AFRL-ML-WP-TR-1998-4143

IMPLEMENTATION OF A NONISOTHERMAL UNIFIED INELASTIC-STRAIN THEORY FOR A TITANIUM ALLOY INTO ABAQUS 5.4 USER GUIDE

JOSEPH L. KROUPA

UNIVERSITY OF DAYTON RESEARCH INSTITUTE
300 COLLEGE PARK
DAYTON, OHIO 45469-0128

MAY 1996

INTERIM REPORT FOR MAY 1994 – MAY 1996

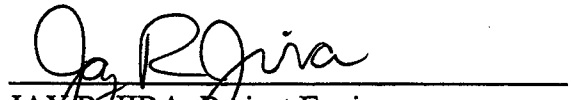
MATERIALS AND MANUFACTURING DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-7750

NOTICE

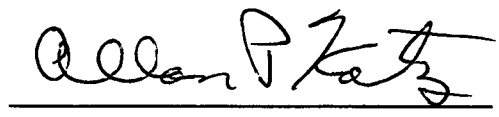
WHEN GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA ARE USED FOR ANY PURPOSE OTHER THAN IN CONNECTION WITH A DEFINITELY GOVERNMENT-RELATED PROCUREMENT, THE UNITED STATES GOVERNMENT INCURS NO RESPONSIBILITY OR ANY OBLIGATION WHATSOEVER. THE FACT THAT THE GOVERNMENT MAY HAVE FORMULATED OR IN ANY WAY SUPPLIED THE SAID DRAWINGS, SPECIFICATIONS, OR OTHER DATA, IS NOT TO BE REGARDED BY IMPLICATION OR OTHERWISE IN ANY MANNER CONSTRUED, AS LICENSING THE HOLDER OR ANY OTHER PERSON OR CORPORATION, OR AS CONVEYING ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY IN ANY WAY BE RELATED THERETO.

THIS REPORT IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

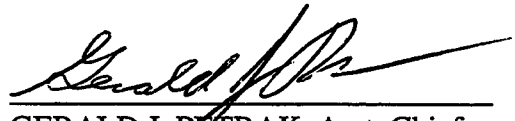
THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.



JAY R. JIRA, Project Engineer
Ceramics, Development, & Materials
Behavior Branch
Metals, Ceramics & NDE Division



ALLAN P. KATZ, Acting Chief
Ceramics, Development, & Materials
Behavior Branch
Metals, Ceramics & NDE Division



GERALD J. PETRAK, Asst. Chief
Metals, Ceramics & Nondestructive
Evaluation Division
Materials and Manufacturing Directorate

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION, PLEASE NOTIFY, AFRL/MLLN, WRIGHT-PATTERSON AFB OH 45433-7817 TO HELP US MAINTAIN A CURRENT MAILING LIST.

COPIES OF THIS REPORT SHOULD NOT BE RETURNED UNLESS RETURN IS REQUIRED BY SECURITY CONSIDERATIONS, CONTRACTUAL OBLIGATIONS, OR NOTICE ON A SPECIFIC DOCUMENT.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	MAY 1996	INTERIM REPORT FOR MAY 1994 - MAY 1996

4. TITLE AND SUBTITLE
IMPLEMENTATION OF A NONISOTHERMAL UNIFIED INELASTIC-STRAIN THEORY FOR A TITANIUM ALLOY INTO ABAQUS 5.4 USER GUIDE

5. FUNDING NUMBERS
C F33615-94-C-5200
PE 61102
PR 2302
TA P1
WU 01

6. AUTHOR(S)
JOSEPH L. KROUPA

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF DAYTON RESEARCH INSTITUTE
300 COLLEGE PARK
DAYTON, OHIO 45469-0128

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
MATERIALS AND MANUFACTURING DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB, OH 45433-7750
POC: JAY JIRA, AFRL/MLLN, 937-255-1358

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-ML-WP-TR-1998-4143

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Simulations with many modern aerospace materials require realistic mechanical response models for use in isothermal and non-isothermal applications. One such model is a unified inelastic-strain theory which has been applied to capture the strain-rate sensitivity and time dependent behavior of the titanium alloy Timetal21S. The Bodner-Partom form of unified theory satisfactorily describes the Timetal 21S stress-strain response for a range of temperatures from 23°C to 815°C and strain rates from 10-3/s to 1.X10-7/s. Practical use of the theory in finite element applications depends on advanced numerical algorithms that rapidly solve the inherently "stiff" constitutive equations. Special "user-defined" subroutines provide the framework for the incorporation of these numerical algorithms into ABAQUS 5.4, a general purpose finite element package. This user-guide contains a brief over-view of the theory, the Timetal21S material parameters, subroutine structure, variable nomenclature, and numerical algorithms. Test cases are described to illustrate the numerical integration schemes and provide guidance for error management. The appendices contain source code of the subroutines and other related input files.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Titanium, Timetal21S, Finite element methods, ABAQUS, Unified inelastic strain theory, Bodner-Partom	84
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

FOREWORD

This report documents computer software that was developed as part of an investigation of the non-isothermal viscoplastic response of the titanium alloy Timetal®21S. The software was completed and tested by Joseph L. Kroupa of the Structural Integrity Division, University of Dayton Research Institute, Dayton, OH. The research was conducted at the Materials Behavior Branch, Metals and Ceramics Division, Materials Directorate, Wright Laboratory (WL/MLLN) Wright-Patterson AFB, OH, under Contract No. F33615-94-C-5200. The contract was administered under the direction of WL/MLLN by Mr. Jay R. Jira. Dr. Noel E. Ashbaugh was the Principal Investigator and Dr. Joseph P. Gallagher was the Program Manager.

SECTION 1.0

INTRODUCTION

Many research and design projects involving structural response at elevated temperatures require a visco-plastic model to adequately describe strain-rate sensitivity and time-dependent behavior of material. In the case of Timetal®21S, a unified-inelastic-strain theory consisting of a modified flow rule of a Bodner-Partom [1]* works well to describe its visco-plastic response. In addition to sophisticated material models, many design projects also require stress-strain analysis in two and three-dimensions (e.g., multiaxial stress states) with complex boundary conditions. This is easily obtained with the finite element method. Thus, the integration of the Bodner-Partom model into a well established finite element package makes for a very powerful tool in solving problems that require visco-plastic material response and multiaxial stress states. Due to the advanced nature of these visco-plastic theories, many finite element programs do not include these formulations directly into their codes. However, most commercially available finite element programs, such as ABAQUS [2], do allow the incorporation of user-defined subroutines that would integrate the visco-plastic theories into their algorithms. This manual describes a unified inelastic strain theory for Timetal®21S and provides the associated user-defined subroutines that integrate the theory into ABAQUS. The manual is designed to assist the designer and researcher in using and understanding these subroutines within ABAQUS.

This document shares several similarities with a predecessor manual that describes the implementation and use of the Bodner-Partom model in ADINA [3]. One similarity is the description of the visco-plastic theory which appears both manuals for completeness. The second similarity is that the same numerical algorithms are used. The most obvious difference is that the present document describes the interface details associated with ABAQUS. In addition, this manual includes more information on error debugging and common usage errors.

In conjunction with improvements to the user manual, improvements to the numerical algorithms have also been completed. One major improvement is the coding of a more generalized form of the Bodner-Partom equations, which allows more temperature-dependent material properties than previously considered. In association with this modification, a second set of material constants is provided that better capture the creep behavior of Timetal®21S than those proposed by Neu [4]. The resultant set of algorithms can readily handle any temperature dependent material parameters for a wide variety of materials.

Other improvements include: 1) a unified set of subroutines for both the two- and three-dimensional elements; 2) new algorithms that accommodate the plane stress case without further modifications; 3) a more straight forward iterative loop; and 4) a special flag to automatically reduce the time-step size.

SECTION 2.0

BODNER-PARTOM CONSTITUTIVE MODEL

This section presents the Bodner-Partom constitutive equations and the material parameters that describe the Timetal®21S response. Several formulations of the Bodner-Partom model can characterize inelastic deformation under a variety of conditions, including anisotropic, isothermal, and/or non-isothermal material response. The formulation presented in the Section 2.1 parallels the isotropic nonisothermal theory of Chan, Bodner, and Lindholm [5,6,7]. The terminology is similar to that of Chan and Lindholm [8]. Section 2.2 contains model parameters that characterize the Timetal®21S material.

2.1 Theory

The first assumption in this analysis is the decomposition of the total strain, ε_{ij}^{tot}, into elastic, thermal, and inelastic components. This decomposition is expressed as:

$$\varepsilon_{ij}^{tot} = \varepsilon_{ij}^{el} + \varepsilon_{ij}^{th} + \varepsilon_{ij}^{in}. \qquad \text{Eq. 1}$$

The elastic strains, ε_{ij}^{el}, depend on the current stress state, elastic modulus E, and Poisson's ratio ν. The thermal strain components, ε_{ij}^{th}, equal the product of the coefficient of thermal expansion and the difference between the current and reference temperatures. The Bodner-Partom flow rule governs the evolution of the inelastic strains, ε_{ij}^{in}.

As opposed to other unified inelastic strain formulations, this theory describes the directional (kinematic) hardening with a special directional hardening term. Other theories represent directional hardening phenomena with a "back-stress" modified equivalent stress [9,10,11]. Introduction of the directional hardening term alters the Bodner-Partom flow rule [1] by replacing the previously known variable "drag-stress" with the sum of isotropic and directional hardening terms, Z^I and Z^D, respectively. These two hardening terms enter into the inelastic strain rate equation or flow law:

$$\dot{\varepsilon}_{ij}^{in} = D_0 \exp\left\{-\frac{1}{2}\left(\frac{\left(Z^I + Z^D\right)^2}{3J_2}\right)^n\right\}\frac{s_{ij}}{\sqrt{J_2}}, \qquad \text{Eq. 2}$$

where D_0 is the limiting strain rate, s_{ij} are the components of deviatoric stress, and $J_2 = s_{ij}\, s_{ij}\, /2$.

The evolution of Z^I and Z^D have similar empirical forms. Each equation consists of a hardening term, a thermal recovery term, and a temperature rate term. The isotropic hardening evolution equation with these three terms is:

$$\dot{Z}^I = m_1\left(Z_1 - Z^I\right)\dot{W}^{in} - A_1 Z_1\left(\frac{Z^I - Z_2}{Z_1}\right)^{r_1} + \left(\frac{Z_1 - Z^I}{Z_1 - Z_2}\right)\frac{\delta Z_2}{\delta \mathrm{T}}\dot{T}, \qquad \text{Eq. 3}$$

where the inelastic work rate is given by:

$$\dot{W}^{in} = \sigma_{ij}\dot{\varepsilon}_{ij}^{in}. \qquad \text{Eq. 4}$$

The initial value of the isotropic hardening, $Z^I(0)$, is Z_0. The material parameters associated with the isotropic hardening evolution are m_1, Z_0, Z_1, Z_2, A_1, and r_1. The thermal differential term $\frac{\delta Z_2}{\delta T}$ appropriately scales the isotropic hardening

variable when inelastic deformation and thermal recovery occur under nonisothermal conditions; thus' Z^I is prevented from passing through maximum or minimum values with temperature changes. The treatment of this thermal differential term is more consistent with the work of McDowell [12] and others [13,14,15,16] than those proposed by Chan, Bodner and Lindholm [5].

The scalar product of a state variable, β_{ij}, and a unit stress vector, u_{ij}, yields the magnitude of the directional hardening term:

$$Z^D = \beta_{ij} u_{ij}, \qquad \text{Eq. 5}$$

where:

$$\dot{\beta}_{ij} = m_2(Z_3 u_{ij} - \beta_{ij})\dot{W}^{in} - A_2 Z_1 \left(\frac{\sqrt{\beta_{kl}\beta_{kl}}}{Z_1}\right)^{r_2} v_{ij} + \frac{\beta_{ij}}{Z_3}\frac{\partial Z_3}{\partial T}\dot{T}, \qquad \text{Eq. 6}$$

$$u_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{kl}\sigma_{kl}}}, \qquad \text{Eq. 7}$$

and:

$$v_{ij} = \frac{\beta_{ij}}{\sqrt{\beta_{kl}\beta_{kl}}}. \qquad \text{Eq. 8}$$

The initial directional hardening variable, $Z^D(0)$ is set to zero. The material constants associated with the directional hardening evolution equation are m_2, Z_3, A_2, and r_2. The temperature differential term $\frac{\delta Z_3}{\delta T}$ appropriately scales the directional hardening variable when inelastic deformation and thermal recovery occurs under nonisothermal conditions. In particular, without these differential

terms the directional hardening accrued at one temperature may exceed the limiting value Z_3 at another temperature, which is not physically realistic.

Table 2-A summarizes the 15 material parameters that characterize the strain-rate sensitivity and time-dependent behavior of Timetal®21S [17]. The number of material parameters is effectively less than those in Table 2-A after applying the usual assumptions: $A_1 = A_2$, $r_1 = r_2$, and $Z_0 = Z_2$. The number of temperature dependent constants is minimized to reduce the amount of required data. Other materials, such as revised Timetal®21S constants [18], Mar-M47 [5] or B1900+Hf [6], require different temperature-dependent parameters. The numerical algorithms provided in Appendix 1 assume that all the material properties are temperature dependent, except D_0 and Z_1. The algorithms also assume that all material properties are unique, i.e., A_1 does not necessarily equal A_2.

Table 2-A

Material Parameters in Bodner-Partom Constitutive Model

Parameter	Units	Description
E	MPa	Elastic modulus
ν	---	Poisson's ratio
α	$(°C)^{-1}$	Coefficient of thermal expansion
D_o	s^{-1}	Limiting shear strain rate
Z_o	MPa	Initial value of the isotropic hardening variable
Z_1	MPa	Limiting (maximum) value of Z
Z_2	MPa	Fully recovered (minimum) value of Z^I
Z_3	MPa	Limiting (maximum) value of Z^D
m_1	$(Mpa)^{-1}$	Hardening rate coefficient of Z^I
m_2	$(Mpa)^{-1}$	Hardening rate coefficient of Z^D
n	---	Strain rate sensitivity parameter
A_1	s^{-1}	Recovery coefficient for Z^I
A_2	s^{-1}	Recovery coefficient for Z^D
r_1	----	Recovery exponent for Z^I
r_2	----	Recovery exponent for Z^D

2.2 Timetal®21S Material Parameters

The material parameters for the Bodner-Partom model with directional hardening were determined from Timetal®21S using monotonic, cyclic, and creep test data [17]. The parameters are valid for a wide range of strain rates (10^{3} to 10^{-7} s^{-1}) and temperatures (23° - 815°C). The strategy for determining the parameters involves a number of steps. First, the temperature-dependent parameters are identified and the values of the temperature-independent parameters are estimated. Then at each temperature where experimental data are available, the temperature-dependent parameters are determined through an iterative process to minimize the differences between model simulations and experiments. *Mathematica* [19] is used to generate the model simulations. Similar to other inelastic strain theories, the set of material parameter for any particular load case is not unique. Thus, for a given set of experimental load responses, a range of values is suitable for each material parameter.

The resulting set of temperature-dependent parameters becomes continuous with temperature as the range of possible values is decreased for each parameter. The response can be very sensitive to small changes in certain material parameters with temperature, especially in the transition regimes between different deformation mechanisms. For Timetal®21S, a transition region for inelastic behavior occurs between 482°C for plasticity and at 650°C for power law creep. Thus, an anomalous change in the saturated stress level can occur if linear interpolation of material parameters is used within this transition region.

Since no experimental data are available within this transition region, values for temperature-dependent parameters are chosen between 482°C and 650°C. Thus, the resulting saturated stress is smooth and continuous with temperature, reducing the anomalous effects of linear interpolation. The final version of the material parameters for Timetal®21S is shown in Table 2-B [4].

Note that at several temperatures Z_3 is larger than Z_2. This introduces a potential convergence problem upon load reversals at high strain rates. A revised series of material parameters for Timetal®21S [18], provided in Table 2-C, is proposed to eliminate this problem. In addition, the revised parameters are calibrated to better capture lower-stress creep response than the previous constants, which are fit to the widest range of stress levels and temperatures possible.

Table 2-B

Bodner-Partom Material Parameters for Timetal®21S [4]

Temperature-Independent Constants

$m_1 = 0$ MPa^{-1} (no isotropic hardening) $Z_1 = 1600$ MPa
$r_1 = r_2 = 3$ $D_o = 1 \times 10^4$ s^{-1}
$\nu = 0.34$

Temperature-Dependent Constants

Temp. °C	E MPa	α^* 10^{-6} $(°C)^{-1}$	n	$Z_o = Z_2$ MPa	Z_3 MPa	m_2 $(Mpa)^{-1}$	$A_1 = A_2$ s^{-1}
23	**112000**	**6.31**	**4.8**	**1550**	**100**	**0.35**	**0**
260	**108000**	**7.26**	**3.5**	**1300**	**300**	**0.35**	**0**
315	ϕ	ϕ	◊	◊	*390*	◊	ϕ
365	ϕ	ϕ	◊	◊	*500*	◊	ϕ
415	ϕ	ϕ	◊	◊	*660*	◊	ϕ
465	ϕ	ϕ	◊	◊	*960*	◊	ϕ
482	**98100**	**8.15**	**1.7**	**1100**	**1100**	**5**	**0.0076**
500	ϕ	ϕ	*1.5*	◊	*1300*	◊	ϕ
525	ϕ	ϕ	*1.28*	◊	*1670*	◊	ϕ
550	ϕ	ϕ	*1.1*	◊	*2100*	◊	ϕ
575	ϕ	ϕ	*0.97*	◊	*2600*	◊	ϕ
600	ϕ	ϕ	**0.82**	◊	**3700**	**10**	ϕ
650	**86600**	**8.83**	**0.74**	**1000**	**3800**	**10**	**0.21**
760	**77200**	**9.27**	**0.58**	**600**	**4000**	**15**	**1.0**
815	**72000**	**9.49**	**0.55**	**300**	**4100**	**30**	**2.0**

* Secant α with $T_o = 25°C$

◊ - Linear interpolate between values given in table.

ϕ - Use functions to determine values.

Bold were values determined based on experiments.

Italics are values that describe smooth and continuous saturated stress change.

Functions

$$A_1 = C_{51} \exp\left(\frac{-C_{61}}{T + 273}\right), \text{ with } C_{51} = 5.8 \times 10^5 \text{ s}^{-1}, C_{61} = 1.37 \times 10^4 \text{ C, and T in °C}$$

$$A_2 = C_{52} \exp\left(\frac{-C_{62}}{T + 273}\right), \text{ with } C_{52} = 5.8 \times 10^5 \text{ s}^{-1}, C_{62} = 1.37 \times 10^4 \text{ C, and T in °C}$$

Table 2-C

Revised Bodner-Partom Material Parameters for Timetal®21S [18]

Temperature-Independent Constants

$m_2 = 4\ MPa^{-1}$
$r_1 = r_2 = 3.5$
$\nu = 0.34$
$Z_1 = 3200MPa$
$D_0 = 1 \times 10^4\ s^{-1}$
$Z_3 = 180\ MPa$

Temperature-Dependent Constants

Temp. °C	E MPa	α^* $10^{-6}\ (°C)^{-1}$	n	$Z_0 = Z_2$ MPa	m_1 $(Mpa)^{-1}$	$A_1 = A_2$ s^{-1}
23	112000	6.31	1.58	2950	0.5	0
260	108000	7.26	1.35	2650	0.5	0
566	92700	8.49	1.09	800	2.5	0.3
650	86600	8.83	0.94	200	7.0	2.0
760	77200	9.27	0.575	190	37.0	4.5
815	72000	9.49	0.55	150	50	7.5

*Secant α with $T_0 = 25°C$

SECTION 3.0

SUBROUTINE OPERATION

3.1 Subroutine Installation and Execution

The installation of the user-defined subroutines into ABAQUS is fairly straight forward. The user-subroutines are compiled and linked automatically when executing the code. Thus, a permanent version of executable code is never created.

The user-subroutines can be incorporated into ABAQUS in two manners. One method is to include the subroutines directly into the ABAQUS input file after a `*USERSUBROUTINE` card. This method works well for one time use, but is not best suited for multiple input files. For use with several input files, the best method is to include the file name in the ABAQUS command statement. For example, if the user-subroutines reside in the file `usub_db.f`, then the command statement for ABAQUS would be:

```
abaqus job=jobname user=usub_db
```

Execution of this command will look for the input files `jobname.inp` and `usub_db.f`. This command will automatically compile and link the user-subroutines in `usub_db.f` to create an executable version of ABAQUS. The program then proceeds to solve the problem described in `jobname.inp`.

3.2 Material Property Input

Different from ADINA, where temperature dependent properties are assumed, ABAQUS assumes nothing special about the material properties. As a result, the properties are read in as a single array, which requires a special subroutine to extract the data. Figure 3.1 gives an example how the Bodner-Partom material constants of Table 2-B appear in the ABAQUS input file.

```
*MATERIAL, NAME=B21S_3
*EXPANSION, ZERO=900.0
 9.7787E-6      23.0
 1.0713E-5     260.0
.
More cards here
.
 1.2323E-5     760.0
 1.2467E-5     815.0
 1.2689E-5     900.0
*USER MATERIAL, CONSTANTS=176
         0        13.      16.      1.0E4     1600.0    5.8E5     1.37E4    5.8E5
     1.37E4       2.0      10.0      0.75     1.E-3       0.0       0.0       0.0
      23.0     260.0     315.0     365.0     415.0     465.0     482.0     500.0
     525.0     550.0     575.0     600.0     650.0     760.0     815.0     900.0
  112000.0  108030.0  106130.0  104090.0  101740.0   99085.0   98113.0   97045.0
   95497.0   93873.0   92172.0   90395.0   86612.0   77000.0   72000.0   63000.0
     0.340     0.340     0.340     0.340     0.340     0.340     0.340     0.340
     0.340     0.340     0.340     0.340     0.340     0.340     0.340     0.340
      4.80      3.50     3.054     2.649     2.243     1.838     1.700     1.500
     1.280     1.100     0.970     0.820     0.740     0.580     0.550     0.550
    1550.0    1300.0    1250.4    1205.4    1160.4    1115.3    1100.0    1089.3
    1074.4    1059.5    1044.6    1029.8    1000.0     600.0     300.0     300.0
     100.0     300.0     390.0     500.0     660.0     960.0    1100.0    1300.0
    1670.0    2100.0    2600.0    3700.0    3800.0    4000.0    4100.0    4300.0
       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
       0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
     0.350     0.350     1.502     2.549     3.597     4.644     5.000     5.763
     6.822     7.881     8.941     10.00      10.0      15.0      30.00     30.00
       3.0       3.0       3.0       3.0       3.0       3.0       3.0       3.0
       3.0       3.0       3.0       3.0       3.0       3.0       3.0       3.0
       3.0       3.0       3.0       3.0       3.0       3.0       3.0       3.0
       3.0       3.0       3.0       3.0       3.0       3.0       3.0       3.0

*DEPVAR
20
*INITIAL CONDITIONS, TYPE=SOLUTION, USER
```

Figure 3.1 Listing of Material Properties within ABAQUS input file.

From Figure 3.1, note that the thermal expansion data are input in the same manner as any other model used in ABAQUS. On the *USER MATERIAL card, CONSTANTS=204 indicates that 204 constants are read in from the remaining cards in this data block which are described below. The *DEPVAR card and the following card indicate that 20 state variables are allocated in the analysis. The next card, *INITIAL CONDITIONS, is required to flag the main program to call the subroutine SDVINI, which initializes state variables.

The variables defined under the *USER MATERIAL card are best described with Figure 3.2. The first constant, IPLANE (=1), is a flag to use plane stress elements. The second constant NTIP is the number of temperature independent material properties and solution control variables. The third constant NTSET is the number of points that define the temperature array. Appearing next are temperature-independent material properties, (e.g., D_0, C_{51}, C_{61}, ...) and the solution control variables (e.g., XNEWDT, INTER...). The temperature array T(1),T(2), ... T(NTSET) appears next, which is preceded by the elastic modulus array TE(1),TE(2), ... TE(NTSET). The following sets of arrays define Poisson's Ratio (ν), n, Z_2, Z_3, m_1, m_2, r_1, and r_2, respectively.

```
*USER MATERIAL, CONSTANTS=NTOTAL
   IPLANE      NTIP      NTSET        D0        Z1        C51        C61        C52
      C62    XNEWDT      INTER     RELAX     TOLER      extra      extra      extra
     T(1)      T(2)       T(3)      ....      ....   T(NTSET)     TE(1)      TE(2)
    TE(3)      ....       ....   TE(NTSET)   TNU(1)     TNU(2)     TNU(3)     ....
....    TNU(NTSET)       TN(1)      TN(2)     TN(3)      ....       ....    TN(NTSET)
  TZ2(1)     TZ2(2)     TZ2(3)      ....      ....    TZ2(NTSET)    TZ3(1)     TZ3(2)
  TZ3(3)      ....       ....   TZ3(NTSET)   TM1(1)     TM1(2)     TM1(3)     ....
  ....   TM1(NTSET)      TM2(1)     TM2(2)    TM2(3)      ....       ....    TM2(NTSET)
  TR1(1)     TR1(2)     TR1(3)      ...       ....   TR1(NTSET)    TR2(1)     TR2(2)
  TR2(3)      ...        ...    TR2(NTSET)
```

Figure 3.2 Bodner-Partom material property array use with ABAQUS.

3.3 Initializing State Variables

For the Bodner-Partom theory presented above, the only state variable that has a non-zero initial value is Z^I. According to the theory, the initial value of Z^I is Z_0, which is temperature dependent. The `SDVINI` subroutine within ABAQUS, which initializes state variables, does not automatically assign a value of Z^I. Thus the initial value of Z^I must be "hard wired" into the subroutine. In other words, the subroutine `SDVINI` must be edited to assign the appropriate values of Z^I. The section of code that must be edited is shown by the *italic* print in Figure 3.3. Note that the initial value of Z^I is assigned to state variable #13.

```
      SUBROUTINE SDVINI(STATEV,COORDS,NSTATV,NCRDS,NOEL,NPT,LAYER,KSPT)
C
      INCLUDE 'ABA_PARAM.INC'
C
       DIMENSION STATEV(NSTATV), COORDS(NCRDS)
c
      COMMON /matconst/  E,  G, a3k, anu, dg, d3k, dnu, d0,
     &        an, am1, am2, z1, z2, z3, a1, a2, r1, r2,
     &        dn, dm1, dm2, dz1,dz2,dz3,da1,da2,dr1,dr2
c
       statev(1)  = 0.0
       statev(2)  = 0.0
       statev(3)  = 0.0
       statev(4)  = 0.0
       statev(5)  = 0.0
       statev(6)  = 0.0
       statev(7)  = 0.0
       statev(8)  = 0.0
       statev(9)  = 0.0
       statev(10) = 0.0
       statev(11) = 0.0
       statev(12) = 0.0
c
c  Initializing ZI to Z0
c
c   Neu's Constants (Table 2-B) [4]
c25
c25        statev(13) = 1550.
c
c650
c650       statev(13) = 1000.
c
c900
c900       statev(13) =  300.
c
c  for Sander's Constants (Table 2-C) [18]
c25
c25        statev(13) = 2950.
c650
      statev(13) =  200.
c815
c815       statev(13) =  150.
c
      statev(14) = 0.0
      statev(15) = 0.0
      statev(16) = 0.0
      statev(17) = 0.0
c
       return
       end
```

Figure 3.3 Listing of subroutine SDVINI.

3.4 Errors and Debugging

The Bodner-Partom subroutines were extensively tested to minimize errors and maximize numerical efficiency. Even though the subroutines were thoroughly tested, some conditions will result in failure of the subroutines to converge to a solution. Documented here are the errors that were most commonly encountered during this test period.

1. **Wrong initial value of Z^I.** This particular error is most likely to occur if several analyses are required with different initial temperatures. The user should be aware that if Z^I is input erroneously, the numerical algorithms may function normally with erroneous output. In addition, the subroutines will not warn the user if an inappropriate value has been assigned. See Section 3.3 about assigning a value to Z^I

2. **Numerical overflow.** This error is most likely to occur if the time step size is too large for the algorithm. To avoid this error assign `INTER` a larger number. For most cases `INTER=10` is adequate.

3. **Nonconvergent Solutions.** There is a couple of reasons that would cause this error to occur. The most common reasons are given here:

   3a. The time step size for the problem may be too great. Increase the value of `INTER`.

   3b. Discontinuities in the temperature history are likely to occur if multiple time steps are not used correctly. The default for the `*AMPLIDUTE CARD` is for time to start from the beginning of the solution step, rather than total time. The flag "`TIME=TOTAL TIME`" will amend this problem.

   3c. A potential problem can occur with stress-reversal at fairly high loading rates with the constants of Table 2-B. This problem is an artifact of the chosen constants at 650°C rather than the numerical algorithms (e.g., $Z_3 > Z_2$). Two solution to the problem are to limit the value to Z^D to be positive or to use the constants of Table 2-C.

SECTION 4.0

EXAMPLES AND VERIFICATION

This section provides an example of the application of the Bodner-Partom user-defined subroutines with the Timetal®21S material parameters. This section presents several numerical exercises that verify the numerical accuracy of the subroutines in Appendix 1. The differential equation solver within *Mathematica* [19] provides a very accurate base-line solution for the evaluation of accuracy. This section provides a general survey of numerical accuracy, while Section 5 provides a detailed investigation.

4.1 Numerical Examples

The file, case1.inp, (see Appendix 2) is an example ABAQUS input file that contains the Timetal®21S material parameters. The example consists of a single axisymmetric element unidirectionally loaded at a constant strain rate of $833.3 \times 10^{-6} s^{-1}$ and constant temperature of 25°C, which has the same conditions as case 1 (discussed below). Input of the Timetal®21S material properties starts with the string `*MATERIAL, NAME=B21S_3` (see Appendix 2). In these example analyses, the thermal expansion coefficients are set to zero (commented out) since the comparisons only consider mechanical strain (total minus thermal strain). The output for this numerical example is provided below, in Figure 4.1.

4.2 Solution Verification

This section presents five test cases that were compared to the solution obtained by *Mathematica,* which has an associated error of approximately 10^{-4} percent. The measure for accuracy is the percent deviation of the finite element stress solution from the *Mathematica* results. The percent stress deviation, %D, is written as:

$$\%D = 100\% * \left| \frac{\sigma_{FE} - \sigma_{math}}{\sigma_{math}} \right|. \qquad \text{Eq. 9}$$

Where σ_{FE} and σ_{math} are the stresses obtained from the finite element and *Mathematica* solutions, respectively. Smaller percent deviation is indicative of a more accurate numerical solution, rather than a better fit with experimental data.

All cases simulate a monotonically increasing tensile loaded element with constant mechanical strain rates of either $833.3 \times 10^{-6} s^{-1}$ or $8.33 \times 10^{-6} s^{-1}$. Table 4-A summarizes all the test conditions and location of the Mathematica solution files in the appendices. The verification cases compare well within 1.0 % deviation for both sets of Bodner-Partom material constants from Tables 2-B and 2-C. Results are shown only for the solutions obtained from the constants of Table 2-B since both sets of constants shared similar solution sensitivity to loading type, time increments, and error tolerances. The default solution control parameters from which these cases were run are provided in Table 4-B.

The percent deviation between the ABAQUS and *Mathematica* solutions depends largely on loading type and number of solution increments. Under isothermal conditions, cases 1, 2, and 3 show good agreement between the ABAQUS and *Mathematica* solutions, as illustrated in Figures 4.1, 4.2, and 4.3, respectively. The percent deviations are below 0.2% for all three cases, which is well within the required accuracy for most applications.

Table 4-A

Verification Test Cases

Test Case Number	Temp. (°C)	Thermal Condition	Strain Rate ($10^{-6}s^{-1}$)	Number of Solution Steps	Solution File	Appendix
1	25	isotherm.	833.3	48	vt1.math	3
2	650	isotherm.	833.3	48	vt2.math	4
3	650	isotherm.	8.333	48	vt3.math	5
4	25/482/25	non-isotherm.	833.3	192	vt4.math	6
5	650/760/650	non-isotherm.	833.3	96	vt5.math	7

Table 4-B

Default Solution Control Parameters

Parameter	Default Values	Description
INTER	10	number of initial sub-increment
RELAX	0.75	relaxation factor
TOLER	10^{-4}	convergence tolerance
XNEWDT	1.0	auto time-step size flag

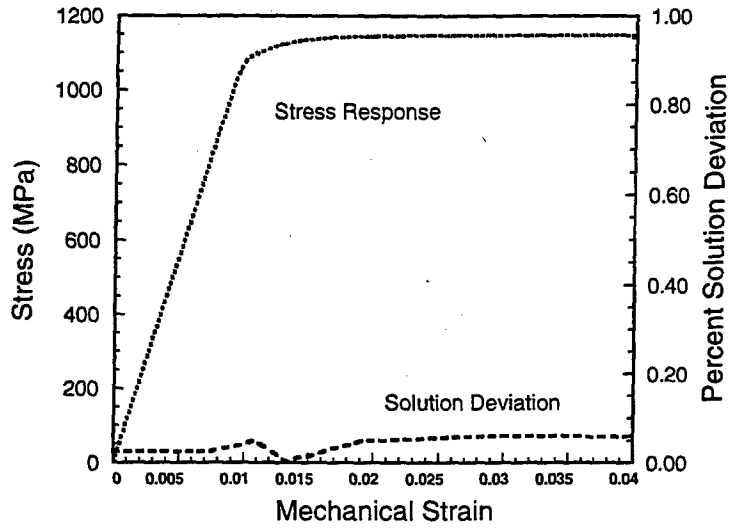


Figure 4.1 Stress-strain response and solution deviation for case 1.

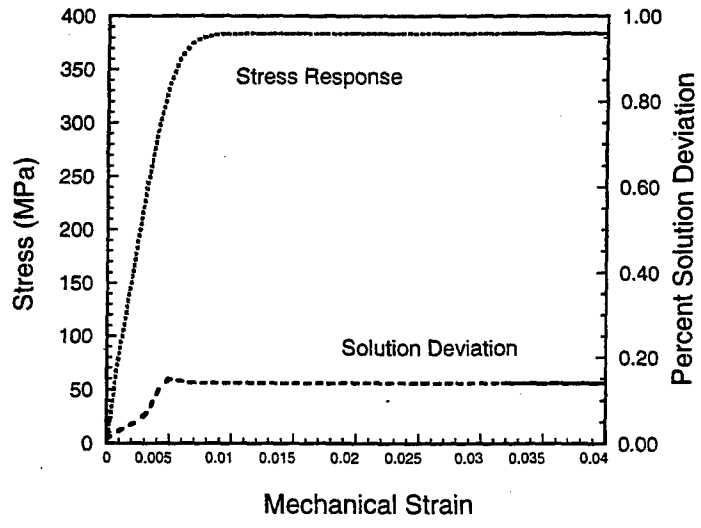


Figure 4.2 Stress-strain response and solution deviation for case 2.

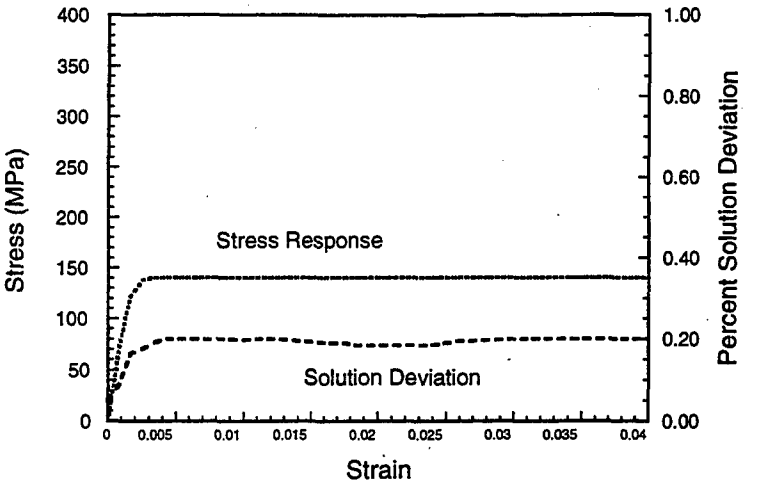


Figure 4.3 Stress-strain response and solution deviation for case 3.

Two nonisothermal cases are proposed to evaluate the performance of the numerical algorithms under combined thermal and mechanical loads. Cases 4 and 5 consist of mechanical loads with two different temperature ranges. The temperature range for case 4 is from 25° to 482°C, which covers a strain-rate insensitive region of Timetal®21S. The temperature profile and stress-strain response from case 4 are illustrated in Figure 4.4, with the associated finite element solution deviation from *Mathematica* presented in Figure 4.5. Note that the error is minimal during the isothermal portion of loading, while the error increases slightly with increasing temperature and then remains relatively constant.

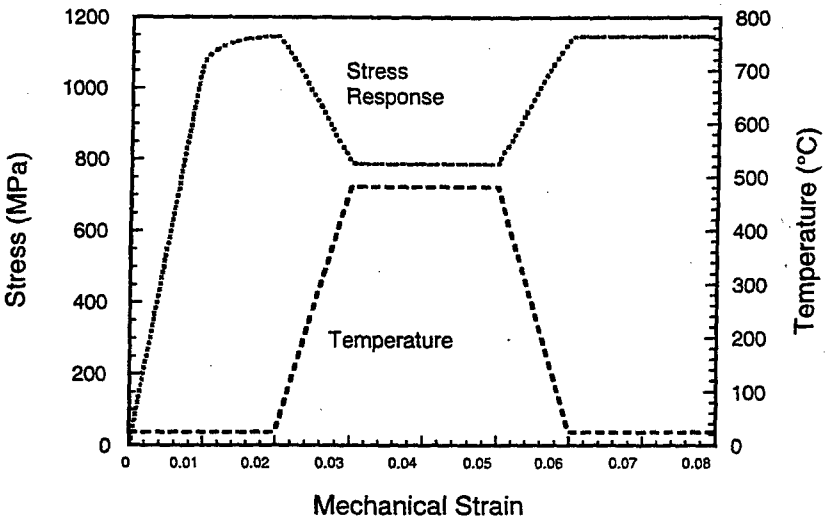


Figure 4.4 Temperature profile and stress response for case 4.

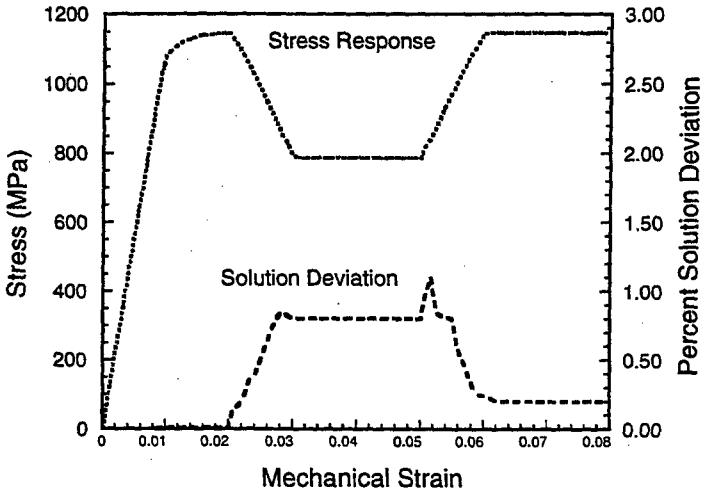


Figure 4.5 Stress-strain response and solution deviation for case 4.

The temperature profile for case 5 is from 650° to 760°C, (see Figure 4.6) where the Timetal®21S exhibits more strain-rate sensitivity and time dependent response. The respective solution deviation, given in Figure 4.7, shows the finite element stresses are well within 0.5 percent of predicting the *Mathematica* results for the isothermal segments of the loading. The percent deviation increases slightly during the non-isothermal periods of loading and reaches a single peak at 2.4%. More solution increments are required to improve the solution accuracy, as discussed in Section 5.

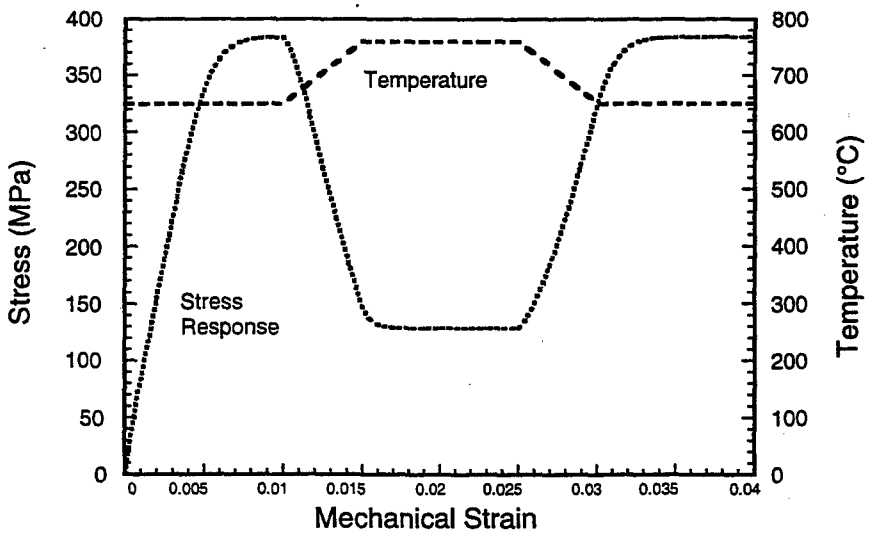


Figure 4.6 Temperature profile and stress response for case 5.

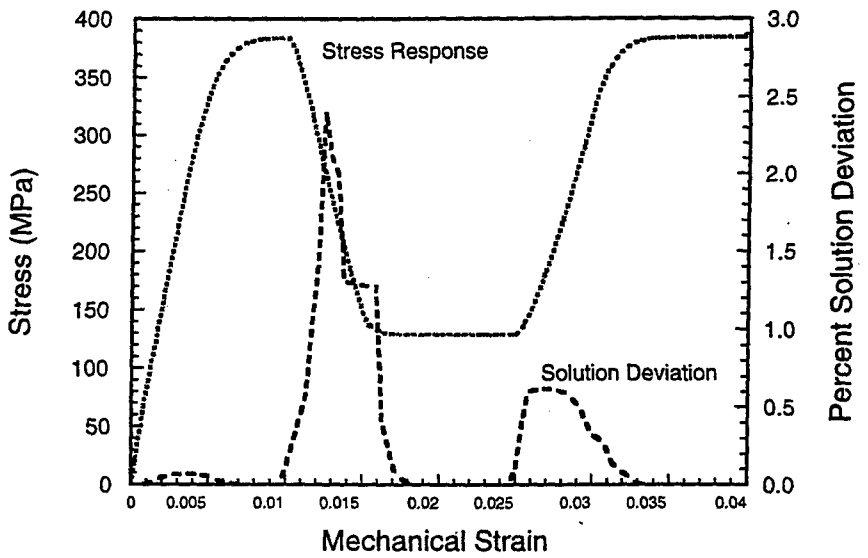


Figure 4.7 Stress-strain response and solution deviation for case 5.

Solution accuracy and computer CPU usage for the test cases are summarized in Table 4-C. For all the isothermal cases, the numerical accuracy is very good (less than 0.5%). The numerical accuracy for the nonisothermal simulations was less than desirable. Thus an investigation was considered to optimize numerical accuracy with respect to computational efficiency. The findings from this investigation are given below.

The computational speed of each of the cases is quite low (≈10 CPU seconds). Note that the computer usage is increased when the simulation conditions are at elevated temperatures and slower loading rates. This increase in CPU usage is consistent with more solution iterations required when significant inelastic deformation occurs within the material.

Table 4-C

Summary of Solution Performance

Case	Solution Increments	Step Size (sec)	CPU* (sec)	Maximum Error (%)
1	48	1.0	3.2	0.07
2	48	1.0	5.5	0.14
3	48	100.0	12.6	0.20
4	192	0.5	12.8	1.20
5	96	0.5	10.0	2.41

*Convex, Metaseries 2

SECTION 5.0
COMPUTATIONAL PERFORMANCE

Case 5 from Section 4 revealed an unacceptable level of solution deviation between finite element and *Mathematica* solutions (2.41%). Thus, a systematic study was conducted to determine the best method to improve the accuracy of the solution. The performance of the subroutines was evaluated with the thermomechanical loading of case 5. The investigation considered three basic types of parameters -- number of solution increments, relaxation factor and convergence tolerances.

5.1 Influence of Number of Solution Increments

The number of solution increments can be changed at two different levels -- major and minor. The global convergence of displacements occurs at the major solution increment, `NSTEP`. In keeping all parameters at their default values, the `NSTEP` was changed from 96 to 192 and 384. The solution accuracy is improved significantly, from 2.41% to 0.12 %, as shown in Figure 5.1. For the improved accuracy, the CPU usage more than doubled, from 10 to 23 CPU seconds†. In reference to Table 5-A, cases 5, 5b and 5c correspond to `NSTEP`=96, 192, and 384, respectively.

At individual Gauss points, the variable `INTER` sub-divides the major increment into equal segments. Similar to increasing the number of global steps, increasing the number of sub-increments from 10 to 40 improves the solution accuracy from 2.41% to 0.092%, as illustrated in Figure 5.2. The improvement in

† CPU time on a Convex Metaseries 2

accuracy did not increase CPU usage significantly. In reference to Table 5-A, cases 5, 5d and 5e correspond to INTER=10, 20, and 40, respectively.

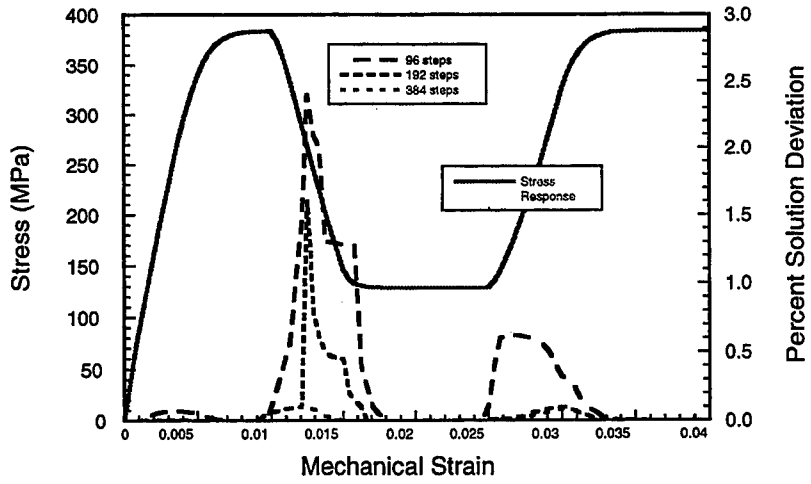


Figure 5.1 Number of major time steps influence on solution deviation.

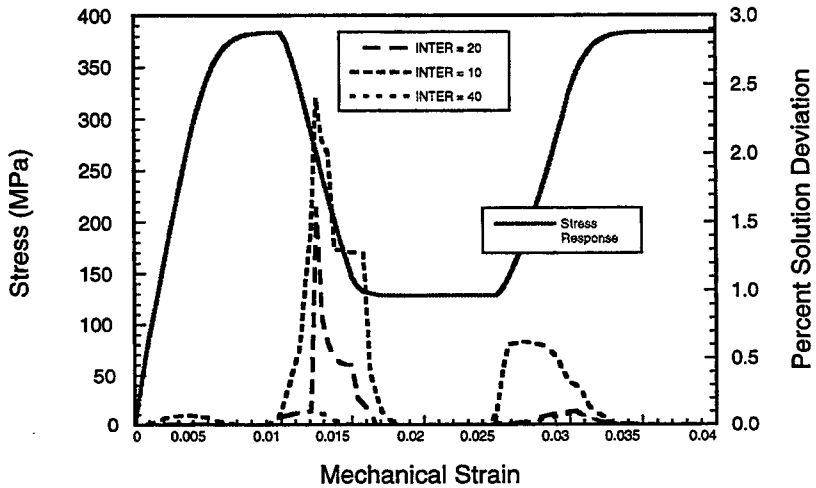


Figure 5.2 Number of sub-increments INTER influence on solution deviation.

5.2 Influence of Relaxation Factor

In principle, changes in the relaxation factor should not influence the accuracy of the solution; however, the factor should reduce the number of iterations required for convergence. In the ideal linear elastic case, a relaxation factor of 1.0 should provide the quickest convergence, while a relaxation factor of 0.5 is best suited for highly non-linear segments of loading. For case 5, the relaxation factor does not significantly improve the accuracy of the solution, as illustrated in Figure 5.3. The relaxation factor of 0.9 provided a slight improvement in CPU usage (9.7 seconds†) compared to that of `RELAX=0.75` (10.0 seconds†). Previous experience [20], showed that a relaxation factor of 0.75 is best suited for general usage. In reference to Table 5-A, the relaxation factors of 0.75, 0.65 and 0.90 correspond to cases 5, 5f and 5g respectively.

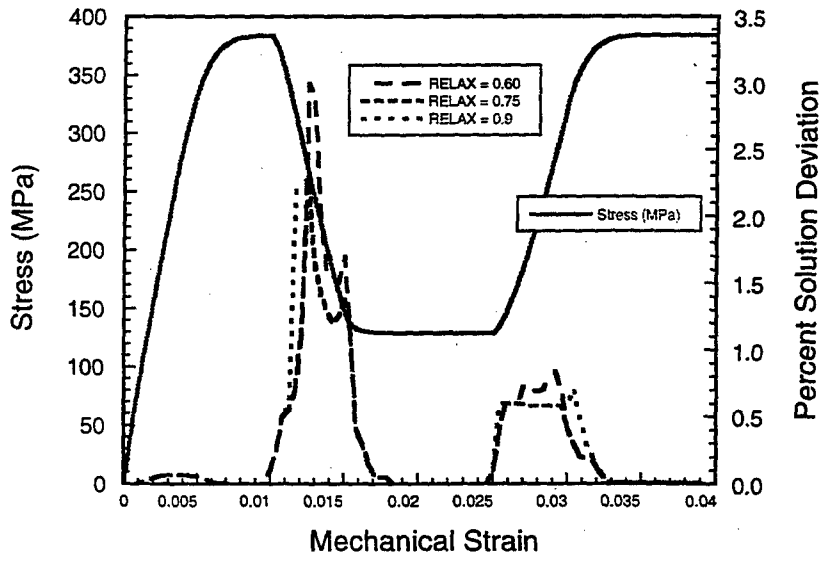


Figure 5.3 Relaxation factor `RELAX` influence on solution deviation.

† CPU time on a Convex Metaseries 2

5.3 Influence of Solution Tolerance

Within the Bodner-Partom iteration algorithm, the tolerance parameter TOLER measures the change of stress state from its previous guess. A change of TOLER, from 10^{-4} to 10^{-5} lowers the level of error by a factor of two, from 2.4% to 1.3%, and with TOLER=10^{-6} minimal improvement is found, as illustrated in Figure 5.4. As expected, the change in TOLER from 10^{-4} to 10^{-6} does increase the computer usage from 10.0 to 23.4 CPU seconds†. Values of TOLER larger than 10^{-3} can produce non-convergent solutions, especially if the finite element configuration contains additional non-linearities (e.g., gap elements). A smaller value for TOLER may be required to achieve convergence for these highly non-linear situations. In reference to Table 5-A, cases 5, 5h and 5i correspond to TOLER = 10^{-4}, 10^{-5}, and 10^{-6} respectively.

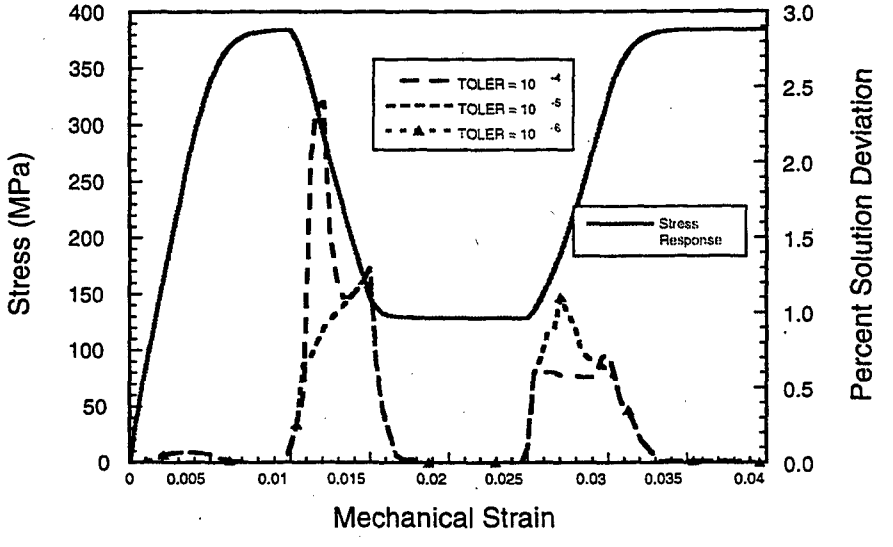


Figure 5.4 Influence of TOLER on solution deviation.

† CPU time on a Convex Metaseries 2

Global tolerance GTOL measures the convergence of displacements and forces (energy) within the finite element mesh for each major time step. For this investigation, GTOL is set by the following input cards:

```
*CONTROLS, PARAMETER=FIELD, FIELD=GLOBAL
   0.0005
```

Tightening the global tolerance (smaller GTOL) did not improve the accuracy of the solution, in fact for GTOL=5.x10^{-6}, the solution deviation actually increased slightly. Under this tolerance, the CPU usage increased 62%, from 10.0 to 16.2 seconds† when compared to the base-line case. In reference to Table 5-A, cases 5, 5h and 5i correspond to GTOL = 5x10^{-3}, 5x10^{-4}, and 5x10^{-5} respectively.

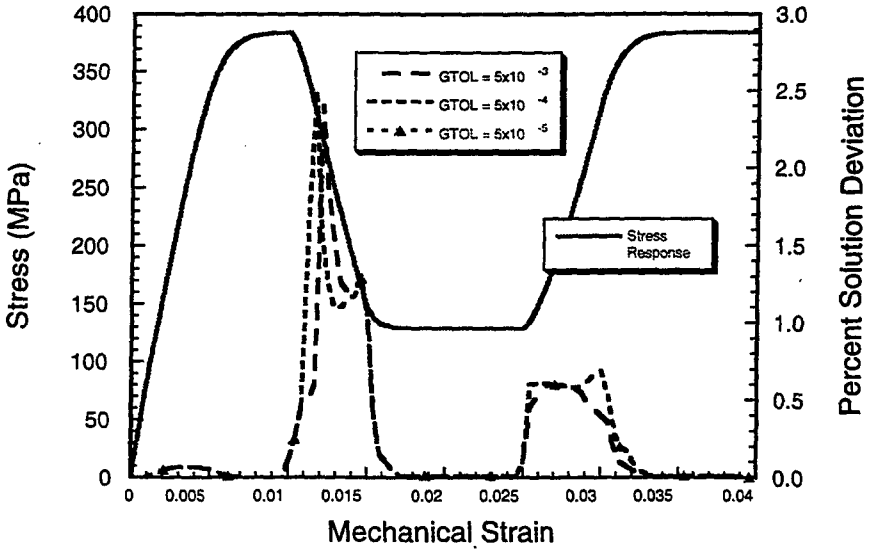


Figure 5.5 Global tolerance GTOL influence on solution deviation.

† CPU time on a Convex Metaseries 2

5.4 Summary of Performance

Overall, the results can be readily summarized by the definition of the performance factor:

$$P_f = \frac{1}{2}\left[\frac{t^*_{cpu}}{t_{cpu}} + \frac{\%D^*}{\%D}\right] \qquad \text{Eq. 10}$$

where t_{cpu} and $\%D$ are the CPU time and percent solution deviation for each test case. The values of t^*_{cpu} and $\%D^*$ are the CPU time and percent solution deviation for the base-line case 5. Higher values of P_f are indicative of improved accuracy and/or lower computer usage. Table 5-A presents a summary of the influence of solution parameters on computer CPU time, solution deviation and performance for the test cases in Section 5.

Increases in the number of solution increments significantly decrease the solution deviation, as shown in Figure 5.6. Some improvement is observed with decreases in TOLER, with minimal improvement with tighter global tolerance GTOL. Additional solution increments and tighter tolerances generally increase CPU usage, as illustrated in Figure 5.7. The comparison of overall performance, P_f, as seen in Figure 5.8, revealed that the best improvement in accuracy with the least debit in CPU usage, is found in increases in INTER.

Table 5-A

Summary of Solution Deviation, CPU Usage and Performance Rating

Case	NSTEP	INTER	RELAX	TOLER	GTOL	%D	CPU time† (sec)	P_f
5	96	10	0.75	10^{-4}	$5x10^{-3}$	2.41	10.0	1.00
5b	192	10	0.75	10^{-4}	$5x10^{-3}$	1.60	13.9	0.98
5c	384	10	0.75	10^{-4}	$5x10^{-3}$	0.077	26.3	0.76
5	96	10	0.75	10^{-4}	$5x10^{-3}$	2.41	10.0	1.00
5d	96	20	0.75	10^{-4}	$5x10^{-3}$	1.55	9.72	1.25
5e	96	40	0.75	10^{-4}	$5x10^{-3}$	0.092	11.4	1.73
5f	96	10	0.60	10^{-4}	$5x10^{-3}$	3.00	10.7	0.87
5	96	10	0.75	10^{-4}	$5x10^{-3}$	2.41	10.0	1.00
5g	96	10	0.90	10^{-4}	$5x10^{-3}$	2.30	9.70	1.04
5a	96	10	0.75	10^{-4}	$5x10^{-3}$	2.41	10.0	1.00
5h	96	10	0.75	10^{-5}	$5x10^{-3}$	1.30	16.2	0.93
5i	96	10	0.75	10^{-6}	$5x10^{-3}$	1.30	23.4	0.71
5a	96	10	0.75	10^{-4}	$5x10^{-3}$	2.41	10.0	1.00
5j	96	10	0.75	10^{-4}	$5x10^{-4}$	2.41	11.7	0.93
5k	96	10	0.75	10^{-4}	$5x10^{-5}$	2.45	16.2	0.77

† CPU time on a Convex Metaseries 2

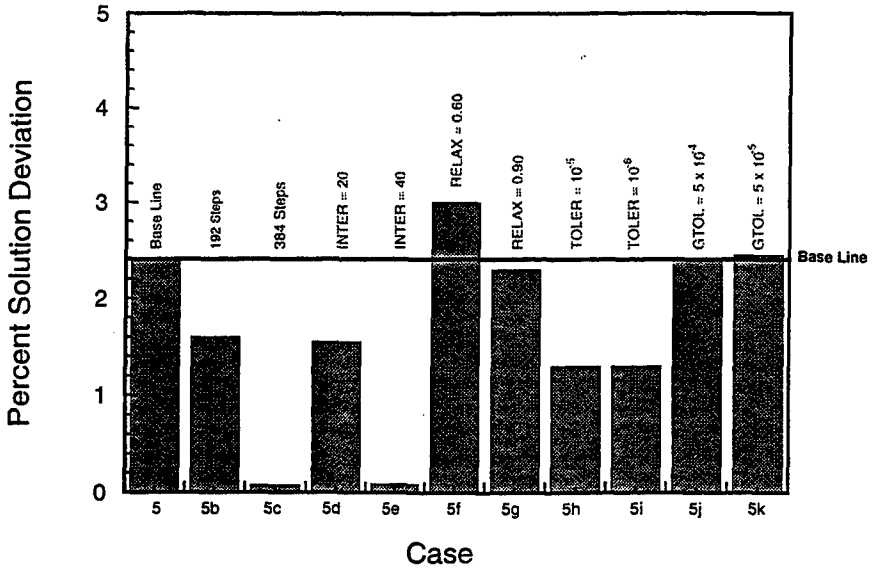


Figure 5.6 Summary of solution deviation.

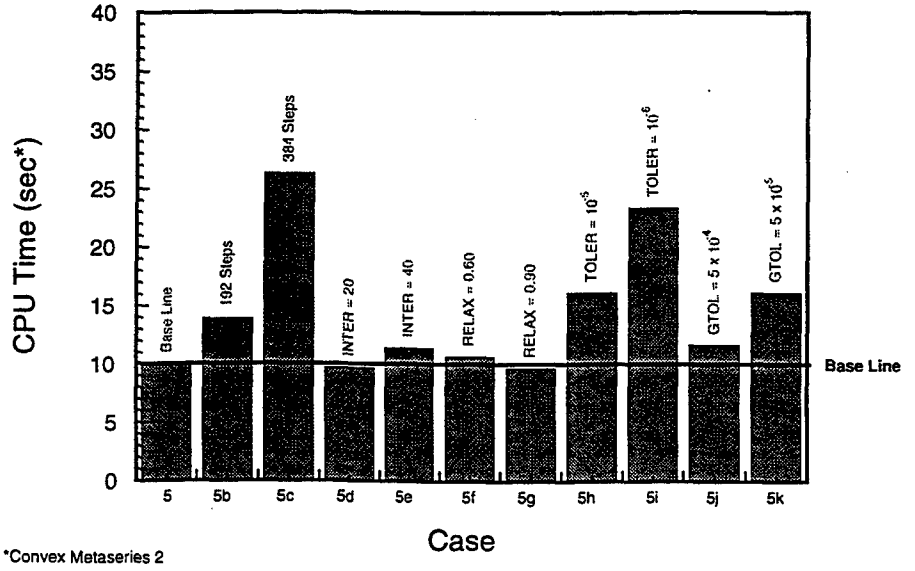


Figure 5.7 Summary of CPU usage.

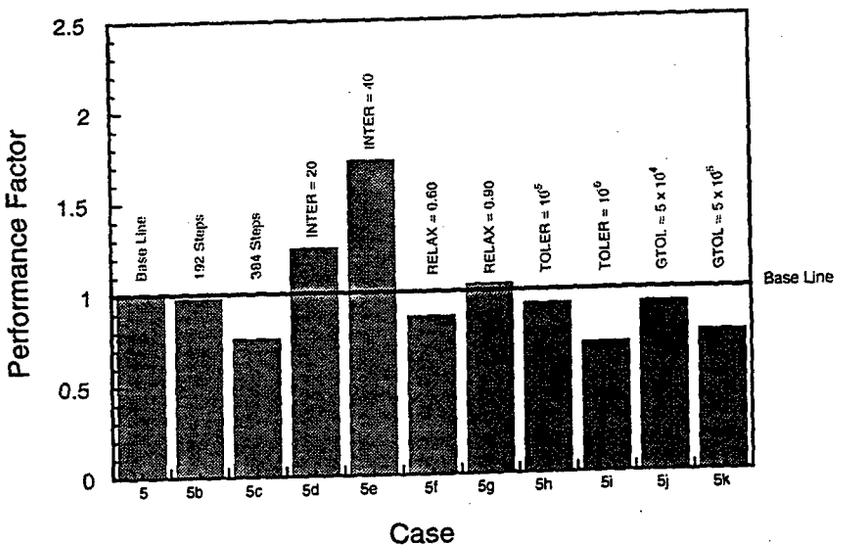


Figure 5.8 Summary of performance factor, P_f.

SECTION 6.0

STRUCTURE AND ALGORITHMS

This section describes the structure and numerical algorithms of the Bodner-Partom subroutines that solve the constitutive equations of Section 2.1. Several tables briefly describe subroutine nomenclature and variable usage. Also, a description of the main subroutine's numerical algorithm is included. The subroutines described in this section are listed in Appendix A and are available by electronic file transfer.

The user-defined material subroutines for ABAQUS are written for general element usage; thus, the use of two different sets of subroutines for 2- and 3-dimensions is not required. The algorithms within ABAQUS account for the active number of normal (`NDI`) and shear (`NSHR`) components of stress and strains. The only exception is for the plane stress case, which requires special input flag. To use the plane-stress elements (e.g., `CPS8`) it is required that `IPLANE = 1`, rather than `IPLANE = 0`.

6.1 Subroutines and Common Blocks

The functionality of subroutines found in `usub_db.f` (see Appendix A) is summarized in Table 6-A. Subroutines `UMAT` and `SDVINI` are native to the ABAQUS code. Subroutines `SETPROPS`, `GETPROPS`, `STGET3`, `STPUT3`, and `DBODNER` are written by the author. The first four of these subroutines control the values of material properties and state variables. The subroutine `DBODNER` contains the numerical algorithms that solve the Bodner-Partom constitutive equations found in Section 2.1.

Table 6-A

Subroutine Names and Functions

Subroutine*	Called From	Function
`UMAT`	`ABAQUS`	Determines stresses from user-defined material behavior models
`SDVINI`	`ABAQUS`	Initializes state variables
`SETPROPS`	`UMAT`	Initializes the material properties
`GETPROPS`	`UMAT`	Assigns material properties
`STGET`	`UMAT`	Retrieves state variables from working variable `STATEV`
`STPUT`	`UMAT`	Stores state variables from array `STATEV`
`DBODNER`	`UMAT`	Solves the constitutive equations of the Bodner-Partom model

The four common blocks -- MATCONST, BPSTATE, MATPROP, and CONTROL -- retain values of certain variables entering or leaving the subroutines in which they appear. MATCONST transfers material constants between UMAT, DBODNER and GETPROPS. Variables that make up common block MATCONST are listed in Table 6-B. Common block BPSTATE transfers the values of the internal state variables (e.g., inelastic strain, isotropic hardening, etc.) between all the subroutines. The state variables of common block BPSTATE are listed in Table 6-C.

Common block CONTROL transfers the values of variables that control various aspects of the solution, e.g., convergence and time step size. The variable names within common block CONTROL are listed Table 6-D. Common block MATPROP stores all the temperature dependent material properties that are initialized in subroutine SETPROPS. The variables contained within common block MATPROP are described in Table 6-E.

Table 6-B

Internal Variables of Common Block MATCONST

Variable Name	Symbol	Variable Description
E	E	elastic modulus
AN	n	kinetic (strain rate sensitivity) parameter
Z0, Z2	Z_0, Z_2	fully recovered (minimum) value of isotropic hardening
Z3	Z_3	limiting value of directional hardening
AM2	m_2	rate coefficient of directional hardening
A1, A2	A_1, A_2	thermal recovery coefficients for hardening,
AM1	m_1	isotropic hardening rate coefficient
Z1	Z_1	limiting value of isotropic hardening
R1, R2	r_1, r_2	thermal recovery exponent for hardening,
D0	D_0	limiting inelastic strain rate
ANU	ν	Poisson's ratio
G	G	shear modulus
A3K	$3 \bullet K$	3 times bulk modulus
DG	$\frac{\partial G}{\partial T}\frac{\partial T}{\partial t}dt$	differential shear modulus with solution increment
D3K	$3 \bullet \frac{\partial K}{\partial T}\frac{\partial T}{\partial t}dt$	three times the differential bulk modulus with solution increment
DNU	$\frac{\partial \nu}{\partial T}\frac{\partial T}{\partial t}dt$	differential Poison's ratio with solution increment
DM1	$\frac{\delta m_1}{\partial T}\frac{\partial T}{\partial t}dt$	differential 'm_1' with solution increment

Table 6-B (continued)

Internal Variables of Common Block MATCONST

Variable Name	Symbol	Variable Description
DM2	$\frac{\delta m_2}{\delta T}\frac{\delta T}{\delta t}dt$	differential 'm_2' with solution increment
DR1	$\frac{\partial r_1}{\partial T}\frac{\partial T}{\partial t}dt$	differential r_1 with solution increment
DR2	$\frac{\partial r_2}{\partial T}\frac{\partial T}{\partial t}dt$	differential r_2 with solution increment
DN	$\frac{\partial n}{\partial T}\frac{\partial T}{\partial t}dt$	differential 'n' with solution increment
DZ0	$\frac{\partial Z_0}{\partial T}\frac{\partial T}{\partial t}dt$	differential Z_0 with solution increment
DZ2	$\frac{\partial Z_2}{\partial T}\frac{\partial T}{\partial t}dt$	differential Z_2 with solution increment
DZ3	$\frac{\partial Z_3}{\partial T}\frac{\partial T}{\partial t}dt$	differential Z_3 with solution increment
DA1	$\frac{\partial A_1}{\partial T}\frac{\partial T}{\partial t}dt$	differential A_1 with solution increment
DA2	$\frac{\partial A_2}{\partial T}\frac{\partial T}{\partial t}dt$	differential A_2 with solution increment

Table 6-C

State Variables of Common Blocks BPSTATE

Variable Name	Symbol	Variable Description
EIN	ε_{ij}^{in}	components of inelastic strain
ZI	Z_I	current isotropic drag stress
SIGEFF	$\sqrt{3J_2}$	effective stress
ZD	Z_D	current magnitude of directional drag stress
BETA	βij	components of directional drag stress
XOUT1,2,3	---	extra state variables that can be used for output (especially nice for debugging)

Table 6-D

Variables of Common Block CONTROL

Variable Name	Common Values	Variable Description
XNEWDT	0.5	ABAQUS time increment cutting factor
RELAX	0.75	relaxation factor for numerical integration
TOLER	10^{-4} to 10^{-6}	tolerance for solution converge for stress and state variables
INTER	10	initial sub-time cutting factor
IPLANE	0	= 1 for plane stress solution = 0 for all other cases

Table 6-E

Variables of Common Block MATPROP

Variable Name	Variable Description
T	Temperature array
TE	Temperature dependent elastic modulus array
TNU	Temperature dependent Poisson's ratio array
TN	Temperature dependent material property n
TZ2	Temperature dependent material property Z_2
TZ3	Temperature dependent material property Z_3
TM2	Temperature dependent material property m_2
C5	Coefficient C_5 for A_1 and A_2 material properties (see Table 2)
C6	Exponential C_6 for A_1 and A_2 material properties (see Table 2-B)
NTSET	Number of members in temperature dependent arrays

6.2 Internal Variable Names

This section summarizes the internal variables of the user-defined subroutines. The ABAQUS internal variables that enter into the user-defined subroutine UMAT are listed in Table 6-F. The descriptions of the other variables are found in the ABAQUS user manuals [2]. The primary variables found in subroutine DBODNER are summarized in Table 6-G.

Table 6-F
ABAQUS Variables Supplied to UMAT Subroutine

Variable Name	Variable Description
STRESS	Components of stress
STATEV	State variables
DDSDDE	Tangent stiffness Matrix
STRAN	Mechanical strain at beginning of solution increment
DSTRAN	Differential mechanical strain
TIME	Time
DTIME	Differential time
TEMP	Temperature
DTEMP	Differential Temperature
NDI	Number of normal components of stress/strain
NSHR	Number of shear components of stress/strain
NTENS	Total number of components of stress/strain
NSTATV	Number of state variables
PROPS	User-defined material properties (see Section 5)
COORDS	Spatial coordinates of Gauss point location
PNEWDT	Flag for time step cutting
NOEL	Current element number
NPT	Current Gauss point number
KSTEP	Current ABAQUS time step
KINC	Current ABAQUS time increment

Table 6-G

Internal Variables in DBODNER

Variable Name	Variable Description
DEVEPS	Deviatoric strains
DDSDDE	Stiffness matrix
AVGEPS	Average (mean) normal strain (bulk strain/3)
DDEPS	Sub-incremental deviatoric strains
DEIEFF	Effective inelastic strain increment
DEPS	Incremental mechanical strain (total minus thermal)
DEPSI	Incremental inelastic deviatoric strain
DEVSIG	Deviatoric stress
ICOUNT	Iteration loop counter
ISUB	Number of sub-increments within DBODNER
RELAX	Relaxation factor for new stress estimate
TOLER	Tolerance for convergence
SIGNEW	Effective stress of previously converged stress state
SIGOLD	Old estimate effective stress
SIGEST	New effective stress estimate
SIGHYD	Hydrostatic (mean) stress
STRES0	Stress of previously converged stress state
STRESS	Current stress
TIME2	Current time value
EINEST	Estimated inelastic strains
EIN0	Inelastic strains of previously converged stress state
BETADOT	Directional drag stress rate vector
BETAEST	Estimated directional drag stress vector
BETA0	Directional drag stress of previously converged stress state
U, V	Directional unit vectors for stress and directional hardening
ZIEST	Estimated isotropic drag stress

6.3 Numerical Algorithms

The algorithms that solve the Bodner-Partom constitutive equations consist of a mixture of iteration loops and sub-incremental schemes. This combination of iterations and sub-incrementation works well for the inherently stiff nature of the equations and Timetal®21S material parameters. The algorithms discussed in this section are found in subroutine DBODNER.

The algorithm, as shown in Figure 6.1, consists of an iteration loop and a sub-incremental solution scheme. Prior to sub-incremental integration, state variable values assume their pre-incremental values; the number of sub-increments (ISUB) initializes to INTER; and constant rate variables are scaled by TFACTOR. The integration of the sub-incremental loop begins at Step III. The primary iteration loop starts at step III.D and converges on stress, inelastic strain, isotropic hardening and the directional hardening parameters. Non-convergent solutions, as defined by a maximum limit set on iteration steps, return to the beginning of the sub-incremental integration (Step I) with an increase in the number of sub-increments, ISUB. When the maximum number of sub-incremental step equals 128, the solution solver ceases operation and prints a diagnostic debugging output.

I. Store variable at the beginning of solution increment
   A. state variables
   B. temperature-dependent material parameters

II. Initialize variables for sub-incremental solution step cutting of `ISUB`
   A. initialize estimated new values for state variables
   B. restore material parameters from Step I
   C. determine new material parameters rates by factors of `ISUB`
   D. restore state variables to values from Step I

III Begin sub-incremental solution integration
   A. update all temperature-dependent parameters to end of sub-incremental step
   B. update deviatoric and mean mechanical strains
   C. step iteration counter `ICOUNT`
   D. begin primary iteration loop
      1. estimate inelastic strain increment (engineering shear strains are computed)
      2. compute new stress state and inelastic work rate
      3. compute new estimate for beta
      4. compute new estimate for isotropic hardening
      5. check for convergence of effective stress, incremental inelastic strains, isotropic and directional hardening
         a. if converged, then continue with conclude current sub-increment via. STEP III.E
         b. if not, make new estimate for effective stress and increase iteration count `ICOUNT` by 1.
         c. check for excessive iteration count
            (1) if excessive, then increase cutting factor `ISUB` by factor of 2.0 and precede to Step II.A
            (2) if not, to Step III.D
   E. Update converged solution with estimates, then continue at STEP III.A

IV. Complete all sub-incremental cycles and then return to `UMAT`

Figure 6.1 Numerical algorithm for solving of Bodner-Partom equations.

SECTION 7

REFERENCES

[1] Bodner, S.R., and Partom, Y., "Constitutive Equations for Elastic-Viscoplastic Strain Hardening Materials," ASME Journal of Applied Mechanics, Vol. 42,1975, p.385.

[2] ABAQUS User Manuals, Verion 5.4, Hibbitt, Karlsson and Sorensen, Inc., 1994, 1080 Main Street, Pawtucket, PI, 02860-4847.

[3] Kroupa, J.K. and Neu, R.W., (1993), "Implementation of a Nonisothermal Unified Inelastic-Strain Theory into ADINA6.0 for a Titanium Alloy - Users Guild," WL-TR934005, Wright Laboratories, Wright-Patterson Air Force Base, Ohio.

[4] New, R.W., "Nonisothermal Material Parameters for the Bodner-Partom Model,", *Material Parameter Estimation for Modern Constitutive Equations*, L.A. Bertram, S.B. Brown, and A.D. Freed, Eds., MD-Vol. 43/AMD-Vol. 168, ASME, 1993, pp. 211-226.

[5] Chan, K.S., Bodner, S.R., and Lindholm, U.S., "Phenomenological Modeling of Hardening and Thermal Recovery in Metals," ASME Journal of Engineering Materials and Technology, Vol. 110, January 1988, pp.1-8.

[6] Chan, K.S., Lindholm, U.S., Bodner, S.R., and Nagy, A., "High Temperature Inelastic Deformation of the B1900+Hf Alloy Under Multiaxial Loading: Theory and Experiment," ASME Journal of Engineering Materials and Technology, Vol. 112, January 1990, pp. 7-14.

[7] Chan, K.S., Lindholm, U.S., Bodner, S.R., "Constitutive Modeling for Isotropic Materials (HOST)", NASA CR-182132 (SwRI-7576), NASA Lewis June 1988.

[8] Chan, K.S. and Lindholm, U.S. "Inelastic Deformation Under Nonisothermal Loading," ASME Journal of Engineering Materials and Technology, Vol. 112, January 1990, pp.15-25.

[9] Sherwood, J.A. and Boyle, M.J., "Investigation of the Thermomechanical Response of a Titanium Aluminide/Silicon Carbide Composite Using a Unified State Variable Model in ADINA," Computers and Structures, Vol. 40, 1991, pp.257-269.

[10] Ramaswamy, V.G., Stouffer, D.C., and Laflen, J.H., "A Unified Constitutive Model for the Inelastic Uniaxial Response of Rene'80 at Temperatures Between 538C and 982C," Transactions of the ASME, Vol. 112, July 1990, pp. 280-286.

[11] Stouffer, D.C., Ramaswamy, V.G., Laflen, J.H., Van Stone, R.H., and Williams, R., "A Constitutive Model for the Inelastic Multiaxial Response of Rene'80 at 871C and 982C," ASME Journal of Engineering Materials and Technology, Vol. 112, April 1990, pp.241-246.

[12] McDowell, D.L., "A Bounding Surface Theory for Cyclic Thermoplasticity," ASME Journal of Engineering Materials and Technology, Vol. 114, July 1992, pp. 297-303.

[13] Walker, K. P., "Research and Development Program for Nonlinear Structural Modeling with Advanced Time-Temperature Dependent Constitutive Relationships," NASA Report CR-165533, 1981.

[14] Freed, A. D., "Thermoviscoplastic Model With Application to Copper," NASA Report TP-2845, 1988.

[15] Chaboche, J. L., "Constitutive Equations for Cyclic Plasticity and Cyclic Viscoplasticity," International Journal of Plasticity, Vol. 5, 1989, pp. 247-302.

[16] Wang, J.-D. and Ohno, N., "Two Equivalent Forms of Nonlinear Kinematic Hardening: Application to Nonisothermal Plasticity," International Journal of Plasticity, Vol. 7, 1991, pp. 637-650.

[17] Ashbaugh, N.E., and Khobaib, M., Unpublished Data, University of Dayton Research Institute, Dayton, Ohio. December 1993.

[18] Sander, B.P., Unpublished Data, Material Directorate, Wright Laboratories Wright-Patterson Air-Force Base, Ohio, January 1995.

[19] Wolfram, S., *Mathematica*-A System for Doing Mathematics by Computer, Sec. Ed., 1991, Addison-Westley Publishing Company, Inc., Redwood City, CA.

[20] J.L. Kroupa, R.W. Neu, T. Nicholas, D. Coker, D.D. Robertson, and S. Mall, "A Comparison of Analysis Tools for Predicting the Inelastic Cyclic Response of Cross-Ply Titanium Matrix Composites," Life Prediction Methodology for Titanium Matrix Composites ASTM STP 1253, Ed. W. S. Johnson, J. M. Larsen, and B. N. Cox, American Society of Testing and Materials, 1995.

APPENDIX A

Listing Of User-Defined Subroutine

```
*USER SUBROUTINE
      SUBROUTINE UMAT(STRESS,STATEV,DDSDDE,SSE,SPD,SCD,
     1 RPL,DDSDDT,DRPLDE,DRPLDT,STRAN,DSTRAN,
     2 TIME,DTIME,TEMP,DTEMP,PREDEF,DPRED,MATERL,NDI,NSHR,NTENS,
     3 NSTATV,PROPS,NPROPS,COORDS,DROT,PNEWDT,CELENT,
     4 DFGRD0,DFGRD1,NOEL,NPT,KSLAY,KSPT,KSTEP,KINC)
C
      INCLUDE 'ABA_PARAM.INC'
      CHARACTER*8 CMNAME
C
      DIMENSION STRESS(NTENS),STATEV(NSTATV),
     1 DDSDDE(NTENS,NTENS),DDSDDT(NTENS),DRPLDE(NTENS),
     2 STRAN(NTENS),DSTRAN(NTENS),TIME(2),PREDEF(1),DPRED(1),
     3 PROPS(NPROPS),COORDS(3),DROT(3,3),
     4 DFGRD0(3,3),DFGRD1(3,3)
C
      PARAMETER (ONE=1.0D0,TWO=2.0D0,THREE=3.0D0,SIX=6.0D0)
C
C    THIS SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION IS RELEASED AS
C    IS.  THE U.S. GOVERNMENT, ITS CONTRACTORS AND THEIR SUBCONTRACTORS
C    MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, CONCERNING THIS
C    SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION, INCLUDING, WITHOUT
C    LIMITATION, ANY WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
C    PARTICULAR PURPOSE.  IN NO EVENT WILL THE U.S. GOVERNMENT, ITS
C    CONTRACTORS AND THEIR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES,
C    INCLUDING LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR
C    CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE
C    THIS SOFTWARE OR ANY ACCOMPANYING DOCUMENTATION, EVEN IF INFORMED
C    IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES.
C
C    "This software is being used at the user's own risk: Neither the
C    Government Agency nor its contractors assure software's accuracy
C    or its appropriate use."
C
c    Directional Bodner-Partom Material Constants:
c
c     Temperature Independent Constants
c
c     d0, Z1
c
c     Temperature Dependent Constants
c
c     an,  am1,  am2,  z0,  z2,  z3,  a1,  a2,  r1,  r2
c
c     Temperature Differentials
c
c     dn,  dm1,  dm2, dz0, dz2, dz3, da1, da2, dr1, dr2
c
      COMMON /matconst/  E,  G, a3k, anu, dg, d3k, dnu, d0,
     &         an, am1, am2, z0, z1, z2, z3, a1, a2, r1, r2,
     &         dn, dm1, dm2,dz0,dz2,dz3,da1,da2,dr1,dr2
C
c     common block for state varibles
c
      common /bpstat/ ein(6),zi,sigeff,betaeff,epeff,
     &               zd, beta(6), xout1, xout2, xout3
```

```
c
      common /control/ xnewdt, relax, toler, inter, Iplane
c
c ***********************************************************
c
c  Set Iplane = 0 for Three-D
c                 for Plane Strain
c                 for Axisymmetric
c
c  Set Iplane = 1 for Plane Stress
c
c ***********************************************************
c
c
      if(inter .eq. 0) then
      call setprops(PROPS,NPROPS)
      endif
c
c   recall state varibles
c
       call stget(STATEV,NSTATV)
c
       call getprops(TEMP,DTEMP)
c
       time2 = time(1)
c
       call dbodner(stran,dstran,stress,ddsdde,dtime,time2,
     $               pnewdt, ndi,nshr,ntens,npt, kstep)
c
c
c   If Bodner Partom solution did not converge
c      then send a cut the major time step defined by
c      abaqus
c
       call stput(STATEV,NSTATV)
c
c   Elastic-Plastic flow correction for DDSDDE
c
c    (Currently Disabled - see Joe, for Details)
c
c       sigave = 0.0
c       do 77 i = 1, NDI
c       sigave = sigave + stress(i)
c 77    continue
c       sigave = sigave / THREE
cc
c       do 78 i = 1, NTENS
c       if(sigeff .lt. 1.0e-10) then
c          flow(I) = 0.0
c       else
c          if(i .gt. ndi) then
c          flow(i)   = stress(i)/sigeff
c          else
c          flow(i)   = (stress(i)-sigave)/sigeff
c          endif
c       endif
c 78    continue
cc
c          YIELD   = SIGEFF
cc
c          EFFG3   = THREE*G
```

```
c           EFFHRD  = EFFG3
cc
c           DO 250 K1=1,NTENS
c             DO 240 K2=1,NTENS
c                 DDSDDE(K2,K1)=DDSDDE(K2,K1)+FLOW(K2)*FLOW(K1)
c     1                                          *(EFFHRD-EFFG3)
c 240         CONTINUE
c 250       CONTINUE
cC
      RETURN
      END
C
      SUBROUTINE SDVINI(STATEV,COORDS,NSTATV,NCRDS,NOEL,NPT,LAYER,KSPT)
C
      INCLUDE 'ABA_PARAM.INC'
C
       DIMENSION STATEV(NSTATV), COORDS(NCRDS)
c
      COMMON /matconst/  E,  G, a3k, anu, dg, d3k, dnu, d0,
     &         an, am1, am2, z0, z1, z2, z3, a1, a2, r1, r2,
     &         dn, dm1, dm2,dz0,dz2,dz3,da1,da2,dr1,dr2
c
       common /control/ effhrd, xnewdt, relax, toler, inter, Iplane
c
       inter = 0
c
       statev(1)  = 0.0
       statev(2)  = 0.0
       statev(3)  = 0.0
       statev(4)  = 0.0
       statev(5)  = 0.0
       statev(6)  = 0.0
       statev(7)  = 0.0
       statev(8)  = 0.0
       statev(9)  = 0.0
       statev(10) = 0.0
       statev(11) = 0.0
       statev(12) = 0.0
c
c  Initializing ZI to Z0
c
c   for Neu's Constants, Table 2-B [4]
c
c25
c25        statev(13) = 1550.
c
c650
c650       statev(13) = 1000.
c
c900
           statev(13) =  300.
c
c  for Sander's Constants, Table 2-C [18]
c
c25
c25        statev(13) = 2950.
c650
c650       statev(13) =  200.
c815
c815       statev(13) =  150.
c
```

```
        statev(14) = 0.0
        statev(15) = 0.0
        statev(16) = 0.0
        statev(17) = 0.0
c
        return
        end
C
      subroutine dbodner(eps,deps,stress,ddsdde,dtime,time2,
     $                   pnewdt, ndi,nshr,ntens,npt, kstep)
c
c ***************************************************
c
c      Copyright 1995 The University of Dayton
c                All rights reserved.
c
c  This material may be reproduced by or for the
c  U.S. Govenment pursuant to the copyright license
c  under the clause at DFARS 252.227-7013 (Oct 1988)
c
c  Programmed by:
c
c   Joseph L. Kroupa  (513) 255-9871
c   University of Dayton Research Institute
c   Dayton, Ohio 45469-0128
c
c ***************************************************
c
C   THIS SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION IS RELEASED AS
C   IS.  THE U.S. GOVERNMENT, ITS CONTRACTORS AND THEIR SUBCONTRACTORS
C   MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, CONCERNING THIS
C   SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION, INCLUDING, WITHOUT
C   LIMITATION, ANY WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
C   PARTICULAR PURPOSE.  IN NO EVENT WILL THE U.S. GOVERNMENT, ITS
C   CONTRACTORS AND THEIR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES,
C   INCLUDING LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR
C   CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE
C   THIS SOFTWARE OR ANY ACCOMPANYING DOCUMENTATION, EVEN IF INFORMED
C   IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES.
C
C   "This software is being used at the user's own risk: Neither the
C   Government Agency nor its contractors assure software's accuracy
C   or its appropriate use."
C
      INCLUDE 'ABA_PARAM.INC'
C-----------------------------------------------------------------------
c
      real*8      Deps(ntens),    STRESS(ntens),  DEPSI(ntens)
      real*8      eps(ntens),     ddeps(ntens),   DEVSIG(ntens)
      real*8      STRES0(ntens),  einest(ntens),  ein0(ntens)
      real*8      betaest(ntens), beta0(ntens)
      real*8      eps0(ntens),    ddsdde(ntens,ntens)
c
C-----------------------------------------------------------------------
C
      COMMON /matconst/  E,  G, a3k, anu, dg, d3k, dnu, d0,
     &        an, am1, am2, z0, z1, z2, z3, a1, a2, r1, r2,
     &        dn, dm1, dm2,dz0,dz2,dz3,da1,da2,dr1,dr2
c
      common /bpstat/ ein(6), zi, sigeff, betaeff, epeff,
     &               zd, beta(6), xout1, xout2, xout3
```

```
C
      common /control/ effhar, xnewdt, relax, toler, inter, Iplane
c
      xsqrt3 = sqrt(3.)
c
C     store old state varibles
c
      DO 10 I=1,ntens
      stres0(I) = stress(I)
      ein0(I)   = ein(I)
      beta0(i)  = beta(i)
      eps0(i)   = eps(i)
10    CONTINUE
c
         do  11 k1=1,NTENS
         do  11 k2=1,NTENS
          DDSDDE(k2,k1) = 0.0
 11     continue
c
      zi0      = zi
      zd0      = zd
c
c   store old material constants
c
      g0   = g
      a3k0 = a3k
      anu0 = anu
      z00  = z0
      z20  = z2
      z30  = z3
      a10  = a1
      a20  = a2
      am10 = am1
      am20 = am2
      r10  = r1
      r20  = r2
c
c    initialize other varibles
c
      isub    = inter
      isubmax = inter * 64
      idbug   = 0
      isecond = 0
c
c    initialize variables for sub-time cutting
c
100   continue
c
      sigest  = sigeff
      zi      = zi0
      ziest   = zi0
      zold    = zi0 + zd0
      zd      = zd0
c
c   restore old material constants
c
       g   =   g0
      a3k  =  a3k0
      anu  =  anu0
      z0   =  z00
      z2   =  z20
```

```
      z3    =  z30
      a1    =  a10
      a2    =  a20
      am1   = am10
      am2   = am20
      r1    =  r10
      r2    =  r20
      time = time2
c
c    update new rate material constants with subincrement isub
c
      tfactor = 1./ dfloat(isub)
c
      dtsub   = dtime * tfactor
      dgsub   = dg    * tfactor
      d3ksub  = d3k   * tfactor
      dnusub  = dnu   * tfactor
       dnsub  =  dn   * tfactor
      dz0sub  = dz0   * tfactor
      dz2sub  = dz2   * tfactor
      dz3sub  = dz3   * tfactor
      da1sub  = da1   * tfactor
      da2sub  = da2   * tfactor
      dm1sub  = dm1   * tfactor
      dm2sub  = dm2   * tfactor
      dr1sub  = dr1   * tfactor
      dr2sub  = dr2   * tfactor
c
c  compute devitoric stress
c
      sighyd = 0.
      do 19  i = 1, ndi
      sighyd = sighyd + stres0(i)
 19   continue
      sighyd = sighyd / 3.0
c
      do 20, i = 1,ntens
c
      ein(I)     = ein0(i)
      beta(i)    = beta0(i)
      betaest(i) = beta0(i)
      eps(i)     = eps0(i)
      ddeps(i)   = deps(i)  * tfactor
c
      IF (I.gt.ndi) THEN
      fact1 = 0.
      ELSE
      fact1 = 1.
      ENDIF
c
       devsig(I) =  stres0(I) - fact1*sighyd
c
 20   CONTINUE
c
c
      DO 200 Jsub=1,isub
c
c     update all temperature dependent material constants to
c     end of subtime increment step
c
        g  = g    +   dgsub
```

```
      a3k = a3k  +  d3ksub
      anu = anu  +  dnusub
      an  = an   +   dnsub
      z0  = z0   +  dz0sub
      z2  = z2   +  dz2sub
      z3  = z3   +  dz3sub
      a1  = a1   +  da1sub
      a2  = a2   +  da2sub
     am1  = am1  +  dm1sub
     am2  = am2  +  dm2sub
      r1  = r1   +  dr1sub
      r2  = r2   +  dr2sub
     time = time + dtsub
c
          EFFG2  =  2.0*G
          EFFG3  =  3.0*G
c
          if(Iplane .eq. 1) then
          EFFLAM =  EFFG2 * (anu/(1.-anu))
          else
          EFFLAM =  (A3K-EFFG2)/3.0
          endif
c
          DO 220 K1=1,NDI
             DO 210 K2=1,NDI
                DDSDDE(K2,K1)=EFFLAM
 210         CONTINUE
             DDSDDE(K1,K1)=EFFG2+EFFLAM
 220      CONTINUE
          DO 230 K1=NDI+1,NTENS
             DDSDDE(K1,K1)=G
 230      CONTINUE
c
     eff_el = 0
      do 30 i = 1,ntens
      if(i .gt. ndi) then
      eff_el = eff_el + ddeps(i)*ddeps(i)*2
      else
      eff_el = eff_el + ddeps(i)*ddeps(i)
      endif
      eps(i)    = eps(i) + ddeps(i)
      einest(I) = ein(I)
 30   continue
c
      eff_el = sqrt(0.5*eff_el)
c
      icount  = 0
      deiold  = 0.
c
       ztot = ziest + zd
c
 300   continue
c
c  For Elastic Test case
c  Set plasticity off
c
coff         d0 = 0.0
c
      IF(sigest .lt. 1.0E-30) THEN
      sigest = 1.0e-30
      deieff = 0.0
```

```
      xlam    = 0.0
      ELSE
      xtmp1 = (ztot/sigest)**2
      xtmp2 = -0.5*xtmp1**an
      deieff = d0*exp(xtmp2)
      xlam    = xsqrt3 * deieff/sigest
      ENDIF
c
       deieff = deieff * dtsub
c
        effhrd = 0
        if (eff_el .gt. 0.0)  effhrd = deieff/eff_el
        xout1  = effhrd
c
         DO 40 I=1,ntens
            IF (I.gt.ndi) THEN
               fact1 = 2
               fact2 = 0
            ELSE
               fact1 = 1
               fact2 = 1
            ENDIF
C
c estimate plastic strains and stresses
c (Engineering Plastic Shear Strains are Computed)
c
       DEPSI(I) = xlam *  devsig(I) * fact1
c
c    Defined thermal differentail terms
c
       theta3 = 0.0
c
       einest(i) = ein(i) + (depsi(i)*dtsub) +theta3
c
  40    CONTINUE
c
c
      ssum    = 0.
      pwdot   = 0.
      sigavg = 0.
c
         DO 41 I=1,ntens
            IF (I.gt.ndi) THEN
               fact1 = 2
               fact2 = 0
            ELSE
               fact1 = 1
               fact2 = 1
            ENDIF
C
      stress(I)  = 0.0
      do 39 j = 1, NTENS
      stress(I) = stress(I) + ddsdde(i,j)*(eps(j)-einest(j))
 39   continue
c
      pwdot = pwdot    + (stress(I)*depsi(I))
      ssum    = ssum   + fact1 * stress(I)**2
      sigavg  = sigavg + fact2 * stress(I)
c
c Extra output, nice for Debugging
c
```

```
        if(i .eq. 2) then
        xout1 = depsi(I)
        xout2 = stress(I)
        endif
c
 41     continue
c
       ssum   = sqrt(ssum)
       sigavg = sigavg/3.0
c
       bnew  = 0.0
       signew = 0.0
c
          DO 42 I=1,ntens
             IF (I.gt.ndi) THEN
                fact1 = 2
                fact2 = 0
             ELSE
                fact1 = 1
                fact2 = 1
             ENDIF
c
        bnew      = bnew   + fact1 * beta(i)**2
        devsig(i) = stress(i) -  fact2 * sigavg
        signew    = signew + fact1 * devsig(I)**2
c
 42     continue
c
        bnew   = sqrt(bnew)
c
c   For plane stress solution
c   Add extra deviatoric stress for third plane
c
       if(Iplane .eq. 1) signew = signew + sigavg**2
c
       signew = sqrt(1.5*signew)
c
       zd     = 0.
       do 44 i = 1,ntens
       IF (I.gt.ndi) THEN
         fact=2
       ELSE
         fact=1
       ENDIF
c
c  compute drag stress vectors
c
       v = beta(i)
       if(bnew .ge. 1.E-30) v =  beta(i)/ bnew
       u = stress(i)
       if(ssum .ge. 1.E-30) u =  stress(i) / ssum
c
       bterm1 = ((z3 * u) - betaest(i)) * pwdot
       bterm2 =  z1 * v *((bnew/z1)**r2)
       theta2     =  betaest(i) * dz3sub / z3
       betadot    = am2*bterm1 - a2*bterm2
       betaest(i) = beta(i) + betadot*dtsub + theta2
c       write(6,*) 'flag 6'
c
       zd = zd + betaest(i) * u * fact
c       write(6,*) 'flag 7'
```

```
 44    continue
c
 46     zterm1 =  z1*(abs(ziest-z2)/z1)**r1
c
        theta1 = ((zi-z2)/(z1-z2)) * dz2sub
c
        ZiDOT = am1*(z1-ziest)*pwdot
     &        -  a1*zterm1
c
C      COMPUTE UPDATED Ztot VALUE
c
       ziest = zi +  zidot*dtsub + theta1
       ztot  = zd + ziest
c
       if(ztot .lt. toler) ztot= toler
c
C      NOW INVESTIGATE CONVERGENCE
c
       serror = dABS(sigest-signew)
       sigmax = dmax1(signew,sigest)
       IF(sigmax .gt. toler) serror = serror / sigmax
c
       deperr = dabs(deieff-deiold)
       depmax = dmax1(deieff,deiold)
       IF( depmax .gt. toler) deperr = deperr / depmax
c
       zerror = dABS(ztot-zold)
       zmax = dmax1(ztot,zold)
       IF( zmax .gt. toler)  zerror = zerror / zmax
c
        error = serror + deperr + zerror
c
c -------------------------------------------------------------
c
        if(idbug .eq. 1) then
        write(6,*) 'icount isub errors s ep ztot ',icount, isub
        write(6,*) sigest, zd, serror,zerror,deperr
        endif
c
c -------------------------------------------------------------
c
c  check for convergence
c
        IF( error .gt. toler) THEN
c
c  nonconvergent solution arrives here
c
          sigest = sigest * (1.-relax) + signew * relax
c         sigest = signew
          deiold = deieff
          zold   =  ztot
c
c   update and check convergence count
c
           icount = icount+1
c
        IF (icount .ge. 10 .or. error .ge. 2.9)  THEN
c
c  restart iteration scheme with new time-step cut
c   make additional time-step cuts
c
```

```
 999     isub = isub * 2
c
c  terminate nonconverge solutions
c
      if (isub .gt. isubmax) then
       write(6,*)
       write(6,*) ' Bodner Solution Refused to Converge '
       write(6,*)
       write(6,*) ' icount jsub isub ' , icount, jsub, isub
       write(6,*) ' error ', error
       write(6,*) ' depsi ' , depsi
       write(6,*) ' ein ', ein
       write(6,*) ' einest ', einest
       write(6,*) ' dtsub ',  dtsub
       write(6,*) 'devsig ', devsig
       write(6,*) 'stress ', stress
       write(6,*) ' zd ' , zd
       write(6,*) ' ztot ' , ztot
       stop
      endif
c
c  restart iteration scheme with new time-step cut
c
            goto 100
c
          endif
c
        goto 300
c
          else
c
c  convergence solution arrives here
c
         zi     = ziest
c
       do 60  i = 1, ntens
         ein(I)    = einest(I)
         beta(I)   = betaest(I)
 60    continue
c
          endif
c
c   end of subtime loops
c
 200       continue
c
        epeff = 0.0
        do 70  i = 1, ntens
             IF (I.gt.ndi) THEN
             fact2 = 2.
             ELSE
             fact2 = 1.
             ENDIF
          epeff = epeff + ein(i) * ein(i) * fact2
 70     continue
          epeff = sqrt(2.0*epeff/3.0)
c
        SIGEFF  = sigest
        betaeff = bnew
c
c  adjust increment size parameter
```

```
c
          pnewdt = 1.0
c
          if (isub .eq. inter .or. effhrd .lt. 0.80) then
          pnewdt = xnewdt
          endif
c
        RETURN
        END
c
       subroutine stget(Array,Nstate)
c
c ***************************************************
c
c      Copyright 1995 The University of Dayton
c                All rights reserved.
c
c  This material may be reproduced by or for the
c  U.S. Govenment pursuant to the copyright license
c  under the clause at DFARS 252.227-7013 (Oct 1988)
c
c  Programmed by:
c
c   Joseph L. Kroupa  (513) 255-9871
c   University of Dayton Research Institute
c   Dayton, Ohio 45469-0128
c
c ***************************************************
c
C   THIS SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION IS RELEASED AS
C   IS.  THE U.S. GOVERNMENT, ITS CONTRACTORS AND THEIR SUBCONTRACTORS
C   MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, CONCERNING THIS
C   SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION, INCLUDING, WITHOUT
C   LIMITATION, ANY WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
C   PARTICULAR PURPOSE.  IN NO EVENT WILL THE U.S. GOVERNMENT, ITS
C   CONTRACTORS AND THEIR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES,
C   INCLUDING LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR
C   CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE
C   THIS SOFTWARE OR ANY ACCOMPANYING DOCUMENTATION, EVEN IF INFORMED
C   IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES.
C
C   "This software is being used at the user's own risk: Neither the
C   Government Agency nor its contractors assure software's accuracy
C   or its appropriate use."
C

      INCLUDE 'ABA_PARAM.INC'
c
      real*8 array(Nstate)
c
      common /bpstat/ ein(6), zi, sigeff, betaeff, epeff,
     &               zd, beta(6), xout1, xout2, xout3
c
       ein(1)  = array(1)
       ein(2)  = array(2)
       ein(3)  = array(3)
       ein(4)  = array(4)
       ein(5)  = array(5)
       ein(6)  = array(6)
       beta(1) = array(7)
       beta(2) = array(8)
       beta(3) = array(9)
```

```
        beta(4) = array(10)
        beta(5) = array(11)
        beta(6) = array(12)
        zi      = array(13)
        sigeff  = array(14)
        epeff   = array(15)
        betaeff = array(16)
        zd      = array(17)
c
         return
         end
c
      subroutine stput(Array,Nstate)
c
c
c ***************************************************
c
c      Copyright 1995 The University of Dayton
c                All rights reserved.
c
c  This material may be reproduced by or for the
c  U.S. Govenment pursuant to the copyright license
c  under the clause at DFARS 252.227-7013 (Oct 1988)
c
c  Programmed by:
c
c   Joseph L. Kroupa  (513) 255-9871
c   University of Dayton Research Institute
c   Dayton, Ohio 45469-0128
c
c ***************************************************
c
C   THIS SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION IS RELEASED AS
C   IS.  THE U.S. GOVERNMENT, ITS CONTRACTORS AND THEIR SUBCONTRACTORS
C   MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, CONCERNING THIS
C   SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION, INCLUDING, WITHOUT
C   LIMITATION, ANY WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
C   PARTICULAR PURPOSE.  IN NO EVENT WILL THE U.S. GOVERNMENT, ITS
C   CONTRACTORS AND THEIR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES,
C   INCLUDING LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR
C   CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE
C   THIS SOFTWARE OR ANY ACCOMPANYING DOCUMENTATION, EVEN IF INFORMED
C   IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES.
C
C   "This software is being used at the user's own risk: Neither the
C   Government Agency nor its contractors assure software's accuracy
C   or its appropriate use."
C
      INCLUDE 'ABA_PARAM.INC'
c
      real*8 array(Nstate)
c
      common /bpstat/ ein(6), zi, sigeff,  betaeff, epeff,
     &               zd, beta(6), xout1, xout2, xout3
c
       array(1)  = ein(1)
       array(2)  = ein(2)
       array(3)  = ein(3)
       array(4)  = ein(4)
       array(5)  = ein(5)
       array(6)  = ein(6)
```

```
        array(7)  = beta(1)
        array(8)  = beta(2)
        array(9)  = beta(3)
        array(10) = beta(4)
        array(11) = beta(5)
        array(12) = beta(6)
        array(13) = zi
        array(14) = sigeff
        array(15) = epeff
        array(16) = betaeff
        array(17) = zd
        array(18) = xout1
        array(19) = xout2
        array(20) = xout3
c
        return
        end
c
      subroutine setprops(PROPS,NPROPS)
c
c
c ***************************************************
c
c      Copyright 1995 The University of Dayton
c                All rights reserved.
c
c  This material may be reproduced by or for the
c  U.S. Govenment pursuant to the copyright license
c  under the clause at DFARS 252.227-7013 (Oct 1988)
c
c  Programmed by:
c
c   Joseph L. Kroupa  (513) 255-9871
c   University of Dayton Research Institute
c   Dayton, Ohio 45469-0128
c
c ***************************************************
c
      INCLUDE 'ABA_PARAM.INC'
c
      PARAMETER  (MAXP=20)
      common /matprop/ T(MAXP),  TE(MAXP), TNU(MAXP),  TN(MAXP),
     &               TZ2(MAXP), TZ3(MAXP), TM1(MAXP), TM2(MAXP),
     &               TR1(MAXP), TR2(MAXP), TINTER (MAXP),
     &               C51, C61, C52, C62, NTSET
c
      COMMON /matconst/  E,  G, a3k, anu, dg, d3k, dnu, d0,
     &        an, am1, am2, z0, z1, z2, z3, a1, a2, r1, r2,
     &        dn, dm1, dm2,dz0,dz2,dz3,da1,da2,dr1,dr2
c
      common /control/ effhrd, xnewdt, relax, toler, inter, Iplane
c
       real*8  Props(NPROPS)
c
c
      NOUT=6
c
c     Extract Control Variables
c
       Iplane  = IFIX(PROPS(1))
       NTIP    = IFIX(PROPS(2))
```

```
        NTSET   = IFIX(PROPS(3))
c
      write(NOUT, '(//,34h Input Control Variables         ,//)')
      write(NOUT,'(15h IPLANE          , 2x, I5 )') iplane
      write(NOUT,'(15h NTIP            , 2x, I5 )') NTIP
      write(NOUT,'(15h NTSET           , 2x, I5 )') NTSET
c
        d0      = Props(4)
        z1      = Props(5)
        c51     = Props(6)
        c61     = Props(7)
        c52     = Props(8)
        c62     = Props(9)
        xnewdt  = Props(10)
        inter   = Props(11)
        relax   = Props(12)
        toler   = Props(13)
c
c
c  Write out material constants
c
      write(NOUT, '(//,34h Temperature Independent Constants ,//)')
      write(NOUT,'(15h DNOT            , 2x, F10.3 )') d0
      write(NOUT,'(15h Z1              , 2x, F10.3 )') z1
      write(NOUT,'(15h C51 and C61     , 2(2x, 1PE10.1)') C51, C61
      write(NOUT,'(15h C52 and C62     , 2(2x, 1PE10.1)') C52, C62
      write(NOUT,'(16h Error Tolerance, 1x, 1PE10.1)') Toler
      write(NOUT,'(16h Relax. Factor   , 1x, F10.3)') relax
      write(NOUT,'(16h XNEWDT          , 1x, F10.3)') XNEWDT
c
        ipt    = ntip + 4
c
        do 5  i = 1, NTSET
        T(i)   =  PROPS(ipt)
        ipt    = ipt + 1
 5      continue
c
        do 6  i = 1, NTSET
        TE(i)  =  PROPS(ipt)
        ipt    = ipt + 1
 6      continue
c
        do 7  i = 1, NTSET
        TNU(i)  =  PROPS(ipt)
        ipt    = ipt + 1
 7      continue
c
        do 8  i = 1, NTSET
        TN(i)  =  PROPS(ipt)
        ipt    = ipt + 1
 8      continue
c
        do 11  i = 1, NTSET
        TZ2(i)  =  PROPS(ipt)
        ipt    = ipt + 1
11      continue
c
        do 12  i = 1, NTSET
        TZ3(i)  =  PROPS(ipt)
        ipt    = ipt + 1
12      continue
```

```
c
        do 13  i = 1, NTSET
        TM1(i)  =  PROPS(ipt)
        ipt    = ipt + 1
13      continue
c
        do 16  i = 1, NTSET
        TM2(i)  =  PROPS(ipt)
        ipt    = ipt + 1
16      continue
c
        do 18  i = 1, NTSET
        TR1(i)  =  PROPS(ipt)
        ipt    = ipt + 1
18      continue
c
        do 19  i = 1, NTSET
        TR2(i)  =  PROPS(ipt)
        ipt    = ipt + 1
19      continue
c
        do 20  i = 1, NTSET
        if (inter .gt. 0) then
        TINTER(i) = inter
        else
        TINTER(i) = PROPS (ipt)
        ipt    = ipt + 1
20      continue
c
c Write Matrix Elastic Properties
c
      write(NOUT,100)
 100   FORMAT(///,36h     Matrix Elastic Properties       ,//,
     >         18h   Temperature    ,18h Elastic Modulus ,
     >         18h Poissons Ratio  ,18h Subcut Factor    ,/)
c
      do 101 i = 1, NTSET
      write(NOUT,102) T(i),TE(i),TNU(I), TINTER(I)
 101  continue
 102   FORMAT( 3(F10.2, 8x),13x, I5)
c
      write(NOUT,103)
 103   FORMAT(///,36h      Bodner-Partom Properties  I      ,//,
     > 18h   Temperature     ,12h      N       ,12h     z2     ,
     >       12h     z3      ,12h     r1       ,12h     r2     ,/ )
c
      do 104 i = 1, NTSET
      write(NOUT,105)  T(i),TN(I),TZ2(I),
     >                 TZ3(I),Tr1(I),Tr2(I)
 104  continue
 105   FORMAT(( F10.2, 4x),6(F12.3))
c
      write(NOUT,106)
 106   FORMAT(///,36h      Bodner-Partom Properties II    ,//,
     > 18h   Temperature     ,12h     M1       ,12h    M2      ,/)
c
      do 107 i = 1, NTSET
      write(NOUT,108) T(i),TM1(I),TM2(I)
 108   FORMAT(( F10.2, 4x),6(F12.3))
 107  continue
c
```

```
          return
          end
c
       function tprop(T, PROPS, Tget)
c
      INCLUDE 'ABA_PARAM.INC'
c
       parameter (MAXP=20)
       real*8   T(MAXP), PROPS(MAXP)
c
        do  10  i = 1, (MAXP-1)
       if(tget .eq. t(i) then
       tprop = propS(i)
       return
       elseif (tget .eq. t(i+1)) then
       tprop = propS(i+1)
       return
       elseif (tget .gt. t(i)  .and. tget .lt. t(i+1)) then
         tprop = propS(i)
     &         + (PROPS(I+1)-PROPS(I))
     &         * (Tget - T(I))
     &         / (T(I+1)-T(I))
       return
          endif
10      continue
c
        write(6,*)
        write(6,*) ' Temperature Outside of Material Data Range '
        write(6,*)
        stop
        end
c
      subroutine getprops(T1,DTEMP)
c
c ***************************************************
c
c      Copyright 1995 The University of Dayton
c                All rights reserved.
c
c  This material may be reproduced by or for the
c  U.S. Govenment pursuant to the copyright license
c  under the clause at DFARS 252.227-7013 (Oct 1988)
c
c  Programmed by:
c
c   Joseph L. Kroupa  (513) 255-9871
c   University of Dayton Research Institute
c   Dayton, Ohio 45469-0128
c
c ***************************************************
c
C   THIS SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION IS RELEASED AS
C   IS.  THE U.S. GOVERNMENT, ITS CONTRACTORS AND THEIR SUBCONTRACTORS
C   MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, CONCERNING THIS
C   SOFTWARE AND ANY ACCOMPANYING DOCUMENTATION, INCLUDING, WITHOUT
C   LIMITATION, ANY WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
C   PARTICULAR PURPOSE.  IN NO EVENT WILL THE U.S. GOVERNMENT, ITS
C   CONTRACTORS AND THEIR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES,
C   INCLUDING LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR
C   CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE, OR INABILITY TO USE
C   THIS SOFTWARE OR ANY ACCOMPANYING DOCUMENTATION, EVEN IF INFORMED
```

```
C    IN ADVANCE OF THE POSSIBILITY OF SUCH DAMAGES.
C
C    "This software is being used at the user's own risk: Neither the
C    Government Agency nor its contractors assure software's accuracy
C    or its appropriate use."
C

       INCLUDE 'ABA_PARAM.INC'
c
       PARAMETER  (MAXP=20)
       common /matprop/ T(MAXP),  TE(MAXP), TNU(MAXP),  TN(MAXP),
      &                 TZ2(MAXP), TZ3(MAXP), TM1(MAXP), TM2(MAXP),
      &                 TR1(MAXP), TR2(MAXP), TINTER(MAXP),
      &                 C51, C61, C52, C62, NTSET
c
       COMMON /matconst/  E,  G, a3k, anu, dg, d3k, dnu, d0,
      &         an, am1, am2, z0, z1, z2, z3, a1, a2, r1, r2,
      &         dn, dm1, dm2,dz0,dz2,dz3,da1,da2,dr1,dr2
c
        common /control/ effhrd, xnewdt, relax, toler, inter, Iplane
c
       T2    = T1 + DTEMP
c
c  Extract Elastic Properties
c
       E    = tprop(T,TE,T1)
       E2   = tprop(T,TE,T2)
       anu  = tprop(T,TNU,T1)
       anu2 = tprop(T,TNU,T2)
       G    =    E/(2.*(1.+ anu))
c
c    note: a3k is defined as 3X of Bulk Modulus
c
       a3k   =   E/(1.-2.* anu)
       dg    =   E2/(2.*(1.+ anu2)) - g
       d3k   =   E2/(1.-2.* anu2)  - a3k
       dnu   =   anu - anu2
C
c  Special Functional curves for a1 and a2  defined by C5 and C6
c
        a1  = c51 * exp(-c61/(T1+273.))
        da1 = c51 * exp(-c61/(t2+273.))-a1
c
        a2  = c52 * exp(-c62/(T1+273.))
        da2 = c52 * exp(-c62/(t2+273.))-a2
c
c  Get other temperature dependent variables
c
        an  =  Tprop(T,TN,T1)
        dn  =  Tprop(T,TN,T2) - an
c
        z2  =  Tprop(T,TZ2,T1)
       dz2  =  Tprop(T,TZ2,T2) - z2
c
        z3  =  Tprop(T,TZ3,T1)
       dz3  =  Tprop(T,TZ3,T2) - z3
c
       am1  =  Tprop(T,TM1,T1)
       dm1  =  Tprop(T,TM1,T2) - am1
c
       am2  =  Tprop(T,TM2,T1)
```

```
      dm2   =  Tprop(T,TM2,T2) - am2
c
       r1   =  Tprop(T,TR1,T1)
      dr1   =  Tprop(T,TR1,T2) - r1
c
       r2   =  Tprop(T,TR2,T1)
      dr2   =  Tprop(T,TR2,T2) - r2
c
       inter = 1+IFIX(Tprop(T,TINTER,T1))
c
        return
        end
c end of user subroutines
```

APPENDIX B

Listing Of Sample Input File

```
*HEADING
 Test input file for umat_db - Bodner Partom, uniaxial tension
*WAVEFRONT MINIMIZATION,SUPPRESS
*NODE,NSET=ALLN
    1       0.        0.        0.
    2       1.        0.        0.
    3       1.        1.        0.
    4       0.        1.        0.
    5      0.5        0.        0.
    6       1.       0.5        0.
    7      0.5        1.        0.
    8       0.       0.5        0.
*ELEMENT,TYPE=CAX8,ELSET=ALLE
    1    1    2    3    4    5    6    7    8
*SOLID SECTION,ELSET=ALLE,MATERIAL=B21S_3
**
** Timetal21S UDRI Tref=900C
**  Special Elastic Case
**
*MATERIAL, NAME=B21S_0
*EXPANSION, ZERO=900.
 9.770E-6      23.0
10.719E-6     260.0
11.500E-6     482.0
12.030E-6     650.0
12.348E-6     760.0
12.501E-6     815.0
12.271E-6     900.0
*ELASTIC
 114000.0      0.34     23.0
 114000.0      0.34    260.0
  90000.0      0.34    482.0
  78000.0      0.34    650.0
  70000.0      0.34    760.0
  64000.0      0.34    815.0
  54700.0      0.34    900.0
**
** NEU'S (1993) Timetal21S CONSTANTS
**
*MATERIAL, NAME=B21S_3
***EXPANSION, ZERO=900.0
** 9.7787E-6      23.0
** 1.0713E-5     260.0
** 1.0915E-5     315.0
** 1.1093E-5     365.0
** 1.1267E-5     415.0
** 1.1436E-5     465.0
** 1.1492E-5     482.0
** 1.1550E-5     500.0
** 1.1631E-5     525.0
** 1.1710E-5     550.0
** 1.1788E-5     575.0
** 1.1865E-5     600.0
** 1.2014E-5     650.0
** 1.2323E-5     760.0
```

```
** 1.2467E-5      815.0
** 1.2689E-5      900.0
*USER MATERIAL, CONSTANTS=176
         1          9         16       1.0E4     1600.0     5.8E5     1.37E4     5.8E5
     1.37E4        1.0         10       0.75      1.E-4       0.0        0.0       0.0
      23.0       260.0      315.0      365.0      415.0      465.0      482.0     500.0
      525.0      550.0      575.0      600.0      650.0      760.0      815.0     900.0
   112000.0   108030.0   106130.0   104090.0   101740.0    99085.0    98113.0   97045.0
    95497.0    93873.0    92172.0    90395.0    86612.0    77216.0    71964.0   63122.0
      0.340      0.340      0.340      0.340      0.340      0.340      0.340     0.340
      0.340      0.340      0.340      0.340      0.340      0.340      0.340     0.340
       4.80       3.50      3.054      2.649      2.243      1.838      1.700     1.500
      1.280      1.100      0.970      0.820      0.740      0.580      0.550     0.550
     1550.0     1300.0     1250.4     1205.4     1160.4     1115.3     1100.0    1089.3
     1074.4     1059.5     1044.6     1029.8     1000.0      600.0      300.0     300.0
      100.0      300.0      390.0      500.0      660.0      960.0     1100.0    1300.0
     1670.0     2100.0     2600.0     3700.0     3800.0     4000.0     4100.0    4300.0
        0.0        0.0        0.0        0.0        0.0        0.0        0.0       0.0
        0.0        0.0        0.0        0.0        0.0        0.0        0.0       0.0
      0.350      0.350      1.502      2.549      3.597      4.644      5.000     5.763
      6.822      7.881      8.941      10.00      10.00      15.00      30.00     30.00
        3.0        3.0        3.0        3.0        3.0        3.0        3.0       3.0
        3.0        3.0        3.0        3.0        3.0        3.0        3.0       3.0
        3.0        3.0        3.0        3.0        3.0        3.0        3.0       3.0
        3.0        3.0        3.0        3.0        3.0        3.0        3.0       3.0
*DEPVAR
20
***
*MATERIAL, NAME=B21S_4
**
** SANDER'S (1995) Timetal21S  CONSTANTS
**
*EXPANSION, ZERO=900
 9.770E-6      23.0
10.719E-6      260.0
11.500E-6      482.0
12.030E-6      650.0
12.348E-6      760.0
12.501E-6      815.0
12.271E-6      900.0
*USER MATERIAL, CONSTANTS=76
         1         13          6       1.0E4     3200.0   5.5875E6     1.37E4   5.5875E6
     1.37E4        2.0         10        0.75      1.E-4       0.00        0.0        0.0
      23.0       260.0      566.0      650.0      760.0      815.0   112000.0   108030.0
    92700.0    86612.0    77216.0    71964.0      0.340      0.340      0.340      0.340
      0.340      0.340       1.58       1.35       1.09       0.94      0.575       0.55
     2950.0     2650.0      800.0      200.0      190.0      150.0      180.0      180.0
      180.0      180.0      180.0      180.0       0.50       0.50      2.500      7.000
     37.000     50.000        4.0        4.0        4.0        4.0        4.0        4.0
      3.500      3.500      3.500      3.500      3.500      3.500      3.500      3.500
      3.500      3.500      3.500      3.500
**
*BOUNDARY
    1    1
    1    2
    2    2
    5    2
    4    1
    8    1
***
*** Temperature curve for case 1
*AMPLITUDE, NAME=TCUR1
      0.00       25.0   48.0000       25.0
***
*** Temperature curve for case 2 & 3
```

```
*AMPLITUDE, NAME=TCUR2
      0.00      650.0  4800.000      650.0
***
*** Temperature curve for case 4
*AMPLITUDE, NAME=TCUR3
      0.00      25.00   24.0000      25.00      36.00      482.0      60.000      482.0
    72.000      25.00   96.0000      25.00
***
*** Temperature curve for case 5
*AMPLITUDE, NAME=TCUR4
      0.00      650.0   12.0000      650.0      18.00      760.0      30.000      760.0
    36.000      650.0   48.0000      650.0
***
*** Strain curve for case 1, 2, 4, & 5
*AMPLITUDE, NAME=MCUR1
      0.00       0.00    96.000      0.080
***
*** Strain curve for case 3
*AMPLITUDE, NAME=MCUR2
      0.00       0.00  4800.000      0.040
***
*STEP, MONOTONIC=NO, INC=48
 This is a Unidirectional Test of the Directional Bodner Partom Routines
*STATIC, DIRECT
       1.0       48.0
***CONTROLS, PARAMETER=FIELD, FIELD=GLOBAL
****     0.0005
*BOUNDARY, AMPLITUDE=MCUR1
    3    2          1.00
    4    2          1.00
    7    2          1.00
*TEMPERATURE, AMPLITUDE=TCUR2
 ALLN      1.000
*EL PRINT
S
SINV
E
*NODE PRINT, FREQUENCY=10
U,RF
*FILE FORMAT, ASCII
*EL FILE, POSITION=AVERAGED AT NODES, FREQUENCY=1
S,E,SINV,TEMP, SDV
*NODE FILE, FREQUENCY=1
COORD, NT
*END STEP
```

APPENDIX C

Verification Solution of Case 1
(file vcase1.math)

Strain	Stress (MPa)	Strain	Stress (MPa)
0.0000000	0.0000	0.020832	1145.6
0.0008333	93.330	0.021666	1145.9
0.0016666	186.66	0.022499	1146.2
0.0024999	279.99	0.023332	1146.3
0.0033332	373.32	0.024166	1146.4
0.0041665	466.65	0.024999	1146.5
0.0049998	559.98	0.025832	1146.6
0.0058331	653.31	0.026666	1146.6
0.0066664	746.64	0.027499	1146.7
0.0074997	839.97	0.028332	1146.7
0.0083330	933.30	0.029166	1146.7
0.0091663	1026.6	0.029999	1146.7
0.0099996	1083.9	0.030832	1146.7
0.010833	1098.6	0.031665	1146.7
0.011666	1110.4	0.032499	1146.7
0.012500	1119.6	0.033332	1146.7
0.013333	1126.7	0.034165	1146.7
0.014166	1132.0	0.034999	1146.7
0.014999	1136.0	0.035832	1146.7
0.015833	1138.9	0.036665	1146.7
0.016666	1141.1	0.037499	1146.7
0.017499	1142.7	0.038332	1146.7
0.018333	1143.8	0.039165	1146.7
0.019166	1144.6	0.039998	1146.7
0.019999	1145.2		

APPENDIX D

Verification Solution Of Case 2

Strain	Stress (MPa)
0.0000000	0.0000
0.0008333	72.163
0.0016666	133.04
0.0024999	192.07
0.0033332	248.97
0.0041665	298.93
0.0049998	337.18
0.0058331	361.49
0.0066664	374.15
0.0074997	379.78
0.0083330	382.07
0.0091663	382.96
0.0099996	383.30
0.010833	383.43
0.011666	383.48
0.012500	383.50
0.013333	383.51
0.014166	383.51
0.014999	383.51
0.015833	383.51
0.016666	383.51
0.017499	383.51
0.018333	383.51
0.019166	383.51
0.019999	383.51
0.020832	383.51
0.021666	383.51
0.022499	383.51
0.023332	383.51
0.024166	383.51
0.024999	383.51
0.025832	383.51
0.026666	383.51
0.027499	383.51
0.028332	383.51
0.029166	383.51
0.029999	383.51
0.030832	383.51
0.031665	383.51
0.032499	383.51
0.033332	383.51
0.034165	383.51
0.034999	383.51
0.035832	383.51
0.036665	383.51
0.037499	383.51
0.038332	383.51
0.039165	383.51
0.039998	383.51

APPENDIX E

Verification Solution Of Case 3

Strain	Stress (MPa)
0.0000000	0.0000
0.0008333	72.106
0.0016666	121.44
0.0024999	138.00
0.0033332	140.14
0.0041665	140.36
0.0049998	140.38
0.0058331	140.38
0.0066664	140.38
0.0074997	140.38
0.0083330	140.38
0.0091663	140.38
0.0099996	140.38
0.010833	140.38
0.011666	140.38
0.012500	140.38
0.013333	140.38
0.014166	140.38
0.014999	140.38
0.015833	140.38
0.016666	140.38
0.017499	140.38
0.018333	140.38
0.019166	140.38
0.019999	140.38
0.020832	140.38
0.021666	140.38
0.022499	140.38
0.023332	140.38
0.024166	140.38
0.024999	140.38
0.025832	140.38
0.026666	140.38
0.027499	140.38
0.028332	140.38
0.029166	140.38
0.029999	140.38
0.030832	140.38
0.031665	140.38
0.032499	140.38
0.034165	140.38
0.034999	140.38
0.035832	140.38
0.036665	140.38
0.037499	140.38
0.038332	140.38
0.039165	140.38
0.039998	140.38

APPENDIX F

Verification Solution Of Case 4

Strain	Stress (MPa)	Temp (°C)
0.00000000	0.0000	25.000
0.00083333	93.333	25.000
0.0016667	186.67	25.000
0.0025000	280.00	25.000
0.0033333	373.33	25.000
0.0041667	466.67	25.000
0.0050000	560.00	25.000
0.0058333	653.33	25.000
0.0066667	746.67	25.000
0.0075000	840.00	25.000
0.0083333	933.33	25.000
0.0091667	1026.7	25.000
0.0100000	1083.9	25.000
0.010833	1098.6	25.000
0.011667	1110.4	25.000
0.012500	1119.6	25.000
0.013333	1126.7	25.000
0.014167	1132.0	25.000
0.015000	1136.0	25.000
0.015833	1138.9	25.000
0.016667	1141.1	25.000
0.017500	1142.7	25.000
0.018333	1143.8	25.000
0.019167	1144.6	25.000
0.020000	1145.2	25.000
0.020833	1122.6	63.100
0.021667	1097.2	101.20
0.022500	1070.3	139.30
0.023333	1041.9	177.30
0.024167	1011.7	215.40
0.025000	979.53	253.50
0.025833	949.64	291.60
0.026667	917.28	329.70
0.027500	882.53	367.80
0.028333	850.18	405.80
0.029167	825.88	443.90
0.030000	790.59	482.00
0.030833	787.05	482.00
0.031667	787.03	482.00
0.032500	787.03	482.00
0.033333	787.03	482.00
0.034167	787.03	482.00
0.035000	787.03	482.00
0.035833	787.03	482.00
0.036667	787.03	482.00
0.037500	787.03	482.00
0.038333	787.03	482.00
0.039167	787.03	482.00
0.040000	787.03	482.00
0.040833	787.03	482.00
0.041667	787.03	482.00
0.042500	787.03	482.00
0.043333	787.03	482.00
0.044167	787.03	482.00
0.045000	787.03	482.00
0.045833	787.03	482.00
0.046667	787.03	482.00
0.047500	787.03	482.00
0.048333	787.03	482.00
0.049167	787.03	482.00
0.050000	787.03	482.00
0.050833	821.57	443.90
0.051667	843.77	405.80
0.052500	879.05	367.80
0.053333	913.54	329.70
0.054167	945.70	291.60
0.055000	975.69	253.50
0.055833	1008.1	215.40
0.056667	1038.7	177.30
0.057500	1067.4	139.30
0.058333	1094.6	101.20
0.059167	1120.3	63.100
0.060000	1144.7	25.000
0.060833	1146.7	25.000

Strain	Stress (MPa)	Temp (°C)
0.061667	1146.7	25.000
0.062500	1146.7	25.000
0.063333	1146.7	25.000
0.064167	1146.7	25.000
0.065000	1146.7	25.000
0.065833	1146.7	25.000
0.066667	1146.7	25.000
0.067500	1146.7	25.000
0.068333	1146.7	25.000
0.069167	1146.7	25.000
0.070000	1146.7	25.000
0.070833	1146.7	25.000
0.071667	1146.7	25.000
0.072500	1146.7	25.000
0.073333	1146.7	25.000
0.074167	1146.7	25.000
0.075000	1146.7	25.000
0.075833	1146.7	25.000
0.076667	1146.7	25.000
0.077500	1146.7	25.000
0.078333	1146.7	25.000
0.079167	1146.7	25.000
0.080000	1146.7	25.000

APPENDIX G

Verification Solution Of Case 5

Strain	Stress (MPa)	Temp (°C)	Strain	Stress (MPa)	Temp (°C)
0.0000	0.0000	650.00	0.015417	136.98	760.00
0.00041667	36.095	650.00	0.015833	132.73	760.00
0.00083333	72.190	650.00	0.016250	130.64	760.00
0.0012500	104.07	650.00	0.016667	129.59	760.00
0.0016667	133.08	650.00	0.017083	129.05	760.00
0.0020833	162.60	650.00	0.017500	128.76	760.00
0.0025000	192.14	650.00	0.017917	128.62	760.00
0.0029167	221.16	650.00	0.018333	128.54	760.00
0.0033333	249.09	650.00	0.018750	128.50	760.00
0.0037500	275.30	650.00	0.019167	128.48	760.00
0.0041667	299.15	650.00	0.019583	128.47	760.00
0.0045833	320.03	650.00	0.020000	128.46	760.00
0.0050000	337.52	650.00	0.020417	128.46	760.00
0.0054167	351.44	650.00	0.020833	128.46	760.00
0.0058333	361.94	650.00	0.021250	128.46	760.00
0.0062500	369.48	650.00	0.021667	128.46	760.00
0.0066667	374.65	650.00	0.022083	128.46	760.00
0.0070833	378.08	650.00	0.022500	128.46	760.00
0.0075000	380.30	650.00	0.022917	128.46	760.00
0.0079167	381.71	650.00	0.023333	128.46	760.00
0.0083333	382.60	650.00	0.023750	128.46	760.00
0.0087500	383.15	650.00	0.024167	128.46	760.00
0.0091667	383.49	650.00	0.024583	128.46	760.00
0.0095833	383.70	650.00	0.025000	128.46	760.00
0.0100000	383.83	650.00	0.025417	134.59	750.80
0.010417	374.19	659.20	0.025833	144.80	741.70
0.010833	357.33	668.30	0.026250	157.42	732.50
0.011250	336.61	677.50	0.026667	171.74	723.30
0.011667	314.21	686.70	0.027083	187.43	714.20
0.012083	291.34	695.80	0.027500	204.29	705.00
0.012500	268.60	705.00	0.027917	222.20	695.80
0.012917	246.28	714.20	0.028333	241.07	686.70
0.013333	224.60	723.30	0.028750	260.85	677.50
0.013750	203.67	732.50	0.029167	281.50	668.30
0.014167	183.61	741.70	0.029583	302.96	659.20
0.014583	164.50	750.80	0.030000	325.21	650.00
0.015000	146.43	760.00	0.030417	341.88	650.00

Strain	Stress (MPa)	Temp (°C)
0.030833	354.79	650.00
0.031250	364.38	650.00
0.031667	371.18	650.00
0.032083	375.79	650.00
0.032500	378.83	650.00
0.032917	380.78	650.00
0.033333	382.01	650.00
0.033750	382.78	650.00
0.034167	383.26	650.00
0.034583	383.56	650.00
0.035000	383.74	650.00
0.035417	383.86	650.00
0.035833	383.93	650.00
0.036250	383.97	650.00
0.036667	383.99	650.00
0.037083	384.01	650.00
0.037500	384.02	650.00
0.037917	384.03	650.00
0.038333	384.03	650.00
0.038750	384.03	650.00
0.039167	384.04	650.00
0.039583	384.04	650.00
0.040000	384.04	650.00