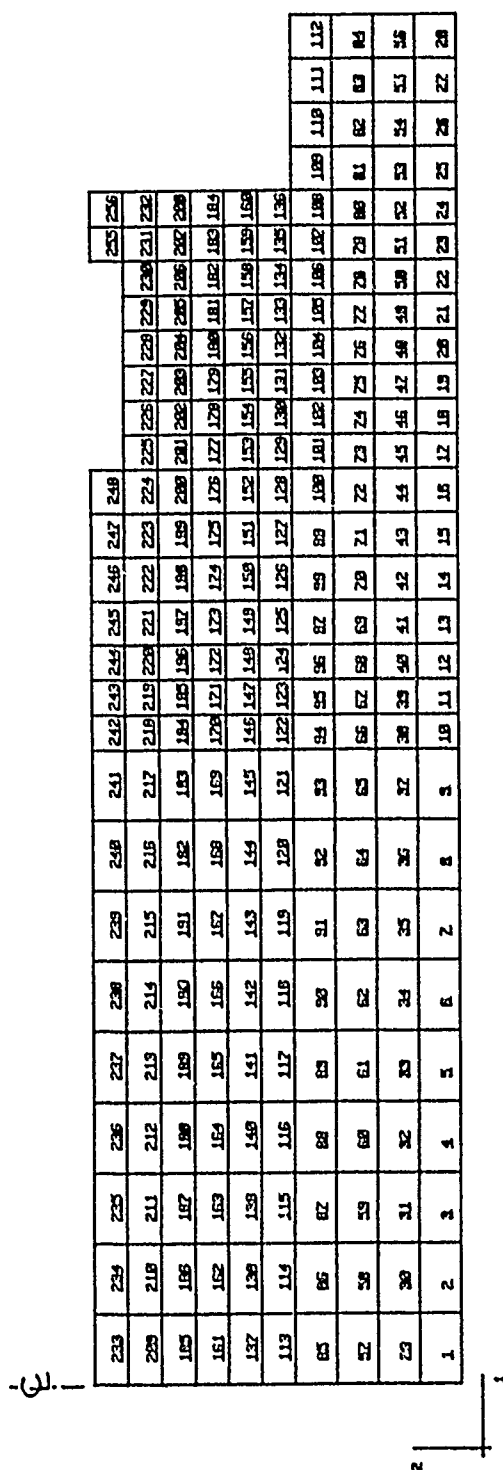
61. Before an input file for the heat-transfer analysis can be generated, the following information must be obtained.

- a. Geometry of the sections to be analyzed.
- b. Annual cycle of average ambient temperatures for the area.
- c. Depth at which soil temperatures remain constant (usually 10 to 20 ft) and the temperature at that depth.
- d. Thermal properties of the soil and concrete (density, specific heat, and conductivity). If any voids are to be included in the analysis, thermal properties must also be determined for air.
- e. Adiabatic curve for each concrete mixture to be simulated.
- f. Expected lift height and placement schedule.
- g. Variables necessary for calculating film coefficients, such as type of formwork and insulation to be used, times for formwork removal, insulation requirements, average wind speed for the area, etc.

62. A sample input file for a 2-D analysis of a lockwall monolith floor and the corresponding DFLUX subroutine are presented in Appendix A. Lift height for this structure was 4 ft, and three lifts were placed at 10-day intervals. Soil was included for a depth of 10 ft below the base of the structure and 10 ft beyond the outer edge. Further information for setting up an input file can be found in the ABAQUS User's Manual Version 4.7 (Hibbitt, Karlsson and Sorensen, Inc. 1988). The structure is shown in Figure 9, and the FE grid is shown in Figure 10.



Figure 9. Typical chamber monolith structure



63. Results from the heat-transfer run were stored in an output file with the extension ".FIL". This is normally a binary file, but can be output in ASCII if requested by the user in the NODE FILE command. This file was then used as the loading for a stress analysis.

Stress Analysis

64. The UMAT subroutine is called by the *USER MATERIAL keyword in the input data file. The material name is given by Mxx where xx is a number from 1 to 99. This allows the material name to be easily converted to an integer in UMAT, which facilitates the modeling of more than one material in the subroutine. The number of material parameters required for the model is 13, and the parameters are listed in the next two lines, with eight parameters per line. A listing of parameters is given in Part III of this report. The number of solution state variables required by the model (specified using the *DEPVAR keyword) is 57. An example call to a user material model is illustrated.

```
*SOLID SECTION,MATERIAL=M1,ELSET=LIFT1
*MATERIAL,NAME=M1
*USER MATERIAL,CONSTANTS=13
3.E6,.15,1000.,100.E-6,5.5E-6,85.,.25,1.
1.,0.,0.,2.,0.
*DEPVAR
57
```

65. For the stress analysis plane-strain elements were used, soil elements were eliminated, and the structure was supported on rollers along the base and axis of symmetry. Prior to removal of the forms (at 2 days), the gravity loading for each new lift was simulated as a pressure on existing concrete. After 2 days the gravity loading was simulated as a body force per unit volume. The input file for the stress analysis is presented in Appendix B. The UMAT subroutine used in this analysis is presented in Appendix C.

66. Various types of output are available in ABAQUS. Output for the stress run in Appendix B was in the form of stress contour and displacement plots at specified time increments and a binary output file containing stresses, strains, and principal stresses at all integration points for each

time increment. A postprocessing routine was then used to convert this binary information into stress-time plots at various locations.

Factors Affecting the Accuracy of the Calculations

67. Several factors can affect the accuracy of calculations using UMAT. The first and most obvious is the accuracy of the information used to develop the creep compliance, modulus, and shrinkage curves in the model. Frequently an analyst will try to cut costs by assuming properties of the concrete. It has been our experience at WES that the properties of mass concrete mixtures are heavily dependent on the amount of cement used, the type and amount of fly ash used, the type and modulus of aggregate, the water-cement ratio of the mixture, and the use of chemical admixtures. Predictions of material behavior are almost impossible to make based on "similar mixtures". For example, two concrete mixtures using the same components and proportions and the same type of aggregate obtained from different sources could still have very different specific creep curves if the moduli of the aggregates were very different.

68. Time-step size also affects accuracy of the results. Small time-steps (0.25 to 0.5 day) must be used after large changes in load to ensure the accuracy of creep predictions. This is true even for loadings applied after the first few days. This can be demonstrated by modeling a creep test using various time-stepping schemes. In the test, a 6- by 12-in. cylinder was loaded to 665 psi at 7 days after placement. The load was removed 9 days later. The test was simulated in four ABAQUS analyses using the time-stepping schemes shown in Table 4. Axial strains predicted for the period prior to unloading are compared with test strains in Figure 11. Predicted strains compared well with test results in the first two analyses (with time-steps less than or equal to 0.5 day for the first 5 days). The third analysis (with 1-day time-steps) overpredicted creep. Predicted and test strains after unloading are compared in Figure 12. Although creep recovery predicted in both Runs 1 and 4 was greater than test creep recovery, Run 1 results closely agreed with strains calculated using superposition, an accepted method of predicting creep recovery (Neville, Dilger, and Brooks 1983). In Run 4, 1-day time-steps were used after unloading, and predicted creep relief was much

Table 4
Time-stepping Schemes for Runs 1 through 4

Run No.	Step No.	<u>Prior to Unloading</u>		<u>After Unloading</u>	
		<u>Increment days</u>	<u>Step days</u>	<u>Increment days</u>	<u>Step days</u>
1	2	0.25	2		
	3	0.50	3		
	4	1.00	4		
	6			0.25	2
	7			0.50	3
2	8			1.00	10
	2	0.50	5		
	3	1.00	4		
	5			0.25	2
	6			0.50	3
3	7			1.00	10
	2	1.00	9		
	4			0.25	2
	5			0.50	3
	6			1.00	10
4	2	0.25	2		
	3	0.50	3		
	4	1.00	4		
	5			1.00	15

greater than in Run 1. In general, the use of time-steps that are too large will result in overpredicting creep strains.

69. The accuracy of predictions may also be affected by the type of elements used in the stress analysis. In a plane-strain analysis, strains in the out-of-plane direction are assumed to be constant along the length of the structure. This type of analysis is considered to be valid for very long structures. In practice, however, the out-of-plane strain is always zero. This condition corresponds to total restraint of out-of-plane strains, a condition which likely does not exist in real mass concrete structures. Because stresses due to this restraint are calculated in the UMAT model, a plane-strain analysis can result in excessive out-of-plane stresses and out-of-plane cracking. In cases where out-of-plane cracking causes convergence problems, plane-stress analyses can be used. In a plane-stress analysis,

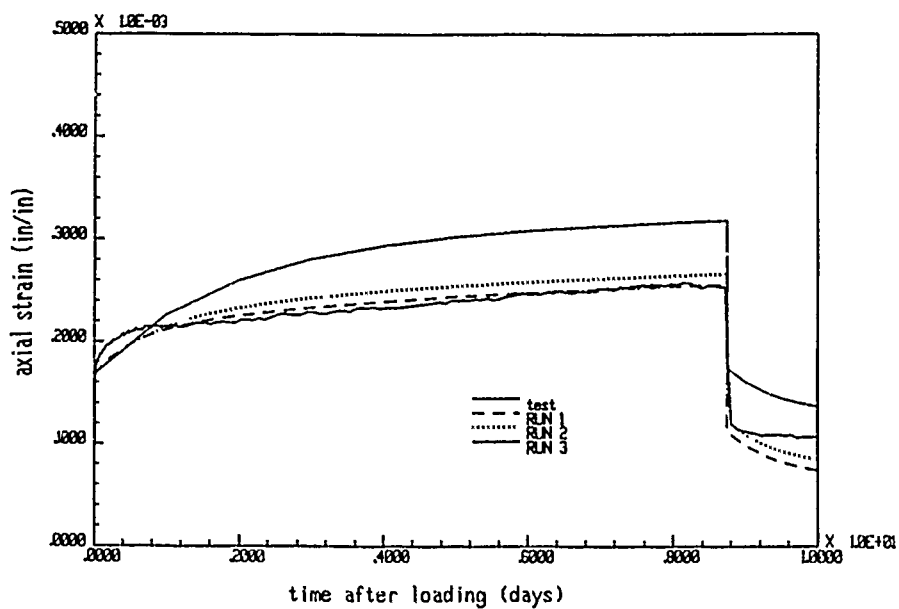


Figure 11. Predicted and test axial strains, 7-day creep test analysis

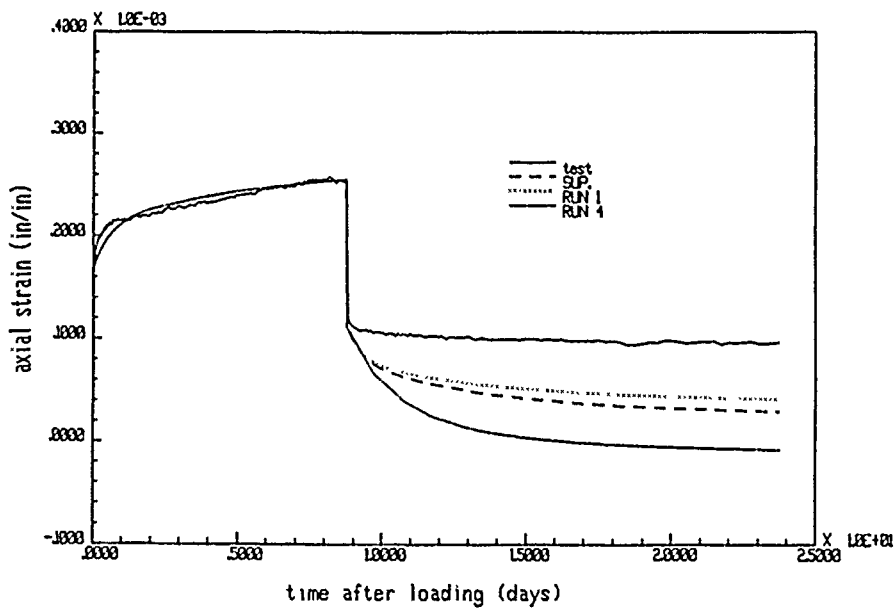


Figure 12. Predicted and test axial strains, 7-day creep test analysis, showing comparison after unloading

out-of-plane stresses are assumed to be constant (or zero) along the length of the structure. This corresponds to no restraint in the out-of-plane direction. Obviously, neither of these analyses gives a complete picture of stresses in most mass concrete structures. However, they can be used to determine the upper and lower limits for in-plane stresses and an upper limit for out-of-plane stresses.

70. To demonstrate the effect of element type on analysis results, three additional analyses were run on the chamber monolith floor shown in Figure 10: (a) plane-stress analysis using 8-node elements with reduced integration; (b) 3-D analysis using 20-node brick elements and a total monolith length of 43 ft; and (c) 3-D analysis using 20-node brick elements and a total monolith length of 86 ft.

71. Grids for the 3-D analyses modeled quarter-symmetric sections of the chamber monolith floor. Full 3 by 3 by 3 integration was used in the heat-transfer analyses and reduced integration in the stress analyses. The grid used in the 43-ft monolith analyses is shown in Figure 13. Integration point locations for the stress elements are shown in Figure 14.

72. Since the highest tensile stresses in the plane-strain analysis occurred at the center of Lift 3, stresses in the x- and z-directions at the center of Lift 3 in the four analyses have been compared in Figures 15 and 16. Stresses in the x-direction in the two 3-D analyses were almost exactly the same. Plane-strain predictions were slightly higher than those for the 3-D analyses, and plane-stress predictions were lower. Stresses in the z-direction increased with monolith length in the 3-D analyses but never reached those predicted by a plane-strain analysis. The sudden drop in stress in Figure 16 for the plane-strain analysis indicates cracking at that integration point.

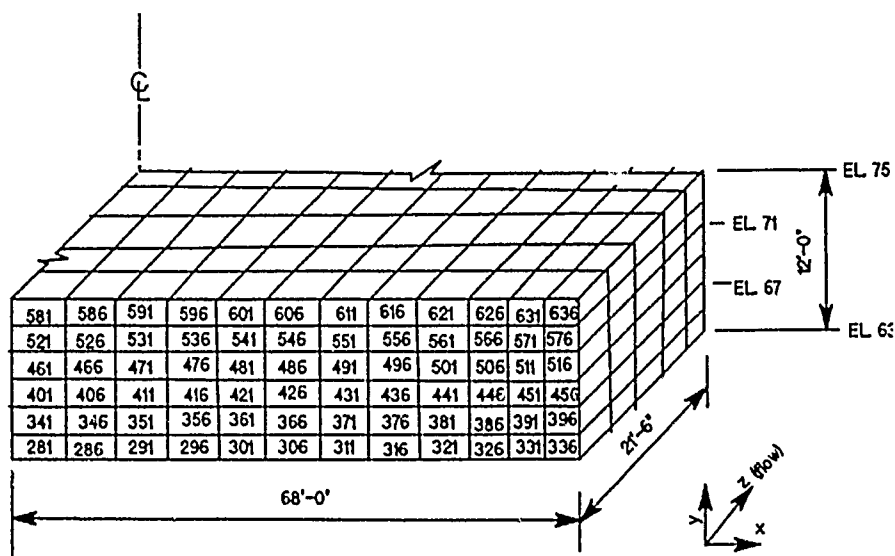


Figure 13. 3-D grid for FR stress analysis

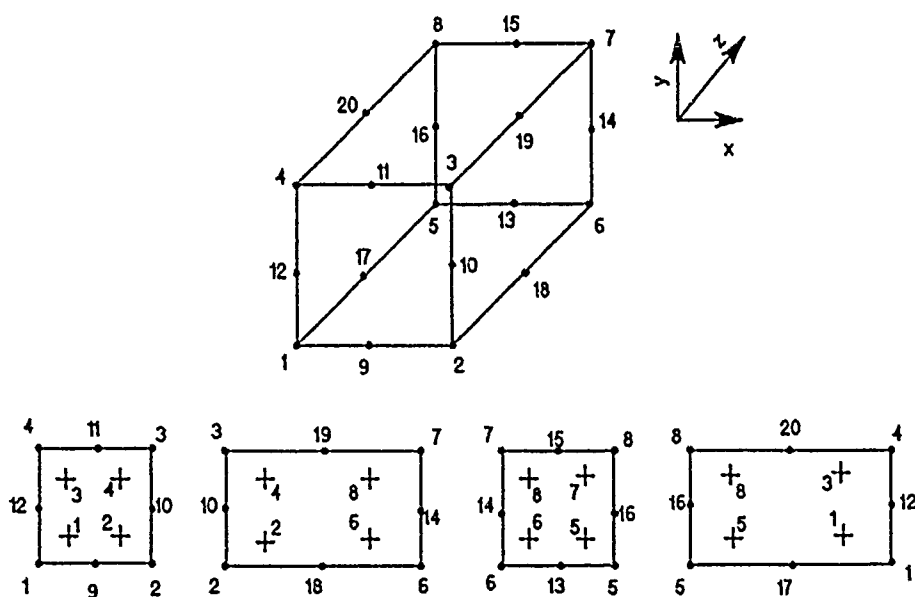


Figure 14. Location of integration points for 20-node element with reduced integration

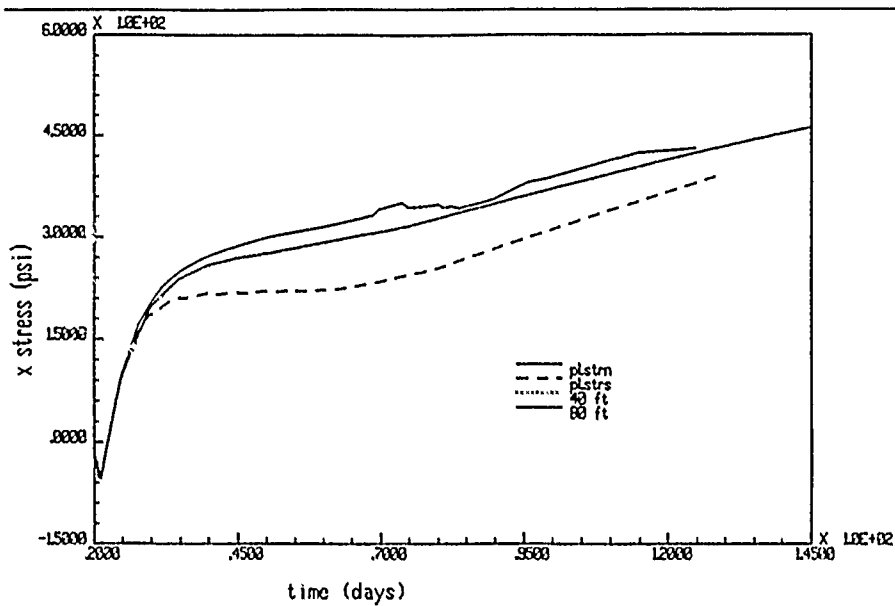


Figure 15. Stresses in the x-direction at the center of Lift 3

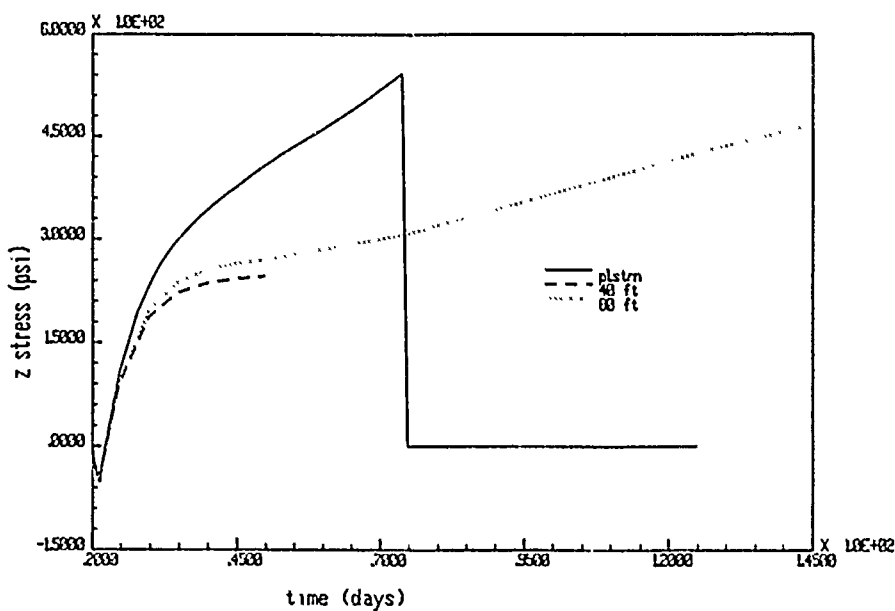


Figure 16. Stresses in the z-direction at the center of Lift 3.

PART VII: CONCLUSIONS AND RECOMMENDATIONS

Conclusions

73. Cracking in mass concrete is due to the restraint of volume change. These volume changes are due to thermal expansion and contraction, shrinkage, and creep/stress relaxation. Restraint of these volume changes is due to external boundary conditions and/or internal thermal gradients.

74. In the construction of mass concrete structures, due consideration should be given to reducing construction-related cracking. Although no Corps of Engineers structures have failed catastrophically due to construction-related cracking, a number of structures have required costly remedial repairs to prevent leakage or to increase the service life of the structure. Therefore, it is prudent to take measures prior to construction of the structure to reduce the potential for cracking.

75. Toward this end, a modern, computationally efficient analysis tool has been developed to predict cracking in concrete. This tool is a constitutive model which keeps track of the time-dependent changes in material response parameters such as elastic modulus, creep, and shrinkage. An interactive cracking criterion is included in the model based upon the smeared-crack approach. Both 2- or 3-D versions of the model are currently available. The model has been developed for use with ABAQUS, a modern, general-purpose heat-transfer and structural analysis FE code. ABAQUS features an option which allows user-defined material models to be easily incorporated as well as user-selected time-stepping for solution of the incremental construction problem. Through the MODEL CHANGE option, ABAQUS also allows the addition of lifts of concrete at user determined intervals of time.

76. The results from the use of this model are sensitive to the input values of the various material parameters. To accurately simulate time-dependent material behavior, the model requires accurate test data for calibration of the user-defined algebraic functions which govern the material properties. These data are critical for obtaining a meaningful representation of material behavior. Because concrete mixtures for mass concrete construction are site- and material-specific, no known data base of test data exists which would allow the model user to confidently estimate changes in

material properties with time. Therefore, until such an extensive data base of material properties data is developed, mechanical tests must be conducted to develop the information needed to calibrate the creep, shrinkage, and elastic modulus formulations in the model. These data should be obtained as soon as possible, beginning no later than 1 day after time of final setting.

77. The results from incremental construction analyses should be used to develop construction specification for projects which will reduce the potential for construction-related cracking. In addition, these analyses will provide information on the characteristics of concrete mixtures which are advantageous for reducing cracking. This information could be used as additional guidance for the proportioning of concrete mixtures for future mass concrete construction.

Recommendations

78. We recommend that accurate early-time material properties data be obtained on a project-by-project basis in the laboratory on project-specific materials and concrete mixtures when possible. If the exact project materials and concrete mixtures are unknown, the analyst should seek assistance from materials experts on the most likely materials to be used to construct the structure. Material properties from these can be developed as an estimate for project materials and then verified at a later date when information on project-specific information is available. The use of material properties from a generic material is not recommended.

79. As more early-time material properties data are gathered from a variety of concrete mixtures and materials, a data base of these properties should be maintained. This data base could be used for reference in the future to possibly establish bounding material properties for use in incremental construction analyses.

80. The analytical formulations presented for creep, elastic modulus, and shrinkage could be further refined as additional data and experience are obtained.

81. To realize the maximum benefit from incremental construction analysis, we recommend that the mixture proportioning phase of the project be integrated with the incremental construction analysis phase. This will lead

✓
to a more cost-effective, crack-free structure. It has been our experience that too often consideration of cracking has been delayed until after many key decisions have been made about the selection of materials, mixture proportions, and other parameters. Much can be gained from timely consideration of the effects of these factors prior to initiating the incremental construction analysis.

82. We recommend that the procedures presented in this report be extended to include roller-compacted concrete (RCC) applications. An investigation into the early-time mechanical properties of RCC mixtures along with an analytical study of the construction procedures used to construct RCC structures should be conducted.

83. The material model developed in this investigation incorporates sound theory. However, a disadvantage of the approach used in developing this model is that verification of the predictions of the model under field conditions is quite difficult and expensive and, therefore, has not been accomplished. We recommend that a comprehensive evaluation of the model be conducted on a mass concrete structure in the field. This would require extensive instrumentation of the structure and analysis of the data to verify model predictions.

REFERENCES

- American Concrete Institute, Committee 209. 1990. "Prediction of Creep, Shrinkage, and Temperature Effects in Concrete Structures," ACI Manual of Concrete Practice, American Concrete Institute, Detroit, MI.
- American Concrete Institute. 1990. "Cement and Concrete Terminology," ACI 116R, Special Publication-19(90), Detroit, MI.
- Bombich, A. A., Norman, C. D., and Jones, H. W. 1987 (Jul). "Thermal Stress Analyses of Mississippi River Lock and Dam 26(R)," Technical Report SL-87-21, US Army Engineer Waterways Experiment Station, Vicksburg, MS.
- Chen, W. F. 1982. Plasticity in Reinforced Concrete. McGraw-Hill, New York.
- Garner, S. B., Hammons, M. I., and Bombich, A. A. "Red River Waterway Thermal Studies, Report 2: Thermal Stress Analyses" (in preparation), US Army Engineer Waterways Experiment Station, Vicksburg, MS.
- Hammons, M. I., Garner, S. B., and Smith, D. M. 1989 (Jun). "Thermal Stress Analysis of Lock Wall, Dashields Locks, Ohio River," Technical Report SL-89-6, US Army Engineer Waterways Experiment Station, Vicksburg, MS.
- Hibbitt, Karlsson and Sorensen, Inc. 1988. ABAQUS User's Manual Version 4.7. Providence, RI.
- Holland, T. C., Liu, T. C., and Bombich, A. A. 1982. "Determination of Properties of Concrete Used in Thermal Studies for Lock and Dam No. 2, Red River Waterway," Miscellaneous Paper SL-82-5, US Army Engineer Waterways Experiment Station, Vicksburg, MS.
- Neville, A. M., Dilger, W. H., and Brooks, J. J. 1983. Creep of Plain and Structural Concrete. Construction Press, New York.
- Norman, C. D., Campbell, R. L., and Garner, S. 1988 (Aug). "Analysis of Concrete Cracking in Lock Wall Resurfacing," Technical Report REMR-CS-15, US Army Engineer Waterways Experiment Station, Vicksburg, MS.
- Norman, C. D., and Anderson, F. A. 1985. "Reanalysis of Cracking in Large Concrete Dams in the US Army Corps of Engineers," Commission Internationale Des Grande Barrages, Quinzieme Congres Des Grande Barrages, Lausanne, Switzerland.
- Rashid, Y. R., and Dunham, R. S. "Development of a General Three-Dimensional UMAT Model for Concrete Considering Aging, Viscoplasticity, and Cracking" (in preparation), Technical Report ANA-85-0041, ANATECH International Corporation, La Jolla, CA.

APPENDIX A

EXAMPLE HEAT-TRANSFER ANALYSIS DECK

C RED RIVER CHAMBER MONOLITH FLOOR 2-DIMENSIONAL MODEL
 C TEMPERATURE DECK - 112 SOIL ELEMENTS
 C 138 CONCRETE ELEMENTS
 C PLACEMENT SCHEDULE:
 C CHAMBER FLOOR, LIFTS 1-3 PLACED AT 10-DAY INCR., E.G.
 C PLACED AT 0, 10, 20, & 30 DAYS.
 C FORM REMOVAL: FORMS ON VERTICAL SURFACES REMOVED AT 2 DAYS AFTER PLMT
 C STEP/INCREMENT SCHEME (SCHEME BEYOND 5 DAYS ONLY AFTER PLACING L1-4,14
 C FIRST 2 DAYS AFTER PLACEMENT- 8 INCREMENTS AT DT=0.25 DAYS EACH.
 C DAYS 3 - 5 AFTER PLACEMENT - 6 INCREMENTS AT DT=0.50 DAYS EACH.
 C DAYS 6 - 10 AFTER PLACEMENT - 5 INCREMENTS AT DT=1.00 DAYS EACH.
 C DAYS 11- 20 AFTER PLACEMENT - 5 INCREMENTS AT DT=2.00 DAYS EACH.
 C DAYS 21- 45 AFTER PLACEMENT - 5 INCREMENTS AT DT=5.00 DAYS EACH.
 C DAYS 46- 95 AFTER PLACEMENT - 5 INCREMENTS AT DT=10.00 DAYS EACH.
 C LIFT ELEMENTS - LNN, WHERE NN=LIFT NO. FOR ALL ELEMENTS IN A LIFT
 C TOP SURFACE ELEMENTS - LNNT; WHERE NN=LIFT NO., T=TOP SURFACE ELEMENT
 C UPON WHICH ADDITIONAL CONCRETE IS PLACED, AND
 C - LNNF; WHERE NN=LIFT NO., F=PERMANENTLY EXPOSED
 C FLOOR SURFACE(SUCH AS CHAMBER FLOOR, CULVERT FLOOR)
 C
 C COMBINED LIFT ELEMENT SETS:
 C 1. FULL LIFTS: L1_2 -ALL ELEMENTS IN LIFTS 1 AND 2
 C L1_3 -ALL ELEMENTS IN LIFTS 1 - 3
 C
 *HEADING
 2-D CHAMBER MONOLITH FLOOR, HEAT TRANSFER DECK #TR1
 *NODE
 2,0,636
 38,432,636
 50,504,636
 66,624,636
 98,816,636
 114,936,636
 1202,0,756
 1238,432,756
 1250,504,756
 1266,624,756
 1298,816,756
 1314,936,756
 3002,0,900
 3038,432,900
 3050,504,900
 3066,624,900
 3098,816,900
 *NGEN
 2,38,2
 38,50,2
 50,66,2
 66,98,2
 98,114,2
 1202,1238,2
 1238,1250,2

1250,1266,2
1266,1298,2
1298,1314,2
3002,3038,2
3038,3050,2
3050,3066,2
3066,3098,2
2,1202,150
38,1238,150
50,1250,150
66,1266,150
98,1298,150
114,1314,150
152,188,4
188,200,4
200,216,4
216,248,4
248,264,4
302,338,2
338,350,2
350,366,2
366,398,2
398,414,2
452,488,4
488,500,4
500,516,4
516,548,4
548,564,4
602,638,2
638,650,2
650,666,2
666,698,2
698,714,2
752,788,4
788,800,4
800,816,4
816,848,4
848,864,4
902,938,2
938,950,2
950,966,2
966,998,2
998,1014,2
1052,1088,4
1088,1100,4
1100,1116,4
1116,1148,4
1148,1164,4
1202,3002,150
1238,3038,150
1250,3050,150
1266,3066,150

1298,3098,150
1352,1388,4
1388,1400,4
1400,1416,4
1416,1448,4
1502,1538,2
1538,1550,2
1550,1566,2
1566,1598,2
1652,1688,4
1688,1700,4
1700,1716,4
1716,1748,4
1802,1838,2
1838,1850,2
1850,1866,2
1866,1898,2
1952,1988,4
1988,2000,4
2000,2016,4
2016,2048,4
2102,2138,2
2138,2150,2
2150,2166,2
2166,2198,2
2252,2288,4
2288,2300,4
2300,2316,4
2316,2348,4
2402,2438,2
2438,2450,2
2450,2466,2
2466,2498,2
2552,2588,4
2588,2600,4
2600,2616,4
2616,2648,4
2852,2888,4
2888,2900,4
2900,2916,4
2916,2948,4
2702,2738,2
2738,2750,2
2750,2766,2
2766,2798,2
*ELEMENT,TYPE=DC2D8
1,2,6,306,302,4,156,304,152
113,1202,1206,1506,1502,1204,1356,1504,1352
255,2790,2794,3094,3090,2792,2944,3092,2940
*ELGEN
1,28,4,1,4,300,28
113,24,4,1,5,300,24

209,2,300,24
 233,16,4,1
 255,2,4,1
 *ELSET,ELSET=SL,GENERATE
 1,112
 *ELSET,ELSET=L1,GENERATE
 113,160
 *ELSET,ELSET=L1T,GENERATE
 137,160
 *ELSET,ELSET=L1R
 136,160
 *ELSET,ELSET=L2,GENERATE
 161,208
 *ELSET,ELSET=L2T,GENERATE
 185,208
 *ELSET,ELSET=L2R
 184,208
 *ELSET,ELSET=L1_2R
 L1R,L2R
 *ELSET,ELSET=L3,GENERATE
 209,248
 255,256
 *ELSET,ELSET=L3T,GENERATE
 233,248
 255,256
 *ELSET,ELSET=L3TA,GENERATE
 225,230
 *ELSET,ELSET=L3R
 232,248,256
 *ELSET,ELSET=L3L
 255
 *ELSET,ELSET=L1_2
 L1,L2
 *ELSET,ELSET=L1_3
 L1_2,L3
 *ELSET,ELSET=L1_3R
 L1R,L2R,L3R
 *ELSET,ELSET=SOILT
 109,110,111,112
 *ELSET,ELSET=REMV
 L2,L3
 *ELSET,ELSET=ALL
 SOIL,L1_3
 *NSET,NSET=SL1,GENERATE
 2,114,2
 *NSET,NSET=SL2,GENERATE
 152,264,4
 *NSET,NSET=SL3,GENERATE
 302,414,2
 *NSET,NSET=SL4,GENERATE
 452,564,4
 *NSET,NSET=SL5,GENERATE

602,714,2
 *NSET,NSET=SL6,GENERATE
 752,864,4
 *NSET,NSET=SL7,GENERATE
 902,1014,2
 *NSET,NSET=SL8,GENERATE
 1052,1164,4
 *NSET,NSET=SL9,GENERATE
 1202,1314,2
 *NSET,NSET=NL1,GENERATE
 1352,1448,4
 1502,1598,2
 1652,1748,4
 1802,1898,2
 *NSET,NSET=NL2,GENERATE
 1952,2048,4
 2102,2198,2
 2252,2348,4
 2402,2498,2
 *NSET,NSET=NL3,GENERATE
 2552,2648,4
 2702,2798,2
 2852,2948,4
 3002,3066,2
 3090,3098,2]
 *SOLID SECTION,MATERIAL=SOIL,ELSET=SL
 *MATERIAL,NAME=SOIL
 *DENSITY
 0.04285
 *CONDUCTIVITY
 2.2
 *SPECIFIC HEAT
 0.266
 *SOLID SECTION,MATERIAL=CONCR,ELSET=L1_3
 *MATERIAL,NAME=CONCR
 *DENSITY
 0.0865
 *CONDUCTIVITY
 2.3
 *SPECIFIC HEAT
 .21
 *INITIAL CONDITIONS,TYPE=TEMPERATURE
 SL1,70.0
 SL2,70.5
 SL3,71.0
 SL4,72.0
 SL5,73.0
 SL6,76.0
 SL7,79.0
 SL8,81.0
 SL9,82.0
 NL1,85.0

NL2,85.0
 NL3,85.0
 *BOUNDARY
 SL1,11,,70
 *AMPLITUDE,NAME=AMB,TIME=HEAT,VALUE=ABS

0.0	82.9	8.25	83.6	16.5	83.8	36.5	83.8
43.1	83.5	47.5	83.0	58.3	82.0	64.0	81.0
68.5	80.0	78.0	77.7	108.5	67.3	139.0	57.4
169.5	50.8	178.8	49.0	187.0	48.5	194.5	48.3
206.3	48.3	215.2	48.7	217.4	49.0	230.0	52.1
259.5	59.0	290.0	67.4	320.5	74.1	351.0	80.5

 *PLOT
 CM FLOOR PREPROCESSOR PLOT
 200,180,190,160,10,15,10,5
 3,,,1,12,.5
 *DETAIL,ELSET=ALL
 *DRAW,ELNUM
 *WAVEFRONT MINIMIZATION,SUPPRESS
 *STEP,INC=8
 PLACE LIFT 1, EL63-67, T=0, DAY=1-2, DT=0.25D
 *HEAT TRANSFER
 0.25,2.0
 *MODEL CHANGE,REMOVE
 REMV
 *FILM,AMPLITUDE=AMB,OP=NEW
 SOILT,F3,,0.53867
 L1R,F2,,0.16549
 L1T,F3,,0.53867
 *DFLUX
 L1_3,BFNU
 *NODE FILE
 NT
 *PLOT,FREQUENCY=3
 CM HEAT TRANSFER RUN, JULY 1 START, L1
 200,180,190,160,10,15,10,5
 3,,,1,12,.5
 *DETAIL,ELSET=L1
 *CONTOUR
 TEMP,10,,
 *END STEP
 *STEP,INC=6
 LIFT 1, REMOVE FORMS AT T=2.0 DAYS, RUN DAYS 3-5, DT=0.5
 *HEAT TRANSFER
 0.5,3.0
 *FILM,AMPLITUDE=AMB
 L1R,F2,,0.53867
 *END STEP
 *STEP,INC=5
 LIFT 1, CONTINUE CALC AT T=5.0 DAYS, RUN DAYS 6-10, DT=1.0
 *HEAT TRANSFER
 1.0,5.0
 *END STEP

```

*STEP,INC=8
PLACE LIFT 2, EL67-71, T=2.00, DAY=3-4, DT=0.5D
*HEAT TRANSFER
0.25,2.0
*MODEL CHANGE,INCLUDE
L2
*FILM,AMPLITUDE=AMB,OP=NEW
SOILT,F3,,0.53867
L1R,F2,,0.53867
L2R,F2,,0.16549
L2T,F3,,0.53867
*FLOT,FREQUENCY=3
CM HEAT TRANSFER RUN, JULY 1 START, L1_2
200,180,190,160,10,15,10,5
3,,,1,12,.5
*DETAIL,ELSET=L1_2
*CONTOUR
TEMP,10,,
*END STEP
*STEP,INC=6
LIFT 2, REMOVE FORMS AT T=12.0 DAYS, RUN DAYS 13-15, DT=0.5
*HEAT TRANSFER
0.5,3.0
*FILM,AMPLITUDE=AMB
L2R,F2,,0.53867
*END STEP
*STEP,INC=5
LIFT 2, CONTINUE CALC AT T=15.0 DAYS, RUN DAYS 16-20, DT=1.0
*HEAT TRANSFER
1.0,5.0
*END STEP
*STEP,INC=8
PLACE LIFT 3, EL71-75, T=20.00, DAY=21-22, DT=0.25D
*HEAT TRANSFER
0.25,2.0
*MODEL CHANGE,INCLUDE
L3
*FILM,AMPLITUDE=AMB,OP=NEW
SOILT,F3,,0.53867
L1_2R,F2,,0.53867
L3L,F4,,0.16549
L3R,F2,,0.16549
L3TA,F3,,0.16549
L3T,F3,,0.53867
*PLOT,FREQUENCY=3
CM HEAT TRANSFER RUN, JULY 1 START, L1_3
200,180,190,160,10,15,10,5
3,,,1,12,.5
*DETAIL,ELSET=L1_3
*CONTOUR
TEMP,10,,
*END STEP

```

*STEP,INC=6
LIFT 3, REMOVE FORMS AT T=22.0 DAYS, RUN DAYS 23-25, DT=0.5
*HEAT TRANSFER
0.5,3.0
*FILM,AMPLITUDE=AMB
L3L,F4,,0.53867
L3R,F2,,0.53867
L3TA,F3,,0.53867
*END STEP
*STEP,INC=5
LIFT 3, CONTINUE CALC AT T=25.0 DAYS, RUN DAYS 26-30, DT=1.0
*HEAT TRANSFER
1.0,5.0
*END STEP
*STEP,INC=5
LIFT 3, CONTINUE CALC AT T=30. DAYS, RUN DAYS 31-40, DT=2.
*HEAT TRANSFER
2.,10.
*END STEP
*STEP,INC=5
LIFT 3, CONTINUE CALC AT T=40. DAYS, RUN DAYS 41-65, DT=5.
*HEAT TRANSFER
5.,25.
*END STEP
*STEP,INC=5
LIFT 3, CONTINUE CALC AT T=65. DAYS, RUN DAYS 66-106, DT=10.
*HEAT TRANSFER
10.,50.
*END STEP

```

      subroutine dflux(flux,temp,kstep,kinc,time,noel,npt,coords,
        $              jltyp)
c*****
c this version interpolates along the adiabatic curve for more than
c one lift of any number of elements
c
c it also permits the use of two adiabatic curves in the same model.
c the concrete represented by the two curves must exist in two distinct
c groups of single or multiple lifts in which the elements in each
c of the two groups of lifts is consecutive that is, the two concretes
c cannot exist in the same lift or exist in a lower and then a higher
c numbered lift so as to alternate between successive lifts. in
c other words, there must be a single lift interface between the two
c concretes. appropriate dimension statements, data statements, and
c coding must be modified to include:
c   q1,t1,q2,t2,prop1,prop2,lmix2
c
c note: although two curves were not required for this analysis, the option
c has been left in to demonstrate its use
c
c units in the t array are hours.
c units in the q array are temperature in degrees F and will be
c converted to btu/(lb-in**3)
c nq      = no. of points in t & q arrays
c entime  = endtime for dflux (last time in t array + .01)
c sttime  = array of starttimes for each element in hours
c          + one dummy time. the array must be dimensioned
c          number of elements + 1.
c prop(1)= density (lb/cu.in.)
c prop(2)= specific heat
c nmax    = number of integration points per element
c nlifts  = number of lifts
c nstme   = number of start times (nlifts + 1 dummy time)
c nelem(nlifts) = array of number of elements in each lift
c stm(i)  = array of starttimes for each lift + 1 dummy time(hrs)
c nstel   = number of first element using dflux
c prop1   = density and specific heat of concrete mix 1
c prop2   = density and specific heat of concrete mix 2
c lmix2   = lowest lift including concrete mix2
c*****
c
c for double precision versions of ABAQUS 4.7
c
      implicit real*8(a-h,o-z)
c parameter statement to hold no. lifts and no lifts +1
      parameter (nlifts=3,nstme=4,nq1=9,nq2=26)
      common/eldef/sttime(139)
      dimension coords(3),q(26),t(26),prop(2),nelem(nlifts),
&               oltime(nlifts),oldq(nlifts),nolincr(nlifts),
&               nseter(nlifts),stm(nstme),q1(nq1),t1(nq1),
&               q2(nq2),t2(nq2),prop1(2),prop2(2)
      save nolincr,oltime,oldq,nseter,nwhere,nnoel

```

```

c hansen's al3 curve (hh)
  data q1 /7.33,14.67,22.,25.9,30.5,35.,35.9,36.4,36.4/
  data t1 /5.,10.,15.,25.,40.,70.,80.,140.,672./
c
  data q2 / 11.18, 27.86, 42.11, 49.95, 53.48, 55.35, 56.86,
&          58.09, 59.15, 59.89, 60.58, 61.17, 62.35, 63.31, 64.15,
&          64.97, 66.10, 66.99, 67.68, 68.75, 69.57, 70.31, 70.94,
&          72.13, 72.88, 73.51/
  data t2 / 6.00, 12.00, 18.00, 24.00, 30.00, 36.00, 42.00,
&          48.00, 54.00, 60.00, 66.00, 72.00, 84.00, 96.00,108.00,
&          120.00,144.00,168.00,192.00,240.00,288.00,336.00,384.00,
&          480.00,576.00,672.00/
c
  data prop1/0.0865,0.21/
  data prop2/0.08000,0.20/
  data entime/672.01/
  data lmix2/4/
  data nmax/9/
  data nelem/48,48,42/
  data stm/0.,240.,480.,720./
  data nstel/113/
c
c renumber elements
c
  if(noel.lt.nstel)return
  if(noel.eq.nstel.and.npt.eq.1)nnoel=0
  if(npt.eq.1)nnoel=nnoel+1
  noel=nnoel
c
c fill start-time array
c
  nst=1
  ntot=0
  do 200 i=1,nlifts
    ntot=nelem(i)+ntot
    do 201 j=nst,ntot
      sttime(j)=stm(i)
201    continue
    nst=nst+nelem(i)
200  continue
    sttime(ntot+1)=stm(nstme)
c
c determine lift number (k)
c
  ne=0
  nem=0
  do 202 k=1,nlifts
    ne=ne+nelem(k)
    if(k.eq.1.and.noel.le.ne)go to 24
    if(k.gt.1)nem=nem+nelem(k-1)
    if(noel.gt.nem.and.noel.le.ne)go to 24
202  continue

```

```

24  if(kstep.eq.1.and.kinc.eq.1)then
      nolincr(k)=1
      oltime(k)=0.
      oldq(k)=0.
    end if
c
c  set up multiple adiabatics & thermal properties
c
    if(k.lt.lmix2)then
c      assign t and q for mix 1
        nq=nq1
        do 203 i=1,nq
            t(i)=t1(i)
            q(i)=q1(i)
203      continue
c      assign props for mix 1
        prop(1)=prop1(1)
        prop(2)=prop2(2)
    else
c      assign t and q for mix 2
        nq=nq2
        do 204 i=1,nq
            t(i)=t2(i)
            q(i)=q2(i)
204      continue
c      assign props for mix 2
        prop(1)=prop2(1)
        prop(2)=prop2(2)
    end if
c  *****
    trel = time - sttime(noel) / 24.
    end = entime / 24.
    flux = 0.0
    if( trel.gt.0.0.and.trel.lt.end ) go to 10
    return
c
10  continue
    do 20 i=1,nq
        nch=0
        td = t(i) / 24.
        if(i.gt.1)tb=t(i-1)/24.
        dif=abs(trel-td)
        if(trel.lt.td.or,dif.lt.0.01)go to 30
20  continue
c
    write(6,35) kstep,kinc,time,noel
35  format(/," warning - passed through dflux without assigning",
&        /," flux. step =",i5," inc =",i5,
&        /," time =",f12.2," element =",i5)
    return
30  continue
c

```

```

c calculate flux
c
    if(i.eq.1)then
        tq=q(i)*trel/td
    end if
    if(i.gt.1)tq=(q(i)-q(i-1))*(trel-tb)/(td-tb)+q(i-1)
    flux=(tq-oldq(k))/(trel-oltime(k))*prop(1)*prop(2)
c
c set pointers
c
    if(sttime(noel+1).gt.sttime(noel))then
        if(kinc.eq.nolincr(k).and.npt.eq.nmax.and.nsetter(k).eq.4)
&      go to 100
        if(kinc.eq.nolincr(k).and.npt.eq.nmax)then
            nsetter(k)=4
        end if
    end if
    if(kinc.ne.nolincr(k))nolincr(k)=kinc
    go to 999
100  oltime(k)=trel
    oldq(k)=tq
    nsetter(k)=1
999  continue
    return
    stop
end

```

APPENDIX B
EXAMPLE STRESS ANALYSIS DECK

C RED RIVER CHAMBER MONOLITH FLOOR 2-DIMENSIONAL MODEL
 C STRESS DECK - 138 CONCRETE ELEMENTS
 C PLACEMENT SCHEDULE:
 C CHAMBER FLOOR, LIFTS 1-3 PLACED AT 10-DAY INCR., E.G.
 C PLACED AT 0, 10, 20, & 30 DAYS.
 C FORM REMOVAL: FORMS ON VERTICAL SURFACES REMOVED AT 2 DAYS AFTER PLMT
 C STEP/INCREMENT SCHEME (SCHEME BEYOND 5 DAYS ONLY AFTER PLACING L1-4,14
 C FIRST 2 DAYS AFTER PLACEMENT- 8 INCREMENTS AT DT=0.25 DAYS EACH.
 C DAYS 3 - 5 AFTER PLACEMENT - 6 INCREMENTS AT DT=0.50 DAYS EACH.
 C DAYS 6 - 10 AFTER PLACEMENT - 5 INCREMENTS AT DT=1.00 DAYS EACH.
 C DAYS 11- 20 AFTER PLACEMENT - 5 INCREMENTS AT DT=2.00 DAYS EACH.
 C DAYS 21- 45 AFTER PLACEMENT - 5 INCREMENTS AT DT=5.00 DAYS EACH.
 C DAYS 46- 95 AFTER PLACEMENT - 5 INCREMENTS AT DT=10.00 DAYS EACH.
 C LIFT ELEMENTS - LNN, WHERE NN=LIFT NO. FOR ALL ELEMENTS IN A LIFT
 C TOF SURFACE ELEMENTS - LNNT; WHERE NN=LIFT NO., T=TOP SURFACE ELEMENT
 C UPON WHICH ADDITIONAL CONCRETE IS PLACED, AND
 C - LNNF; WHERE NN=LIFT NO., F=PERMANENTLY EXPOSED
 C FLOOR SURFACE(SUCH AS CHAMBER FLOOR, CULVERT FLOOR)
 C
 C COMBINED LIFT ELEMENT SETS:
 C 1. FULL LIFTS: L1_2 -ALL ELEMENTS IN LIFTS 1 AND 2
 C L1_3 -ALL ELEMENTS IN LIFTS 1 - 3
 C
 *HEADING
 2-D CHAMBER MONOLITH FLOOR, STRESS DECK #SR1
 *NODE
 1202,0,756
 1238,432,756
 1250,504,756
 1266,624,756
 1298,816,756
 3002,0,900
 3038,432,900
 3050,504,900
 3066,624,900
 3098,816,900
 *NGEN
 1202,1238,2
 1238,1250,2
 1250,1266,2
 1266,1298,2
 3002,3038,2
 3038,3050,2
 3050,3066,2
 3066,3098,2
 1202,3002,150
 1238,3038,150
 1250,3050,150
 1266,3066,150
 1298,3098,150
 1352,1388,4
 1388,1400,4

1400,1416,4
1416,1448,4
1502,1538,2
1538,1550,2
1550,1566,2
1566,1598,2
1652,1688,4
1688,1700,4
1700,1716,4
1716,1748,4
1802,1838,2
1838,1850,2
1850,1866,2
1866,1898,2
1952,1988,4
1988,2000,4
2000,2016,4
2016,2048,4
2102,2138,2
2138,2150,2
2150,2166,2
2166,2198,2
2252,2288,4
2288,2300,4
2300,2316,4
2316,2348,4
2402,2438,2
2438,2450,2
2450,2466,2
2466,2498,2
2552,2588,4
2588,2600,4
2600,2616,4
2616,2648,4
2852,2888,4
2888,2900,4
2900,2916,4
2916,2948,4
2702,2738,2
2738,2750,2
2750,2766,2
2766,2798,2
*ELEMENT,TYPE=CPE8R
113,1202,1206,1506,1502,1204,1356,1504,1352
255,2790,2794,3094,3090,2792,2944,3092,2940
*ELGEN
113,24,4,1,5,300,24
209,2,300,24
233,16,4,1
255,2,4,1
*ELSET,ELSET=L1,GENERATE
113,160

```

*ELSET, ELSET=L1T, GENERATE
137, 160
*ELSET, ELSET=L2, GENERATE
161, 208
*ELSET, ELSET=L2T, GENERATE
185, 208
*ELSET, ELSET=L3, GENERATE
209, 248
255, 256
*ELSET, ELSET=L1_2
L1, L2
*ELSET, ELSET=L1_3
L1_2, L3
*ELSET, ELSET=REMV
L2, L3
*NSSET, NSSET=CL, GENERATE
1202, 3002, 150
*NSSET, NSSET=BASE, GENERATE
1202, 1298, 2
*NSSET, NSSET=SUP
CL, BASE
*NSSET, NSSET=NLFT1, GENERATE
1352, 1448, 4
1502, 1598, 2
1652, 1748, 4
1802, 1898, 2
*NSSET, NSSET=NLFT2, GENERATE
1952, 2048, 4
2102, 2198, 2
2252, 2348, 4
2402, 2498, 2
*NSSET, NSSET=NLFT3, GENERATE
2552, 2648, 4
2702, 2798, 2
2852, 2916, 4
2940, 2948, 4
3002, 3066, 2
3090, 3098, 2
*SOLID SECTION, MATERIAL=M1, ELSET=L1
*MATERIAL, NAME=M1
*USER MATERIAL, CONSTANTS=13
1.88E6, .15, 600., 100.E-6, 5.5E-6, 85., .24, 1.,
1.00, 0., 0., 2., 0.
*DEPVAR
57
*SOLID SECTION, MATERIAL=M2, ELSET=L2
*MATERIAL, NAME=M2
*USER MATERIAL, CONSTANTS=13
1.88E6, .15, 600., 100.E-6, 7.E-6, 85., .24, 1.,
1.00, 0., 10., 2., 0.
*DEPVAR
57

```

```

*SOLID SECTION,MATERIAL=M3,ELSET=L3
*MATERIAL,NAME=M3
*USER MATERIAL,CONSTANTS=13
1.88E6, .15,600.,100.E-6,7.E-6,85.,.24,1.,
1.00,0.,20.,2.,0.
*DEPVAR
57
*INITIAL CONDITIONS,TYPE=TEMPERATURE
NLFT1,85.0
NLFT2,85.0
NLFT3,85.0
*BOUNDARY
BASE,2
CL,1
*WAVEFRONT MINIMIZATION,SUPPRESS
*STEP
PLACE LIFT 1, EL63-67, T=0, DAY=1-2, DT=0.25D
*STATIC,PTOL=10.,DIRECT=NOSTOP
0.01,.01
*MODEL CHANGE,REMOVE
REMV
*TEMPERATURE,FILE=15,BSTEP=1(INC=1),ESTEP=1(INC=1)
*EL FILE,ELSET=L1
S,E
*EL PRINT,ELSET=L1,FREQUENCY=0
*PLOT,FREQUENCY=3
CM RUN 1, JULY 1 START, PL STRN, L1
200,180,190,160,10,15,10,5
3,,140,,1,12,.5
*DETAIL,ELSET=L1
*CONTOUR
S11,6
*CONTOUR
S22,6
*CONTOUR
S33,6
*CONTOUR
PRIN3,6
*DISPLACED
U,
*END STEP
*STEP,INC=7
PLACE LIFT 1, EL63-67, T=0, DAY=1-2, DT=0.25D
*STATIC,PTOL=10.,DIRECT=NOSTOP
.25,1.75
*DLOAD
L1,BY,-.0865
*TEMPERATURE,FILE=15,BSTEP=1(INC=2),ESTEP=1(INC=8)
*END STEP
*STEP,INC=6
LIFT 1, REMOVE FORMS AT T=2.0 DAYS, RUN DAYS 3-5, DT=0.5
*STATIC,PTOL=10.,DIRECT=NOSTOP

```

0.5,3.0
 *TEMPERATURE,FILE=15,BSTEP=2(INC=1),ESTEP=2(INC=6)
 *END STEP
 *STEP,INC=5
 LIFT 1, CONTINUE CALC AT T=5.0 DAYS, RUN DAYS 6-10, DT=1.0
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 1.0,5.0
 *TEMPERATURE,FILE=15,BSTEP=3(INC=1),ESTEP=3(INC=5)
 *END STEP
 *STEP
 LIFT 1, CONTINUE CALC AT T=11. DAYS, RUN DAY 11, DT=.24
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 .24,.24
 *TEMPERATURE,FILE=15,BSTEP=4(INC=1),ESTEP=4(INC=1)
 *END STEP
 *STEP
 PLACE LIFT 2, EL67-71, T=11.00, DT=0.01D
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 .01,.01
 *DLOAD
 L1T,P3,4.152
 *TEMPERATURE,FILE=15,BSTEP=4(INC=1),ESTEP=4(INC=1)
 *MODEL CHANGE,INCLUDE
 L2
 *EL FILE,ELSET=L1_2
 S,E
 *EL PRINT,ELSET=L1_2,FREQUENCY=0
 *PLOT,FREQUENCY=3
 CM RUN 1, JULY 1 START, PL STRN, L1_2
 200,180,190,160,10,15,10,5
 3,,140,,1,12,.5
 *DETAIL,ELSET=L1_2
 *CONTOUR
 S11,6
 *CONTOUR
 S22,6
 *CONTOUR
 S33,6
 *CONTOUR
 PRIN3,6
 *DISPLACED
 U,
 *END STEP
 *STEP,INC=7
 LIFT 2, EL67-71, T=11.00, DAYS=11-12, DT=.25D
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 0.25,1.75
 *TEMPERATURE,FILE=15,BSTEP=4(INC=2),ESTEP=4(INC=8)
 *END STEP
 *STEP,INC=1
 LIFT 2, REMOVE FORMS AT T=12.0 DAYS, RUN DAYS 13, DT=0.5
 *STATIC,PTOL=10.,DIRECT=NOSTOP

0.5,.50
 *DLOAD,OP=NEW
 L1_2,BY,-.0865
 *TEMPERATURE,FILE=15,BSTEP=5(INC=1),ESTEP=5(INC=1)
 *END STEP
 *STEP,INC=5
 LIFT 2, REMOVE FORMS AT T=12.0 DAYS, RUN DAYS 13-15, DT=0.5
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 0.5,2.5
 *TEMPERATURE,FILE=15,BSTEP=5(INC=2),ESTEP=5(INC=6)
 *END STEP
 *STEP,INC=5
 LIFT 2, CONTINUE CALC AT T=16.0 DAYS, RUN DAYS 16-20, DT=1.0
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 1.0,5.0
 *TEMPERATURE,FILE=15,BSTEP=6(INC=1),ESTEP=6(INC=5)
 *END STEP
 *STEP
 LIFT 2, CONTINUE CALC AT T=21.0 DAYS, RUN DAY 21, DT=.24
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 .24,.24
 *TEMPERATURE,FILE=15,BSTEP=7(INC=1),ESTEP=7(INC=1)
 *END STEP
 *STEP
 PLACE LIFT 3, EL71-75, T=21.00, DAY=21, DT=0.01D
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 .01,.01
 *DLOAD
 L2T,P3,4.152
 *MODEL CHANGE,INCLUDE
 L3
 *TEMPERATURE,FILE=15,BSTEP=7(INC=1),ESTEP=7(INC=1)
 *EL FILE,ELSET=L1_3
 S,E
 *EL PRINT,ELSET=L1_3,FREQUENCY=0
 *PLOT,FREQUENCY=3
 CM RUN 1, JULY 1 START, PL STRN, L1_3
 200,180,190,160,10,15,10,5
 3,,140,,1,12,.5
 *DETAIL,ELSET=L1_3
 *CONTOUR
 S11,6,
 *CONTOUR
 S22,6,
 *CONTOUR
 S33,6,
 *CONTOUR
 PRIN3,6,
 *DISPLACED
 U,
 *END STEP
 *STEP,INC=7

PLACE LIFT 3, EL71-75, T=21.00, DAY=21-22, DT=0.25D
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 .25,1.75
 *TEMPERATURE,FILE=15,BSTEP=7(INC=2),ESTEP=7(INC=8)
 *END STEP
 *STEP,INC=1
 LIFT 3, REMOVE FORMS AT T=22.0 DAYS, RUN DAYS 23, DT=0.5
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 0.5,.50
 *DLOAD,OP=NEW
 L1_3,BY,-.0865
 *TEMPERATURE,FILE=15,BSTEP=8(INC=1),ESTEP=8(INC=1)
 *END STEP
 *STEP,INC=5
 LIFT 3, REMOVE FORMS AT T=22.0 DAYS, RUN DAYS 23-25, DT=0.5
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 0.5,2.5
 *TEMPERATURE,FILE=15,BSTEP=8(INC=2),ESTEP=8(INC=6)
 *END STEP
 *STEP,INC=5
 LIFT 3, CONTINUE CALC AT T=26.0 DAYS, RUN DAYS 26-30, DT=1.0
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 1.0,5.0
 *TEMPERATURE,FILE=15,BSTEP=9(INC=1),ESTEP=9(INC=5)
 *END STEP
 *STEP,INC=5
 LIFT 3, CONTINUE CALC AT T=30. DAYS, RUN DAYS 31-40, DT=2.D
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 2.,10.
 *TEMPERATURE,FILE=15,BSTEP=10(INC=1),ESTEP=10(INC=5)
 *END STEP
 *STEP,INC=5
 LIFT 3, CONTINUE CALC AT T=40. DAYS, RUN DAYS 41-65, DT=5.
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 5.,25.
 *TEMPERATURE,FILE=15,BSTEP=11(INC=1),ESTEP=11(INC=5)
 *END STEP
 *STEP,INC=5
 LIFT 3, CONTINUE CALC AT T=65. DAYS, RUN DAYS 66-106, DT=10
 *STATIC,PTOL=10.,DIRECT=NOSTOP
 10.,50.
 *TEMPERATURE,FILE=15,BSTEP=12(INC=1),ESTEP=12(INC=5)
 *END STEP

APPENDIX C
UMAT SUBROUTINE

```

SUBROUTINE UMAT(STRESS,STATEV,HH,SSE,SPD,SCD,DUM1,
$ DUM2,DUM3,DUM4,TEPS,DEP,TYME,DELT,TEMP,DTEMP,
$ PREDEF,DPRED,CMAT,NDI,NSHR,NTENS,NSTATV,PROPS,
$ NPROPS,COORDS,DUM5)
C
C FOR DOUBLE PRECISION VERSIONS OF ABAQUS 4.7
C
C IMPLICIT REAL*8 (A-H,O-Z)
C
C UMAT86: COMBINED UMAT FOR HANDLING
C
C JPROPS=1 2-D CONCRETE WITH ELASTIC CRACKING ONLY
C JPROPS=2 2-D CONCRETE WITH CRACKING, PLASTICITY, CREEP & AGING
C JPROPS=3 3-D CONCRETE WITH CRACKING, PLASTICITY, CREEP & AGING
C
C THE FOLLOWING ARE ABAQUS COMMON BLOCKS. THESE ARE FOR 4.5 ONLY
C
COMMON /CSP/SINT(513)
COMMON /CELGI/ IDUM(7),IEDBR,JDUM(110)
COMMON /CEL/ LGEL(75)
COMMON /COUNT/ ICOUNT(4),ACOUNT(14),JCOUNT(6),BCOUNT,KSTIF,
$ KDUM,DDUM(3),LDUM(2),EDUM(6),NDUM(8),FDUM(4),
$ MDUM(2),GDUM(3)
C
COMMON /RSDINF/ NOUT,JELNO,INT,NSTPAB,INCRAB,NPASS
COMMON /RSDPR/ IPRINT
COMMON /RSDDBG/ NBUG
LOGICAL IPRINT
C
COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVENTH,EIGHTH,NINETH,
$ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPAN
REAL*8 NINE,NINETH,MILLON
CHARACTER*8 CMAT
C
PARAMETER (IAR=78,MAXSV=57)
C
DIMENSION STRESS(NTENS),STATEV(NSTATV),PROPS(NPROPS),NPRINT(100),
$ TEPS(NTENS),DEP(NTENS),HH(NTENS,NTENS),IORDER(6),DTIM(2),
$ PROPI(100),AR(IAR),EP(6),PH(6,6),COORDS(3),DEPS(4),SIG(4),D(11),
$ LORDER(4),DE(4,4),DDEPS(4),DSIG(4),KRK1(3),KRK2(3),JORDER(6),
$ IARSV(IAR),DUM2(NTENS),DUM3(NTENS),DUM5(3,3)
C
LOGICAL FIRST(100),FIRSTE,BAD,XAGE
C
SAVE
DATA FIRST/100*.TRUE./, IORDER/1,2,3,6,5,4/,
$ MPRINT/0/, FIRSTE/.TRUE./, BAD/.FALSE./, NPRINT/100*0/,
$ LORDER/1,2,4,3/, KRK1/3*0/, KRK2/3*0/, NUMITR/1/, NCRACK/0/,
$ IARSV/12*0,1,2,3,9*0,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
$ 20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,

```

```

$ 41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57/
EPSMACH=1.E-9
NOUT=6
JPROPS=NEAR(PROPS(1))

C
C      CONVERT MATERIAL NAME TO NUMBER
C
      IF(CMAT(1:1).NE.'M')WRITE(6,888)
888  FORMAT('ERROR IN MATERIAL NAME. MUST BE MXX WHERE XX IS',
        /'NUMBER FROM 1 TO 99.')
      IF(CMAT(3:3).EQ.' ')THEN
        READ\CMAT(2:2),'(I1')MATERL
      ELSE
        READ(CMAT(2:3),'(I2')MATERL
      END IF

C
C      DETERMINE WHICH UMAT MODEL IS TO BE USED
C
      IF (JPROPS.NE.2.OR.JPROPS.NE.3)THEN
        WRITE(NOUT,977)
977  FORMAT('OERROR IN USER SUBROUTINE CARD')
      ELSEIF (JPROPS.EQ.2.OR.JPROPS.EQ.3) THEN
        XAGE=.TRUE.
        IF (NSTATV.LT.MAXSV) THEN
          WRITE(NOUT,11) JPROPS,NSTATV,MAXSV,IAR
11  FORMAT('OBAD CONTROL DATA TO UMAT.  JPROPS,NSTATV,MAXSV,IAR =',
$    I9,3I5)
          STOP 'BAD CONTROL DATA FOR UMATAGE'
        ENDIF
      ENDIF

C
C      SET CONTROL PARAMETERS FROM ABAQUS
C
      CALL ACOPDI(SINT(IEDBR+1),JELNO,1)
      NINT=LCEL(23)
      INT=LCEL(5)
      NSEC=LCEL(8)
      INCRAB=ICOUNT(1)
      NSTPAB=JCOUNT(1)
      IF (FIRSTE) THEN
        IF (NSTPAB.EQ.1.AND.INCRAB.EQ.1) THEN
          INCRMT=0
        ELSE
          INCRMT=1
        ENDIF
      CALL FLOATN
      AGEMIN=0.
      JELNO1=JELNO
      FIRSTE=.FALSE.
C      CALL SECOND(TZERO)
      NPASS=0
      WRITE(NOUT,12)

```

```

12     FORMAT('OANATECH UMAT87 VERSION - SEP 16, 1987')
    ENDIF
C
    IF (JELNO.EQ.JELNO1.AND.INT.EQ.1) THEN
C
C         FIRST ELEMENT & FIRST INTEGRATION PT ONLY
C
        NPASS=NPASS+1
C
        IF (KSTIF.EQ.1) THEN
            IF (NSTPAB.EQ.1.AND.INCRAB.EQ.1.AND.INCRMT.GT.0) THEN
                INCRMT=1
                WRITE(NOUT,21)
                PRINT 21
12         FORMAT(' WARNING NSTPAB & INCRAB = 1, BUT INCRMT .GT. 0')
            ELSE
                MPASS=NPASS
                NPASS=0
                INCRMT=INCRMT+1
                PRINT 25, NSTPAB, INCRAB
25         FORMAT(' START STEP=',I4,' INCR=',I4)
                IF (NCRACK.NE.0) WRITE(NOUT,26)
                $      (KRK2(I),I=1,3),(KRK1(I),I=1,3)
26         FORMAT(' NUMBER OF INTEGRATION POINTS IN PREVIOUS INCREMENT ',
                $      'WITH OPEN 1 2 & 3 CRACKS = ',3I5/
                $      ' NUMBER OF INTEGRATION POINTS IN PREVIOUS INCREMENT ',
                $      'WITH CLOSED 1 2 & 3 CRACKS = ',3I5)
            ENDIF
        ENDIF
C
        NT=MIN(2,INCRMT)
C
        DO 27 I=1,3
            KRK2(I)=0
27         KRK1(I)=0
C
        ENDIF
C
C     CONCRETE PROPERTIES
C
        IF (NPROPS.EQ.0) THEN
            ECONC=ZERO
            IPROPS=4
            GO TO 560
        ENDIF
C
        IFLAG=0
        IF (JPROPS.LT.10.AND.JPROPS.GT.0) IFLAG=1
        IPROPS=NPROPS-IFLAG
        IPROPS=MIN(IPROPS,13)
        IF (IPROPS.EQ.1 .AND. PROPS(1).LT.ONE) THEN
            IPROPS=0

```

```

      GO TO 560
    ENDIF
    GO TO (559,558,557,556,555,554,553,552,551,550,549,548,547),IPROPS
547 IHANOP=NEAR(PROPS(13+IFLAG))
548 IPRUM=NEAR(PROPS(12+IFLAG))
549 TIMREF=PROPS(11+IFLAG)
550 EPSHRK=PROPS(10+IFLAG)
551 CREEP=PROPS(9+IFLAG)
552 SHRINK=PROPS(8+IFLAG)
553 AGE=PROPS(7+IFLAG)
554 TREF=PROPS(6+IFLAG)
555 ALFAC=PROPS(5+IFLAG)
556 EPFRAC=PROPS(4+IFLAG)
557 CRUSH=PROPS(3+IFLAG)
558 XVC=PROPS(2+IFLAG)
559 ECONC=PROPS(1+IFLAG)
560 IPROPS=IPROPS+1
C
C      ALL VALUES ARE IN PSI, DEG F & DAY UNITS.
C
      GO TO (561,562,563,564,565,566,567,568,569,570,571,572,573,574),
      $ IPROPS
561 ECONC=3.E6
562 XVC=0.15
563 CRUSH=(ECONC/57600.)*2
564 EPFRAC=(6.7*SQRT(CRUSH))/ECONC
565 ALFAC=ZERO
566 TREF=ZERO
567 AGE=THOU
568 SHRINK=ZERO
569 CREEP=ZERO
570 EPSHRK=ZERO
571 TIMREF=ZERO
572 IPRUM=2
573 IHANOP=0
C
574 G=ECONC/(ONE+XVC)
C
      IF (ECONC.NE.ZERO) THEN
        IF (XVC.LT.ZERO.OR.XVC.GT.0.49.OR.ABS(CRUSH/ECONC).LT.1.E-4
        $ .OR.EPFRAC.LE.ZERO.OR.CRUSH.LE.ZERO) BAD=.TRUE.
      ENDIF
C
      IPRUM=MAX(1,MIN(4,IPRUM))
      IHANOP=MAX(0,MIN(2,IHANOP))
      IF (IPRUM.NE.4) IHANOP=0
C
      IF (MATERL.GT.100) GO TO 123
      IF (.NOT.FIRST(MATERL)) GO TO 123
C
      IF (AGE.LT.AGEMIN) THEN
        WRITE(NOUT,56) AGE,AGEMIN

```

```

56  FORMAT('OTHE UMAT AGE IS TOO SMALL. AGE,AGEMIN=',1P2E11.3,
$   ' DAYS')
    STOP 'UMAT AGE TOO SMALL'
    ENDIF

C
    WRITE(NOUT,57) CMAT,NPROPS,ECONC,XVC,CRUSH,EPFRAC,ALFAC,
$   TREF,AGE,SHRINK,CREEP,EPShrk,TIMREF,IPRUM,IHANOP
57  FORMAT('OUMAT PROPERTIES:/' MATERL = ',A/' NPROPS = ',
$   I5/' ECONC = ',1PE11.3,' (PSI)/' XVC = ',0PF7.3/' CRUSH = ',
$   1PE11.3,' (PSI)/' EPFRAC = ',1PE11.3,' (IN/IN)/' ALFAC = ',
$   1PE11.3,' (IN/IN/DEG)/' TREF = ',1PE11.3,' (DEG)/' AGE = ',
$   1PE11.3,' (DAYS)/' SHRINK FACTOR = ',1PE11.3/
$   ' CREEP FACTOR = ',1PE11.3/' INITIAL SHRINKAGE = ',1PE11.3,
$   ' (IN/IN)/' TIMREF = ',1PE11.3,' (DAYS)/' IPRUM = ',I5/
$   ' IHANOP = ',I5)

C
C
C
    IF (BAD) THEN
        WRITE(NOUT,70)
70  FORMAT('OBAD CONCRETE MATERIAL PROPERTIES IN UMAT')
        STOP 'BAD CONCRETE MATERIAL PROPERTIES IN UMAT'
    ENDIF

C
    FIRST(MATERL)=.FALSE.
    JPRINT=0

C
    IAE=7
    MPROPS=7 + IAE

C
C
C
    PRINTING CONTROL

    IF (NPROPS.GE.MPROPS+1) THEN
        IPR=NEAR(PROPS(MPROPS))
        IF (IPR.EQ.999) THEN
            MPROPS=MPROPS+1
            JPRINT=NEAR(PROPS(MPROPS))
            IF (JPRINT.LE.0.OR.NPROPS.LT.(MPROPS+JPRINT).AND.FIRST(MATERL))
$       THEN
                WRITE(NOUT,80) NPROPS,JPRINT,MPROPS
80      FORMAT('OBAD VALUES FOR PRINT CONTROL IN UMAT. ',
$       'NPROPS,JPRINT,MPROPS=',3I5)
                STOP 'BAD VALUE OF PRINT CONTROL IN UMAT'
            ELSE
                MPRINT=MIN(JPRINT,100)
                DO 90 I=1,MPRINT
                    MPROPS=MPROPS+1
90         NPRINT(I)=NEAR(PROPS(MPROPS))
                WRITE(NOUT,112) MPRINT,(NPRINT(I),I=1,MPRINT)
112      FORMAT('OUMAT INFORMATION PRINTED FOR THE FOLLOWING',
$       ' ELEMENTS. MPRINT = ',I5/(5X,10I8))

```

```

        ENDIF
        ENDIF
        ENDIF
        IF (MPROPS.LT.NPROPS.AND.FIRST(MATERL)) WRITE(NOUT,122)
        $ (PROPS(I),I=MPROPS+1,NPROPS)
122 FORMAT('OUNRECOGNIZED USER PROPERTIES IN UMAT.'/
        $ (1P10E11.3))
C
123 IPRINT=.FALSE.
    IF (MPRINT.GT.0.AND.INT.EQ.1) THEN
        DO 130 I=1,MPRINT
            IF (JELNO.NE.NPRINT(I)) GO TO 130
            IPRINT=.TRUE.
            GO TO 135
130 CONTINUE
135 CONTINUE
        ENDIF
C
C CALL TO AGING CREEP & CRACKING MODEL
C
    DO 142 I=1,IAR
        N=IARSV(I)
        IF (N.GT.0) THEN
            AR(I)=STATEV(N)
        ELSE
            AR(I)=ZERO
        ENDIF
142 CONTINUE
C
    IBUG=0
    IF (IPRINT) THEN
        IBUG=1
        WRITE(NOUT,146) JELNO,JELNO1,NPASS,INCRMT,KSTIF,NSTPAB,INCRAB,
        $ INT,NT,NDI
146 FORMAT('OBEFORE STRN3D CALL. JELNO,JELNO1,NPASS,INCRMT,',
        $ 'KSTIF,NSTPAB,INCRAB,INT,NT,NDI'/10I5)
    ENDIF
C
    DTIM(1)=TYME-TIMREF
    EPSMACH=1.E-14
    IF(DABS(DTIM(1)).LT.EPSMACH)NT=1
    IF (DTIM(1).LT.ZERO) THEN
        DO 120 I=1,NTENS
            STRESS(I)=ZERO
        DO 110 J=1,NTENS
110 HH(I,J)=ZERO
120 HH(I,I)=ECONC
        RETURN
    ENDIF
C
    DTIM(2)=DTIM(1)+DELTM
C

```

```

DO 721 I=1,6
AR(I)=ZERO
AR(I+6)=ZERO
AR(I+18)=ZERO
EP(I)=ZERO
721 JORDER(I)=0
N=0
DO 722 I=1,NDI
N=N+1
722 JORDER(N)=IORDER(I)
DO 723 I=1,NSHR
N=N+1
723 JORDER(N)=IORDER(I+3)
NTEN=6
DO 126 I=1,NTEN
J=JORDER(I)
IF (J.NE.0) THEN
AR(J)=STRESS(I)
AR(J+6)=STRESS(I)
AR(J+18)=TEPS(I)
EP(J)=DEP(I)
ENDIF
126 CONTINUE
C
MAXITR=NUMITR
IF (KSTIF.EQ.1) MAXITR=1
DO 127 NITER=1,MAXITR
KITER=1
IF (NITER.EQ.MAXITR) KITER=2
CALL STRN3D(AR,ALFAC,CRUSH,TREF,ECONC,EP,NT,XVC,PH,EPFRAC,TEMP,
$ DTEMP,DTIM,AGE,SHRINK,CREEP,EPSHRK,KITER,NDI,IPRUM,IHANOP,KSTIF)
127 CONTINUE
C
DO 175 I=1,NTENS
II=JORDER(I)
STRESS(I)=AR(II+6)
DO 175 J=1,NTENS
JJ=JORDER(J)
175 HH(I,J)=PH(II,JJ)
C
DO 180 I=1,IAR
N=IARSV(I)
IF (N.GT.0) STATEV(N)=AR(I)
180 CONTINUE
C
KRAK=NEAR(AR(26))
IF (KRAK.GT.0) THEN
NCRACK=1
KMOD=10
KDIV=1
DO 190 I=1,3
K=MOD(KRAK,KMOD)/KDIV

```

```

      KMOD=KMOD*10
      KDIV=KDIV*10
      IF (K.EQ.2) THEN
        KRK2(I)=KRK2(I)+1
      ELSEIF (K.EQ.1) THEN
        KRK1(I)=KRK1(I)+1
      ENDIF
190    CONTINUE
      ENDIF
C
C CRACKING REPORT
C
      IF(AR(26).NE.0.)WRITE(6,888)DTIM(2),AR(26),COORDS(1),COORDS(2),
& (AR(I),I=37,42)
888    FORMAT('T,CRK.FLG,X,Y,DIR.COS:',F7.2,9E11.4)
C
      RETURN
      END

      FUNCTION NEAR(X)
      REAL*8 X
      NEAR=NINT(X)
      RETURN
      END

      SUBROUTINE FLOATN
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$  NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTH,EIGHTH,NINETH,
$  TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
      REAL*8 NINE,NINETH,MILLON
C
      ZERO=0.0D0
      ONE=1.0D0
      TWO=ONE+ONE
      THREE=TWO+ONE
      FOUR=TWO*TWO
      FIVE=FOUR+ONE
      SIX=FIVE+ONE
      SEVEN=SIX+ONE
      EIGHT=FOUR*TWO
      NINE=THREE*THREE
      TEN=FIVE*TWO
      HUNDRD=TEN*TEN
      THOU=HUNDRD*TEN
      MILLON=THOU*THOU
      HALF=ONE/TWO
      THIRD=ONE/THREE
      FOURTH=ONE/FOUR
      FIFTH=ONE/FIVE
      SIXTH=ONE/SIX
      SEVNTH=ONE/SEVEN

```

```

EIGHTH=ONE/EIGHT
NINETH=ONE/NINE
TENTH=ONE/TEN
PIFAC=ATAN(ONE)
PI=PIFAC*FOUR
PIFAC=PIFAC/(NINE*FIVE)
PIFAC1=ONE/PIFAC
EXPN=EXP(ONE)
RETURN
END

```

```

SUBROUTINE SYMINV(H,NDIM,NN)
C
  IMPLICIT REAL*8 (A-H,O-Z)
C
  COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$  NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVENTH,EIGHTH,NINETH,
$  TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
  REAL*8 NINE,NINETH,MILLON
C
  DIMENSION H(NDIM,NDIM)
C
  IF (NN.LT.3.OR.NN.GT.6) STOP 'BAD SIZE TO SYMINV'
C
  H11=H(1,1)
  H22=H(2,2)
  H33=H(3,3)
  H12=H(1,2)
  H13=H(1,3)
  H23=H(2,3)
C
  DET=H11*(H22*H33-H23*H23)+H12*(H23*H13-H12*H33)+
$  H13*(H12*H23-H22*H13)
C
  IF (DET.LE.ZERO) THEN
    WRITE(6,10) DET,((H(I,J),J=1,3),I=1,3)
10    FORMAT(' BAD DET IN SYMINV = ',1PE11.3/(1P3E11.3))
    STOP 'BAD DET IN SYMINV'
  ENDIF
C
  H(2,1)=- (H12*H33-H23*H13)/DET
  H(3,1)=- (H12*H23-H22*H13)/DET
  H(3,2)=- (H11*H23-H12*H13)/DET
  H(1,2)=H(2,1)
  H(1,3)=H(3,1)
  H(2,3)=H(3,2)
C
  H(1,1)=(H22*H33-H23*H23)/DET
  H(2,2)=(H11*H33-H13*H13)/DET
  H(3,3)=(H11*H22-H12*H12)/DET
C
  IF (NN.GE.4) H(4,4)=ONE/H(4,4)

```

```

      IF (NN.GE.5) H(5,5)=ONE/H(5,5)
      IF (NN.GE.6) H(6,6)=ONE/H(6,6)
C
      RETURN
      END

      SUBROUTINE INVERT(AA,NDIM,NN)
C
C          GENERAL MATRIX INVERSION SUBROUTINE
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$  NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVENTH,EIGHTH,NINETH,
$  TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
      REAL*8 NINE,NINETH,MILLON
C
      DIMENSION AA(1),A(36),M(6),C(6)
C
      IF (NN.LE.0.OR.NN.GT.6) STOP 'BAD CALL TO INVERT'
C
      N=0
      DO 10 J=1,NN
      L=(J-1)*NDIM
      DO 10 I=1,NN
      N=N+1
      L=L+1
10  A(N)=AA(L)
C
      DO 90 I=1,NN
      90  M(I)=-I
      DO 140 I=1,NN
C      LOCATE LARGEST ELEMENT
      D=0.0
      DO 112 L=1,NN
      IF (M(L)) 100,100,112
100  J=L
      DO 110 K=1,NN
      IF (M(K)) 103,103,108
103  IF (ABS(D) - ABS(A(J))) 105,105,108
105  LD=L
      KD=K
      D=A(J)
108  J=J+NN
110  CONTINUE
112  CONTINUE
C      INTERCHANGE ROWS
      TEMP=-M(LD)
      M(LD)=M(KD)
      M(KD)=TEMP
      L=LD
      K=KD

```

```

        DO 114 J=1,NN
        C(J)=A(L)
        A(L)=A(K)
        A(K)=C(J)
        L=L+NN
114 K=K+NN
C      DIVIDE COLUMN BY LARGEST ELEMENT
        NR=(KD-1)*NN+1
        NH=NR+NN-1
        DO 115 K=NR,NH
115 A(K)=A(K)/D
C      REDUCE REMAINING ROWS AND COLUMNS
        L=1
        DO 135 J=1,NN
        IF (J-KD) 130,125,130
125 L=L+NN
        GO TO 135
130 DO 134 K=NR,NH
        A(L)=A(L)-C(J)*A(K)
134 L=L+1
135 CONTINUE
C      REDUCE ROW
        C(KD)=-1.0
        J=KD
        DO 140 K=1,NN
        A(J)=-C(K)/D
140 J=J+NN
C      INTERCHANGE COLUMNS
        DO 200 I=1,NN
        L=0
150 L=L+1
        IF(M(L)-I) 150,160,150
160 K=(L-1)*NN+1
        J=(I-1)*NN+1
        M(L)=M(I)
        M(I)=I
        DO 200 L=1,NN
        TEMP=A(K)
        A(K)=A(J)
        A(J)=TEMP
        J=J+1
200 K=K+1
C
        N=0
        DO 210 J=1,NN
        L=(J-1)*NDIM
        DO 210 I=1,NN
        N=N+1
        L=L+1
210 AA(L)=A(N)
C
        RETURN

```

```

END

SUBROUTINE MATCON(TEMP2,EE,YIELD,FACTR,EPSEFF,CON,ESEC,EPFRAC,
$ AGEFAC,CRUSH,ECONC,IPROP,IHANOP,DTIM,ULTTEN,BAND,TEMP1,INCRMT,
$ IHARD)
  IMPLICIT REAL*8 (A-H,O-Z)

C
  COMMON /RSDDBG/ IBUG
C
  COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
  REAL*8 NINE,NINETH,MILLON
C
  DIMENSION DTIM(2),A(4,4),X(4),DX(4),XPR(4),BAND(6)
C
C  DATA FOR ELASTIC MODULUS CURVE FIT FOR NEW HANFORD CONCRETE
C
  DATA ((A(I,J),J=I,4),I=1,4),A1,A2,A3,A4,S/
1 3.0226E-1,-3.5888E-2,-3.1970E-4,-1.0128E-2,1.0323E-2,
2 -4.6499E-6,-2.1691E-4,8.9565E-7,3.2632E-6,2.0813E-3,
3 5.3947,1.233E-1,-6.751E-3,-1.786E-1,3.5/
C
  IF (IPROP.EQ.4) GO TO 100
C
C  GENERAL, WES & OLD HANFORD CONCRETE
C
  IF (ECONC.GT.ONE) THEN
C
C  USER DEFINED CONCRETE PROPERTIES
C
    EE=ECONC*AGEFAC
    IF (CRUSH.LE.ZERO) THEN
      WRITE(6,10)
10    FORMAT('OCRUSH .LE.0 IN MATCON.  EXECUTION TERMINATED.')
```

```

C          EEFF1=2240. MICRONS
C
      IHARD=1
      EE=3.350E6*AGEFAC
      SIGULT=4650.*AGEFAC
      EPFRAC=138.8E-6
      EEFF1=2240.E-6
      E1=SIGULT/EEFF1
      EODE1=EE/E1
      ETA0=TWO-EODE1
      ETA0=MIN(ETA0,ONE)
      ETAMIN=0.05
      ETA0=MAX(ETA0,ETAMIN)
      FACTR=-0.02
      ELSEIF (ECONC.LT.-ONE) THEN
C
C          USER DEFINED ELASTIC-PLASTIC PROPERTIES WITH FRACTURE
C
      IHARD=0
      EE=-ECONC*AGEFAC
      ESEC=EE
      CON=ESEC*EE
      IF (CRUSH.LE.ZERO) THEN
        WRITE(6,10)
        STOP 'CRUSH .LE. 0 IN MATCON'
      ENDIF
      IF (EPFRAC.LT.ZERO) THEN
        WRITE(6,15)
15      FORMAT('OBAD EPFRAC IN MATCON.  EXECUTION TERMINATED. ')
        STOP 'EPFRAC .LT. 0 IN MATCON'
      ENDIF
      YIELD=CRUSH*AGEFAC
      FACTR=-0.02
      IF (IBUG.EQ.1) WRITE (6,98) IHARD,ECONC,CRUSH,EPFRAC,AGEFAC,
$  EPSEFF,EE,EEFF1,FACTR,ETA0,YIELD,SPLUS,ESEC
C
      RETURN
      ELSE
20      WRITE(6,20)
      FORMAT('OBAD ECONC IN MATCON.  EXECUTION TERMINATED. ')
      STOP 'BAD ECONC IN MATCON'
      ENDIF
C
      EPSULT=SIGULT/EE
      EEFF1=(TWO-ETA0)*EPSULT
      PSI=EPSEFF/EEFF1
      IF (PSI.LE.ETA0) THEN
        YIELD=ETA0*EEFF1*EE
        ESEC=EE
        CON=EE*EE
      ELSE
        SPLUS=EE/(ONE-ETA0*PSI+PSI**2)

```

```

      EPS=(SPLUS)**2
      YIELD=SPLUS*EPSEFF
      EPN=(YIELD/EEFF1)**2
      ETAN=(EPS-EPN)/EE
      CON=ETAN*EE
      CON=CON/(EE-ETAN)
      ETOL=EE/THOU
      IF (ABS(CON).LT.ETOL) CON=CON*ETOL
      IF (CON.LT.ETOL) CON=ETOL
      ESEC=YIELD/EPSEFF
ENDIF
C
      IF (EPSEFF.GT.EEFF1) EE=EE*((YIELD/SIGULT)**2)
      ESEC=(TWO*ESEC+EE)/THREE
C
      IF (IBUG.EQ.1) WRITE (6,98) IHARD,ECONC,CRUSH,EPFRAC,AGEFAC,
$ EPSEFF,EE,EEFF1,FACTR,ETA0,YIELD,SPLUS,ESEC
98  FORMAT (' IN MATCON: IHARD,ECONC,CRUSH,EPFRAC,AGEFAC,EPSEFF, ',
$ 'EE,EEFF1,FACTR,ETA0,YIELD,SPLUS,ESEC-',I3/(1P10E11.3))
C
      RETURN
C
      NEW HANFORD CONCRETE - MAY 1987
C
100  IHARD=0
      DTEMP=TEMP2-TEMP1
      IF (IBUG.NE.0) WRITE(6,101) EE,BANMOD,TEMP2,DTEMP,DTIM
101  FORMAT(' IN MATCON BEFORE 1ST CALL TO CRVFIT: EE,BANMOD,TEMP2, ',
$ 'DTEMP,DTIM'/(1P10E11.3))
C
      TEMMAX=EPSEFF
      IF (TEMP2.LT.TEMMAX) GO TO 170
      DTEMP=MIN(DTEMP,TEMP2-TEMMAX)
      EE=EE/MILLON
      CALL CRVFIT(EE,BAND(1),BAND(2),TEMP2,ZERO,DTEMP,ZERO,DTIM,INCRMT,
$ IHANOP,1)
      EE=EE*MILLON
      TA=MAX(ZERO,TEMP2-350.0)
      DTA=TA-MAX(ZERO,TEMP1-350.0)
      DTA=MAX(DTA,ZERO)
      CALL CRVFIT(ULTTEN,BAND(5),BAND(6),TA,ZERO,DTA,ZERO,DTIM,INCRMT,
$ IHANOP,3)
C
      TEMLOW=250.0
      TEMP3=MAX(TEMP2,TEMLOW)
      TEMP4=MAX(TEMP1,TEMLOW)
      TB=MAX(ZERO,350.0-TEMP3)
      TBM1=MAX(ZERO,350.0-TEMP4)
      DTB=TB-TBM1
      DTB=MAX(DTB,ZERO)
      CALL CRVFIT(YIELD,BAND(3),BAND(4),TA,TB,DTA,DTB,DTIM,INCRMT,
$ IHANOP,2)

```

```

C      TEMMAX=TEMP2
170    EPFRAC=ULTTEN/ECONC
      ESEC=EE
      CON=EE*EE

C      IF (IBUG.NE.0) WRITE (6,198) EE,YIELD,ULTTEN,EPFRAC,ESEC,ECONC
198    FORMAT (' MATCON(IPROP=4): EE,YIELD,ULTTEN,EPFRAC,ESEC,ECONC'/
$ (1P10E11.3))

C      RETURN
      END

      SUBROUTINE USHRNK(TIME,SHRINK)

C      IMPLICIT REAL*8 (A-H,Q-Z)

C      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVENTH,EIGHTH,NINETH,
$ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
      REAL*8 NINE,NINETH,MILLON

C      COMMON /RSDDBG/ IBUG

C      SHRINK=204.91E-6*(ONE-EXP(-0.15*TIME))+
$ 145.09E-6*(ONE-EXP(-0.0226348*TIME))
      SHRINK=SHRINK*.32
      RETURN
      END

      SUBROUTINE CRPROP(GE,DTIM,IN,CREEP,TEMP,IPROP)

C      IMPLICIT REAL*8 (A-H,O-Z)

C      COMMON /RSDDBG/ IBUG

C      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVENTH,EIGHTH,NINETH,
$ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
      REAL*8 NINE,NINETH,MILLON

C      DIMENSION DTIM(2),GE(4)

C      DTM=DTIM(1)
      DTP=DTIM(2)
      DELTM=DTP-DTM
      XK=DTM
      TS1=ZERO
      TS2=ZERO
      TIM=DTP-XK
      XKN=XK+DELTM*HALF
      IF(IPROP.NE.4) THEN

```

```

CALL UAGE(XKN,TEMP,AFAC,TEMFAC,IPROP)
CON=CREEP
CALL SHIFT1(TEMP,R,PRIME,SECOND,CON,XK,IN,IPROP)
C
C      THE FOLLOWING TWO LINES ARE TO BE USED WITH THE
C      STANDARD FIT
C
IF(AFAC.GT.1.)THEN
  A=PRIME/(AFAC**2)
  D=SECOND/(AFAC**2)
ELSE
C
C      THE FOLLOWING TWO LINES ARE TO BE USED WITH
C      LOW AGING FACTOR FOR EARLY TIME
C
  A=PRIME/AFAC
  D=SECOND/AFAC
END IF
C
ELSE
  CON=ONE
  CALL SHIFT1(TEMP,R,A,D,CON,XK,IN,IPROP)
ENDIF
TIM=MIN(DELM,0.63/R)
TS=TIM
CONST=R*TS
IF (CONST.GT.85.0) CONST=85.0
GE4=EXP(-CONST)
GE(4)=GE4
TS1=A*(ONE-GE4)
TS2=D*DELM
GE(1)=3.0*TS1
GE(2)=ZERO
IF (IN.EQ.1.AND.IPROP.NE.4) GE(2)=3.0*TS2
GE(3)=DELM
RETURN
END

SUBROUTINE SHIFT1(TEMP,R,A,D,CON,XK,IN,IPROP)
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$  NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVENTH,EIGHTH,NINETH,
$  TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
REAL*8 NINE,NINETH,MILLON
C
C      IPROP=1  CREEP DATA FOR GENERAL CONCRETE
C      IPROP=2  WES CREEP DATA FOR YOUNG CONCRETE
C      IPROP=3  CREEP DATA FOR OLD HANFORD CONCRETE - 1979
C      IPROP=4  CREEP DATA FOR NEW HANFORD CONCRETE - 1987
C

```

```

      T=TEMP
C
      IF (TEMP.LT.60.0) TEMP=60.0
C
      IF (IPROP.NE.1) GO TO 200
C
      GENERAL CONCRETE
C
      AP0=0.10425
      AP1=-0.02775
      AP2=0.0062667
      CON=ONE
      FF=(AP0+AP1*LOG10(XK))*0.625
      F=FF*EXP(AP2*T)
      F=F*CON*1.E-6
      R=0.07
      DT=T-200.0
      IF (DT.LT.ZERO) DT=ZERO
      D1=0.00075+DT*1.0E-5
      D2=0.0015
      D=MIN(D1,D2)*F
      IF (IN.EQ.1) THEN
        A=HALF*F
        R=0.6
        D=ZERO
      ELSEIF (IN.EQ.2) THEN
        A=1.5*F
        R=0.07
        D=ZERO
      ELSEIF (IN.EQ.3) THEN
        A=(0.056*DT-DT*DT*1.511E-4)*F
        R=0.0046
      ENDIF
C
      RETURN
C
200  IF (IPROP.NE.2) GO TO 300
C
      WES CONCRETE
C
      TK=(T-32.0)*FIVE/NINE+273.0
      RTK=(70.0-32.0)*FIVE/NINE+273.0
      EQRT=EXP(-4345.0/(1.98*TK))/EXP(-4345.0/(1.98*RTK))
      CONN=EQRT*1.E-6
C
C MODIFIED FOR RED RIVER MIX All (NO LINEAR TERM) 1/25/89
C
C      D=5.2477E-4*CON*CONN
      IF (IN.EQ.1) THEN
        A=CONN*CON*.13887
        R=1.7661058
        D=ZERO

```

```

ELSEIF (IN.EQ.2) THEN
  A=CONN*CON*.15890
  R=.18922563
  D=ZERO
ELSEIF (IN.EQ.3) THEN
  A=CONN*CON*.10576
  R=.05887019
  D=ZERO
ENDIF
RETURN

```

C

C MATERIAL DATA FOR OLD HANFORD CONCRETE - 1979

C

```

300 IF (IPROP.NE.3) GO TO 400
    TK=(T-32.0)*FIVE/NINE + 273.0
    EQRT=EXP(-4345.0/(1.98*TK))
    A=1.11E-4*EQRT
    D=3.8265E-7*EQRT
    R=0.23
    ET=A*SIG
    EM=D*SIG
    RETURN

```

C

C MATERIAL DATA FOR NEW HANFORD CONCRETE - 1987

C

```

400 XT=226.09 - 0.00429*T + 147.52*T**(-0.367) - 309.26*T**(-0.044)
    XT=XT*1.0E-6
    D=ZERO
    IF(IN.EQ.1) THEN
      A=0.000934*XT
      R=6.9
    ELSEIF(IN.EQ.2) THEN
      A=0.097*XT
      R=0.69
    ELSEIF(IN.EQ.3) THEN
      A=0.0957*XT
      R=0.069
    ELSEIF(IN.EQ.4) THEN
      A=0.28*XT
      R=0.0069
    ELSEIF(IN.EQ.5) THEN
      A=0.375*XT
      R=0.00069
    ELSEIF(IN.EQ.6) THEN
      A=0.348*XT
      R=0.000069
    ENDIF
    RETURN
END

```

SUBROUTINE COEF(TIME,TEMPT,AGEFAC,TEMFAC,IPROP)

C

```

C      CALCULATION OF AGE AND TEMPERATURE DEPENDENT COEFFICIENTS
C      OF THE CREEP FORMULA
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
COMMON /FLTNUM/ ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT,
$  NINE, TEN, HALF, THIRD, FOURTH, FIFTH, SIXTH, SEVENTH, EIGHTH, NINTH,
$  TENTH, HUNDRED, THOU, MILLON, PI, PIFAC, PIFAC1, EXPN
REAL*8 NINE, NINTH, MILLON
C
IF (IPROP.EQ.4) RETURN
XKN=ONE/TIME
C
C      WES CONCRETE
C
IF (IPROP.NE.2) GO TO 200
TEMP=TEMPT
IF (TEMP.LT.60.0) TEMP=60.0
T=(TEMP-32.)*FIVE/NINE
XK=ONE/28.0
FF1=ZERO
DO 10 I=1,2
F1=FF1
  A0=.201881+T*(-1.98117E-4+T*(5.31521E-5-T*4.23376E-7))
  A1=-5.41464+T*(.40337+T*(-5.36256E-3+T*2.33622E-5))
  A2=69.245+T*(-3.38663+T*(-6.42465E-3+2.34411E-4*T))
  A3=-583.108+T*(19.8205+T*(.683703-.67766E-2*T))
  A4=3356.7+T*(-83.0216+T*(-5.91922+5.39942E-2*T))
  A5=-7664.4+T*(.132439E3+T*(17.0764-.150322*T))
  FF1=A0+XK*(A1+XK*(A2+XK*(A3+XK*(A4+XK*A5))))
  T=(70.0-32.0)*FIVE/NINE
10 CONTINUE
F1=F1*1.E-6
FF1=FF1*1.E-6
ET28=ONE/F1
ERT28=ONE/FF1
TSHAPE=ET28/ERT28
TM=ONE/XKN-ONE
IF(TM.LT.0.)TM=0.
ETATAU=1.E6*(3.58802*(ONE-EXP(-0.03250502*TM))+1.29715*
. (ONE-EXP(-0.40756288*TM))+.332774*(ONE-EXP(-2.649*TM))+
. .6)*TSHAPE
IF(TM.EQ.0.)THEN
  ETATAU=ONE/XKN*ETATAU
END IF
TM=TWO
ETA3=1.E6*(3.58802*(ONE-EXP(-0.03250502*TM))+1.29715*
. (ONE-EXP(-0.40756288*TM))+.332774*(ONE-EXP(-2.649*TM))+
. 0.6)*TSHAPE
AGEFAC=ETATAU/ETA3
TEMFAC=TSHAPE
RETURN

```

```

C
C      REGULAR & OLD HANFORD CONCRETE
C
200  ALFA=0.926+4.444*XKN
      ESTAR=ET28/ALFA
      AGEFAC=ESTAR/ERT28
      F1=F1*ALFA
C
      T=(TEMP-32.)*FIVE/NINE
      BETA=0.56+12.245*XKN
      XK= ONE/28.0
      TIME=ONE/XK
C
15  AP0=-.076945+T*(7.70542E-3+T*(-8.38733E-5+T*3.82484E-7))
      AP1= 14.0236+T*(-.713801+T*( 1.12008E-2-5.71309E-5*T))
      AP2=-422.681+T*( 21.8111+T*(-.347742+1.78503E-3*T))
      AP3= 6016.54+T*(-304.577+T*( 4.84830-2.48844E-2*T))
      AP4=-39844.4+T*( 1986.93+T*(-31.4952+.161156*T))
      AP5= 98581.7+T*(-4865.40+T*( 76.8028-.391529*T))
      F2= 1.E-6*(AP0+XK*(AP1+XK*(AP2+XK*(AP3+XK*(AP4+XK*AP5))))
      IF (F2.GT.ZERO) GO TO 20
      TIME=TIME+HALF
      XK=ONE/TIME
      GO TO 15
20  CONTINUE
      F2=F2*BETA
      RETURN
      END

      SUBROUTINE UAGE(TIME,TEMPT,AGEFAC,TEMFAC,IPROP)
C
C      CALCULATION OF AGING & TEMPERATURE FACTORS
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$      NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$      TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPAN
      REAL*8 NINE,NINETH,MILLON
C
      S1(X,A)=ONE-THREE*(X/A)**2+TWO*(X/A)**3
      S2(X,A)=X-TWO*X*(X/A)+X*(X/A)**2
      S3(X,A)=THREE*(X/A)**2-TWO*(X/A)**3
      S4(X,A)=-X*(X/A)+X*(X/A)**2
C
      IF (IPROP.EQ.4) RETURN
C
      LINEAR VARIATION OF AGEFAC I  M 0.033 AT .25 DAYS TO
      STANDARD VALUE AT 1 DAY (0.8)
C
      IF (TIME.LT.ONE) THEN
C
      CALL COEF(ONE,TEMPT,AFAC1,TEMFAC,IPROP)

```

```

C      AFACO=HUNDRD/(THREE*THOU)
C      TIME0=0.25
C      AGEFAC=AFACO+(AFAC1-AFACO)*(TIME-TIME0)/(ONE-TIME0)
C      IF (TIME.LE.TIME0) AGEFAC=AFACO
C      CALL COEF(TIME,TEMPT,AGEDUM,TEMFAC,IPROP)
C      ELSE
C      CALL COEF(TIME,TEMPT,AGEFAC,TEMFAC,IPROP)
C      ENDIF
C
C      CUBIC FIT OF AGEFAC FROM 0.1 AT 0.2 DAYS TO 1.0 AT 3 DAYS
C
C      IF (TIME.LT.THREE) THEN
C      TYME=TIME-0.2
C      XLEN=THREE-0.2
C      AGEFAC=0.1*S1(TYME,XLEN)+0.8*S2(TYME,XLEN)+S3(TYME,XLEN)+
C      $ 0.03*S4(TYME,XLEN)
C      CALL COEF(TIME,TEMPT,AGEDUM,TEMFAC,IPROP)
C      ELSE
C      CALL COEF(TIME,TEMPT,AGEFAC,TEMFAC,IPROP)
C      ENDIF
C      RETURN
C      END

SUBROUTINE CRVFIT(PBAND,P,TSTRIN,TA,TBIN,DTA,DTBIN,DTIM,
$ INCRMT,IFIT,ICRV)
RETURN
END

SUBROUTINE STRN3D(AR,ALFAC,CRUSH,TREF,ECONC,DEP,INCRMT,XVC,PH,
$ EPFR,TEMP1,DTEMP,DTIME,AGE,SHRINK,CREEP,EPShrk,ITER,NDI,IPROP,
$ IHANOP,KSTIF)
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVENTH,EIGHTH,NINETH,
$ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
C      REAL*8 NINE,NINETH,MILLON
C
C      COMMON /RSDDBG/ IBUG
C      COMMON /RSDINF/ NOUT,JELNO,INT,NSTPAB,INCRAB,NPASS
C
C      DIMENSION EPP(6),DTIM(2),AR(1),PH(6,6),GE(4,6),EP(6),DTIME(2),
$ STRESS(6),DEP(6),BAND(6)
C
C      LOGICAL ABAQUS
C
C      DATA ABAQUS/.TRUE./
C
C      ECONC      = ELASTIC MODULUS
C      XVC         = POISSON'S RATIO
C      CRUSH       = COMPRESSIVE STRENGTH

```

```

C   ALFAC   = COEFFICIENT OF CONCRETE THERMAL EXPANSION
C   TREF    = REFERENCE (STRESS-FREE) TEMPERATURE
C   EPFRAC  = TENSION FRACTURE STRAIN
C   TEMP1   = TEMPERATURE AT START OF STEP
C   DTEMP   = CHANGE IN TEMPERATURE
C   DTIME(1)= TIME AT START OF STEP
C   DTIME(2)= TIME AT END OF STEP
C   PH      = TANGENT CONSTITUTIVE MATRIX
C   INCRMT  = LOAD STEP NUMBER
C   ITER    = 1  INTERMEDIATE ITERATION
C             2  FINAL ITERATION
C   KSTIF   = 1  FOR 1ST ITERATION WHERE DEP=0
C             > 1 FOR SUBSEQUENT ITERATIONS
C   ABAQUS  = .TRUE.  MEANS TOTAL STRAINS ARE STORED IN AR
C             .FALSE. MEANS MECHANICAL STRAINS ARE STORED IN AR
C
C   AR      = STATE VARIABLES
C   (1-6)   STRESSES AT START OF STEP
C   (7-12)  STRESSES AT END OF STEP
C   (13-15) CLOSED CRACK STRAIN (LOCAL)
C   (19-24) TOTAL STRAINS
C   (25)    PLASTICITY FLAG
C   (26)    CRACKING FLAG
C   (27)    EFFECTIVE STRAIN (MAX TEMP FOR IPROP=4)
C   (28)    MODULUS
C   (29)    YIELD STRESS (ULTIMATE COMPRESSION FOR IPROP=4)
C   (30)    TENSION STRENGTH FOR IPROP=4 ONLY (ULTTEN)
C   (31-36) BAND HISTORIES FOR IPROP=4 ONLY
C   (37-42) DIRECTION COSINES OF FIRST TWO CRACKS
C   (43-78) CREEP HISTORY PARAMETERS
C
C   STRESS & STRAIN ORDERING: 11,22,33,23,31,12
C   SHEAR STRAINS ARE GAMMAS: DU/DY+DV/DX
C
C   NCOS=36
C   NCREEP=42
C   JBAND=6
C   NBAND=30
C   JCREEP=3
C   IF (IPROP.EQ.4) JCREEP=6
C   IF (CREEP.LE.1.0E-6) JCREEP=0
C   IF (IBUG.NE.0) WRITE(NOUT,13) INCRMT,ITER,NDI,IPROP,IHANOP,JCREEP
13  FORMAT(' START STRN3D: INCRMT,ITER,NDI,IPROP,IHANOP,JCREEP=',6I5)
C   NTENS=6
C   EPFRAC=EPFR
C   TEMP2=TEMP1+DTEMP
C   AVGTEM=TEMP1+HALF*DTEMP
C   DELTM=DTIME(2)-DTIME(1)
C   IF (JCREEP.NE.0.AND.DELTM.LT.1.E-6) DELTM=1.E-6
C   DTIM(1)=DTIME(1)+AGE
C   DTIM(2)=DTIM(1)+DELT
C   EPSEFF=AR(27)

```

```

AGEFAC=ONE
DAGE=ONE

C
C      1ST INCREMENT INITIALIZATION
C
IF (INCRMT.EQ.1) THEN
  DELTM=DTIM(2)

C
C      AGING (1ST INCREMENT)
C
IF (AGE.LT.999.0) THEN
  CALL UAGE(DTIM(2),TEMP1,AGEFAC,TEMFAC,IPROP)
  AGEFAC=AGEFAC*TEMFAC
ENDIF

C
C      PRESET OPEN CRACKS FOR NDI .LT. 3
C      ONLY 1ST TWO VECTOR DIRECTIONS SET
C
N=NCOS
DO 50 I=1,6
  N=N+1
50  AR(N)=ZERO
  AR(NCOS+1)=ONE
  AR(NCOS+5)=ONE
  KRACK=0
  DO 60 I=NDI+1,3
60  KRACK=KRACK+2*(10**(I-1))
  AR(26)=KRACK
ELSE

C
C      AGING (GENERAL INCREMENT)
C
IF (AGE.LT.999.0) THEN
  CALL UAGE(DTIM(2),TEMP2,AGEFAC,TEMFAC,IPROP)
  AGEFAC=AGEFAC*TEMFAC
  CALL UAGE(DTIM(1),TEMP1,AGEM1,TEMFAC,IPROP)
  AGEM1=AGEM1*TEMFAC
  DELAGE=AGEFAC-AGEM1
  DAGE=ONE+DELAGE/AGEM1
ENDIF
ENDIF

C
C      CONCRETE PROPERTIES
C
IF (IPROP.EQ.4) THEN
  EI=AR(28)
  YIELD=AR(29)
  ULTFEN=AR(30)
  N=NBAND
  DO 61 I=1,JBAND
  N=N+1
61  BAND(I)=AR(N)

```

```

ENDIF
C
CALL MATCON(TEMP2,EI,YIELD,FACTR,EPSEFF,CON,ESEC,EPFRAC,AGEFAC,
$ CRUSH,ECONC,IPROP,IHANOP,DTIM,ULTTEN,BAND,TEMP1,INCRMT,IHARD)
YIELD=YIELD*SQRT(ONE+THREE*FACTR)
IF (INCRMT.EQ.1) THEN
    AR(28)=EI
    AR(29)=YIELD
ENDIF
IF (IPROP.EQ.4.AND.ITER.NE.1) THEN
    AR(27)=EPSEFF
    AR(28)=EI
    AR(29)=YIELD
    AR(30)=ULTTEN
    N=NBAND
    DO 62 I=1,JBAND
        N=N+1
62    AR(N)=BAND(I)
ENDIF
C
C    INCREMENTAL SHRINKAGE (TOTAL FOR 1ST INCREMENT)
C
SHRINK1=ZERO
SHRINK2=ZERO
IF (AGE.LT.999.0.AND.IPROP.NE.4) THEN
    CALL USHRNK(DTIM(1),SHRINK1)
    CALL USHRNK(DTIM(2),SHRINK2)
ENDIF
C
EXPND1=-SHRINK*SHRINK1-EPSHRK+ALFAC*(TEMP1-TREF)
IF (INCRMT.EQ.1) EXPND1=ZERO
EXPND2=-SHRINK*SHRINK2-EPSHRK+ALFAC*(TEMP2-TREF)
ADT=EXPND2-EXPND1
C
C    SET CREEP PARAMETERS
C
GE(2,1)=ZERO
DO 130 J=1,JCREEP
    DO 120 I=1,NTENS
120    GE(I,J)=ZERO
    CALL CRPROP(GE(1,J),DTIM,J,CREEP,AVGTEM,IPROP)
130    CONTINUE
C
IF (IBUG.NE.0) THEN
C    WRITE(NOUT,131) AGE,DTIM(1),DTIM(2),DELT, TREF,TEMP1,DTEMP,
C    $ TEMFAC,AGEM1,DELAGE,AGEFAC,DAGE,DSHRNK,ADT,EPSHRK,SHRINK,SHRINK1,
C    $ SHRINK2,EXPND1,EXPND2
C131    FORMAT(' STRN3D AFTER 130: AGE,DTIM(1),DTIM(2),DELT, TREF,TEMP1',
C    $ ',DTEMP,TEMFAC,AGEM1,DELAGE',' AGEFAC,DAGE,DSHRNK,ADT,',
C    $ 'EPSHRK,SHRINK,SHRINK1,SHRINK2,EXPND1,EXPND2'/(1P10E11.3))
    WRITE(NOUT,132) (AR(I),I=1,NCOS)
132    FORMAT(' AR'/(1P10E11.3))

```

```

ENDIF
C
C      DEP - INCRMENT IN STRAIN (DU/DX)
C      EP  - INCRMENT IN MECHANICAL STRAIN (DU/DX-ADT)
C      EPP - OLD TOTAL MECHANICAL STRAIN (DU/DX-ADT) GOING TO CONC3D
C      AR(1-5) & STRESS - STRESSES AT START OF INCREMENT
C      AR(7-12)          - STRESSES AT END OF INCREMENT
C      AR(19-24)         - TOTAL MECHANICAL STRAINS
C
C      IF (.NOT.ABAQUS) EXPND1=ZERO
C      SUM=ZERO
C      DO 140 I=1,3
C      STRESS(I)=AR(I)
C      STRESS(I+3)=AR(I+3)
C      EP(I)=DEP(I)-ADT
C      EP(I+3)=DEP(I+3)
C      EPP(I)=AR(I+18)-EXPND1
140  EPP(I+3)=AR(I+21)
C      IF (IBUG.NE.0) WRITE(NOUT,141) EP,EPP,STRESS
141  FORMAT(' STRN3D AFTER 140: EP/EPP/STRESS/'/(1P6E11.3))
C
C      CALL CONC3D(AR,PH,EPFRAC,EP,EPP,GE,EI,CON,FACTR,XVC,YIELD,
C      $ ITER,STRESS,NTENS,JCREEP,NCOS,NCREEP,KSTIF,ECONC,NDI,DEP,
C      $ DAGE,TEMP2,AGEFAC,CRUSH,IPROP,IHANOP,DTIM,ULTTEN,BAND,TEMP1,
C      $ INCRMT,IHARD,ABAQUS)
C
C      RETURN
C      END

C      SUBROUTINE CONC3D(AR,PH,EPFRAC,EP,EPP,GE,EI,CON,FACTR,XVC,YIELD,
C      $ ITER,STRESS,NTENS,JCREEP,NCOS,NCREEP,KSTIF,ECONC,NDI,DEP,
C      $ DAGE,TEMP2,AGEFAC,CRUSH,IPROP,IHANOP,DTIM,ULTTEN,BAND,TEMP1,
C      $ INCRMT,IHARD,ABAQUS)
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
C      $ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTNTH,EIGHTH,NINETH,
C      $ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
C      REAL*8 NINE,NINETH,MILLON
C
C      COMMON /RSDDBG/ IBUG
C      COMMON /RSDINF/ NOUT,JELNO,INT,NSTPAB,INCRAB,NPASS
C
C      DIMENSION ECC(6),EPP(6),BF(6),HH(6,6),Q(6,6),DF(6),SR(6),STN(6),
C      $ HK(6,6),AS(6,6),SIJ(6),H2(6,6),HB(6,6),DIJ(6),GE(4,6),AR(1),
C      $ PH(6,6),EP(6),STR(6),TF(6),SG(6),H(6,6),KRK(3),PSTRS(6),PSTRN(6),
C      $ PSTRNO(6),DPSTRN(6),IPERM(3),A(3,3),STRESS(6),EPSFAC(6),TAU(6),
C      $ SIGFAC(6),PSTRSO(6),ECLOSE(3),KOC(3),DEP(6)
C
C      LOGICAL PLAST,ABAQUS
C

```

```

DATA IPERM/2,3,1/, SQ3/1.732/
C
C      NPL(1) THE PLASTICITY FLAGS
C
C      NPL = 0  MEANS STEP IS ELASTIC
C             1  MEANS STEP IS PLASTIC
C             2  MEANS ELASTIC PREDICTOR IS PLASTIC BUT STEP ELASTIC
C             3  MEANS STEP IS ELASTIC BUT CRACKING STRESSES PLASTIC
C
C      KRK = K1 + 10*K2 + 100*K3
C
C      K1,K2,K3 .EQ. 0 MEANS POINT IS UNCRACKED
C                   1 MEANS PREVIOUSLY OPENED CRACK IS CLOSED
C                   2 MEANS CRACK IS OPEN
C
C      RETRIVE TWO DIRECTIONS OF PREVIOUS CRACKING
C
C      N=NCOS
C      DO 40 J=1,2
C      DO 40 I=1,3
C      N=N+1
40  A(I,J)=AR(N)
C
C      KRKFLG=NEAR(AR(26))
C      KRKOLD=MOD(KRKFLG,10000)
C
C      COMPUTE THIRD CRACK DIRECTION
C
C      CALL CROSS(A(1,1),A(1,2),A(1,3),DET,IPERM)
C
C      STORE PREVIOUS STEP CRACK STATUS
C
C      MMOD=10
C      MDIV=1
C      DO 55 I=1,3
C      KRK(I)=MOD(KRKOLD,MMOD)/MDIV
C      MMOD=MMOD*10
55  MDIV=MDIV*10
C      K123=0
C      NK=0
C      NOPEN=0
C      NCLOSE=0
C
C      ESTABLISH STIFFNESS(ECC), STRESS(SIGFAC) & STRAIN(EPSFAC)
C      FACTORS FOR CRACKING FOR NORMAL DIRECTIONS
C
C      DO 80 I=1,3
C      K=KRK(I)
C      K123=MAX(K123,K)
C      IF (K.NE.0) NK=NK+1
C      IF (K.EQ.2) THEN
C          NOPEN=NOPEN+1

```

```

      ECC(I)=1.E-3
      SIGFAC(I)=ZERO
      EPSFAC(I)=ZERO
    ELSE
      IF (K.EQ.1) NCLOSE=NCLOSE+1
      ECC(I)=ONE
      SIGFAC(I)=1.E20
      EPSFAC(I)=ONE
    ENDIF
C
C      CHECK FOR CONSISTENCY OF CRACKING ORDER
C
    IF (I.GT.1.AND.K.NE.0) THEN
      DO 70 II=1,I-1
      IF (KRK(II).EQ.0) THEN
        KRK(II)=K
        KRK(I)=0
        DO 60 J=1,3
          SUM=A(J,I)
          A(J,I)=A(J,II)
60      A(J,II)=SUM
          KRKOLD=KRK(1)+10*KRK(2)+100*KRK(3)
          GO TO 50
        ENDIF
70      CONTINUE
      ENDIF
80      CONTINUE
C
C      ROTATE OLD & NEW TOTAL MECHANICAL STRAINS
C      THESE STRAINS ARE NOT PURIFIED
C      TO LOCAL (PRINCIPAL OR CRACKED) DIRECTIONS
C
      DO 85 I=1,NTENS
      STN(I)=E'P(I)
85      EPP(I)=EPP(I)+EP(I)
      CALL PVAL3D(1,NK,EPP,PSTRN,A)
C
C      STORE NEW PRINCIPAL DIRECTIONS
C
      N=NCOS
      DO 88 J=1,2
      DO 88 I=1,3
      N=N+1
88      AR(N)=A(I,J)
      CALL PVAL3D(1,3,STN,PSTRNO,A)
C
C      ESTABLISH STIFFNESS(ECC), STRESS(SIGFAC) & STRAIN(EPSFAC)
C      FACTORS FOR CRACKING FOR SHEAR
C
      SIGO=EI*EPFRAC
      DO 90 K=1,3
      KP3=K+3

```

```

I=IPERM(K)
J=IPERM(I)
KIJ=MAX(KRK(I),KRK(J))
IF (KIJ.EQ.2) THEN
  EPSFAC(KP3)=ZERO
  SIGFAC(KP3)=SIGO/MAX(ONE,ABS(PSTRN(K)))/EPFRAC)
  EPPMAX=MAX(PSTRN(I),PSTRN(J))
  GRATIO=MAX(EPPMAX/EPFRAC,ONE)
  ECC(KP3)=0.4/GRATIO
ELSE
  SIGFAC(KP3)=1.E20
  EPSFAC(KP3)=ONE
  ECC(KP3)=ONE
ENDIF
90  CONTINUE
   IF (IBUG.NE.0) WRITE(NOUT,95) KRKOLD,KRK,K123,NOPEN,NCLOSE,EPP,
$  PSTRN,SIGFAC,EPFAC,ECC
95  FORMAT(' CONCD3D AFTER 90: KRKOLD,K1,K2,K3,K123,NOPEN,NCLOSE/EPP/ ',
$  ' PSTRN/SIGFAC/EPFAC/ECC=',7I5/(1P12E11.3))
C
C      INITIALIZATION
C
XVCP1=ONE+XVC
DO 110 I=1,NTENS
  TF(I)=ZERO
  DF(I)=ZERO
  BF(I)=ZERO
  STN(I)=ZERO
  STR(I)=ZERO
DO 110 J=1,NTENS
  HH(I,J)=ZERO
  HB(I,J)=ZERO
  H(I,J)=ZERO
  H2(I,J)=ZERO
  Q(I,J)=ZERO
110 HK(I,J)=ZERO
DO 130 I=1,3
DO 120 J=1,3
  H2(I,J)=-XVC/THREE
120 HK(I,J)=-XVC
  H2(I,I)=ONE/THREE
  HK(I,I)=ONE
  H2(I+3,I+3)=TWO*XVCP1/THREE
130 HK(I+3,I+3)=TWO*XVCP1
C
  C1=XVC*EI/(XVCP1*(ONE-TWO*XVC))
  C2=(ONE-XVC)*EI/(XVCP1*(ONE-TWO*XVC))
  C3=HALF*EI/XVCP1
  H(1,1)=C2
  H(1,2)=C1
  H(2,1)=C1
  H(1,3)=C1

```

```

      H(3,1)=C1
      H(2,2)=C2
      H(2,3)=C1
      H(3,2)=C1
      H(3,3)=C2
      H(4,4)=C3
      H(5,5)=C3
      H(6,6)=C3
C
C      MODIFICATIONS TO CONSTITUTIVE MATRIX DUE TO CHANGE IN MODULUS
C      AND CREEP
C
      DELTAE=EI-AR(28)
      IF(AR(28).LT.0.01E6)DEE=-DELTAE/(EI*EI+ONE)
      IF(AR(28).GE.0.01E6)DEE=-DELTAE/(EI*EI+ONE)*EI/AR(28)
      GEN=ZERO
      IF (JCREEP.NE.0) GEN=GE(2,1)
      DO 180 J=1,JCREEP
180    GEN=GEN+GE(1,J)
      DO 190 I=1,NTENS
      DO 190 J=1,NTENS
      PH(I,J)=H(I,J)
      HH(I,J)=HK(I,J)
190    AS(I,J)=DEE*HK(I,J)+GEN*H2(I,J)
C
C      MODIFICATIONS DUE TO CRACKING
C
      IF (K123.NE.2) GO TO 250
      DO 220 K=1,3
      IF (KRK(K).EQ.2) THEN
        DO 210 L=1,3
        HH(L,K)=ZERO
210    HH(K,L)=ZERO
      ENDIF
      I=IPERM(K)
      J=IPERM(I)
      KIJ=MAX(KRK(I),KRK(J))
      IF (KIJ.EQ.2) THEN
        HH(K+3,K+3)=TWO/ECC(K+3)
      ELSE
        HH(K+3,K+3)=TWO*XVCP1
      ENDIF
220    HH(K,K)=ONE/ECC(K)
      DO 230 I=1,NTENS
      DO 230 J=1,NTENS
230    HK(I,J)=HH(I,J)
      IF (IBUG.NE.0) WRITE(NOUT,236) ((HK(I,J),J=1,6),I=1,6)
236    FORMAT(' CONC3D AFTER 230: HK'/(1P6E11.3))
C
      CALL SYMINV(HH,6,NTENS)
C
      DO 240 I=1,NTENS

```

```

DO 240 J=1,NTENS
AS(I,J)=DEE*HK(I,J)+GEN*H2(I,J)
240 H(I,J)=HH(I,J)*EI
C
250 CALL TRNS3D(A,Q,HB)
C
IF (K123.NE.2) GO TO 310
DO 270 K=1,NTENS
DO 270 L=1,NTENS
AS(K,L)=ZERO
HH(K,L)=ZERO
DO 270 M=1,NTENS
HH(K,L)=HH(K,L)+HK(K,M)*HB(M,L)
270 AS(K,L)=AS(K,L)+H(K,M)*Q(M,L)
DO 280 K=1,NTENS
DO 280 L=1,NTENS
HK(K,L)=ZERO
H(K,L)=ZERO
DO 280 M=1,NTENS
HK(K,L)=HK(K,L)+HB(M,K)*HH(M,L)
280 H(K,L)=H(K,L)+Q(M,K)*AS(M,L)
DO 290 K=1,NTENS
DO 290 L=1,NTENS
290 AS(K,L)=DEE*HK(K,L)+GEN*H2(K,L)
C
C      STRESSES DUE TO CREEP
C
310 N=NCREEP
DO 320 J=1,JCREEP
DO 320 I=1,NTENS
N=N+1
320 TF(I)=TF(I)+GE(4,J)*AR(N)*GE(3,J)
IF (IBUG.NE.0) WRITE(6,321) DEE,GEN,TF,GE
321 FORMAT(' CONC3D AFTER 320: DEE,GEN/TF/GE=',1P3E11.3/
$ 1P6E11.3/(1P12E11.3))
IF (ABS(DEE).LT.1.E-9.AND.GEN.LT.1.E-9) GO TO 410
DO 350 I=1,NTENS
DO 350 J=1,NTENS
HH(I,J)=ZERO
DO 350 K=1,NTENS
350 HH(I,J)=HH(I,J)+H(I,K)*AS(K,J)
DO 360 I=1,NTENS
360 HH(I,I)=HH(I,I)+ONE
IF (IBUG.NE.0) WRITE(6,361) H,AS,HH
361 FORMAT(' CONC3D AFTER 360: H/AS/HH'/(1P6E11.3))
C
CALL INVERT (HH,6,NTENS)
C
DO 370 I=1,NTENS
DO 370 J=1,NTENS
PH(I,J)=ZERO
DO 370 K=1,NTENS

```

```

370  PH(I,J)=PH(I,J)+HH(I,K)*H(K,J)
      DO 380 I=1,NTENS
      DO 380 J=1,NTENS
380  H(I,J)=PH(I,J)
      IF (IBUG.NE.0) WRITE(6,381) PH
381  FORMAT(' CONC3D AFTER 380: PH'/(1P6E11.3))
C
410  DO 420 I=1,NTENS
      BF(I)=ZERO
      DO 420 J=1,NTENS
      BF(I)=BF(I)-HK(I,J)*AR(J)*DEE-H2(I,J)*TF(J)
420  PH(I,J)=H(I,J)
C
C      ELASTIC PREDICTION OF THE NEW STRESS STATE
C
      SIGM=ZERO
      DO 430 I=1,NTENS
      SUM=STRESS(I)
      DO 425 J=1,NTENS
425  SUM=SUM+H(I,J)*(EP(J)+BF(J))
      IF (I.LE.3) SIGM=SIGM+SUM
430  SG(I)=SUM
      SIGM=SIGM/THREE
      IF (IBUG.NE.0) WRITE(NOUT,431) ((H(I,J),J=1,6),SG(I),EP(I),
$ BF(I),I=1,6)
431  FORMAT(' CONC3D AFTER 430: H,SG,EP,BF'/(1P9E11.3))
C
C      PLASTICITY CALCULATIONS
C      SKIP FOR STIFFNESS ONLY RECOVERY ( EP = 0 )
C
      IF (KSTIF.EQ.1) GO TO 1155
C
C      GET UPDATED PROPERTIES BASED ON ELASTIC PREDICTOR STRESSES
C      AND UPDATED EFFECTIVE STRAIN
C
      DO 450 I=1,3
450  ECLOSE(I)=AR(I+12)
      CALL PVAL3D(0,3,STRESS,PSTRSO,A)
      IF (IHARD.NE.0) THEN
      CALL PVAL3D(0,3,SG,PSTRS,A)
      ECONCA=ABS(ECONC)
      CALL ESTRN(NTENS,XVC,EI,PSTRN,PSTRS,EPFAC,NOPEN,EPSEFF,KRK,
$ ECLOSE,ECONCA)
      IF (SIGM.GT.ZERO) EPSEFF=ZERO
      CALL MATCON(TEMP2,EI,YIELD,FACTR,EPSEFF,CON,ESEC,EPFRAC,AGEFAC,
$ CRUSH,ECONC,IPROP,IHANOP,DTIM,ULTTEN,BAND,TEMP1,INCRMT,IHARD)
      YIELD=YIELD*SQRT(ONE+THREE*FACTR)
      ENDIF
C
C      TEST YIELD CONDITION - IF EITHER THE OLD OR NEW YIELD
C      CONDITION IS EXCEEDED THE STEP IS PLASTIC
C

```

```

NPL=NEAR(AR(25))
NPL1=NPL/100
NPL=MOD(NPL,100)
ECONCA=ABS(ECONC)
EK=AR(29)/SQ3
EK2=EK*EK
PP=ZERO
CALL DEVIAT(SG,SIJ,SIGM)
C
IF (SIGM.GT.ZERO) THEN
  NPL1=0
  YIELD=AR(29)*DAGE
  EI=AR(28)*DAGE
  GO TO 890
ENDIF
CALL YFUN(SG,SIJ,SIGM,FACTR,EK2,K123,RADFAC,PLAST)
EK=YIELD/SQ3
EK2=EK*EK
NPL1=0
IF (PLAST) NPL1=1
CALL YFUN(SG,SIJ,SIGM,FACTR,EK2,K123,RADFAC,PLAST)
IF (PLAST) THEN
  NPL1=1
ELSE
  IF (NPL1.NE.0) NPL1=2
  IF (NPL1.EQ.0) THEN
    YIELD=AR(29)*DAGE
    EI=AR(28)*DAGE
  ENDIF
  GO TO 890
ENDIF
C
CALL RADRET(SG,SIJ,SIGM,K123,RADFAC)
IF (IBUG.NE.0) WRITE(6,461) NPL,NPL1,RADFAC,SG
461  FORMAT(' IN CONC3D AFTER 460: NPL,NPL1,RADFAC,SG=',
$ 3I3,1PE11.3/(1P6E11.3))
C
C      CURRENT INCREMENT IS PLASTIC
C
600  FA=FACTR
    C11=TWO/THREE*CON
    SUM=THREE*SIGM
    FA=TWO*FA*SUM
    RT=ONE-PP
    DO 610 I=1,3
      STR(I)=SIJ(I)+FA
610  STR(I+3)=SIJ(I+3)
    IF (K123.NE.2) GO TO 670
    DO 620 K=1,NTENS
      DIJ(K)=ZERO
    DO 620 L=1,NTENS
620  DIJ(K)=DIJ(K)+HB(K,L)*SG(L)

```

```

DO 630 M=1,NTENS
630 DIJ(M)=DIJ(M)*EPSFAC(M)
DO 640 I=1,NTENS
640 DIJ(I)=DIJ(I)
DO 650 K=1,NTENS
STR(K)=ZERO
DO 650 L=1,NTENS
650 STR(K)=STR(K)+Q(L,K)*DIJ(L)
SUM=STR(1)+STR(2)+STR(3)
DO 660 K=1,3
660 STR(K)=STR(K)-SUM*(ONE/THREE-TWO*FACTR)
C
670 DO 680 I=1,NTENS
STN(I)=ZERO
DO 680 J=1,NTENS
680 STN(I)=STN(I)+PH(I,J)*STR(J)
RS=ZERO
DO 690 I=1,NTENS
RS=RS+STR(I)*STN(I)+CI1*STR(I)*STR(I)
DO 690 J=1,NTENS
690 HH(I,J)=STN(I)*STN(J)
IF (IBUG.NE.0) WRITE(NOUT,691) NPL,NPL1,STR,STN,CI1,RT,RS,
$ SUM
691 FORMAT(' CONG3D AFTER 690: NPL,NPL1/STR,STN/CI1,RT,RS,SUM',
$ 3I3/(1P12E11.3))
IF (RS.EQ.ZERO) RS=ONE
DO 700 I=1,NTENS
DO 700 J=1,NTENS
700 PH(I,J)=PH(I,J)-HH(I,J)*RT/RS
C
C      MODIFY CREEP & PLASTIC CONSTITUTIVE MATRIX FOR CRACKING
C
IF (K123.NE.2) GO TO 890
DO 730 I=1,NTENS
DO 730 J=1,NTENS
SUM=ZERO
DO 720 K=1,NTENS
720 SUM=SUM+HB(I,K)*PH(K,J)
730 H(I,J)=SUM
DO 750 I=1,NTENS
DO 750 J=1,NTENS
SUM=ZERO
DO 740 K=1,NTENS
740 SUM=SUM+H(I,K)*HB(J,K)
750 HK(I,J)=SUM
DO 780 K=1,3
IF (KRK(K).NE.2) GO TO 780
DO 770 I=1,NTENS
IF (I.EQ.K) GO TO 770
DO 760 J=1,NTENS
IF (J.EQ.K) GO TO 760
HK(I,J)=HK(I,J)-HK(I,K)*HK(K,J)/HK(K,K)

```

```

760  CONTINUE
770  CONTINUE
780  CONTINUE
      DO 790 I=1,NTENS
      DO 790 J=1,NTENS
790  H(I,J)=HK(I,J)
      DO 810 I=1,NTENS
      DO 810 J=1,NTENS
      SUM=ZERO
      DO 800 K=1,NTENS
800  SUM=SUM+H(I,K)*Q(K,J)
810  HH(I,J)=SUM
      DO 830 I=1,NTENS
      DO 830 J=1,NTENS
      SUM=ZERO
      DO 820 K=1,NTENS
820  SUM=SUM+Q(K,I)*HH(K,J)
830  PH(I,J)=SUM
      C
890  IF (KSTIF.EQ.1) RETURN
      C
      C      ROTATE STRESSES TO LOCAL (PRINCIPAL OR CRACKED) DIRECTIONS
      C
      CALL PVAL3D(0,3,SG,PSTRS,A)
      C
      C      ROTATE MECHANICAL STRAIN INCREMENTS TO LOCAL
      C
      CALL PVAL3D(1,3,EP,DPSTRN,A)
      C
      C      CRACKING CRITERIA FOR NEW CRACK STATUS
      C      USING INTERACTION CURVE BASED ON ORIGINAL MODULUS
      C
      CRKCLO=-.1E-6
      NOPEN=0
      NCLOSE=0
      NOPING=0
      NCLING=0
      K123=0
      NCPERM=3-NDI
      DO 950 I=1,3
      IF (I.LE.NCPERM) THEN
        K=2
        GO TO 940
      ENDIF
      PSIG=PSTRS(I)
      PEPS=DPSTRN(I)
      DPEPS=DPSTRN(I)
      K=KRK(I)
      KOC(I)=0
      IF (K.EQ.0) THEN
        EPSF=MAX((TWO*EPFRAC-PSIG/ECONCA),ZERO)
        SIGF=MAX((TWO*SIGO-PEPS*ECONCA),ZERO)

```

```

C
C      MULTIAXIAL TENSION STRENGTH CUTOFF
C
      J=IPERM(I)
      L=IPERM(J)
      SIGT=MAX(PSTRS(J)*EPSFAC(J),PSTRS(L)*EPSFAC(L),ZERO)
      SIGT=MIN(SIGT,SIG0)
      SIGT=TWO*SIG0-SIGT
      SIGF=MIN(SIGF,SIGT)
      ELSE
      EPSF=ZERO
      SIGF=ZERO
      ENDIF
      SIGFAC(I)=1.E20
      EPSFAC(I)=ONE
C
C      CLOSED CRACK
C
      IF ((K.EQ.2 .AND. DPEPS.LE.CRKCLO) .OR.
$ (K.EQ.1 .AND. PSIG.LE.ZERO .AND.
$ (DPEPS.LE.CRKCLO .OR. PEPS.LE.ZERO))) THEN
      IF (K.EQ.2) THEN
      KOC(I)=1
      ECLOSE(I)=PEPS
      NCLING=NCLING+1
      ENDIF
      K=1
      NCLOSE=NCLOSE+1
      GO TO 940
      ENDIF
C
C      UNCRACKED
C
      IF (K.EQ.0.AND.(PSIG.LT.SIGF.OR.PEPS.LT.EPSF)) GO TO 940
C
C      OPEN CRACK
C
      IF (K.NE.2) THEN
      NOPING=NOPING+1
      KOC(I)=-1
      ECLOSE(I)=ZERO
      ENDIF
      K=2
      NOPEN=NOPEN+1
      SIGFAC(I)=ZERO
      EPSFAC(I)=ZERO
C
940   KRK(I)=K
      K123=MAX(K123,K)
950   CONTINUE
C
C      UPDATE CRACK STATUS

```

```

C      KRKNEW=KRK(1)+10*KRK(2)+100*KRK(3)
      AR(26)=KRKOLD+10000*KRKNEW
      IF (ITER.NE.1) AR(26)=KRKNEW+10000*KRKNEW

C
C      SANITIZE LOCAL STRESSES & STRAINS
C
      DO 955 K=1,3
      KP3=K+3
      I=IPERM(K)
      J=IPERM(I)
      KIJ=MAX(KRK(I),KRK(J))
      IF (KIJ.EQ.2) THEN
        SIGFAC(KP3)=SIGO/MAX(ONE,ABS(PSTRN(K))/EPFRAC)
        EPSFAC(KP3)=EPFRAC/MAX(ONE,ABS(PSTRN(K)))
      ELSE
        SIGFAC(KP3)=1.E20
        EPSFAC(KP3)=ONE
      ENDIF
955    CONTINUE
      IF (IBUG.NE.0) WRITE(NOUT,956) KRK,ECLOSE,((A(I,J),J=1,3),I=1,3),
$ PSTRS,PSTRN,DPSTRN,SIGFAC
956    FORMAT(' CONC3D AFTER 955: K1,K2,K3,ECLOSE=',3I5,1P3E11.3,
$ ' /A/PSTRS/PSTRN/DPSTRN/SIGFAC'/1P3E11.3/1P3E11.3/1P3E11.3/
$ (1P6E11.3))
      DO 960 I=1,NTENS
      PSTRN(I)=(PSTRNO(I)+DPSTRN(I))*EPSFAC(I)
960    PSTRS(I)=SIGN(MIN(ABS(PSTRS(I)),SIGFAC(I)),PSTRS(I))
C
C      ADJUST TRANSVERSE STRESSES FOR OPENING CRACKS
C
      IF (NOPING.GT.0.AND.NOPEN.LT.3) THEN
        DO 965 I=1,3
        DO 965 J=1,3
965      H(I,J)=PH(I,J)
        CALL SYMINV(H,6,3)
        IF (NOPEN.EQ.1) THEN
          IF (KRK(1).EQ.2) THEN
            I=1
          ELSEIF(KRK(2).EQ.2) THEN
            I=2
          ELSE
            I=3
          ENDIF
          J=IPERM(I)
          K=IPERM(J)
          DEPSJ=H(J,I)*PSTRSO(I)
          DEPSK=H(K,I)*PSTRSO(I)
          PSTRS(J)=PSTRS(J)-PH(J,I)*DPSTRN(I)+PH(J,J)*DEPSJ+PH(J,K)*DEPSK
          PSTRS(K)=PSTRS(K)-PH(K,I)*DPSTRN(I)+PH(K,J)*DEPSJ+PH(K,K)*DEPSK
        ELSE
          IF (KRK(1).NE.2) THEN

```

```

      K=1
      ELSEIF (KRK(2).NE.2) THEN
      K=2
      ELSE
      K=3
      ENDIF
      I=IPERM(K)
      J=IPERM(I)
      DEPSJ=ZERO
      DEPSK=H(K,I)*PSTRSO(I)+H(K,J)*PSTRSO(J)
      PSTRS(K)=PSTRS(K)-PH(K,I)*DPSTRN(I)-PH(K,J)*DPSTRN(J)+
$   PH(K,K)*DEPSK
      ENDIF
      IF (IBUG.NE.0) WRITE(6,967) NOPING,NOPEN,I,J,K,H(J,I),H(K,I),
$   DEPSJ,DEPSK,PH(J,J),PH(J,K),PH(K,K),(PSTRSO(L),L=1,3),
$   (PSTRS(L),L=1,3),(DPSTRN(L),L=1,3)
967   FORMAT(' CONC3D AFTER TRANS. CORR: NOPING,NOPEN,I,J,K,',
$   'H(J,I K,I),DEPSJ,DEPSK,PH(J,J J,K K,K)/PSTRSO,PSTRS,DPSTRN=' /
$   5I3,1P7E11.3/(1P9E11.3))
      ENDIF

C
C   FURTHER PLASTICITY CHECK (LOCAL)
C   ROTATE OLD STRESSES TO LOCAL SYSTEM
C

      CALL DEVIAT(PSTRS,SIJ,SIGM)
      IF (SIGM.LE.ZERO) THEN
      CALL YFUN(PSTRS,SIJ,SIGM,FACTR,EK2,K123,RADFAC,PLAST)
      ELSE
      PLAST=.FALSE.
      RADFAC=ONE
      ENDIF

C
C   UPDATE PLASTICITY FLAG
C

      IF (PLAST .AND. NPL1.NE.1) NPL1=3
      AR(25)=NPL1+100*NPL1
      IF (IBUG.NE.0) WRITE(6,983) NPL1,RADFAC
983   FORMAT(' CONC3D AFTER FINAL CHECK: NPL1,RADFAC=',I2,1PE11.3)

C
C   PERFORM RADIAL RETURN FOR CRACKED STRESSES
C

      IF (NPL1.EQ.1.AND.RADFAC.LT.ONE)
$   CALL RADRET(PSTRS,SIJ,SIGM,K123,RADFAC)

C
C   ROTATE LOCAL SANITIZED STRESSES TO GLOBAL
C

      CALL PVAL3D(0,4,PSTRS,SG,A)

C
C   UPDATE THE EFFECTIVE STRAIN ONLY FOR HARDENING MATERIALS
C

      IF (IHARD.NE.0) THEN
      IF (SIGM.GT.ZERO) EPSEFF=ZERO

```

```

        AR(27)=EPSEFF
    ENDIF
C
C        UPDATE NEW STRESSES FOR ITER=1
C
1155 IF (ITER.EQ.1.OR.KSTIF.EQ.1) THEN
        DO 1160 I=1,NTENS
1160   AR(I+6)=SG(I)
        GO TO 1200
    ENDIF
C
C        UPDATE ALL STRAINS & STRESSES
C
        DO 1170 I=1,NTENS
        IF (ABAQUS) THEN
            AR(I+18)=AR(I+18)+DEP(I)
        ELSE
            AR(I+18)=AR(I+18)+EP(I)
        ENDIF
        SR(I)=SG(I)-AR(I)
        AR(I+6)=SG(I)
1170   AR(I)=SG(I)
        DO 1175 I=1,3
1175   AR(I+12)=ECLOSE(I)
C
C        UPDATE CREEP PARAMETERS
C
        N=NCREEP
        DO 1180 J=1,JCREEP
        DO 1180 I=1,NTENS
            N=N+1
1180   AR(N)=AR(N)*GE(4,J)+SR(I)*GE(1,J)/GE(3,J)
C
        AR(28)=EI
        AR(29)=AR(29)*DAGE
        IF (NPL1.GT.0) AR(29)=YIELD
C
C        UPDATE PLASTICITY FLAG
C
        AR(25)=NPL1+100*NPL1
C
1200 IF (IBUG.NE.0) WRITE(NOUT,1210) (AR(I),I=1,NCOS)
1210 FORMAT(' CONCD3D BEFORE RETURN: AR'/(1P10E11.3))
C
        RETURN
    END

    SUBROUTINE ESTRN(NTENS,XVC,ESEC,PEPS,PSTRS,EPSFAC,NK,EPSEFF,KRK,
$ ECLOSE,ECONC)
    IMPLICIT REAL*8 (A-H,O-Z)
C
    COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,

```

```

$  NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$  TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
REAL*8  NINE,NINETH,MILLON

C
COMMON /RSDDBG/ IBUG

C
DIMENSION H(6,6),PSTRN(6),PSTRS(6),EPSFAC(6),EPSE(6),EPSP(6),
$ KRK(3),ECLOSE(3),PEPS(6)

C
DATA PTOL/1.0E-14/, SIZTOL/0.1E-6/

C
IF (NK.GE.3) THEN
  EPSEFF=ZERO
  RETURN
ENDIF

C
C      COMPUTE EFFECTIVE STRAIN
C
EI=ESEC
XVCPI=ONE+XVC
10  DO 20 I=1,NTENS
    PSTRN(I)=PEPS(I)
    DO 20 J=1,NTENS
20  H(I,J)=ZERO
    DO 40 I=1,3
    DO 30 J=1,3
30  H(I,J)=-XVC/EI
    H(I,I)=ONE/EI
40  H(I+3,I+3)=TWO*XVCPI/EI

C
C      SANATIZE STRAINS FOR CLOSED CRACKS
C
DO 45 I=1,3
IF (KRK(I).EQ.1) PSTRN(I)=PSTRN(I)-ECLOSE(I)
45 CONTINUE

C
C      COMPUTE PURIFIED MECHANICAL ELASTIC & PLASTIC STRAINS
C
EPSEM=ZERO
EPSPM=ZERO
ESTR=ZERO
PSTR=ZERO
ESIZE=ZERO
PSIZE=ZERO
PTEST=ZERO
DO 50 I=1,NTENS
SUM=ZERO
DO 50 J=1,NTENS
50 SUM=SUM+H(I,J)*PSTRS(J)
ESTR=SUM*EPSFAC(I)
EPSE(I)=ESTR
PSTR=PSTRN(I)*EPSFAC(I)-ESTR

```

```

EPSP(I)=PSTR
ESIZE=MAX(ESIZE,ABS(ESTR))
PSIZE=MAX(PSIZE,ABS(PSTR))
IF (I.LE.3) THEN
  PTEST=PTEST+ESTR*PSTR
  EPSEM=EPSEM+ESTR
  EPSPM=EPSPM+PSTR
ELSE
  PTEST=PTEST+TWO*ESTR*PSTR
ENDIF
60 CONTINUE
C
IF (ESIZE.LT.SIZTOL .AND. PSIZE.LT.SIZTOL) THEN
  EPSEFF=ZERO
  RETURN
ELSEIF (ESIZE.GT.TEN .OR. PSIZE.GT.TEN) THEN
  WRITE(6,61) PSTRS,PSTRN
61  FORMAT(' ESTRN-STRAINS TOO LARGE: /PSTRS/PSTRN'/(1P6E11.3))
  EPSEFF=ZERO
  RETURN
ENDIF
EPSEM=EPSEM/THREE
EPSPM=EPSPM/THREE
C
FE=THREE/TWO
FP=FE
IF (NK.EQ.0) THEN
  FE=FE/(XVCP1**2)
  FP=TWO/THREE
ELSEIF (NK.EQ.1) THEN
  FE=FE/(XVCP1+XVC*XVC)
  FP=SIX/SEVEN
ELSEIF (NK.GE.3) THEN
  FE=ZERO
  FP=ZERO
ENDIF
C
EFPF=ZERO
EFFE=ZERO
DO 70 I=1,3
EFPF=EFPF+(EPSP(I)-EPSPM)**2+TWO*(EPSP(I+3)**2)
70  EFFE=EFFE+(EPSE(I)-EPSEM)**2+TWO*(EPSE(I+3)**2)
IF (PSIZE.GT.SIZTOL) THEN
  IF (PTEST.LT.PTOL) THEN
    IF (ABS(EI/ECONC-ONE).GT.0.01) THEN
      EI=ECONC
      GO TO 10
    ENDIF
    IF (IBUG.NE.0) WRITE(6,69) ESIZE,PSIZE,PTEST,ESEC,ECONC,EPSE,
$   EPSP,PSTRS,PSTRN
69  FORMAT(' ESTRN BAD PLASTIC STRAIN TEST: ESIZE,PSIZE,PTEST,',
$   'ESEC,ECONC/EPSE/EPSP/PSTRS/PSTRN=' /1P5E11.3/(1P6E11.3))

```

```

      FP=ZERO
    ENDIF
  ELSE
    FP=ZERO
  ENDIF
  EPSEFF=SQRT(FP*EFFP)+SQRT(FE*EFFE)
  IF (IBUG.NE.0) WRITE(6,71) NK,EPSFAC,FE,FP,EPSEFF,EFFE,EFFP,PSTRN,
$ ECLOSE
71  FORMAT(' END ESTRN: NK,EPSFAC/FE,FP,EPSEFF,EFFE,EFFP/PSTRN/ECLOSE'
$ /I5,1P6E11.3/1P5E11.3/(1P6E11.3))
C
    RETURN
    END

    SUBROUTINE DEVIAT(SG,SIJ,SIGM)
    IMPLICIT REAL*8 (A-H,O-Z)
C
    COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPAN
    REAL*8 NINE,NINETH,MILLON
C
    COMMON /RSDDBG/ IBUG
C
    DIMENSION SG(6),SIJ(6)
C
    SIGM=(SG(1)+SG(2)+SG(3))/THREE
    DO 10 I=1,3
    SIJ(I)=SG(I)-SIGM
10  SIJ(I+3)=SG(I+3)
    RETURN
    END

    SUBROUTINE YFUN(SG,SIJ,SIGM,FACTR,EK2,K123,RADFAC,PLAST)
    IMPLICIT REAL*8 (A-H,O-Z)
C
    COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$ NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$ TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPAN
    REAL*8 NINE,NINETH,MILLON
C
    COMMON /RSDDBG/ IBUG
C
    DIMENSION SG(6),SIJ(6)
C
    LOGICAL UNCON,PLAST
C
    DATA PTOL/1.E-20/
C
    IF (IBUG.NE.0) WRITE(6,1) K123,FACTR,EK2,SIGM,SIJ
1  FORMAT(' START YFUN: K123/FACTR,EK2,SIGM,SIJ-',I5/(1P9E11.3))
    PLAN=ZERO

```

```

    PLAS=ZERO
    DO 10 I=1,3
    PLAN=PLAN+HALF*(SIJ(I)**2)
10  PLAS=PLAS+SIJ(I+3)**2
    PLA=PLAN+PLAS
    PLAKK=NINE*FACTR*(SIGM**2)
    UNCON=.FALSE.

C
    IF (K123.EQ.2.OR.UNCON) THEN
C
C      WITH AT LEAST ONE OPEN CRACK OR LOW CONFINING STRESS
C      PERFORM THE RADIAL RETURN ON THE TOTAL STRESSES
C
    IF (EK2.LE.ZERO) THEN
      RADFAC=ZERO
    ELSE
      RADFAC=ONE
      PLA=PLA+PLAKK
      PLAST=.FALSE.
      IF (PLA.GT.PTOL) THEN
        RADFAC=SQRT(EK2/PLA)
        IF (RADFAC.LE.ONE) PLAST=.TRUE.
      ENDIF
    ENDIF
    ELSE
C
C      WITH NO OPEN CRACKS PERFORM THE RADIAL RETURN
C      ON THE DEVIATORIC STRESSES
C
    EKEK=EK2-PLAKK
    IF (EKEK.LE.ZERO) THEN
      RADFAC=ZERO
    ELSE
      RADFAC=ONE
      PLAST=.FALSE.
      IF (PLA.GT.PTOL) THEN
        RADFAC=SQRT(EKEK/PLA)
        IF (RADFAC.LE.ONE) PLAST=.TRUE.
      ENDIF
    ENDIF
    ENDIF

C
    IF (IBUG.NE.0) WRITE(6,99) UNCON,PLAST,PLA,PLAN,PLAS,PLAKK,RADFAC
99  FORMAT(' END YFUN: UNCON,PLAST,PLA,PLAN,PLAS,PLAKK,RADFAC= ',
$ L1,1X,L1/(1P10E11.3))
    RETURN
    END

    SUBROUTINE RADRET(SG,SIJ,SIGM,K123,RADFAC)
    IMPLICIT REAL*8 (A-H,O-Z)

C
    COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,

```

```

$  NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$  TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
REAL*8  NINE,NINETH,MILLON

C
COMMON /RSDDBG/ IBUG

C
DIMENSION SG(6),SIJ(6)

C
IF (K123.EQ.2) THEN

C
    WITH AT LEAST ONE OPEN CRACK PERFORM THE RADIAL RETURN
C
    ON THE TOTAL STRESSES
C

    DO 20 I=1,6
    SIJ(I)=SIJ(I)*RADFAC
20    SG(I)=SG(I)*RADFAC
    SIGM=SIGM*RADFAC
    ELSE

C
    WITH NO OPEN CRACKS PERFORM THE RADIAL RETURN
C
    ON THE DEVIATORIC STRESSES
C

    DO 30 I=1,3
    S=SIJ(I)*RADFAC
    SIJ(I)=S
    SG(I)=S+SIGM
    S=SIJ(I+3)*RADFAC
    SIJ(I+3)=S
30    SG(I+3)=S
    ENDIF

C
IF (IBUG.NE.0) WRITE(6,99) SIGM,SIJ,SG
99  FORMAT(' AFTER RADRET: SIGM,SIJ/SG= '/1P7E11.3/(1P6E11.3))
RETURN
END

SUBROUTINE TRNS3D(A,Q,B)
IMPLICIT REAL*8 (A-H,O-Z)

C
C  THIS ROUTINE CREATES Q AND B (BOTH 6X6) OUT OF THE TRANSFORMATON
C  MATRIX A. Q AND B MAY THEN BE USED TO TRANSFORM A 6X6 TENSOR.
C

COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$  NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$  TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
REAL*8  NINE,NINETH,MILLON

C
DIMENSION A(3,3),Q(6,6),B(6,6)

C
DO 200 K=1,3
DO 100 I=1,3
100  B(K,I)=A(I,K)*A(I,K)

```

```

      B(K,4)=A(2,K)*A(3,K)*TWO
      B(K,5)=A(1,K)*A(3,K)*TWO
      B(K,6)=A(1,K)*A(2,K)*TWO
      B(4,K)=A(K,2)*A(K,3)
      B(5,K)=A(K,1)*A(K,3)
200    B(6,K)=A(K,1)*A(K,2)
      C
      B(4,4)=A(2,2)*A(3,3)+A(3,2)*A(2,3)
      B(4,5)=A(1,2)*A(3,3)+A(3,2)*A(1,3)
      B(4,6)=A(1,2)*A(2,3)+A(2,2)*A(1,3)
      B(5,4)=A(2,1)*A(3,3)+A(3,1)*A(2,3)
      B(5,5)=A(1,1)*A(3,3)+A(3,1)*A(1,3)
      B(5,6)=A(1,1)*A(2,3)+A(2,1)*A(1,3)
      B(6,4)=A(2,1)*A(3,2)+A(3,1)*A(2,2)
      B(6,5)=A(1,1)*A(3,2)+A(3,1)*A(1,2)
      B(6,6)=A(1,1)*A(2,2)+A(2,1)*A(1,2)
      C
      DO 300 I=1,6
      DO 300 J=1,6
300    Q(I,J)=B(I,J)
      DO 400 I=1,3
      DO 400 J=4,6
400    Q(I,J)=HALF*Q(I,J)
      DO 500 I=4,6
      DO 500 J=1,3
500    Q(I,J)=TWO*Q(I,J)
      C
      RETURN
      END

      SUBROUTINE PVAL3D(ISTRN,IOPT,SIG,PSIG,A)
      IMPLICIT REAL*8 (A-H,O-Z)

      C
      C          CALCULATE PRINCIPAL STRESSES AND DIRECTIONS
      C
      C          ISTRN
      C              EQ.0 - STRESSES
      C              NE.0 - STRAINS
      C
      C          IOPT
      C              EQ.0 - COMPUTE PRINC. VALS./DIRECTS.
      C              EQ.1 - COMPUTE PRINC. VALS./DIRECTS.
      C                      (1ST COL. OF A GIVEN)
      C              EQ.2 - COMPUTE PRINC. VALS./DIRECTS.
      C                      (1ST & 2ND COLS. OF A GIVEN)
      C              EQ.3 - ROTATE GLOBAL TO LOCAL
      C                      (A GIVEN)
      C              EQ.4 - ROTATE LOCAL TO GLOBAL
      C                      (A GIVEN)
      C
      C          ORDER OF SIG & PSIG
      C

```

```

C          1 - 1,1
C          2 - 2,2
C          3 - 3,3
C          4 - 2,3
C          5 - 3,1
C          6 - 1,2
C
C          A(I,J)=COS(X(I),X'(J))
C
C          WHERE X IS GLOBAL & X' IS LOCAL(PRINCIPAL)
C          SO THAT COLUMNS IN A CORRESPOND TO THE LOCAL
C          SYSTEM
C
C          COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$          NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTN,EIGHTH,NINETH,
$          TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
          REAL*8 NINE,NINETH,MILLON
C
C          COMMON /RSDINF/ NOUT,JELNO,INT,NSTPAB,INCRAB,NPASS
C
C          DIMENSION SIG(6),PSIG(6),A(3,3),S(3,3),XU(3),SP(3,3),
$          IPERM(3),AT(3,3)
C
C          EQUIVALENCE (S1,S(1,1)),(S2,S(2,2)),(S3,S(3,3)),(S6,S(1,2)),
$          (S5,S(1,3)),(S4,S(2,3))
C
C          DATA TOL/1.E-6/, CTOL/1.E-3/, IPERM/2,3,1/, NERROR/0/
C
C          X27=THREE*NINE
C          X120=FOUR*THREE*TEN*PIFAC
C          XNS=ONE
C          IF (ISTRN.NE.0) XNS=TWO
C          DO 10 N=1,3
C            I=IPERM(N)
C            J=IPERM(I)
C            SUM=SIG(N+3)/XNS
C            S(I,J)=SUM
C            S(J,I)=SUM
10         S(N,N)=SIG(N)
C
C          IGO=IOPT+1
C          GO TO (20,100,200,300,400), IGO
C
C          COMPUTE ALL VALUES & DIRECTIONS
C
20         T1=S4*S4
C          T2=S5*S5
C          T3=S6*S6
C          P=-(S1+S2+S3)
C          Q=S1*S2+S2*S3+S3*S1-T1-T2-T3
C          R=-(S1*S2*S3+TWO*S4*S5*S6-S1*T1-S2*T2-S3*T3)
C          Z=Q-P*P/THREE

```

```

IF (ISTRN.EQ.0) THEN
  ZTOL=TEN**(-6)
ELSE
  ZTOL=TEN**(-10)
ENDIF
IF (ABS(Z).LT.ZTOL) GO TO 42
IF (Z.LE.-ZTOL) GO TO 30
WRITE(NOUT,26) ISTRN,IOPT,Z,ZTOL,SIG
26  FORMAT(' NEGATIVE ROOT NEEDED IN PVAL3D. ISOTROPIC',
$ ' STATE RETURNED.'/' ISTRN,IOPT,Z,ZTOL,SIG='2I6,1P8E11.3)
GO TO 42

C
30  B=TWO*P*P*P/X27-P*Q/THREE+R
    C=SQRT(-X27*B*B/(FOUR*Z*Z*Z))
    IF (ABS(C).GT.ONE) C=ABS(C)/C
    PHI=-SIGN(C,B)
    PHI=ACOS(PHI)
    C=TWO*SQRT(-Z/THREE)

C
C      THE PRINCIPAL VALUES
C
    CRIT=ZERO
    DO 35 K=1,3
    PSIG(K+3)=ZERO
    ANG=X120*FLOAT(K-1)+PHI/THREE
    X=C*COS(ANG)-P/THREE
35  XU(K)=X
    CRIT=TOL*(P/THREE)**2
    IF (CRIT.LE.ZERO) CRIT=TOL*(C**2)

C
    PSIG(1)=MAX(XU(1),XU(2),XU(3))
    PSIG(3)=MIN(XU(1),XU(2),XU(3))
    PSIG(2)=XU(1)+XU(2)+XU(3)-PSIG(1)-PSIG(3)

C
C      TEST FOR EQUAL ROOTS
C
    EQUAL=MAX(ABS(PSIG(1)),ABS(PSIG(3)))*TOL
    NE=0
    N=0
    IF (ABS(PSIG(1)-PSIG(2)).LT.EQUAL) THEN
      NE=NE+1
      N=3
    ENDIF
    IF (ABS(PSIG(2)-PSIG(3)).LT.EQUAL) THEN
      NE=NE+1
      N=1
    ENDIF
    IF (ABS(PSIG(3)-PSIG(1)).LT.EQUAL) THEN
      NE=NE+1
      N=2
    ENDIF
C

```

```

C          ISOTROPIC STATE
C
      IF (NE.LT.2) GO TO 46
42    DO 45 I=1,3
      PSIG(I)=P/THREE
      PSIG(I+3)=ZERO
      DO 44 J=1,3
44    A(J,I)=ZERO
45    A(I,I)=ONE
      GO TO 500
46    IF (NE.EQ.0) THEN
C
C          COMPUTE ALL 3 DIRECTION COSINES
C
      DO 60 I=1,3
      CALL DIRCOS(PSIG(I),S,A(1,I),CRIT,CTOL,DET)
      IF (DET.LE.CRIT) GO TO 600
60    CONTINUE
      ELSE
C
C          1 PAIR OF EQUAL PRINCIPAL VALUES
C
      CALL DIRCOS(PSIG(N),S,A(1,N),CRIT,CTOL,DET)
      IF (DET.LE.CRIT) GO TO 600
      CALL TWOVEC(N,A,XU,IPERM,CTOL,DET)
      IF (DET.LE.CTOL) GO TO 600
      ENDIF
      GO TO 500
C
C          FIRST DIRECTION KNOWN
C          COMPUTE LARGEST IN-PLANE "PRINCIPAL" VALUES
C
100   CALL TWOVEC(1,A,XU,IPERM,CTOL,DET)
      IF (DET.LE.CTOL) GO TO 600
      CALL RSDROT(S,SP,A,PSIG,IPERM)
      PSIG(4)=ZERO
      GEN=(SP(2,2)+SP(3,3))/TWO
      DIF=(SP(2,2)-SP(3,3))/TWO
      TAU=SP(2,3)
      RAD=SQRT(DIF*DIF+TAU*TAU)
      PSIG(2)=GEN+RAD
      PSIG(3)=GEN-RAD
      ANG=ZERO
      IF (RAD.GT.ZERO) ANG=ATAN2(TAU,DIF)/TWO
      CA=COS(ANG)
      SA=SIN(ANG)
      DO 110 I=1,3
      SP(I,1)=ZERO
      SP(1,I)=ZERO
      DO 110 J=1,3
110   S(I,J)=A(I,J)
      SP(1,1)=ONE

```

```

        SP(2,2)=CA
        SP(3,3)=CA
        SP(2,3)=-SA
        SP(3,2)=SA
        DO 130 I=1,3
        DO 130 J=1,3
        SUM=ZERO
        DO 120 K=1,3
120     SUM=SUM+S(I,K)*SP(K,J)
130     A(I,J)=SUM
        GO TO 500

C
C             TWO DIRECTIONS KNOWN
C
200     CALL CROSS(A(1,1),A(1,2),A(1,3),DET,IPERM)
        IF (DET.LE.CTOL) GO TO 600

C
C             ALL 3 DIRECTIONS KNOWN
C             ROTATE GLOBAL-TO-LOCAL
C
300     CALL RSDROT(S,SP,A,PSIG,IPERM)
        GO TO 500

C
C             ROTATE LOCAL-TO-GLOBAL
C
400     DO 410 I=1,3
        DO 410 J=1,3
410     AT(J,I)=A(I,J)
        CALL RSDROT(S,SP,AT,PSIG,IPERM)

C
C             RESET STRAINS
C
500     IF (ISTRN.EQ.0) RETURN
        DO 510 N=4,6
510     PSIG(N)=XNS*PSIG(N)
        RETURN

C
C             DIRECTION ERRORS
C
600     WRITE(NOUT,601) JELNO,INT,NSTPAB,INCRAB,NPASS,IOPT,ISTRN,CRIT,
$ CTOL,DET,(SIG(I),I=1,6),(PSIG(I),I=1,3),A
601     FORMAT('OPROBLEMS COMPUTING PRINCIPAL VALUES OR DIRECTIONS. ',
$ 'ISOTROPIC VALUES & UNIT VECTORS RETURNED.'/' JELNO,INT,NSTPAB',
$ ',INCRAB,NPASS,IOPT,ISTRN,CRIT,CTOL,DET='/'7I7,1P3E11.3/
$ ' SIG/EPS='/'1P6E11.3/' PSIG/EPS='/'1P3E11.3/' A='/'1P9E11.3)
        NERROR=NERROR+1
        IF (NERROR.LT.50) GO TO 42
        STOP 'PVAL3D ERROR TERMINATION'

C
END

SUBROUTINE RSDROT(S,SP,A,PSIG,IPERM)

```

```

      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /FLTNUM/ ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT,
$     NINE, TEN, HALF, THIRD, FOURTH, FIFTH, SIXTH, SEVNTN, EIGHTH, NINETH,
$     TENTH, HUNDRD, THOU, MILLON, PI, PIFAC, PIFAC1, EXPN
      REAL*8 NINE, NINETH, MILLON
C
      DIMENSION S(3,3), SP(3,3), A(3,3), PSIG(6), IPERM(3)
C
      DO 20 I=1,3
      DO 20 J=I,3
      SUM=ZERO
      DO 10 K=1,3
      AKI=A(K,I)
      DO 10 L=1,3
10     SUM=SUM+AKI*A(L,J)*S(K,L)
      SP(J,I)=SUM
20     SP(I,J)=SUM
      DO 30 N=1,3
      PSIG(N)=SP(N,N)
      I=IPERM(N)
      J=IPERM(I)
30     PSIG(N+3)=SP(I,J)
      RETURN
      END

      SUBROUTINE CROSS(A,B,C,DET,IPERM)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      COMMON /FLTNUM/ ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT,
$     NINE, TEN, HALF, THIRD, FOURTH, FIFTH, SIXTH, SEVNTN, EIGHTH, NINETH,
$     TENTH, HUNDRD, THOU, MILLON, PI, PIFAC, PIFAC1, EXPN
      REAL*8 NINE, NINETH, MILLON
C
      DIMENSION A(3), B(3), C(3), IPERM(3)
C
      DET=ZERO
      DO 10 N=1,3
      I=IPERM(N)
      J=IPERM(I)
      CC=A(I)*B(J)-A(J)*B(I)
      DET=DET+CC*CC
10     C(N)=CC
      IF (DET.LE.ZERO) RETURN
      DET=SQRT(DET)
      DO 20 N=1,3
20     C(N)=C(N)/DET
      RETURN
      END

      SUBROUTINE DIRCOS(PVAL,S,XU,CRIT,CTOL,DET)
      IMPLICIT REAL*8 (A-H,O-Z)

```

```

C      COMMON /FLTNUM/ ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT,
$      NINE, TEN, HALF, THIRD, FOURTH, FIFTH, SIXTH, SEVENTH, EIGHTH, NINETH,
$      TENTH, HUNDRD, THOU, MILLON, PI, PIFAC, PIFAC1, EXPN
      REAL*8 NINE, NINETH, MILLON

C      DIMENSION S(3,3), SP(3,3), XU(3)

C      NP=0
      DO 20 I=1,3
      XU(I)=ZERO
      DO 10 J=1,3
10      SP(J,I)=S(J,I)
20      SP(I,I)=S(I,I)-PVAL
C
30      DO 50 I=1,2
      IP1=I+1
      DO 40 J=IP1,3
      OFF=SP(I,J)
      DET=(SP(I,I)*SP(J,J)-OFF*OFF)
      IF (ABS(DET).LT.CRIT) GO TO 40

C
C      NORMAL RANK 2 MATRIX
C
      K=6-I-J
      XU(K)=ONE
      F1=-SP(I,K)
      F2=-SP(J,K)
      XU(I)=(SP(J,J)*F1-OFF*F2)/DET
      XU(J)=(SP(I,I)*F2-OFF*F1)/DET
      GO TO 60

C
40      CONTINUE
50      CONTINUE
C
C      REDUCE CRIT & TRY AGAIN
C
      DET=ABS(DET)
      IF (NP.NE.0) RETURN
      NP=1
      CRIT=CTOL*CRIT
      GO TO 30

C
60      FAC=SQRT(XU(1)**2+XU(2)**2+XU(3)**2)
      DO 70 I=1,3
70      XU(I)=XU(I)/FAC
      DET=ABS(DET)
      RETURN
      END

      SUBROUTINE TWOVEC(N,A,XU,IPERM,CTOL,DET)
      IMPLICIT REAL*8 (A-H,O-Z)

```

```

C      COMMON /FLTNUM/ ZERO,ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT,
$     NINE,TEN,HALF,THIRD,FOURTH,FIFTH,SIXTH,SEVNTH,EIGHTH,NINETH,
$     TENTH,HUNDRD,THOU,MILLON,PI,PIFAC,PIFAC1,EXPN
      REAL*8 NINE,NINETH,MILLON

C      DIMENSION A(3,3),XU(3),IPERM(3)

C      I=IPERM(N)
      J=IPERM(I)
      XU(1)=ONE
      XU(2)=ZERO
      XU(3)=ZERO
      CALL CROSS(A(1,N),XU,A(1,I),DET,IPERM)
      IF (DET.GT.CTOL) GO TO 10
      XU(1)=ZERO
      XU(2)=ONE
      CALL CROSS(A(1,N),XU,A(1,I),DET,IPERM)
      IF (DET.GT.CTOL) GO TO 10
      XU(2)=ZERO
      XU(3)=ONE
      CALL CROSS(A(1,N),XU,A(1,I),DET,IPERM)
      IF (DET.LE.CTOL) RETURN

C
10     CALL CROSS(A(1,N),A(1,I),A(1,J),DET,IPERM)
      RETURN
      END

C      SUBROUTINE DUMMY
C      IMPLICIT REAL*8 (A-H,O-Z)
C      ENTRY LEON
C      ENTRY STRNEC
C
C      WRITE(6,10)
C10    FORMAT('O***BAD CALL TO DUMMY UMAT SUBROUTINE***')
C      STOP 'BAD CALL TO DUMMY UMAT SUBROUTINE'
C      END
^Z

```