NASA/TM—1999-209070

Micromechanics Analysis Code With Generalized Method of Cells (MAC/GMC)

User Guide: Version 3.0

S.M. Arnold
Glenn Research Center, Cleveland, Ohio

B.A. Bednarcyk
Ohio Aerospace Institute, Cleveland, Ohio

T.E. Wilt and D. Trowbridge
The University of Akron, Akron, Ohio

National Aeronautics and
Space Administration

Glenn Research Center

March 1999

Acknowledgments

The authors would like to thank Mr. Robert Goldberg and Mrs. Cheryl Bowman for their diligent efforts and assistance in checking the accuracy and consistency of this manual by running numerous test cases. Also, we would like to thank Prof. J. Aboudi for many fruitful discussions and technical assistance.

Trade names or manufacturers' names are used in this report for identification only. This usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076
Price Code: A11

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100
Price Code: A11

1.0 Introduction

The ability to accurately predict the thermomechanical deformation response of advanced heterogeneous materials (e.g., multi-phase composites, cellular solids, etc.) plays a critical role in the development and practical implementation of these strategic materials. Analytical models that predict the effective behavior of composites are used not only by engineers performing structural analysis of large-scale composite components, but also by materials scientists in developing new material systems. For an analytical model to fulfill these two distinct functions it must be based on a micromechanics approach which utilizes physically based deformation and life constitutive models and generates the average (macro) response of a composite material given the properties of the individual constituents and their geometric arrangement. Only then can such a model be used by a materials scientist to investigate the effect(s) of different deformation mechanisms on the overall response of the composite in order to identify the appropriate constituents for a given application. Alternatively, if a micromechanical model is to be used in a large-scale structural analysis it must be 1) computationally efficient, 2) able to generate accurate displacement and stress fields at **both** the macro and the micro levels and 3) be compatible with the finite element method. Also, as advancements in processing and fabrication techniques make it possible to more accurately engineer (tailor) the architectures of these advanced composite systems, development of a computationally efficient micromechanics analysis tool capable of accurately predicting the effect of microstructural details on the internal and macroscopic behavior of composites becomes even more important. Such computational efficiency is absolutely required because 1) the large number of parameters which must be varied in the course of engineering (or designing) composite (heterogeneous) materials, and 2) the optimization of a material's microstructure requires the integration of the micromechanics model with optimization algorithms. From this perspective, analytical approaches which use closed-form expressions to describe the effect of a material's internal architecture on the overall material behavior are preferable to numerical methods, such as finite element or finite difference schemes.

A number of models presently exist that can fulfill certain aspects of the aforementioned tasks. However, there are very few working models that are both computationally efficient and sufficiently accurate both at the micro- and macro-levels. One such micromechanics model with the potential of fulfilling both tasks is Aboudi's method of cells [1] and its continuous reinforcement [2] and discontinuous reinforcement [3] generalizations. The comprehensive capabilities and efficiency of this method have been documented in references [4] and [5]. Consequently, the recently developed, computationally efficient and comprehensive micromechanics analysis code, **MAC/GMC**, will now be described: with a predictive capability resting entirely upon the fully analytical micromechanics model, herein referred to as the generalized method of cells, **GMC**, [2 and 3] . **MAC/GMC** is a versatile form of research software that "drives" the double or triple periodic micromechanics constitutive models based upon **GMC**. **GMC** is capable of predicting the response of both continuous and discontinuous multi-phase composites with arbitrary internal microstructures and reinforcement shapes. **GMC** is a continuum-based micromechanics model that provides closed-form expressions for the macroscopic composite response in terms of the properties, size, shape, distribution, and response of the individual constituents or phases that make up the material. **GMC** also utilizes physically based viscoplastic deformation and life models for each constituent. Furthermore, expressions relating the internal stress and strain fields in the individual constituents in terms of the macroscopically applied stresses and strains are available through strain or stress concentration factors. These expressions make possible the investigation of failure processes at the microscopic level at each step of an applied load history. Similarly, **GMC** provides the capability of studying the influence of bond strength at the fiber/matrix interface which has been shown to be an important damage mechanism in composites [6].

MAC/GMC enhances the basic capabilities of **GMC** by providing a modular framework wherein 1) various thermal, mechanical (stress or strain control) and thermomechanical load histories can be imposed, 2) different integration algorithms may be selected, 3) a variety of material constitutive models (both deformation and life) may be utilized and/or implemented and 4) a variety of fiber architectures (both unidirectional, laminate and woven) may be easily accessed through their corresponding representative volume elements contained within the supplied library of RVEs or input directly by the user, and 5) graphical post processing of the macro and/or micro field quantities is made available. Consequently, the availability of **MAC/GMC** now provides industry, academia and government engineers and materials scientists with a comprehensive, computationally efficient, user-friendly micromechanics analysis tool that can easily and accurately design/analyze multi-phase (composite) materials for a given application. **MAC/GMC** is also ideally suited for conducting sensitivity/parametric studies (i.e., "what-if" scenarios) in the design/analysis of advanced composite materials (e.g., MMC/IMCs, PMCs, and CMCs). Figure 1 illustrates the basic flow diagram for this modular framework code. Furthermore, since **GMC** is a continuum-based micromechanics model that provides closed form expressions for the macroscopic composite response in terms of the properties, size, shape, distribution, and response of the individual constituents or phases that make up the material, **MAC/GMC** can be interfaced directly with standard linear and nonlinear finite element analysis packages (through their respective user definable constitutive routines) for cost-effective large scale component design and analysis. Currently, such an interface exists only for HKS's nonlinear finite element code, **ABAQUS.** For clarity purposes, when this finite element implementation of **MAC/GMC** is discussed within this manual it will be referred to as **FEAMAC**, for example see section **4.1.2**.

The capabilities of this version of **MAC/GMC** are listed in section 2, whereas theoretical and background information on the basic capabilities previously itemized are given in section 3. Section 4 describes in detail the required input for using **MAC/GMC** and will be the most referenced section in the entire manual. Section 5 gives some insights into the future modifications planned for **MAC/GMC** and in section 6 examples A-S illustrate and describe the various features of **MAC/GMC**.

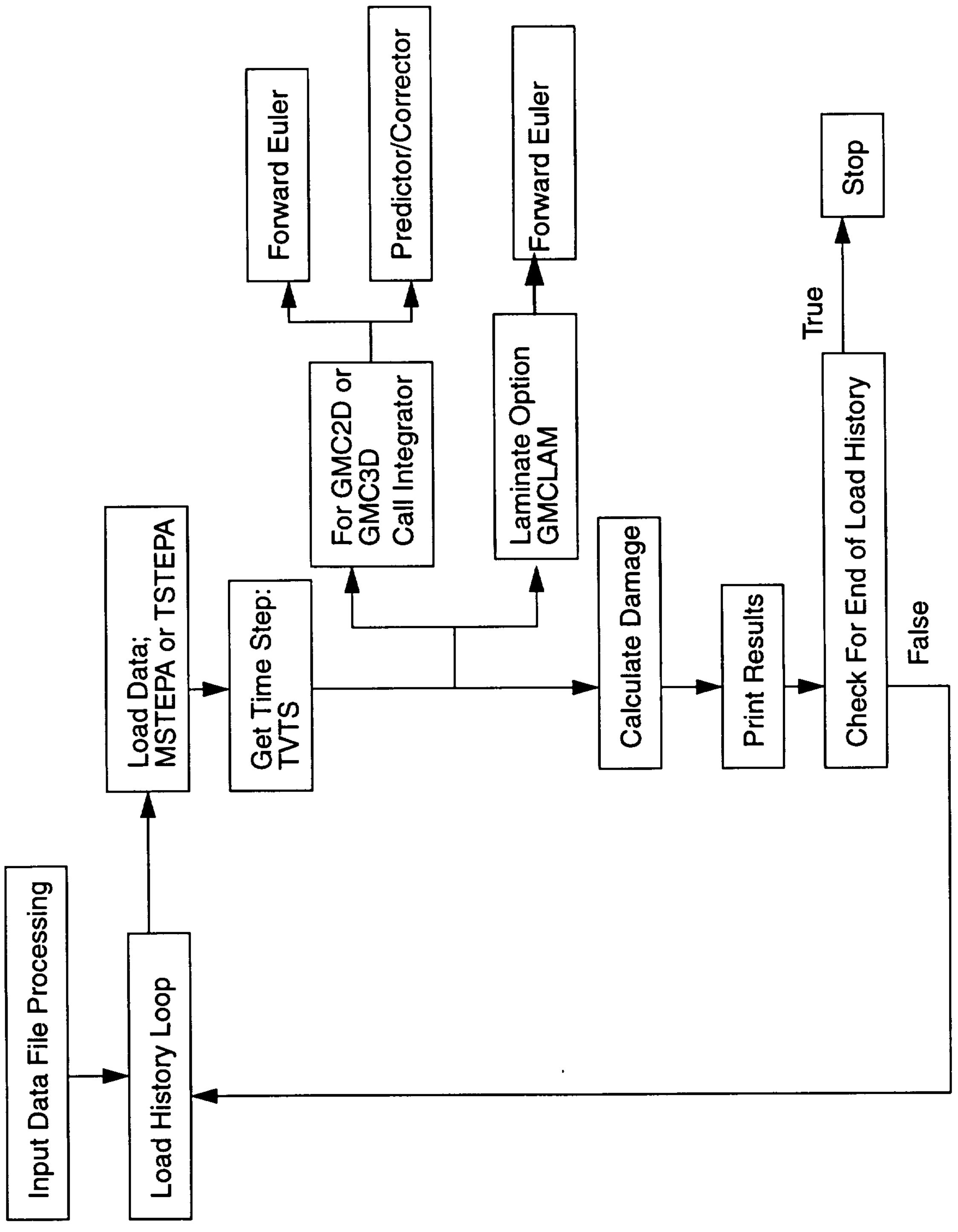


Figure1: ***MAC*** Flowchart

2.0 Current Capabilities

In this section the current features/capabilities of **MAC/GMC** are itemized, with those new to this version being denoted by **bold type face.**

- Load Types:
  - Thermal, Mechanical, Thermomechanical; **Generalized Loading Option**
- Aboudi GMC Models (**Reformulated for computational speed**):
  - Double Periodicity Model for continuous reinforcement
  - Triple Periodicity Model for discontinuous/**woven** reinforcement
- Library of Representative Volume Elements:
  - Double Periodicity Model: **circular fiber representations**
  - Triple Periodicity Model: **ellipsoidal inclusion, open cell**
- Input / Graphical Output
  - Free Format Input Data
  - Up to 5 x-y data plot files generated for macro and micro (subcell) quantities
  - PATRAN for subcell geometry and color results evaluation, e.g. stress, strains, inelastic strains, J_2, etc., via **MACPOST**
- Integration Options:
  - Forward Euler
  - Predictor/Corrector
- Constitutive Models (**expanded internal material database**)
  - elastic: transversely isotropic and **anisotropic**
  - inelastic viscoplastic models:
    - Bodner-Partom
    - Robinson
    - GVIPS isotropic form
    - **TGVIPS transversely isotropic form**
    - user defined constitutive model via subroutines **USRMAT** (inelastic), **USRFORMDE** and **USRCPEVAL** (elastic); user defined functional dependence of material constants via subroutine **USRFUN**
- Interface Modeling
  - fiber/matrix interface layer (distinct interfacial material)
  - **fiber/matrix debonding conditions** (interfacial displacement discontinuity)
- **Fatigue Damage Analysis**
- **Yield Surface Calculations**
- Symmetric and **Nonsymmetric** Laminate Analysis
- **FEAMAC - allows easy finite element access to MAC/GMC capabilities**

3.0 Background

As stated in the introduction, **MAC/GMC**'s predictive capabilities rest upon the fully analytical micromechanics model known as **GMC** which is capable of predicting the inelastic response of both continuous (double periodicity) and discontinuous/woven (triple periodicity) multi-phase composites with arbitrary internal microstructures and reinforcement shapes. Included in this section is a brief overview of the theoretical foundations of **GMC**; a more complete discussion is given in references [1-3]. Also, included in this section are discussions of 1) the available integration algorithms, 2) various constitutive models, 3) theory underlying the laminate, 4) fatigue damage, and 5) interfacial modeling approaches incorporated into **MAC/GMC**. Besides describing the available constitutive models, explanations of the user defined inelastic and elastic material models are included. Finally, the available architectures (Repeating Volume Elements, or RVEs) and required input are then described in section 4.0.

3.1 Micromechanics Models

In the original formulation of the method of cells, a continuously (or discontinuously) reinforced, unidirectional fibrous composite was modeled as a rectangular, double-periodic (or triple-periodic) array of fibers embedded in a matrix phase. The periodic character of the assemblage allowed identification of a repeating unit cell that can be used as a building block to construct the entire composite. The properties of the repeating cell were thus representative of the properties of the entire assemblage. The unit cell consisted of a single fiber subcell surrounded by three matrix subcells for continuous, and seven for discontinuous, composites, hence the name **method of cells**. The rectangular geometry of the repeating unit cell allowed one to obtain an approximate solution for the stresses and strains in the individual subcells given some macroscopically homogeneous state of strain or stress applied to the composite. The approximate solution to the posed boundary value problem was, in turn, used to determine macroscopic (average) properties, traditionally referred to as effective properties, of the composite and the macro (global) stress-strain response in the inelastic region.

In the **generalized method of cells** for continuous (or discontinuous) fibrous composites, the repeating unit cell can consist of an arbitrary number of phases, see Fig. 2. Hence, the generalized method of cells is capable of modeling a multiphase composite. This generalization extends the modeling capability of the original method of cells to include the following: 1) inelastic thermomechanical response of multi-phase metal matrix composite, 2) modeling of various fiber architectures (including both shape and packing arrangements), 3) modeling of porosities and damage, and 4) modeling of interfacial regions around inclusions including interfacial degradation.

The basic homogenization approach taken in the micromechanical analysis consists essentially of four steps. First, the RVE, of the periodic composite is identified. Second, the macroscopic or average stress and strain states in terms of the individual microscopic (subcell) stress and strain states is defined. Third, the continuity of tractions and displacements are imposed at the boundaries between the constituents. These three steps, in conjunction with micro-equilibrium, establish the relationship between micro (subcell) total, thermal and inelastic strains and macro (composite) strains via the relevant concentration tensors. In the fourth and final step, the overall macro constitutive equations of the composite are determined. These four steps form the basis of the micro-to-macromechanics analysis which describes the behavior of heterogeneous media. The resulting micromechanical analysis establishes the overall (macro) behavior of the multi-phase composite which is expressed as a constitutive relation between the average stress, strain, thermal, and inelastic strains, in conjunction with the effective elastic stiffness matrix.

That is,

$$\bar{\underset{\sim}{\sigma}} = \underset{\sim}{B}^{*}(\bar{\underset{\sim}{\varepsilon}} - \bar{\underset{\sim}{\varepsilon}}^{I} - \bar{\underset{\sim}{\varepsilon}}^{T}) \qquad \text{(EQ 1)}$$

where, for the most general case of discontinuous reinforcement with N_α by N_β by N_γ number of subcells, the effective elastic stiffness matrix, $\underset{\sim}{B}^{*}$, of the composite is given by,

$$\underset{\sim}{B}^{*} = \frac{1}{dhl}\sum_{\alpha=1}^{N_\alpha}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} d_\alpha h_\beta l_\gamma \underset{\sim}{C}^{(\alpha\beta\gamma)}\underset{\sim}{A}^{(\alpha\beta\gamma)} \qquad \text{(EQ 2)}$$

the composite inelastic strain vector is defined as,

$$\bar{\underset{\sim}{\varepsilon}}^{I} = \frac{-\underset{\sim}{B}^{*-1}}{dhl}\sum_{\alpha=1}^{N_\alpha}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} d_\alpha h_\beta l_\gamma \underset{\sim}{C}^{(\alpha\beta\gamma)}(\underset{\sim}{D}^{(\alpha\beta\gamma)}\underset{\sim}{\varepsilon}_s^{I} - \underset{\sim}{\varepsilon}^{I(\alpha\beta\gamma)}) \qquad \text{(EQ 3)}$$

the average thermal strain vector as,

$$\bar{\underset{\sim}{\varepsilon}}^{T} = \frac{-\underset{\sim}{B}^{*-1}}{dhl}\sum_{\alpha=1}^{N_\alpha}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} d_\alpha h_\beta l_\gamma \underset{\sim}{C}^{(\alpha\beta\gamma)}(\underset{\sim}{D}^{(\alpha\beta\gamma)}\underset{\sim}{\varepsilon}_s^{T} - \underset{\sim}{\varepsilon}^{T(\alpha\beta\gamma)}) \qquad \text{(EQ 4)}$$

and $\bar{\underset{\sim}{\varepsilon}}$ is the uniform applied macro (composite) strain. For the case of continuous reinforcements with N_β by N_γ number of subcells, EQS. 2 - 4 reduce to the following:

$$\underset{\sim}{B}^{*} = \frac{1}{hl}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} h_\beta l_\gamma \underset{\sim}{C}^{(\beta\gamma)}\underset{\sim}{A}^{(\beta\gamma)} \qquad \text{(EQ 5)}$$

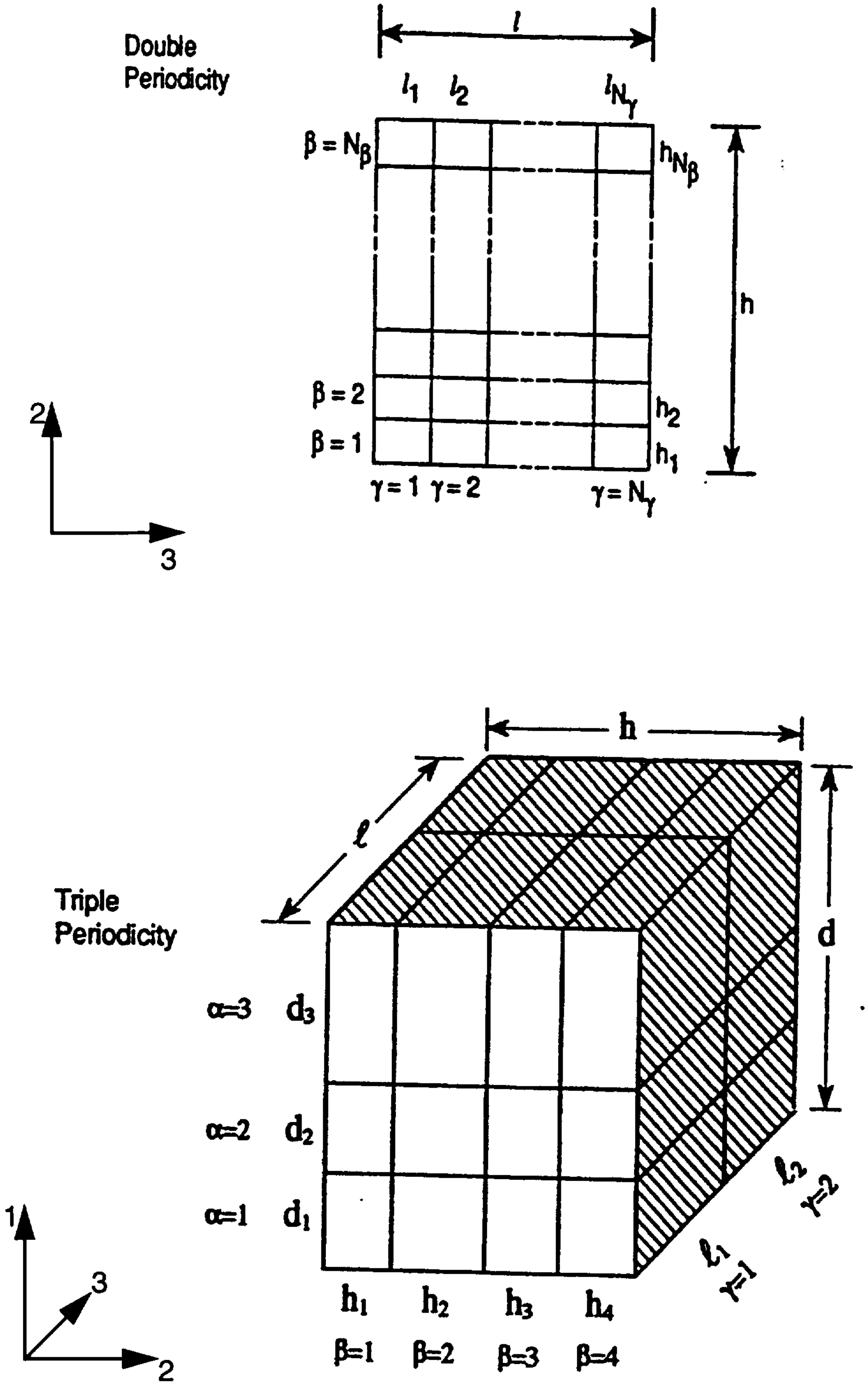


Figure 2: Subcell Dimension Nomenclature

$$\bar{\underset{\sim}{\varepsilon}}^{I} = \frac{-\underset{\sim}{B}^{*-1}}{hl}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} h_\beta l_\gamma \underset{\sim}{C}^{(\beta\gamma)}(\underset{\sim}{D}^{(\beta\gamma)}\underset{\sim}{\varepsilon}_s^{I} - \bar{\underset{\sim}{\varepsilon}}^{I(\beta\gamma)}) \tag{EQ 6}$$

$$\bar{\underset{\sim}{\varepsilon}}^{T} = \frac{-\underset{\sim}{B}^{*-1}}{hl}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} h_\beta l_\gamma \underset{\sim}{C}^{(\beta\gamma)}(\underset{\sim}{D}^{(\beta\gamma)}\underset{\sim}{\varepsilon}_s^{T} - \bar{\underset{\sim}{\varepsilon}}^{T(\beta\gamma)}) \tag{EQ 7}$$

In the above equations matrix notation is employed; where, for example, the average stress, $\bar{\underset{\sim}{\sigma}}$, average strain, $\bar{\underset{\sim}{\varepsilon}}$, and inelastic subcell strain, $\underset{\sim}{\varepsilon}_s^{I}$, vectors represent,

$$\bar{\underset{\sim}{\sigma}} = \{\bar{\sigma}_{11}, \bar{\sigma}_{22}, \bar{\sigma}_{33}, \bar{\sigma}_{23}, \bar{\sigma}_{13}, \bar{\sigma}_{12}\} \tag{EQ 8}$$

$$\bar{\underset{\sim}{\varepsilon}} = \{\bar{\varepsilon}_{11}, \bar{\varepsilon}_{22}, \bar{\varepsilon}_{33}, 2\bar{\varepsilon}_{23}, 2\bar{\varepsilon}_{13}, 2\bar{\varepsilon}_{12}\} \tag{EQ 9}$$

$$\underset{\sim}{\varepsilon}_s^{I} = \{\bar{\underset{\sim}{\varepsilon}}^{I(111)}, \ldots, \bar{\underset{\sim}{\varepsilon}}^{I(N_\alpha N_\beta N_\gamma)}\} \tag{EQ 10}$$

where the six components of the vector $\bar{\underset{\sim}{\varepsilon}}^{I(\alpha\beta\gamma)}$ are arranged as in EQ. 9. Similar definitions for $\underset{\sim}{\varepsilon}_s^{T}$, $\bar{\underset{\sim}{\varepsilon}}^{T(\alpha\beta\gamma)}$ also exist. Note that the key ingredient in the construction of this macro constitutive law is the derivation of the appropriate concentration matrices, $\underset{\sim}{A}^{(\alpha\beta\gamma)}$ and $\underset{\sim}{D}^{(\alpha\beta\gamma)}$ having the dimensions 6 by 6 and 6 by $N_\alpha N_\beta N_\gamma$ respectively, at the micro (subcell) level. The definitions of $\underset{\sim}{A}$ and $\underset{\sim}{D}$, although not given here, may be found in references [2] and [3]. Finally, the matrix $\underset{\sim}{C}^{(\alpha\beta\gamma)}$ represents the elastic stiffness tensor of each subcell $(\alpha\beta\gamma)$ and d_α, h_β, l_γ the respective subcell dimensions (see Fig. 2) wherein,

$$d = \sum_{\alpha=1}^{N_\alpha} d_\alpha \qquad h = \sum_{\beta=1}^{N_\beta} h_\beta \qquad l = \sum_{\gamma=1}^{N_\gamma} l_\gamma$$

Similarly, given the concentration matrices $\underset{\sim}{A}^{(\alpha\beta\gamma)}$ and $\underset{\sim}{D}^{(\alpha\beta\gamma)}$, expressions for the average strain in each subcell can be constructed, i. e.,

$$\bar{\underset{\sim}{\varepsilon}}^{(\alpha\beta\gamma)} = \underset{\sim}{A}^{(\alpha\beta\gamma)}\bar{\underset{\sim}{\varepsilon}} + \underset{\sim}{D}^{(\alpha\beta\gamma)}(\underset{\sim}{\varepsilon}_s^{I} + \underset{\sim}{\varepsilon}_s^{T}) \tag{EQ 11}$$

as well as average stress

$$\bar{\underset{\sim}{\sigma}}^{(\alpha\beta\gamma)} = \underset{\sim}{C}^{(\alpha\beta\gamma)}[\underset{\sim}{A}^{(\alpha\beta\gamma)}\bar{\underset{\sim}{\varepsilon}} + \underset{\sim}{D}^{(\alpha\beta\gamma)}(\underset{\sim}{\varepsilon}_s^{I} + \underset{\sim}{\varepsilon}_s^{T}) - (\underset{\sim}{\varepsilon}^{I(\alpha\beta\gamma)} + \underset{\sim}{\varepsilon}^{T(\alpha\beta\gamma)})] \tag{EQ 12}$$

The analytic constitutive law, see EQ. 1, may be readily applied to investigate the behavior of various types of composites, given knowledge of the behavior of the individual phases. Note that within **MAC/GMC** an incremental tangent solution scheme is utilized, therefore the appropriate rate forms of EQS. 1-12 are employed.

Numerous advantages can be stated regarding the current macro/micro constitutive laws as compared to the other numerical micromechanical approaches in the literature, e.g. the finite element unit cell approach. One advantage is that any type of simple or combined loading (multiaxial state of stress) can be applied irrespective of whether symmetry exists or not, as well as without resorting to different boundary condition application strategies as in the case of the finite element unit cell procedure. Another, advantage concerns the availability of an analytical expression representing the macro elastic-thermo-inelastic constitutive law, thus ensuring a reduction in memory requirements when implementing this formulation into a structural finite element analysis code. Furthermore, this formulation has been shown to predict accurate macro behavior given only a few subcells, within the repeating cell (see references [2], and [4]). Whereas, if one employs the finite element unit cell procedure, a significant number of finite elements are required within a given repeating unit cell to obtain the same level of global accuracy as with the present formulation. Consequently, it is possible to utilize this formulation to efficiently analyze metal matrix composite structures subjected to complex thermomechanical load histories. This is particularly important when analyzing realistic structural components, since different loading conditions exist throughout the structure, thus necessitating the application of the macromechanical equations repeatedly at these locations.

The equations of **GMC-3D** (and consequently through appropriate specialization **GMC-2D**) have recently been reformulated [7] in a way that significantly increases the computational efficiency of the model. This new reformulation has now been implemented into **MAC/GMC** and is the default computational mode. By nature of the traction continuity conditions within the original generalized method, all six stress components are not unique in every subcell. Normal stress components are constant in certain rows of subcells, while shear stress components are constant in certain layers of subcells. The unique subcell stress components are denoted as,

$$T_{11}^{(\beta\gamma)}, T_{22}^{(\alpha\gamma)}, T_{33}^{(\alpha\beta)}, T_{23}^{(\alpha)}, T_{13}^{(\beta)}, T_{12}^{(\gamma)}. \qquad \text{(EQ 13)}$$

Consequently, a more efficient formulation of **GMC** can be obtained by applying traction continuity directly (i.e. recognizing that traction continuity conditions require no more and no less than the aforementioned reduction in subcell stress components) and using subcell stresses rather than strains as the basic unknown quantities. Accordingly, the continuity of displacement conditions are formed in terms of subcell stresses (through the use of the subcell constitutive and kinematic equations), and the *mixed concentration equations* for the unit cell are constructed,

$$\underset{\sim}{T} = \underset{\sim}{G}^{(\alpha\beta\gamma)}\underset{\sim}{\varepsilon} + \underset{\sim}{G}^{IT(\alpha\beta\gamma)}(\underset{\sim}{\varepsilon}_s^I + \underset{\sim}{\varepsilon}_s^T) \qquad \text{(EQ 14)}$$

Here, $\underset{\sim}{T}$ is the vector of all subcell stress components (listed in EQ 13), $\underset{\sim}{G}^{(\alpha\beta\gamma)}$ is the subcell mixed concentration matrix, and $\underset{\sim}{G}^{IT(\alpha\beta\gamma)}$ is the subcell inelastic-thermal mixed concentration matrix. The term *mixed* is used here because EQ 14 relates local (subcell) stresses to global strains. Clearly this equation contrasts with its original formulation counterpart, EQ 11, which relates local strains to global strains and is thus the unit cell strain concentration equation. As one could obtain EQ 1 from EQ. 12 in the original formulation, similar expressions for the global quantities ($\underset{\sim}{B}^{*}$, $\bar{\varepsilon}^{I}$ and $\bar{\varepsilon}^{T}$) can be determined easily from EQ 14 (see [7] for details).

The increased efficiency of the reformulation of **GMC** emerges mainly due to the increased efficiency of forming EQ 14 versus forming EQ 11. The formation of EQ 14 requires solution (of linear equations) for the unknown independent subcell stress components listed in EQ 13, numbering $N_{\beta}N_{\gamma} + N_{\alpha}N_{\gamma} + N_{\alpha}N_{\beta} + N_{\alpha} + N_{\beta} + N_{\gamma}$. The formation of EQ 11 requires solution for 6 unknown strain components for each subcell, or a total of $6N_{\alpha}N_{\beta}N_{\gamma}$ unknowns. Solution of linear equations, in essence, amounts to the inversion of a matrix which has the rank of the number of unknown quantities. Since the computational effort associated with matrix inversion increases approximately as the **cube** of the matrix rank, reducing the number of unknown quantities has a major impact on computational efficiency. The reduction in unknowns due to the reformulation is shown in Figure 3 and can be quite significant. For example, a 10×10 ×10 unit cell originally required solution for 6000 unknowns. In the reformulated version of **GMC-3D** the number of unknowns is reduced to 330. Consequently, the corresponding execution times are reduced as well. Table I gives sample CPU times for identical cases executed using the original and reformulated versions of **GMC** with continuous reinforcement. Clearly, as the number of subcells in the repeating unit cell increases, the increase in efficiency attributable to the reformulation becomes astronomical. It is important to note that:

(1) The reformulation of **GMC** gives **identical results** as the original formulation of GMC **in every case.**

(2) The significant increase in the computational efficiency of GMC due to the reformulation allows:
   (a) **Analysis** of simple unit cells (i.e. few subcells) **in a fraction of the time.**
   (b) **Analysis of refined unit cells** (i.e. many subcells) which was **previously impossible** (due to excessive execution times and memory requirements) can now be performed in times comparable to those previously required for analysis of simple unit cell, see Table I.

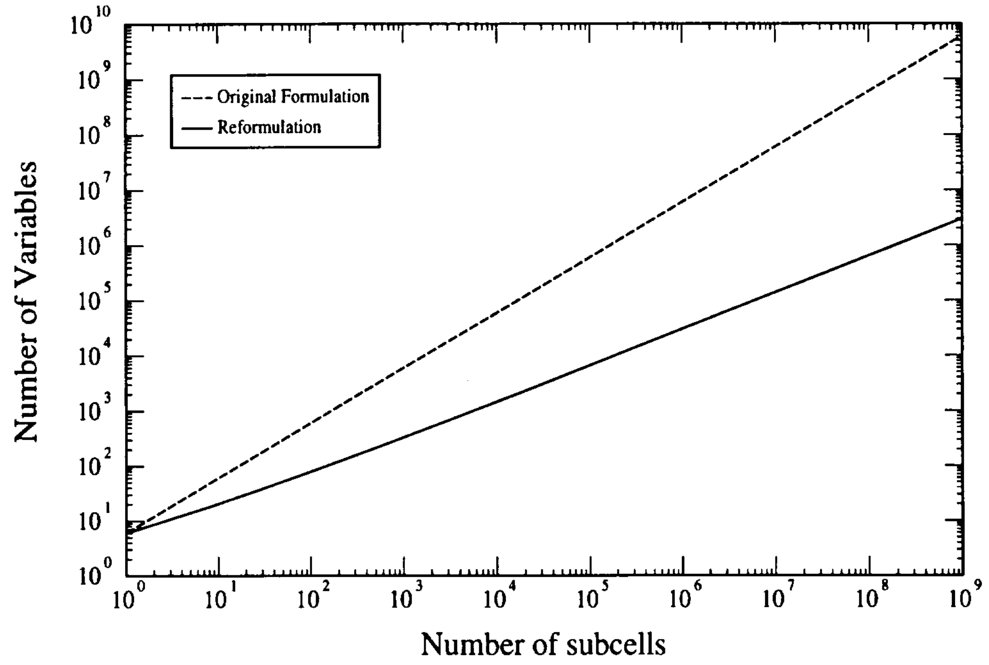


Figure 3. Number of subcells vs. number of unknown variables for the original and reformulated versions of GMC-3D for $N_\alpha = N_\beta = N_\gamma$

Table I: CPU Times (seconds) for Heat-Up of Unidirectional SiC/TiAl

GMC Version	Subcell Discretization							
	2x2	4x4	6x6	8x8	10x10	12x12	20x20	100x100
original	0.87	19	182	508	8,679	43,781	-	-
reformulated	0.18	0.25	0.5	0.9	1.5	2.3	8.3	796
speed-up ratio	4.8	76	364	564	5,786	19,035	-	-

☞ **Note**: These execution times are for a specialized version of **GMC-2D** and not **MAC/GMC**, therefore one should not expect to see the same execution times when running **MAC/GMC** due to increased overhead. However, speed-up ratios should be comparable.

3.2 Integration Algorithms

There are two integration algorithms currently available within **MAC/GMC.** The first is the standard, explicit Forward Euler algorithm, which can be expressed as,

$$w_{i+1} = w_i + \Delta t f(t_i, w_i)$$

where

$$w_i = \underset{\sim}{y}(t_i)$$
$$w_{i+1} = \underset{\sim}{y}(t_{i+1})$$

and $f(t_i, w_i)$ is the rate of change with respect to time of the vector, $\underset{\sim}{y}(t_i)$, i.e.

$$f(t_i, w_i) = \dot{\underset{\sim}{y}} = \frac{d\underset{\sim}{y}(t_i)}{dt}$$

The second is a predictor/corrector algorithm with a self-adaptive time step which uses:

1) a 4th order Runge-Kutta starter:

$$w_i = w_{i-1} + \frac{(K_1 + 2K_2 + 2K_3 + K_4)}{6}$$

where

$$K_1 = \Delta t f(t_{i-1}, w_{i-1})$$
$$K_2 = \Delta t f\left(t_{i-1} + \frac{\Delta t}{2}, w_{i-1} + \frac{K_1}{2}\right)$$
$$K_3 = \Delta t f\left(t_{i-1} + \frac{\Delta t}{2}, w_{i-1} + \frac{K_2}{2}\right)$$
$$K_4 = \Delta t f(t_{i-1} + \Delta t, w_{i-1} + K_3)$$

2) with an Adams-Bashforth four-step predictor:

$$w_0 = \alpha_1 \qquad w_1 = \alpha_2 \qquad w_2 = \alpha_3 \qquad w_3 = \alpha_4$$

$$w_{i+1}^P = w_i + \frac{\Delta t}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2})$$
$$-9f(t_{i-3}, w_{i-3})]$$

and 3) an Adams-Moulton four step corrector:

$$w_{i+1}^{c} = w_i + \frac{\Delta t}{24}[9f(t_{i+1}, w_{i+1}^{p}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})]$$

where the α's come from the 4th order Runge-Kutta starter. Further details may be found in [8].

☞ **Note**: It has been found, based on experience, that for relatively rapid monotonic or cyclic loadings it may be more efficient to use the Forward Euler integrator since the predictor/corrector requires 5 evaluations per step, as shown above. However, in the case of creep, relaxation or slow monotonic or cyclic loading histories, significant increases in solution speeds can be obtained using the predictor/corrector algorithm.

Finally, within **MAC/GMC** the vector $\underset{\sim}{w}$ as used above contains the following macro quantities:

position	contents
1 - 6	Macro Total Strain
7 - 12	Macro Stress
13 - 18	Macro Inelastic Strain
19 - 30	(currently empty space for 2 6x1 vectors)
31 - 36	Macro Thermal Strain
37	Current Temperature
38- 46	Future Use

and the quantities associated with each subcell are stored sequentially in $\underset{\sim}{w}$, such that

position	contents
47 - 52	Micro Total Strain
53 - 58	Micro Stress
59 - 64	Micro Inelastic Strain
65 - 76	Micro Internal State Variables (space for 2 6x1 vectors)
77 - 82	Micro Thermal Strain
83 - 88	Debonding Parameters

The above 42 positions (i.e., positions 47-88) are repeated for the total number of subcells (N) thus bringing the total length of the $\underset{\sim}{w}$ vector to 46+42N. It follows that a second vector of similar length that contains the corresponding macro and micro rates, e.g., $f(t_i, w_i)$, is also utilized.

3.3 Available Constituent Constitutive Models

Currently **MAC/GMC** provides two elastic and five inelastic constitutive models. These models have been selected purely based upon the availability of material parameters for the materials of interest. However, **MAC/GMC** is designed in a modular fashion thus allowing the implementation of additional constitutive models through a user defined subroutine. Two of the five available inelastic models are capable of representing transversely isotropic material behavior, thus allowing one to investigate the reinforcement of an anisotropic matrix allowing idealization of a heterogeneous material via a pseudo-homogenous anisotropic material (e.g. fiber tow). In all five inelastic models a purely elastic response is possible by modifying a single material parameter for each model as noted below.

3.3.1 Transversely Isotropic Elastic Model

Reference: Jacob Aboudi, Mechanics of Composite Materials, Elsevier, 1991

The following transversely isotropic model is provided for those materials that have an elastic only response and whose strong direction is aligned with the 1 axis shown in Fig. 6, such as the fiber constituent in a composite.

$$\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{23}\\ \sigma_{13}\\ \sigma_{12}\end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0\\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0\\ C_{12} & C_{23} & C_{22} & 0 & 0 & 0\\ 0 & 0 & 0 & C_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & C_{44} & 0\\ 0 & 0 & 0 & 0 & 0 & C_{66}\end{bmatrix}\left(\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \gamma_{23}\\ \gamma_{13}\\ \gamma_{12}\end{bmatrix} - \begin{bmatrix}\alpha_L\Delta T\\ \alpha_T\Delta T\\ \alpha_T\Delta T\\ 0\\ 0\\ 0\end{bmatrix}\right)$$

where components C_{ij} can be expressed in terms of five independent constants,

$$E_A, E_T, \nu_A, \nu_T, G_A$$

thus,

$$C_{11} = E_A + 4\kappa\nu_A^{\ 2}$$
$$C_{12} = 2\kappa\nu_A$$
$$C_{22} = \kappa + \frac{0.5E_T}{(1+\nu_T)}$$
$$C_{23} = \kappa - \frac{0.5E_A}{(1+\nu_T)}$$
$$C_{44} = G_A$$
$$C_{66} = \frac{(C_{22} - C_{23})}{2}$$

with

$$\kappa = 0.25E_A/[0.5(1-\nu_T)(E_A/E_T) - \nu_A^{\ 2}]$$

3.3.2 Anisotropic Elastic Model

Reference: S. M. Arnold, A Transversely Isotropic Thermoelastic Theory, NASA TM 101302, 1988

An alternative transversely isotropic model has also been provided for those materials that have an elastic only response, but whose strong direction may not be aligned with the 1 axis shown in Fig. 6 (i.e., the plane of isotropy is allowed to rotate), for example in the case of woven composites, see **Example N**. This elastic model is consistent with the anisotropic inelastic models described subsequently and is defined within the global coordinate system via a direction cosine vector, d_i.

$$\underset{\sim}{\sigma} = [\underset{\sim}{C}]\{\underset{\sim}{\varepsilon} - \underset{\sim}{\varepsilon}^{th}\}$$

where the stiffness matrix, $\underset{\sim}{C}$, is a function of the five independent material parameters $E_A, E_T, \nu_A, \nu_T, G_A$ and the direction cosine vector, d_i, and, in general is fully populated. Note the multiaxial thermal strain tensor is assumed to have the following form,

$$\varepsilon_{ij}^{th} = [(\alpha_L - \alpha_T)d_i d_j + \delta_{ij}\alpha_T]\Delta T$$

Further details can be found in the above reference.

3.3.3 Bodner-Partom Model

Reference: Jacob Aboudi, Mechanics of Composite Materials, Elsevier, 1991

This model represents a Bodner-Partom viscoplastic material with isotropic hardening, Z, and can be used for an initially isotropic metallic material.

The flow law is given as:

$$\dot{\varepsilon}_{ij}^{I} = \Lambda s_{ij}$$

where

$$\Lambda = \sqrt{\frac{D_2^{PL}}{J_2}}$$

$$D_2^{PL} = D_0^2 exp\left[-\left(\frac{A^2}{J_2}\right)^n\right]$$

$$A^2 = \frac{1}{3} Z_{eff}^2 \left(\frac{n+1}{n}\right)^{\frac{1}{n}}$$

$$J_2 = \frac{1}{2} S_{ij} S_{ij}$$

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}$$

The evolution law for isotropic hardening is given as:

$$\dot{Z} = m(Z_1 - Z_{eff})\frac{\dot{W}^{PL}}{Z_0}$$

where Z_0, Z_1 and m are inelastic constants and the plastic work rate, $\dot{W}^{PL}$, is given by;

$$\dot{W}^{PL} = \sigma_{ij}\dot{\varepsilon}_{ij}^{I}$$

$$Z_{eff} = Z_0 + q\int_0^t \dot{Z}(\tau)d\tau + (1-q)\sum_{i,j=1}^{3} r_{ij}\int_0^t \dot{Z}(\tau) r_{ij}(\tau)d\tau$$

$$r_{ij}(t) = \sigma_{ij}(t)/[\sigma_{kl}(t)\sigma_{kl}(t)]^{1/2}$$

An elastic only response may be obtained by setting the material parameter D_0 to zero.

3.3.4 Modified Bodner-Partom Model

Reference: R. W. Neu, "Nonisothermal Material Parameters For the Bodner-Partom Model", MD-Vol. 43, Material Parameter Estimation for Modern Constitutive Equations, L. A. Betram, S.B. Brown, and A.D. Freed, Eds., ASME, Book No. H00848, 1993.

This model represents a nonisothermal Bodner-Partom viscoplastic material with isotropic, Z^I, and directional hardening, Z^D, and can be used for an initially isotropic metallic material.

The flow law is given as:

$$\dot{\varepsilon}_{ij}^{I} = \Lambda s_{ij}$$

where

$$\Lambda = D_0 exp\left[-0.5\left(\frac{Z^2}{3J_2}\right)^n\right]\frac{1}{\sqrt{J_2}}$$

$$Z = Z^I + Z^D$$

$$J_2 = \frac{1}{2}S_{ij}S_{ij}$$

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}$$

The evolution law for the isotropic hardening, Z^I, is given as:

$$\dot{Z^I} = m_1(Z_1 - Z^I)\dot{W}^{PL} - A_1 Z_1\left(\frac{Z^I - Z_2}{Z_1}\right)^{r_2} + \left(\left(\frac{Z^I - Z_2}{Z_1 - Z_2}\right)\frac{\partial Z_1}{\partial T} + \left(\frac{Z_2 - Z^I}{Z_1 - Z_2}\right)\frac{\partial Z_2}{\partial T}\right)\dot{T}$$

where Z_0, Z_1,Z_2,A_1, r$_1$and m_1 are the material parameters associated with isotropic hardening. The magnitude of the directional hardening is defined as the scalar product of a state variable,β_{ij}, and unit stress vector, u_{ij}, as given below.

$$Z^D = \beta_{ij}u_{ij}$$

$$\dot{\beta_{ij}} = m_2(Z_3 u_{ij} - \beta_{ij})\dot{W}^{PL} - A_1 Z_1\left(\frac{\sqrt{\beta_{ij}\beta_{ij}}}{Z_1}\right)^{r_2} v_{ij} + \left(\left(\frac{\beta_{ij}}{Z_3}\right)\frac{\partial Z_3}{\partial T}\right)\dot{T}$$

$$u_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{kl}\sigma_{kl}}}$$

$$v_{ij} = \frac{\beta_{ij}}{\sqrt{\beta_{kl}\beta_{kl}}}$$

where, the plastic work rate is defined as; $\dot{W}^{PL} = \sigma_{ij}\dot{\varepsilon}^{I}_{ij}$ and with, Z_3, A_2, r_2 and m_2 being the material parameters defining the directional evolution. An elastic only response may be obtained by setting the material parameter D_0 to zero.

3.3.5 Robinson Viscoplastic Model

Reference: S.M. Arnold, D.N. Robinson, and P.A. Bartlotta, "Unified Viscoplastic Behavior of Metal Matrix Composites", NASA TM 105819, 1992

This model represents a **transversely isotropic material** wherein the vector of direction cosines, d_i, defines the preferred material direction. In this model the strength of anisotropy is specified by the parameters ω and η; where ω is the ratio of the normal longitudinal and transverse yield stress and η is the ratio of longitudinal and transverse threshold shear stress.

Flow Law:

$$\dot{\varepsilon}^{I}_{ij} = \frac{\langle F^n \rangle}{2\mu}\Gamma_{ij}$$

Evolution Law:

$$\dot{a}_{ij} = \frac{H}{G^{\beta}}\dot{\varepsilon}^{I}_{ij} - RG^{m-\beta}\Pi_{ij}$$

where

$$\Gamma_{ij} = \Sigma_{ij} - \xi(D_{ki}\Sigma_{jk} + D_{jk}\Sigma_{ki} - 2I_0D_{ij}) - \frac{1}{2}\zeta I_0(3D_{ij} - \delta_{ij})$$

$$\Pi_{ij} = a_{ij} - \xi(D_{ki}a_{jk} + D_{jk}a_{ki} - 2\hat{I}_0D_{ij}) - \frac{1}{2}\zeta\hat{I}_0(3D_{ij} - \delta_{ij})$$

and

$$F = \frac{1}{\kappa_T^2}\left[I_1 + \frac{1}{\eta^2}I_2 + \frac{9}{4(4\omega^2-1)}I_3\right] - 1$$

$$\hat{G} = \frac{1}{\kappa_T^2}\left[\hat{I}_1 + \frac{1}{\eta^2}\hat{I}_2 + \frac{9}{4(4\omega^2-1)}\hat{I}_3\right]$$

$$G = \langle \hat{G} - \hat{G}_0 \rangle Hv[S_{ij}\pi_{ij}] + \hat{G}_0$$

$$I_1 = J_2 - I - \frac{1}{4}I_3 \qquad I_2 = I - I_3 \qquad I_3 = I_0^2$$

$$J_2 = \frac{1}{2}\Sigma_{ij}\Sigma_{ji} \qquad I = D_{ij}\Sigma_{ji} \qquad D_{ij} = d_i d_j \qquad \Sigma_{ij} = S_{ij} - a_{ij}$$

$$\xi = \frac{\eta^2 - 1}{\eta^2} \qquad \zeta = \frac{4(\omega^2-1)}{4\omega^2 - 1}$$

with $\kappa_T, \mu, n, H, \beta, R, m, G_o, \eta$ and ω representing the associated required inelastic material parameters and < > denoting a Macauley bracket, as defined below.

$$\langle\ \rangle = \begin{Bmatrix} 0 & x < 0 \\ x & x \geq 0 \end{Bmatrix}$$

The invariants $\hat{I}_1, \hat{I}_2, \hat{I}_3$ are the same I_1, I_2, I_3 as those given above but with Σ_{ij} replaced by a_{ij}. Special cases involving an isotropic material and/or elastic only response can be obtained by defining $\omega = \eta = 1$ and/or by setting κ_T to an extremely large number.

3.3.6 Generalized Viscoplastic Potential Structure (GVIPS) Model

References: S.M. Arnold, A.F. Saleeb, and M.G. Castelli, "A Fully Associative, Non-Linear Kinematic, Unified Viscoplastic Model for Titanium Based Matrices", NASA TM 106609, 1994.

S.M. Arnold, A.F. Saleeb, and M.G. Castelli, "A Fully Associative, Nonisothermal, NonLinear Kinematic, Unified Viscoplastic Model for Titanium Alloys", NASA TM 106926, 1994.

This model is a fully associative, multiaxial, nonlinear kinematic hardening viscoplastic model for use with initially isotropic metallic materials. A unique aspect of this model is the inclusion of non-linear hardening through the use of a compliance operator Q_{ijkl} in the evolution law for the back stress. This non-linear tensorial operator is significant in that it allows both the flow and evolutionary laws to be fully associative (and therefore easily integrated) and greatly influences the multiaxial response under non-proportional loading paths.

Flow Law:

$$\dot{\varepsilon}_{ij}^{I} = \frac{3}{2}\left\|\dot{\varepsilon}_{ij}^{I}\right\|\frac{\Sigma_{ij}}{\sqrt{J_2}} \qquad if \qquad F \geq 0$$

where

$$\left\|\dot{\varepsilon}_{ij}^{I}\right\| = \sqrt{\frac{2}{3}\dot{\varepsilon}_{ij}^{I}\dot{\varepsilon}_{ij}^{I}} = \frac{\mu F^{n}}{\kappa}$$

Internal constitutive rate equation:

$$\dot{a}_{ij} = L_{ijrs}\dot{A}_{rs}$$

Evolution Law:

$$\dot{A}_{rs} = \dot{\varepsilon}_{rs}^{I} - \frac{3}{2}\frac{\beta\kappa}{\kappa_o^2}\left\|\dot{\varepsilon}_{ij}^{I}\right\|\frac{a_{rs}}{\sqrt{G}}Hv[Y] - \frac{3R_\alpha B_0 G^q}{\kappa_o^{\;2}}a_{rs} \qquad if \qquad a_{ij}\Sigma_{ij} \geq 0$$

$$\dot{A}_{rs} = Q_{rslm}E_{lmnp}\left(\dot{\varepsilon}_{np}^{I} - \frac{3}{2}\frac{\beta\kappa}{\kappa_o^2}\left\|\dot{\varepsilon}_{ij}^{I}\right\|\frac{a_{np}}{\sqrt{G}}Hv[Y] - \frac{3R_\alpha B_0 G^q}{\kappa_o^{\;2}}a_{np}\right) \qquad if \qquad a_{ij}\Sigma_{ij} < 0$$

where

$$F = \langle \frac{\sqrt{J_2}}{\kappa} - Y \rangle$$

$$Y = \langle 1 - \beta\sqrt{G} \rangle$$

$$G = \frac{I_2}{\kappa_o^2}$$

$$L_{ijrs} = [Q_{ijrs}]^{-1} = \frac{\kappa_o^2}{3B_0(1 + B_1 p G^{p-1})}\left(I_{ijrs} - \frac{3B_1 p(p-1)G^{p-2}}{\kappa_o^2(1 + B_1 p G^{p-1}(6p-5))} a_{rs} a_{ij}\right)$$

and

$$I_2 = \frac{3}{2}a_{ij}a_{ij} \qquad \Sigma_{ij} = S_{ij} - a_{ij}$$

$$J_2 = \frac{3}{2}\Sigma_{ij}\Sigma_{ij} \qquad S_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}$$

with $\kappa, \mu, n, \kappa_o, B_0, B_1, p, R_\alpha, q$ and β being the associated required inelastic material parameters. Typically, $\kappa, \mu, B_0, R_\alpha$ and β are taken to be functions of temperature and $\kappa_o = \kappa(T_{ref})$ is the initial drag stress at the reference temperature. The special case of an elastic only response maybe obtained by setting κ to an extremely large value.

3.3.7 Transversely Isotropic GVIPS Model (TGVIPS)

Reference: A.F. Saleeb and T.E. Wilt, Analysis of the Anisotropic Viscoplastic-Damage Response of Composite Laminates-Continuum Basis and Computational Algorithms, Int. J. Numer. Meth. Enging., Vol. 36, pp. 1629-1660, 1993.

This model is a fully associative, multiaxial, isothermal, nonlinear kinematic hardening viscoplastic model for use with **initially transversely isotropic** metallic materials. A unique aspect of this model is the inclusion of non-linear hardening through the use of a compliance operator Q_{ijkl} in the evolution law for the back stress. This non-linear tensorial operator is significant in that it allows both the flow and evolutionary laws to be fully associative (and therefore easily integrated) and greatly influences the multiaxial response under non-proportional loading paths.

The flow law for the inelastic strain, $\dot{\underset{\sim}{\varepsilon}}^I$, is given by,

$$\dot{\underset{\sim}{\varepsilon}}^I = \frac{\langle F^n \rangle}{2\mu}\underset{\sim}{\Gamma}$$

and the evolution law for internal stress, $\underset{\sim}{\alpha}$ is given by,

$$\dot{\underset{\sim}{\alpha}} = \left[\underset{\sim}{Z}_m + \frac{h'}{h(1+2\beta)}(\underset{\sim}{\alpha} \otimes \underset{\sim}{\alpha})\right]\left(\frac{H}{G^\beta}\dot{\underset{\sim}{\varepsilon}}^I - RG^m\underset{\sim}{\Pi}\right)$$

where $h = H/G^\beta$, $h' = -\beta/G$ and $Z_m = \underset{\sim}{M}^{-1}$

$$F = \frac{1}{2\kappa_t^2}(\underset{\sim}{\sigma} - \underset{\sim}{\alpha}) : \underset{\sim}{M} : (\underset{\sim}{\sigma} - \underset{\sim}{\alpha}) - 1$$

$$G = \frac{1}{2\kappa_t^2}\underset{\sim}{\alpha} : \underset{\sim}{M} : \underset{\sim}{\alpha}$$

$$\underset{\sim}{\Gamma} = \underset{\sim}{M} : (\underset{\sim}{\sigma} - \underset{\sim}{\alpha})$$

$$\underset{\sim}{\Pi} = \underset{\sim}{M} : \underset{\sim}{\alpha}$$

$$\underset{\sim}{M} = \underset{\sim}{P} - \xi\underset{\sim}{Q} - \frac{1}{2}\zeta\underset{\sim}{R}$$

The anisotropy of the material is introduced through the $\underset{\sim}{M}$ matrix, specifically the parameters, ζ, ξ which are defined as,

$$\xi = \frac{\eta^2 - 1}{\eta^2} \qquad \zeta = \frac{4(\omega^2 - 1)}{4\omega^2 - 1}$$

$$\eta = \frac{K_l}{K_t} \qquad \omega = \frac{Y_l}{Y_t}$$

In the above, $0 \le \xi \le 1$ and $0 \le \zeta \le 1$ are the material strength ratios, in which the constants K_l (K_t) are the threshold strengths in longitudinal (transverse) shear, and Y_l (Y_t) are the threshold strengths in longitudinal (transverse) tension. Note that for an isotropic material, $\omega = \eta = 1$.

In addition, the fourth-order tensors $\underset{\sim}{P}$, $\underset{\sim}{Q}$, and $\underset{\sim}{R}$ are defined as,

$$\underset{\sim}{P} = \underset{\sim}{I} - \frac{1}{3}(\underset{\sim}{\delta} \otimes \underset{\sim}{\delta}) \qquad I_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{ji} + \delta_{il}\delta_{jk})$$

$$Q_{ijkl} = \frac{1}{2}(D_{ik}\delta_{jl} + D_{il}\delta_{jk} + D_{jk}\delta_{il} + D_{jl}\delta_{ik}) - 2D_{ij}D_{kl}$$

$$R_{ijkl} = 3D_{ij}D_{kl} - (D_{ij}\delta_{kl} + \delta_{ij}D_{kl}) + \frac{1}{3}(\delta_{ij}\delta_{kl})$$

In the above, the vector of direction cosines d_i defines the orientation of the material fiber, which leads to the material directionality tensor $\underset{\sim}{D}$, ($\underset{\sim}{D} = d_i d_j$). Also, $\underset{\sim}{\delta}$ is the Kronecker delta (second-order identity tensor) and $\underset{\sim}{I}$ is the fourth-order identity tensor. Finally, the symbol $:$ indicates double-contraction and $\otimes$ cross product.

☞ **Note:** When calculating Z_m for the three dimension case one needs to replace $\underset{\sim}{P}$ with $\hat{\underset{\sim}{P}}$ = diag[1,1,1,2,2,2] as $\underset{\sim}{P}$ is singular for the three dimensional case.

3.4 Laminate Theory

MAC/GMC includes the capability to analyze general (symmetric and nonsymmetric) composite laminates [9], see Fig. 4. Mid-plane strains and resultant forces in the plane of the laminate may be applied. That is, the global laminate stress-strain relation that is solved within **MAC/GMC** is expressed as,

$$\begin{bmatrix} \bar{N}_{XX} \\ \bar{N}_{YY} \\ \bar{N}_{XY} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} \bar{\varepsilon}_{xx} \\ \bar{\varepsilon}_{yy} \\ \bar{\gamma}_{xy} \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} \bar{\kappa}_{xx} \\ \bar{\kappa}_{yy} \\ \bar{\kappa}_{xy} \end{bmatrix} - \begin{bmatrix} \bar{N}^{I}_{XX} \\ \bar{N}^{I}_{YY} \\ \bar{N}^{I}_{XY} \end{bmatrix} - \begin{bmatrix} \bar{N}^{T}_{XX} \\ \bar{N}^{T}_{YY} \\ \bar{N}^{T}_{XY} \end{bmatrix} \qquad \text{(EQ 15)}$$

$$\begin{bmatrix} \bar{M}_{XX} \\ \bar{M}_{YY} \\ \bar{M}_{XY} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} \bar{\varepsilon}_{xx} \\ \bar{\varepsilon}_{yy} \\ \bar{\gamma}_{xy} \end{bmatrix} + \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \begin{bmatrix} \bar{\kappa}_{xx} \\ \bar{\kappa}_{yy} \\ \bar{\kappa}_{xy} \end{bmatrix} - \begin{bmatrix} \bar{M}^{I}_{XX} \\ \bar{M}^{I}_{YY} \\ \bar{M}^{I}_{XY} \end{bmatrix} - \begin{bmatrix} \bar{M}^{T}_{XX} \\ \bar{M}^{T}_{YY} \\ \bar{M}^{T}_{XY} \end{bmatrix} \qquad \text{(EQ 16)}$$

or

$$\begin{bmatrix} \underset{\sim}{\bar{N}} \\ \underset{\sim}{\bar{M}} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{A} & \underset{\sim}{B} \\ \underset{\sim}{B} & \underset{\sim}{D} \end{bmatrix} \begin{bmatrix} \underset{\sim}{\bar{\varepsilon}} \\ \underset{\sim}{\bar{\kappa}} \end{bmatrix} - \begin{bmatrix} \underset{\sim}{\bar{N}}^{I} \\ \underset{\sim}{\bar{M}}^{I} \end{bmatrix} - \begin{bmatrix} \underset{\sim}{\bar{N}}^{T} \\ \underset{\sim}{\bar{M}}^{T} \end{bmatrix} \qquad \text{(EQ 17)}$$

where $\underset{\sim}{\bar{N}}$, $\underset{\sim}{\bar{N}}^{I}$, $\underset{\sim}{\bar{N}}^{T}$, and $\underset{\sim}{\bar{M}}$, $\underset{\sim}{\bar{M}}^{I}$, $\underset{\sim}{\bar{M}}^{T}$ are the global laminate total, inelastic and thermal force and moment resultants, respectively. The matrices $\underset{\sim}{A}$,$\underset{\sim}{B}$, and $\underset{\sim}{D}$ are the global laminate extensional, coupling and bending stiffnesses, respectively, and, $\underset{\sim}{\bar{\varepsilon}}$ and $\underset{\sim}{\bar{\kappa}}$ the global laminate mid-plane strain and mid-plane curvature, respectively.

In forming the laminate extensional stiffness $\underset{\sim}{A}$ the generalized method of cells model, **GMC**, is utilized to calculate the individual lamina properties. In this regard, the individual laminate stiffness, in lamina coordinates, $\underset{\sim}{Q}$ is given by,

$$\underset{\sim}{Q} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{21} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \qquad \text{(EQ 18)}$$

in which the components of $\underset{\sim}{Q}$ are given as,

$$Q_{11} = C_{11} - \frac{C_{13}C_{31}}{C_{33}} \qquad Q_{12} = C_{12} - \frac{C_{13}C_{23}}{C_{33}}$$
$$Q_{22} = C_{22} - \frac{C_{23}C_{32}}{C_{33}} \qquad Q_{33} = C_{66} \tag{EQ 19}$$

The C_{ij} in the above are the effective macro properties for the unidirectional composite lamina obtained from **GMC**.

It follows, employing a Kirchhoff-Love hypothesis, that the lamina stress-strain relation in global (laminate) coordinates denoted by x-y is given by the relation,

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix}_k = \begin{bmatrix} \bar{Q}_{11} & \bar{Q}_{12} & \bar{Q}_{13} \\ \bar{Q}_{12} & \bar{Q}_{22} & \bar{Q}_{23} \\ \bar{Q}_{13} & \bar{Q}_{23} & \bar{Q}_{33} \end{bmatrix}_k \left[\begin{bmatrix} \bar{\varepsilon}_{xx} \\ \bar{\varepsilon}_{yy} \\ \bar{\gamma}_{xy} \end{bmatrix} + z \begin{bmatrix} \bar{\kappa}_{xx} \\ \bar{\kappa}_{yy} \\ \bar{\kappa}_{xy} \end{bmatrix} - \begin{bmatrix} \varepsilon^I_{xx} \\ \varepsilon^I_{yy} \\ \gamma^I_{xy} \end{bmatrix}_k - \begin{bmatrix} \varepsilon^T_{xx} \\ \varepsilon^T_{yy} \\ \gamma^T_{xy} \end{bmatrix}_k \right] \tag{EQ 20}$$

or

$$\underset{\sim}{\sigma}_k = \overline{\underset{\sim}{Q}_k}(\underset{\sim}{\bar{\varepsilon}} + z\underset{\sim}{\bar{\kappa}} - \underset{\sim}{\varepsilon}^I_k - \underset{\sim}{\varepsilon}^T_k) \tag{EQ 21}$$

where

$$\overline{\underset{\sim}{Q}_k} = R_k^{-1}\underset{\sim}{Q}_k R_k \tag{EQ 22}$$

$$R_k = \begin{bmatrix} \cos^2\theta & \sin^2\theta & 2\sin\theta\cos\theta \\ \sin^2\theta & \cos^2\theta & (-2)\sin\theta\cos\theta \\ -\sin\theta\cos\theta & \sin\theta\cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \tag{EQ 23}$$

and θ is the orientation of the longitudinal lamina axis with respect to the global x-direction, see Fig. 4, thus $\underset{\sim}{\bar{Q}}_k$ is the transformed lamina stiffness, i.e. from local lamina to global laminate coördinates. In addition, $\underset{\sim}{\sigma}_k$ is the lamina stress vector in laminate coordinates. It then follows that the global laminate extensional stiffness $\underset{\sim}{A}$ is given by,

$$\underset{\sim}{A} = \sum_{k=1}^{nly} \underset{\sim}{\bar{Q}}_k t_k \tag{EQ 24}$$

in which nly is the total number of layers in the laminate and t_k is the thickness of the k^{th} lamina. The coupling and bending stiffnesses can be similarly developed and given by the following expressions:

$$\underset{\sim}{B} = \frac{1}{2}\sum_{k=1}^{nly} \underset{\sim}{\overline{Q}}_k(z_k^2 - z_{k-1}^2) \qquad \text{(EQ 25)}$$

$$\underset{\sim}{D} = \frac{1}{3}\sum_{k=1}^{nly} \underset{\sim}{\overline{Q}}_k(z_k^3 - z_{k-1}^3) \qquad \text{(EQ 26)}$$

where z_k is the distance (considering the sign) to the top of layer k from the mid-plane.

Returning to EQ. 17, the quantities $\underset{\sim}{\overline{N}}^I$ and $\underset{\sim}{\overline{N}}^T$ (the laminate inelastic and thermal force resultants, respectively) are calculated from the individual lamina contributions through the following relations,

$$\underset{\sim}{\overline{N}}^I = \sum_{k=1}^{nly} \underset{\sim}{\overline{Q}}_k \int_{z_{k-1}}^{z_k} \underset{\sim}{\varepsilon}_k^I dz \qquad \underset{\sim}{\overline{M}}^I = \sum_{k=1}^{nly} \underset{\sim}{\overline{Q}}_k \int_{z_{k-1}}^{z_k} \underset{\sim}{\varepsilon}_k^I z dz \qquad \text{(EQ 27)}$$

and

$$\underset{\sim}{\overline{N}}^T = \sum_{k=1}^{nly} \underset{\sim}{\overline{Q}}_k \underset{\sim}{\varepsilon}_k^T t_k \qquad \underset{\sim}{\overline{M}}^T = \frac{1}{2}\sum_{k=1}^{nly} \underset{\sim}{\overline{Q}}_k \underset{\sim}{\varepsilon}_k^T (z_k^2 - z_{k-1}^2) \qquad \text{(EQ 28)}$$

where the integrals in EQ. 27 are performed using second order gauss quadrature which requires two integration points per layer. Thus all field quantities are tracked at the two gauss quadrature points in each layer of the laminate in **MAC/GMC.**

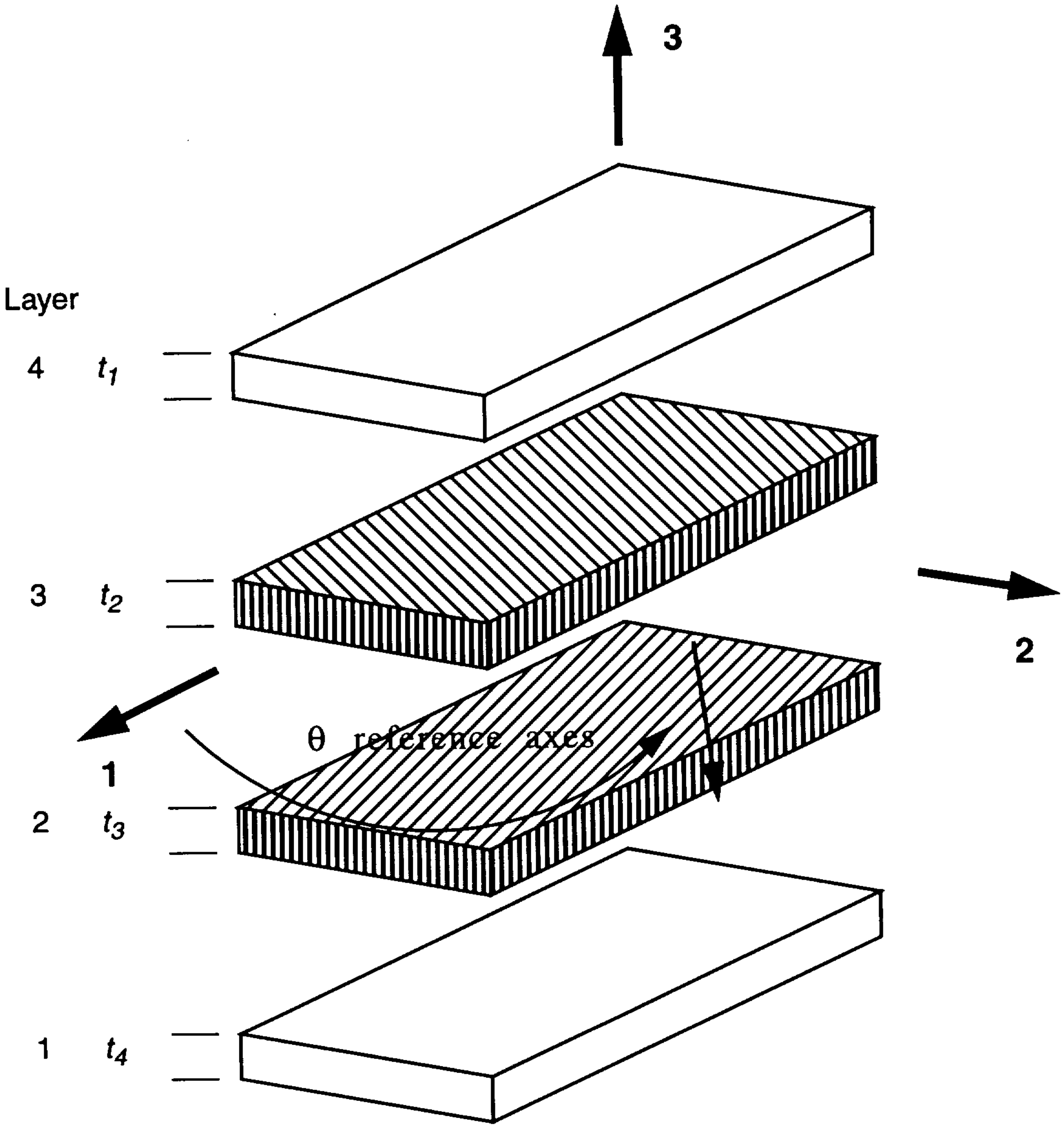


Figure 4: Laminate Coordinate System

3.5 Fatigue Damage Analysis

The fatigue damage calculations utilize a recently developed multiaxial, isothermal, continuum damage mechanics model for the fatigue of unidirectional metal matrix composites [10]. The model is phenomenological, stress based, and assumes a single scalar internal damage variable, D. Note that for an initially anisotropic material, the evolution of the damage, although a scalar, is directionally dependent. As will be shown, this directional dependence is accounted for in the terms, $\hat{F}_m$, Φ_{fl}, and Φ_u. The present multiaxial, isothermal, continuum damage model for **initially transversely isotropic materials** (e.g., unidirectional metal matrix composites) may be expressed as, [10]

$$\int_{D_{k-1}}^{D_k} dD = \int_0^N [1-(1-D)^{\beta+1}]^{\alpha}\left[\frac{\hat{F}_m}{1-D}\right]^{\beta} dN \qquad \text{(EQ 29)}$$

where N is the number of cycles at the current stress state, (σ_k), and D_k and D_{k-1} are the amount of damage at the current and previous increments, respectively. The quantity α which is a function of the current stress state is defined as,

$$\alpha = 1 - a\frac{\langle \Phi_{fl}\rangle}{\langle \Phi_u\rangle} \qquad \text{(EQ 30)}$$

where $\langle\ \rangle$ are the Macauley brackets. In the above, the fatigue limit surface, Φ_{fl}, and the static fracture surface, Φ_u, are defined as

$$\Phi_{fl} = \frac{1}{2}\underset{t_0}{max}\,\underset{t}{max}\; F_{(\sigma_{fl})}(\sigma_{ij}(t) - \sigma_{ij}(t_0)) - 1 \qquad \text{(EQ 31)}$$

$$\Phi_u = 1 - \underset{t}{max}\; F_{(\sigma_u)}(\sigma_{ij}(t)) \qquad \text{(EQ 32)}$$

and the quantity $\hat{F}_m$, used in EQ. 29 is the normalized stress amplitude, and is defined as,

$$\hat{F}_m = \frac{1}{2}\underset{t}{max}\,\underset{t_0}{max}\; F_{(M)}(\sigma_{ij}(t) - \sigma_{ij}(t_0)) \qquad \text{(EQ 33)}$$

Note, the case $\langle\Phi_u\rangle = 0$ indicates static fracture, which is failure, making it unnecessary to perform the fatigue calculations as in this case the subcell is considered to have failed completely. Thus, having to consider the possibility of α being undefined is unnecessary. The case $\langle\Phi_{fl}\rangle = 0$ indicates that the current stress state is below the fatigue limit and thus α is set equal to 1. This presents a special case when integrating the fatigue damage expression, EQ. 29, and will be considered later in this section.

In the above equations, t_0 is the time at the beginning of the current load cycle, and t, is some time during the load cycle. The general form for $F_{(\sigma_{fl}),\ (\sigma_u),\ or\ (M)}$ may be expressed as,

$$F_{(\ )} = \sqrt{\frac{1}{(\ )_L^2}\left\{(4\omega_{(\ )}^2 - 1)I_1 + \frac{4\omega_{(\ )}^2 - 1}{\eta_{(\ )}^2}I_2 + \frac{9}{4}I_3\right\}} \qquad \text{(EQ 34)}$$

It is in the above expression, in which the evolution of the damage becomes directionally dependent. This simply amounts to the assumption of partial anisotropy, where the "extent" (magnitude) of damage is affected by the directionality of the stress state. Specifically, the directional dependence enters through the quantities, I_1, I_2, I_3, $\omega_{(\ )}$, and $\eta_{(\ )}$. The quantities, I_1, I_2, I_3 are invariants having the form,

$$\begin{aligned} I_1 &= \frac{1}{2}S_{ij}S_{ij} - d_i d_j S_{jk} S_{ki} + \frac{1}{4}(d_i d_j S_{ij})^2 \\ I_2 &= d_i d_j S_{jk} S_{ki} - (d_i d_j S_{ij})^2 \\ I_3 &= (d_i d_j S_{ij})^2 \end{aligned} \qquad \text{(EQ 35)}$$

which are a function of the current deviatoric stress state, $S_{ij}^k = \sigma_{ij}^k - \frac{1}{3}\sigma_{mm}^k \delta_{ij}$, as well as the vector d_i which defines the materials' preferred direction (e.g., fiber orientation in a composite). In addition, the terms $\omega_{(\ )}$ and $\eta_{(\ )}$ represent the ratios of longitudinal to transverse normal and shear stresses, respectively. Note, the longitudinal direction is parallel to the preferred direction and transverse is perpendicular to the preferred direction. For initially transversely isotropic materials, $\omega_{(\ )}$ and $\eta_{(\ )}$ are > 1 and for isotropic materials $\omega_{(\ )}$ and $\eta_{(\ )}$ are = 1.

In the context of micromechanics analysis within **MAC/GMC**, the isotropic simplification of the above representation will be predominately used for the various constituent phases [11]. This isotropic representation is the previously validated NonLinear Cumulative Damage Rule (NLCDR) developed at **ONERA** (Office Nationale d'Etudes et de Recherches Aerospatiales) for isotropic monolithic metals. However, it maybe desirable to use the transverse isotropic form when dealing with fiber tows in woven composites systems.

3.5.1 Above Initial Fatigue Limit

Given a current state of stress, σ_k, above the fatigue limit, i.e. $\alpha \neq 1$ and integrating EQ. 29 results in an expression for the number of cycles, N, i.e.,

$$N = \frac{([1-(1-D_k)^{\beta+1}]^{1-\alpha} - [1-(1-D_{k-1})^{\beta+1}]^{1-\alpha})}{\hat{F}_m^{\beta}(1-\alpha)(\beta+1)} \quad \text{(EQ 36)}$$

Note that D_{k-1} is the total amount of damage at the beginning of the load block and D_k is the total amount of damage at the end of this load block. Alternatively, rewriting EQ. 36 an expression for the damage, D_k, in terms of the number of cycles and previous damage can be obtained, i.e.,

$$D_k = 1 - \left(1 - \left\{[1-(1-D_{k-1})^{\beta+1}]^{1-\alpha} + (1-\alpha)(\beta+1)\hat{F}_m^{\beta}N\right\}^{\frac{1}{1-\alpha}}\right)^{\frac{1}{\beta+1}} \quad \text{(EQ 37)}$$

In the present computational scheme, since the damage increment is controlled, both D_k and D_{k-1} are known. That is, $D_k = D_{k-1} + \Delta D$ where ΔD is the user specified increment in damage. Thus EQ. 36 is used to predict the increment in the number of cycles for each subcell, N^e, due to the imposed increment in damage.

To calculate the number of cycles to failure, for an initial damage amount, D_{k-1}, let $D_k = 1$, which results in the following,

$$N_F = \frac{(1-[1-(1-D_{k-1})^{\beta-1}]^{1-\alpha})}{\hat{F}_m^{\beta}(1-\alpha)(\beta+1)} \quad \text{(EQ 38)}$$

3.5.2 Below Initial Fatigue Limit

Now consider the case in which the current stress state is below the initial fatigue limit, i.e. $\langle \Phi_{fl} \rangle = 0$, which leads to $\alpha_k = 1$. Thus, EQ. 29 takes the form,

$$\int_{D_{k-1}}^{D_k} \frac{(1-D)^{\beta}}{1-(1-D)^{\beta+1}} dD = \int_0^N \hat{F}_m^{\beta} dN \quad \text{(EQ 39)}$$

Upon integrating the above equation, the increment in cycles, N, with initial damage, D_{k-1}, may be expressed as,

$$N = \left(\frac{\log[1-(1-D_k)^{\beta+1}] - \log[1-(1-D_{k-1})^{\beta+1}]}{\hat{F}_m^{\beta}(\beta+1)}\right) \quad \text{(EQ 40)}$$

Alternatively, the following expression for the damage, D_k, may be expressed as:

$$D_k = 1 - \{1 - [1 - (1 - D_{k-1})^{\beta + 1}]exp((\beta + 1)\hat{F}_m^{\beta}N)\}^{\frac{1}{\beta + 1}} \qquad \text{(EQ 41)}$$

For the number of cycles to failure, let $D_k = 1$,

$$N_F = \frac{-\log[1 - (1 - D_{k-1})^{\beta + 1}]}{\hat{F}_m^{\beta}(\beta + 1)} \qquad \text{(EQ 42)}$$

The effect of damage is included in the present micromechanics analysis utilizing the concept of effective stress and the hypothesis of strain-equivalence [12].

$$\hat{\underset{\sim}{\sigma}} = \frac{\underset{\sim}{\sigma}}{(1 - D_k)} \qquad \text{(EQ 43)}$$

3.6 User Defined Inelastic Material Model

MAC/GMC, has the option for a user to implement their own inelastic constitutive model. This is accomplished by using the subroutine **USRMAT** into which the user writes the necessary FORTRAN code for the particular constitutive model being implemented. The **USRMAT** subroutine is as shown below and is always called whenever ncmd=99 (see section **4.2.12**).

```
c########################################################################
        SUBROUTINE USRMAT(DSA, SA, PE, PV, D, LOCTISO, TIME, TSTEP,
     &     CTEMP, DTEMPR, NIO, NE, NV, NS, MN, CDUM, DMGF,
     &     NEP, NVP, NSASIZE)

c     purpose:  user material constitutive model for determination of
c               the inelastic strain and state variable rates
c               (used when ncmd = 99)

      IMPLICIT DOUBLE PRECISION (A - H, O - Z)


      CHARACTER*2 CDUM
      DIMENSION SS(6), S(6), R(6)
      DIMENSION DSA(NSASIZE), SA(NSASIZE)
      DIMENSION PV(NVP), PE(NEP), D(3)

***********************************************************************
c  note: 1) in this subroutine, [SA] and [DSA] contain the
c           micro (subcell) quantities for aboudi's micromechanics model
c
c        2) arrangement of [dsa] & [sa]  arrays:
c           variable              location
c          +------------------------------------------------------------
c          | strain rate          (1-6)  (contains ENGINEERING shears)
c          |------------------------------------------------------------
c          | stress rate          (7-12)
c          |------------------------------------------------------------
c          | inelastic
c          | strain rate          (13-18)  (contains ENGINEERING shears)
c          |------------------------------------------------------------
c          | 12 "slots"           (19-30)
c          | for state variables
c          |------------------------------------------------------------
c          | thermal strain rate  (31-36)
c          +------------------------------------------------------------
***********************************************************************
c NOTE:  quantities in [SA] and [DSA] are SUBCELL quantities - the
c        values on entry are for the first subcell containing material
c        # MN - the values on exit of this subroutine will be applied to
c        ALL SUBCELLS containing material # MN.  It is thus recommended
c        that, if using the field variables, you assign the appropriate
c        material # to ONE SUBCELL ONLY.  Use of [SA] and [DSA] in this
c        context in conjunction with bending in laminate theory will
c        result in erroneous results as field variables become dependent
```

```
c       on through-thickness position.
************************************************************************
c   on entry:
c       SA          - vector of total (integrated) quantities (see above)
c       PE(NEP)      - vector of elastic constants for material MN
c                      (where NE = # of elastic constants --> 9 MAX)
c       PV(NVP)      - vector of viscoplastic constants for material MN
c                     (where NV = # of viscoplastic constants --> 19 MAX)
c       D(3)        - vector of direction cosines (for models 3, 7, & 9)
c       LOCTISO     - flag indicating if ANY material exhibits
c                     local transverse isotropy (and global anisotropy)
c                   = 0 - all materials are at most globally transversely
c                         isotropic (D not used)
c                   = 1 - at least one material is locally transversely
c                         isotropic (D used)
c       TIME        - current time
c       TSTEP       - current time step
c       CTEMP       - current temperature
c       DTEMPR      - time rate of change of temperature
c       NIO         - unit number of .out file
c       NE          - # of elastic constants --> 9 MAX
c       NV          - # of viscoplastic constants --> 19 MAX
c       NS          - subcell number
c       MN          - material number
c       CDUM        - material character/number designation
c                     (i.e. F1 = fiber #1)
c       DMGF        - damage factor - if damage is included the user
c                     should multiply material stiffness terms by DMGF
c                     when using such terms in their inelastic model.
c
c   expected on exit:
c       DSA         - vector of rate quantities (see above)
************************************************************************
c
************************************************************************
*                           BEGIN USER EDITS                           *
************************************************************************
c     place code defining model here
************************************************************************
*                            END USER EDITS                            *
************************************************************************
      RETURN
      END
```

A description of the input and output required for the usrmat subroutine is as follows

Data supplied to **USRMAT**:

SA   array containing current total quantities for all of the state variables

PE   array containing elastic constants

PV	array containing inelastic constants
D(3)	vector of direction cosines (for models 3, 7, & 9)
LOCTISO	flag indicating if ANY material exhibits local transverse isotropy (and global anisotropy) = 0 - all materials are at most globally transversely isotropic (D not used) = 1 - at least one material is locally transversely isotropic (D used)

TIME	current time
TSTEP	current time increment (step)
CTEMP	current temperature
DTEMPR	current temperature rate
NE	total number of elastic constants
NV	total number of inelastic constants
NIO	output file unit number
NCE	current subcell number
NS	subcell number
MN	material number of current constituent
CDUM	Character string identifying current constituent
DMGF	damage factor - if damage is included the user should multiply material stiffness terms by DMGF when using such terms in their inelastic model.

Output expected from **USRMAT**:

DSA	current increments in all state variables

The state variables are arranged in SA and DSA in the following order:

Position	Quantity
1-6	strain
7-12	stress
13-18	inelastic strain
19-30	available space for 2 6x1 vectors for model dependent internal state variables
31-36	thermal strain
37-42	debonding parameters

Again, SA contains the total quantities and DSA contains the rates.

☞ **Note: Example K** contains a sample input file and the **USRMAT** subroutine containing an implementation of the Bodner-Partom viscoplastic material model as well as two elastic models.

☞ **Note: GMC** utilizes **engineering strains,** in SA and DSA consequently it is the users responsibility to convert tensorial strain quantities to engineering **before exiting** the **USRMAT** routine.

3.7 User Defined Functional Form For Material Properties

An addition to this version of **MAC/GMC** is the option for a user to implement functionally dependent material properties. With this the elastic and/or inelastic material properties may be defined as a function of any variable contained in SA and DSA (e.g., stress, strain, time, temperature, etc.) In certain cases, the user must also provide code for determining the stiffness matrix (USRFORMDE) or the time derivative of the stiffness matrix (USRCPEVAL). This is accomplished by using the subroutines **USRFUN, USRFORMDE, and USRCPEVAL** into which the user writes the necessary FORTRAN code for the particular constitutive model being implemented. All three routines are shown below.

```
c#######################################################################
      SUBROUTINE USRFUN(MN, TIME, TSTEP, CTEMP, DTEMPR, SA, DSA,
     &     DOLD, PEM, PVM, D, LOCTISO, ALPA, ALPT,
     &     NE, NV, NMTS, NEP, NVP, NSASIZE)

c     purpose:  user subroutine to allow elastic and viscoplastic
c               material properties to be functions of TEMP or
c               field variables.  Used for user defined functional
c               form material properties, that is, when:
c               (mat .eq. 'U') .and. (ifm .eq. 2)
c
c     note:     can be used in conjunction with a provided material
c               constitutive model, or a constitutive model input
c               by the user in USRMAT

      INCLUDE 'parm.inc'
      IMPLICIT DOUBLE PRECISION (A - H, O - Z)
      DIMENSION DOLD(6, 6), D(3)
      DIMENSION PEM(NEP, NMTS), PVM(NVP, NMTS)
      DIMENSION ALPA(NMTS), ALPT(NMTS)
      DIMENSION DSA(NSASIZE), SA(NSASIZE)
**********************************************************************
c  note: 1) in this subroutine, [SA] and [DSA] contain the
c           micro (subcell) quantities for aboudi's micromechanics model
c
c        2) arrangement of [dsa] & [sa]  arrays:
c           variable             location
c          +-----------------------------------------------------------
c          | strain rate          (1-6)   (contains ENGINEERING shears)
c          |-----------------------------------------------------------
c          | stress rate          (7-12)
c          |-----------------------------------------------------------
c          | inelastic
c          | strain rate          (13-18)  (contains ENGINEERING shears)
c          |-----------------------------------------------------------
c          | 12 "slots"           (19-30)
c          | for state variables
c          |-----------------------------------------------------------
c          | thermal strain rate  (31-36)
c          +-----------------------------------------------------------
```

```
************************************************************************
c NOTE:  quantities in [SA] and [DSA] are SUBCELL quantities - the
c        values on entry are for the first subcell containing material
c        # MN - the values on exit of this subroutine will be applied to
c        ALL SUBCELLS containing material # MN.  It is thus recommended
c        that, if using the field variables, you assign the appropriate
c        material # to ONE SUBCELL ONLY.  Use of [SA] and [DSA] in this
c        context in conjunction with bending in laminate theory will
c        result in erroneous results as field variables become dependent
c        on through-thickness position.
************************************************************************
c   on entry:
c      MN           - material number
c      TIME         - current time
c      TSTEP        - current time step
c      CTEMP        - current temperature
c      DTEMPR       - time rate of change of temperature
c      SA           - vector of total (integrated) quantities (see above)
c      DSA          - vector of rate quantities (see above)
c      DOLD(6, 6)   - previous elastic material stiffness matrix
c      PEM(NE, MN)  - vector of previous elastic constants for material
c                     # MN (where NE = # of elastic constants --> 9 MAX)
c      PVM(NV, MN)  - vector of previous viscoplastic constants for
c                     material # MN
c                     (where NV = # of viscoplastic constants --> 19 MAX)
c
c   expected on exit:
c     PEM(NE, MN) - vector of current elastic constants for material MN
c      PVM(NV, MN)  - vector of current viscoplastic constants for
c                     material MN
c      D(3)         - vector of direction cosines
c                    (required for models 3, 7, & 9)
c      LOCTISO      - flag indicating if ANY material exhibits
c                     local transverse isotropy (and global anisotropy)
c                   = 0 - all materials are at most globally transversely
c                         isotropic (D not used)
c                   = 1 - at least one material is locally transversely
c                         isotropic (D used)
c      ALPA(MN)     - longitudinal cte for material MN
c      ALPT(MN)     - transverse cte for material MN
************************************************************************

************************************************************************
*                          BEGIN USER EDITS                            *
************************************************************************
c                          place code here
************************************************************************
*                           END USER EDITS                             *
************************************************************************
      RETURN
      END
      SUBROUTINE USRFORMDE(MN, PEM, PVM, D, LOCTISO, DNEW,
     &     NE, NV, NEP, NVP, NMTS)

c     purpose:  user subroutine to allow formation of material stiffness
c               matrices based on a user constitutive model (used when
c               ncmd = 99)
c
C      INCLUDE 'parm.inc'

      IMPLICIT DOUBLE PRECISION (A - H, O - Z)

      DIMENSION DNEW(6, 6)
      DIMENSION PEM(NEP, NMTS), PVM(NVP, NMTS)
      DIMENSION D(3)

***********************************************************************
c   on entry:
c      MN           - material number
c      PEM(NE, MN) - vector of elastic constants for material
c                     # MN (where NE = # of elastic constants --> 9 MAX)
c      PVM(NV, MN) - vector of viscoplastic constants for material # MN
c                     (where NV = # of viscoplastic constants --> 19 MAX)
c      D(3)         - vector of direction cosines
c                    (required for models 3, 7, & 9)
c      LOCTISO      - flag indicating if ANY material exhibits
c                     local transverse isotropy (and global anisotropy)
c                   = 0 - all materials are at most globally transversely
c                         isotropic (D not used)
c                   = 1 - at least one material is locally transversely
c                         isotropic (D used)
c
c   expected on exit:
c      DNEW(6, 6)   - current elastic material stiffness matrix

***********************************************************************
*                          BEGIN USER EDITS                          *
***********************************************************************
c                          place code here
***********************************************************************
*                           END USER EDITS                           *
***********************************************************************

      RETURN
      END
```

```
########################################################################
      SUBROUTINE USRCPEVAL(DSA, SA, MN, TIME, TSTEP, CTEMP, DTEMPR,
     &     DNEW, DOLD, PEM, PVM, D, LOCTISO, ALPA, ALPT, DDOT,
     &     NE, NV, NMTS, NEP, NVP, NSASIZE)


c     purpose:  user subroutine to allow formation of the TIME
c               derivative of the material stiffness matrix.
c               this subroutine is used when:
c               a) material properties are user defined and functional
c                  form.  That is:  (mat .eq. 'U') .and. (ifm .eq. 2)
c               b) the constitutive model is user-defined, and the
c                  material properties are not functional form, and
c                  the material properties are temperature-dependent
c                  that is: (ncmd .eq. 99) .and. (ifm .ne. 2) .and.
c                  (ndpt .eq. 2)

      IMPLICIT DOUBLE PRECISION (A - H, O - Z)

      DIMENSION DNEW(6, 6), DOLD(6, 6)
      DIMENSION DDOT(6, 6)

      DIMENSION PEM(NEP, NMTS), PVM(NVP, NMTS)
      DIMENSION ALPA(NMTS), ALPT(NMTS)
      DIMENSION DSA(NSASIZE), SA(NSASIZE)

************************************************************************
c  note: 1) in this subroutine, [SA] and [DSA] contain the
c           micro (subcell) quantities for aboudi's micromechanics model
c
c        2) arrangement of [dsa] & [sa]  arrays:
c           variable              location
c         +------------------------------------------------------------
c         | strain rate           (1-6)   (contains ENGINEERING shears)
c         |------------------------------------------------------------
c         | stress rate           (7-12)
c         |------------------------------------------------------------
c         | inelastic
c         | strain rate           (13-18)  (contains ENGINEERING shears)
c         |------------------------------------------------------------
c         | 12 "slots"            (19-30)
c         | for state variables
c         |------------------------------------------------------------
c         | thermal strain rate  (31-36)
c         +------------------------------------------------------------
************************************************************************
c NOTE:  quantities in [SA] and [DSA] are SUBCELL quantities - the
c        values on entry are for the first subcell containing material
c        # MN - the values on exit of this subroutine will be applied to
c        ALL SUBCELLS containing material # MN.  It is thus recommended
c        that, if using the field variables, you assign the appropriate
c        material # to ONE SUBCELL ONLY.  Use of [SA] and [DSA] in this
c        context in conjunction with bending in laminate theory will
c        result in erroneous results as field variables become dependent
c        on through-thickness position.
```

```
************************************************************************
c   on entry:
c      SA           - vector of total (integrated) quantities (see above)
c      DSA          - vector of rate quantities (see above)
c      MN           - material number
c      TIME         - current time
c      TSTEP        - current time step
c      CTEMP        - current temperature
c      DTEMPR       - time rate of change of temperature
c      DNEW(6, 6)   - current elastic material stiffness matrix
c      DOLD(6, 6)   - previous elastic material stiffness matrix
c      PEM(NE, MN)  - vector of current elastic constants for material MN
c                     (where NE = # of elastic constants --> 9 MAX)
c      PVM(NV, MN)  - vector of current viscoplastic constants for
c                     material MN
c                     (where NV = # of elastic constants --> 19 MAX)
c      D(3)         - vector of direction cosines
c                     (required for models 3, 7, & 9)
c      LOCTISO      - flag indicating if ANY material exhibits
c                     local transverse isotropy (and global anisotropy)
c                   = 0 - all materials are at most globally transversely
c                         isotropic (D not used)
c                   = 1 - at least one material is locally transversely
c                         isotropic (D used)
c      ALPA(MN)     - longitudinal cte for material MN
c      ALPT(MN)     - transverse cte for material MN
c
c   expected on exit:
c      DDOT(6, 6)   - derivative with respect to TIME of stiffness matrix
************************************************************************
************************************************************************
*                             BEGIN USER EDITS                         *
************************************************************************
c                              place code here
************************************************************************
*                              END USER EDITS                          *
************************************************************************

      RETURN
      END
```

A description of the input and output required for each subroutine is given in the comment statements at the top of each subroutine.

☞ **Note: MAC/GMC** assumes the longitudinal direction for the material is oriented in the x_1- coordinate direction, see Fig. 6.

☞ **Note: Example K** illustrates how one could use each of these routines to define material properties as well as the associated elastic stiffness matrix.

3.8 Interface Modeling

Interfaces in composite materials play a major role in the determination of their mechanical and thermal properties. Consequently, it is important to have the ability to model interface behavior accurately. This is accomplished in **MAC/GMC** in one of two ways. The first is to define an actual interface region with its own constitutive behavior. In this way the influence of initial imperfections (flaws, voids, improper wetting, etc.) and induced interfacial damage (due to stress, environment, chemical reactions, etc.) may be incorporated into the micromechanical analysis of the overall behavior of the composite. The development of proper interfacial constitutive models is an active area of research, and **MAC/GMC**, through the use of its **USRMAT** routine, provides the researcher with a convenient tool for testing new and existing interfacial constitutive models.

The second approach to modeling the effect of imperfect (weak) bonding between two phases (e.g. a fiber and a matrix) is to assume that a jump in the displacement field at an interface may occur given certain conditions, while still maintaining continuity of the traction vector. In the spirit of Jones and Whittier [13] and Achenbach and Zhu [14] we have assumed the following flexible interface model.

$$\begin{bmatrix} (u_n{}^I = R_n \cdot \sigma_n{}^I) \\ (u_t{}^I = R_t \cdot \sigma_t{}^I) \end{bmatrix} \quad if \quad \begin{bmatrix} \sigma_n{}^I \geq \sigma_{DBn} \\ \sigma_t{}^I \geq \sigma_{DBt} \end{bmatrix} \qquad \text{(EQ 44)}$$

$$\begin{bmatrix} (u_n{}^I = 0) \\ (u_t{}^I = 0) \end{bmatrix} \quad if \quad \begin{bmatrix} \sigma_n{}^I < \sigma_{DBn} \\ \sigma_t{}^I < \sigma_{DBt} \end{bmatrix} \qquad \text{(EQ 45)}$$

where R_n, R_t, σ_{DBn} and σ_{DBt} are the interfacial normal and shear, compliance and debond stresses, respectively. Note, the implementation of the various forms for R_n and R_t described below will impact the definition of the concentration matrices, $\underset{\sim}{A}^{(\alpha\beta\gamma)}$ in the original **GMC** formulation or its counterpart in the reformulated version, see **section 3.1.**

This approach to debonding has been implemented in **MAC/GMC** in two forms. In the first, R_n and R_t are assumed to be constants that are independent of time. Therefore, when the time derivative of EQ. 44 is taken (as it is for implementation in the incremental formulation of **MAC/GMC**) the expression becomes,

$$\begin{bmatrix} (\dot{u}_n{}^I = R_n \cdot \dot{\sigma}_n{}^I) \\ (\dot{u}_t{}^I = R_t \cdot \dot{\sigma}_t{}^I) \end{bmatrix} \quad if \quad \begin{bmatrix} \sigma_n{}^I \geq \sigma_{DBn} \\ \sigma_t{}^I \geq \sigma_{DBt} \end{bmatrix} \qquad \text{(EQ 46)}$$

Hence, the debonding is instantaneous, and immediately reaches its full extent. If R_n and R_t are chosen to be sufficiently large (as is customary), the stress at the interface will remain constant, with a value of σ_{DB} after debonding occurs.

In the second form, R_n and R_t are assumed to be functions of time. Thus we obtain

$$\begin{bmatrix} \dot{u}_n{}^I = R_n \cdot \dot{\sigma}_n{}^I + \dot{R}_n \cdot \sigma_n{}^I \\ \dot{u}_t{}^I = R_t \cdot \dot{\sigma}_t{}^I + \dot{R}_t \cdot \sigma_t{}^I \end{bmatrix} \quad if \quad \begin{bmatrix} \sigma_n{}^I \geq \sigma_{DBn} \\ \sigma_t{}^I \geq \sigma_{DBt} \end{bmatrix} \qquad \text{(EQ 47)}$$

in lieu of EQ 46. The additional term present in EQ. 47 for both normal and tangential debonding is significant. If the time-dependence of R_n and R_t is chosen wisely, these additional terms will enable local unloading in the composite. For implementation in **MAC/GMC**, the following functional form of the time-dependence has been employed,

$$R(t) = \Lambda\left[\exp\left(\frac{\hat{t}}{\mathrm{B}}\right) - 1\right] \qquad \text{(EQ 48)}$$

where $\hat{t}$ is the time since debonding, and Λ and B are functional parameters which characterize how the interface unloads.

Figure 5 shows a simple example to illustrate the differences between the two implementations of this debond model. The repeating unit cell used to generate the results shown in this figure, is IDP= 1 as illustrated in Fig. 9. As Fig. 5 shows, the results generated using both implementations of the debond model are the same until debonding occurs at an interfacial stress of approximately 15 ksi. At this point, the stress at the interface in the first implementation becomes constant, while, in the case of the second implementation, the slope of the interfacial stress verses applied strain curve decreases and then the interface begins to unload. The effect of the difference between the two implementations on the predicted composite stress-strain response is shown clearly in Fig. 5, where the softer composite response is a result of additional inelasticity in the remaining matrix subcells due to the local stress redistribution from the debonded subcells.

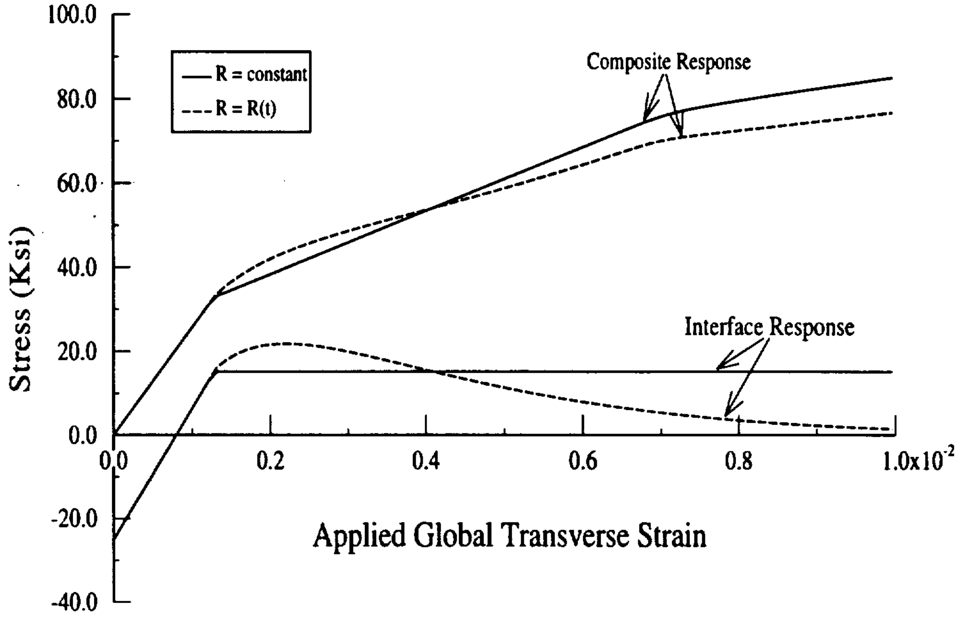


Figure 5. Simulated transverse behavior using the two implementations of the debond model in **MAC/GMC**.

3.9 References

1) Aboudi, J.; Mechanics of Composite Materials: A Unified Micromechanical Approach, Elsevier, Amsterdam, 1991.

2) Paley, M., and Aboudi, J.; "Micromechanical Analysis of Composites by the Generalized Method of Cells Model", Mechanics of Materials, Vol. 14, pp 127-139, 1992.

3) Aboudi, J.; "Micromechanical Analysis of Thermo-Inelastic Multiphase Short-Fiber Composites", NASA CR 195290, 1994, or Composites Engineering, Vol.5, No. 7, pp. 839-850, 1995.

4) Arnold, S.M., Wilt, T.E., Saleeb, A.F., and Castelli, M.G., "An Investigation of Macro and Micromechanical Approaches for a Model MMC System", NASA CP 19117, Vol. II, pp. (52.1) - (52.12), 1993.

5) Arnold, S.M., Pindera, M.J., and Wilt, T.E., "Influence of Fiber Architecture on the Elastic and Inelastic Response of Metal Matrix Composites", NASA TM 106705, 1994, Int. Jnl. of Plasticity, Vol. 12, No. 4, pp. 507-545, 1996.

6) Titanium Matrix Composites, Mechanical Behavior, Eds. S. Mall and T. Nicholas, Technomic, 1998.

7) M.J. Pindera and B.A. Bednarcyk; "An Efficient Implementation of the GMC Micromechanics Model for Multi-Phased Materials with Complex Microstructures", NASA CR 202350, 1997, Composites: Part B, Vol. 30, No. 9, pp. 1213-1238, 1999.

8) Burden, R.L. and Faires, J.D.; Numerical Analysis, Prindle, Weber and Schmidt, 1985.

9) Agarwal, B.D., and Broutman L. J., Analysis and Performance of Fiber Composites, Wiley Inter Science, 1980.

10) Arnold, S. M., and Kruch, S.: Differential Continuum Damage Mechanics Models for Creep and Fatigue of Unidirectional Metal Matrix Composites, Int. J Damage Mech., Vol. 3, n 2, 1994, pp. 170-191.

11) Wilt, T.E., Arnold, S. M., and Saleeb, A.F.: A Coupled/Uncoupled Computational Scheme for Deformation and Fatigue Damage Analysis of Unidirectional Metal Matrix Composites, Applications of Continuum Damage Mechanics to Fatigue and Fracture, ASTM STP 1315, D.L. McDowell, Ed., American Society for Testing and Materials, pp. 65-82, 1997.

12) Lemaitre, J. and Chaboche, J. L., Mechanics of Solid Materials, Cambridge University Press, 1990.

13) J.P. Jones and J.S. Whittier, J. Appl. Mech. Vol. 34, pg. 905, 1967.

14) J.D. Achenbach and H. Zhu, J. Mech. Phys. Solids, Vol. 37, pg. 381, 1989

4.0 Running MAC/GMC

MAC/GMC can be executed in one of two ways; 1) as a stand-alone code, and 2) through the UMAT facilities of the nonlinear finite element code, **ABAQUS**. Upon linking the associated object modules that comprise **MAC/GMC**, an executable version of **MAC/GMC** is generated as well as the file "feamac.o". The file "feamac.o" is one of the required files to be provided to **ABAQUS** for finite element access to **MAC**.

4.0.1 MAC/GMC - stand-alone:

The execution of **MAC/GMC** as a stand-alone code utilizes I/O redirection. Execution of **MAC/GMC** is initiated by typing a line at the command prompt such as;

```
MAC < infilename > outfilename
```

4.0.2 FEAMAC - Accessing MAC through ABAQUS:

To execute **MAC/GMC** through **ABAQUS** several files are required, these are summarized in the next section. The execution of **ABAQUS** for **MAC/GMC** access is as follows;

```
abaqus job=abaqus_input_deck user=feamac.o
```

☞ **Note: Please check the “readme” file included with your version for any last minute changes to this manual!**

OR you can check our web page at

www.lerc.nasa.gov/WWW/LPB/mac/index.html

4.1 Input and Output Files

The input and output files required for the stand-alone version of **MAC/GMC** and **FEAMAC** are specified in different ways. The differences between these two methods of running **MAC/GMC** are detailed below.

4.1.1 MAC stand-alone

Since **MAC/GMC** reads from standard in and writes to standard out, the name of the files used as input and output are left to the user's discretion, i.e., will have no impact on the performance of the **MAC/GMC** code. As will be explained in the following sections, the user will also need to specify the names of the **PATRAN** output files and the files containing the X-Y data for plotting.

Note that a "debug" file may be generated depending on the "PRINT" level the user specifies in the input file (see **4.2.2**). This file will have the name "mac_debug".

Input	Output
infilename	*outfilename*
	debug file (mac_debug)
	PATRAN files (optional, see section **4.2.9**)
	plot data files (see section **4.2.16 and 4.2.17**)
	damage files (optional, see section **4.2.6**)
	yield surface files (optional, see section **4.2.7**)

4.1.2 FEAMAC

Almost all of the capabilities of **MAC/GMC** are available through the user defined material facilities of the **ABAQUS** finite element code. Access to **MAC/GMC** from within **ABAQUS** is outlined here. Currently, **MAC/GMC,** can only be access using 3D elements; however, **FEAMAC** has only been specifically tested using the C3D8 element within **ABAQUS.** Note any options not available or modified for finite element implementation are denoted subsequently by the following notation, for example:

☞ **FEAMAC Note:** *Not required by FEAMAC*

An example containing an **ABAQUS** input deck, required FORTRAN subroutines, and **FEAMAC** input files are given in **Example O**.

<u>ABAQUS Input Deck</u>:

To utilize **FEAMAC**, the **ABAQUS** input deck must contain the "*USER

SUBROUTINE", "*USER DEFINED FIELD","*INITIAL CONDITIONS, TYPE=FIELD" ,"*EXPANSION" (if conducting nonisothermal analysis) and "*USER MATERIAL" card. Where the USER SUBROUTINE card identifies the name of the file containing the FORTRAN subroutines required to access **MAC/GMC** and the "USER MATERIAL" card instructs **ABAQUS** to obtain the materials stress/strain behavior from the supplied subroutines. The name of the user material (defined in the **ABAQUS** input deck), converted to upper case, will be appended with the file extension that has been defined in the user supplied FORTRAN routines, consequently this name must be used as the **MAC/GMC** input filename for the given material. Note: the material name is limited by **ABAQUS** to be 8 characters or less. See the **ABAQUS** Users' Manual for a complete descriptions of these cards

ABAQUS FORTRAN Subroutine Files:

The name of this file is to be provided on the " *USER SUBROUTINE" card inside the **ABAQUS** input deck. These subroutines will be executed for every integration point of every element in the finite element model associated with a given material group. The required subroutine are listed in **Example O**. Only a portion of the FEAMAC_INIT routine should be edited by the user. This portion is bounded by the comments;

```
C *** BEGIN USER EDITS ***
```

and

```
C *** END OF USER EDITS ***
```

The seven variables within this section that can be edited are as follows:

PATH:	CHARACTER*80 Path to working directory
EXTENSION:	CHARACTER*80 File name extension to be appended to material name to form the name of the **MAC/GMC** input deck.
DFNAME:	CHARATER*80 Name of diagnostic file.
NPEL:	INTEGER Number of integration points whose data is to be plotted.
N_PEN(N):	INTEGER ARRAY (N:1->NPEL) Element numbers whose data is to be plotted.
N_PIN(N):	INTEGER ARRAY (N:1->NPEL) Integration numbers whose data is to be plotted.

☞ **Note**: N_PEN() and N_PIN() pairs define locations of data extraction to be utilized with the *MACRO or *MICRO **MAC/GMC** options

AB_PRINT:

INTEGER
Diagnostic print level for **FEAMAC**.

0 - NONE
1 - UMAT: STRESS, STRAIN, INCREMENTAL AND EXECUTION TRACING INFORMATION
3 - FEAMAC_PRE: STRESS, STRAIN, INCREMENTAL AND EXECUTION TRACING INFORMATION
4 - TEMPERATURE INFORMATION
5 - STATE INFORMATION BEFORE AND AFTER CALL TO FEAMAC
25 - CONSTITUTIVE MATERIAL PROPERITES

☞ **Note**: Higher values of AB_PRINT provide all information provided by lower values plus additional information as described above.

<u>**FEAMAC** Input Deck</u>:

The **FEAMAC** input deck can be identical to the stand-alone **MAC/GMC** input deck as discussed in section **4.2**. The associated file name is obtained by converting the material property name, defined within the **ABAQUS** input deck, to uppercase and appending the file extension specified in the FEAMAC_INIT subroutine described previously. Since the load history definition and time integration are dictated by the associated **ABAQUS,** input deck, information in the **MAC/GMC** deck that relates to these issues will be ignored. All entries in section **4.2** that describe **MAC/GMC** options should be considered to be required by **FEAMAC** unless denoted otherwise.

☞ **Note:** Multiple **FEAMAC** input decks can be specific per **ABAQUS** run, so that multiple material systems can be analyzed within a given structure. Utilization of this option will allow the most efficient execution of large problems. For example, if one knows that only a small portion of the problem will be inelastic then define the zone of elements that may go inelastic by a given material group and the remainder by another. In this way maximum speed can be achieved, as a purely isothermal, elastic, problem with no micro fields being printed executes the fastest.

FEAMAC Output Files:

Most diagnostic messages from **FEAMAC** appear in the **ABAQUS** msg file. Files containing X-Y plotting information are generated for each N_PEN/N_PIN pair. The names for these files correspond to those defined in the **MAC/GMC** deck except that the N_PEN and N_PIN number are append to the name.

Example:
If in the FEAMAC_INIT subroutine the user specified;

```
NPEL = 1
N_PEN(1) = 18
N_PIN(1) = 7
```

and in the **FEAMAC** deck one specified;

```
*CURVE
 NP=2 %
*MACRO
 NT=1
 NC=1 X=1 Y=7 NAM=plot_file_sve %
```

FEAMAC would provide the stress and strain data in the 11-direction for integration point 7 of element number 18 in a file named

"PLOT_FILE_SVE.18.7.feamacro.data".

☞ **Note:** see section **4.2.17** regarding curve data input

4.2 Input Requirements (for both MAC/GMC and FEAMAC)

This section describes how the data should appear in the input file, *infilename*, where each line in the input file is limited to 80 characters. In the following, each block of input data will have its own subsection and will typically contain the following information:

1) statement of purpose
2) declaration of input data block
3) example(s)
4) notes

A data block has the following general format:

1) **∗KEYWORD** beginning of data block
2) input line(s)
3) % end of data block

where

1) **∗KEYWORD**: (denotes beginning of data block)

Each input data block begins with the appropriate keyword, which starts with the ∗ symbol. For example, the keyword for selecting the method of integration is **∗SOLVER**. The input routine scans the input file and locates the appropriate keyword and then reads the corresponding input data

2) Input line(s):

The input lines contain the necessary input data. The specific format of these input lines will be given in the following subsections. The capital letters denote actual variable names and lower case letters represent the possible input choices. Multiple inputs can be on one line, just as shown in the following. However, at **least one space must be used** to separate **data sets** on a single line.

Two special characters (**&** and **#**) are provided for entering the input.

The "**&**" (continuation) symbol:

For input data that is too long to fit on a **single 80 character line**, the "&" symbol is used as the continuation character. Thus, a line of data may be divided into a series of lines. For example, when entering the material properties all of the data cannot fit on a single line, thus the continuation character is required:

```
EL=11700.,11700.,0.365,0.365,4287.5,1.,1. &
VI=0.8E-8,0.1,0.1E-5,0.,0.85E-3,0.05,1.,1.,1.,3.3,1.8,1.35,1.,0.01
```

The "#" (comment) symbol:

The input file may also contain "comments" for the users convenience. The "#" symbol is used to mark a comment line, with the requirement that the "#" symbol appears in the first column.

3) % (denotes end of data block)

Each input block ends with the "%" symbol. The % symbol **should be** included as it signals the input routine that a particular data block has been completed.

A sample data block is as follows,

```
*SOLVER
#   NTF=2 ISTM=0.0001 ERR=0.1E-3
    NTF=2 ISTM=0.0000024 ERR=0.1E-2 %
```

Note how the # symbol is used to comment out a line of input data. Thus, the user can change various parameters by simply commenting out input lines containing different input data. Also, note how the data block is terminated with the % symbol and how it need not appear on a separate line.

☞ **Note:** It is suggested that the order of keywords in a given input file follow that given in this manual as some keyword ordering may give rise to problems, since in certain cases, **MAC/GMC** uses input from previous keywords to determine which input to read from later keywords.

4.2.1 Header Line:

Purpose: Define the title of this particular job (80A format).

☞ **Note:** the **Header Line** is limited to one 80 character line, and is always taken as the first line in the input file.

☞ **FEAMAC Note:** *Required by FEAMAC*

problem title

Example: Transverse tensile response of 35% SCS6/Ti-6-4

4.2.2 Output Print Level:

Purpose: To control the output generated.

☞ **FEAMAC Note:** *Required by FEAMAC*

```
*PRINT
   NPL=nplvl

%
```

Where:

nplvl:

-1 = print out macro stiffness matrix, engineering constants and macro-thermal expansion coefficients and stop

0 = Input echo, minimal output (most commonly used)

1 = time and iteration information

3 = Material properties/Stiffness matrix output

5 = Lamination information

7 = Damage information

9 = Ancillary data

10 *=program execution trace and all array data*

***CAUTION:** the choice 10 generates a very large output file*

☞ **Note:** Print levels are inclusive, such that the current level information as well as all lower level information is printed.

Example: (minimal print out)

```
*PRINT
   NPL=0 %
```

4.2.3 Load Type:

Purpose: To select load type

☞ **FEAMAC Note: *Only LCON is required for FEAMAC, LOP, and LSS are Ignored by FEAMAC.***

***LOAD**
LCON=*nsel*   LOP=*lop*   LSS=*iopt*

%

Where

nsel:

1 = Thermal Load
2 = Mechanical Load
3 = Thermomechanical Load

lop: (for 2-D and 3-D GMC)

1 = axial load in 1-direction
2 = axial load in 2-direction
3 = axial load in 3-direction
4 = shear load 23-direction
5 = shear load 13-direction
6 = shear load 12-direction
99 = generalized loading in potentially all 6 directions

For 2-D GMC:

7 = axial loads in 1 and 2 directions
8 = axial loads in 2 and 3 directions
9 = axial load in 1-direction and shear load in 23-direction
10 = axial load in 2-direction and shear load in 13-direction

For 3-D GMC:

7 = axial loads in 1 and 2 directions
8 = axial loads in 2 and 3 directions
9 = axial loads in 1 and 3 directions
10 = axial load in 1-direction and shear load in 23-direction
11 = axial load in 2-direction and shear load in 13-direction
12 = axial load in 3-direction and shear load in 12-direction

iopt:

1 = Strain control
2 = Stress control

☞ **Note:** If lop = 99 one must now enter six iopt values, for example:
LCON= nsel LOP=lop LSS = *iopt1, iopt2, iopt3, iopt4, iopt5, iopt6*

☞ **Note:** see Fig. 6 for the definitions of the directions mentioned above

☞ **Note:** If using laminate option; lop = 1, 2 or 6 are the only valid selections

Example: General loading option

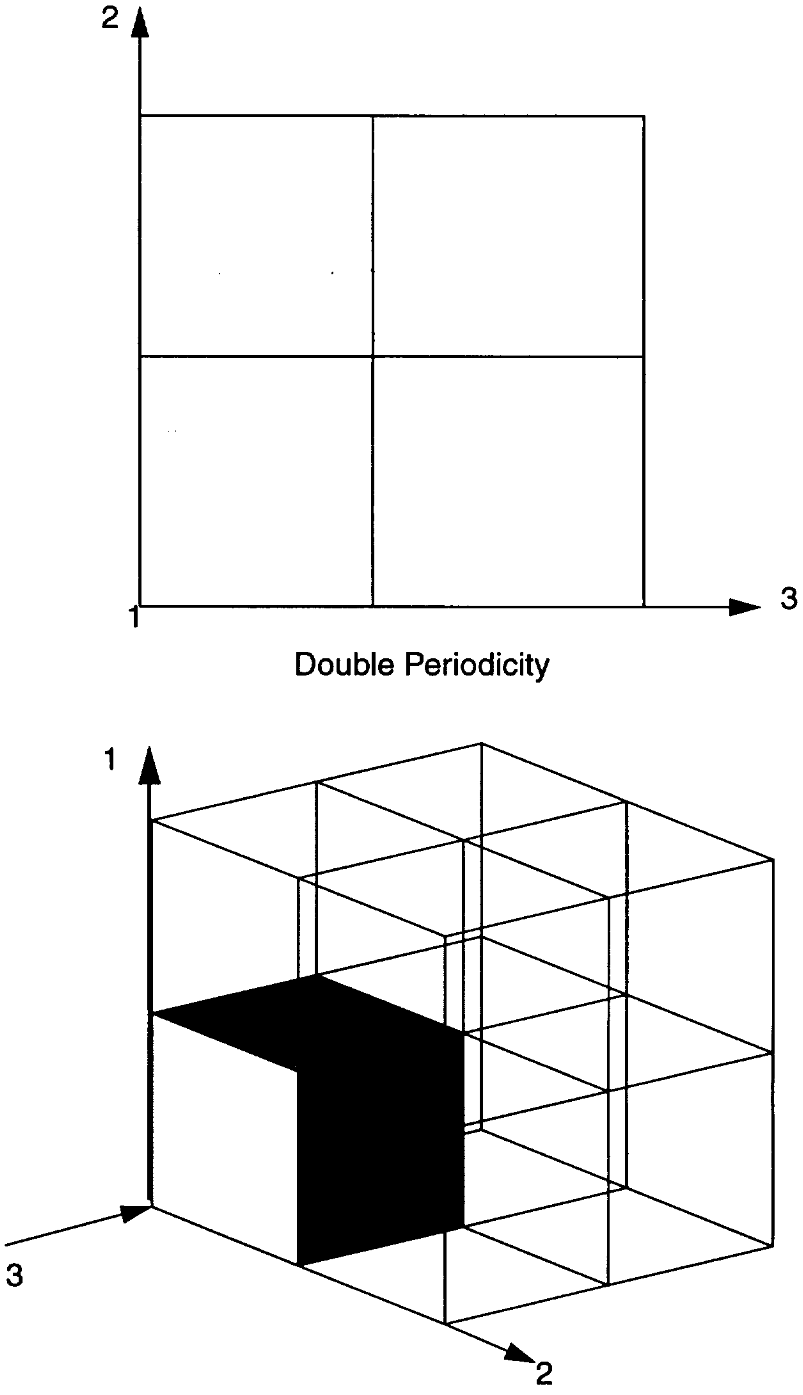
***LOAD**

```
LCON=1 LOP=99 LSS=2,1,2,2,2,2 %
```

☞ **Note:** This would correspond to specifying mixed stress and strain control, i.e.,:

$$\sigma_{11}, \varepsilon_{22}, \sigma_{33}, \tau_{23}, \tau_{31}, \tau_{12}$$

If however, the loads in the ***MECH** section were set to 0, 0.01, 0,0,0,0 this would correspond to a pure strain control problem in the 22 direction.



Double Periodicity

Triple Periodicity

Figure 6: Coordinate Systems

4.2.4 Mechanical Load Control:

Purpose: Select type of load control for mechanical load

☞ **Note: This block is only required if LCON = 2 or 3**

☞ **FEAMAC Note:** ***All the data on the MECH card is ignored by FEAMAC***

***MECH**
NPTW=*nptw* TI=$t_1, t_2....t_{nptw}$ LO= $l_1, l_2,....l_{nptw}$

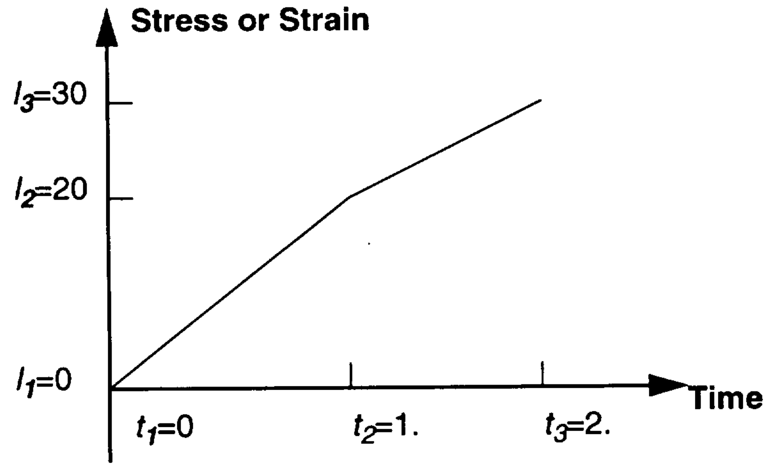
%

Where:

nptw: - number of points on load curve
t_1, t_2, ...t_{nptw}: time values
l_1, l_2, ...l_{nptw}: load curve values

Example:

***MECH**
NPTW=3 TI=0., 1., 2. LO=0., 20., 30. %



☞ **Note:** For LOP=7,8,...etc (i.e., two load/displacement components) the following format is required **(See Example J** for an example**)**:

```
*MECH
   NPTW=2    TI=0., 1.5   LO=0., 0.015          component 1 curve
   NPTW=2    TI=0., 1.5   LO=0., 0.01 %         component 2 curve
```

For LOP=99, six load/displacement components are required.

4.2.5 Temperature Control:

Purpose: Select control for temperature.

☞ **Note: This block is only required if LCON = 1 or 3**

☞ **FEAMAC Note: *All the data on the THERM card is ignored by FEAMAC***

***THERM**
NPTT=*nptt* TI=$t_1,t_2....t_{nptw}$ TE= $te_1,te_2,....te_{nptt}$

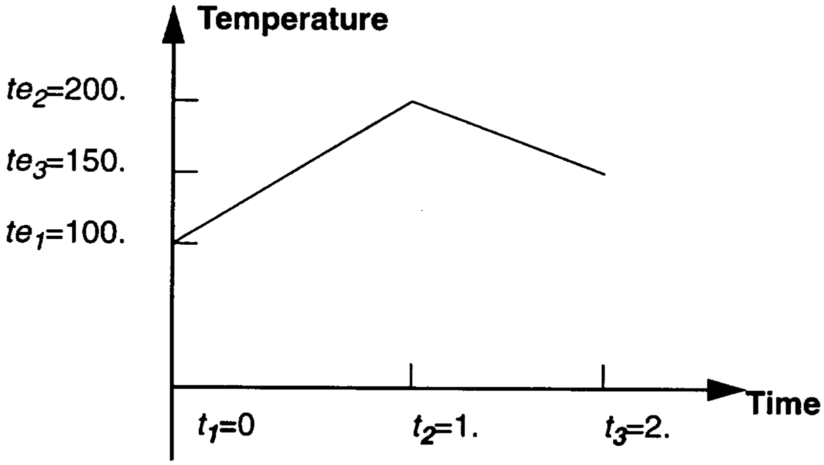
%

Where:

nptt: - number of points on temperature curve
t_1, t_2, ..t_{nptw}: time values
te_1, te_2, ..te_{nptt} : temperature curve values

Example:

***THERM**
NPTT=3 TI=0., 1., 2. TE=100., 200., 150. %



☞ **Note:** For the thermomechanical load $t_{nptw} \equiv t_{nptt}$, and both curves must have $t_1 \equiv 0$. But the number and time value of the data points in-between maybe different, see figure 7.

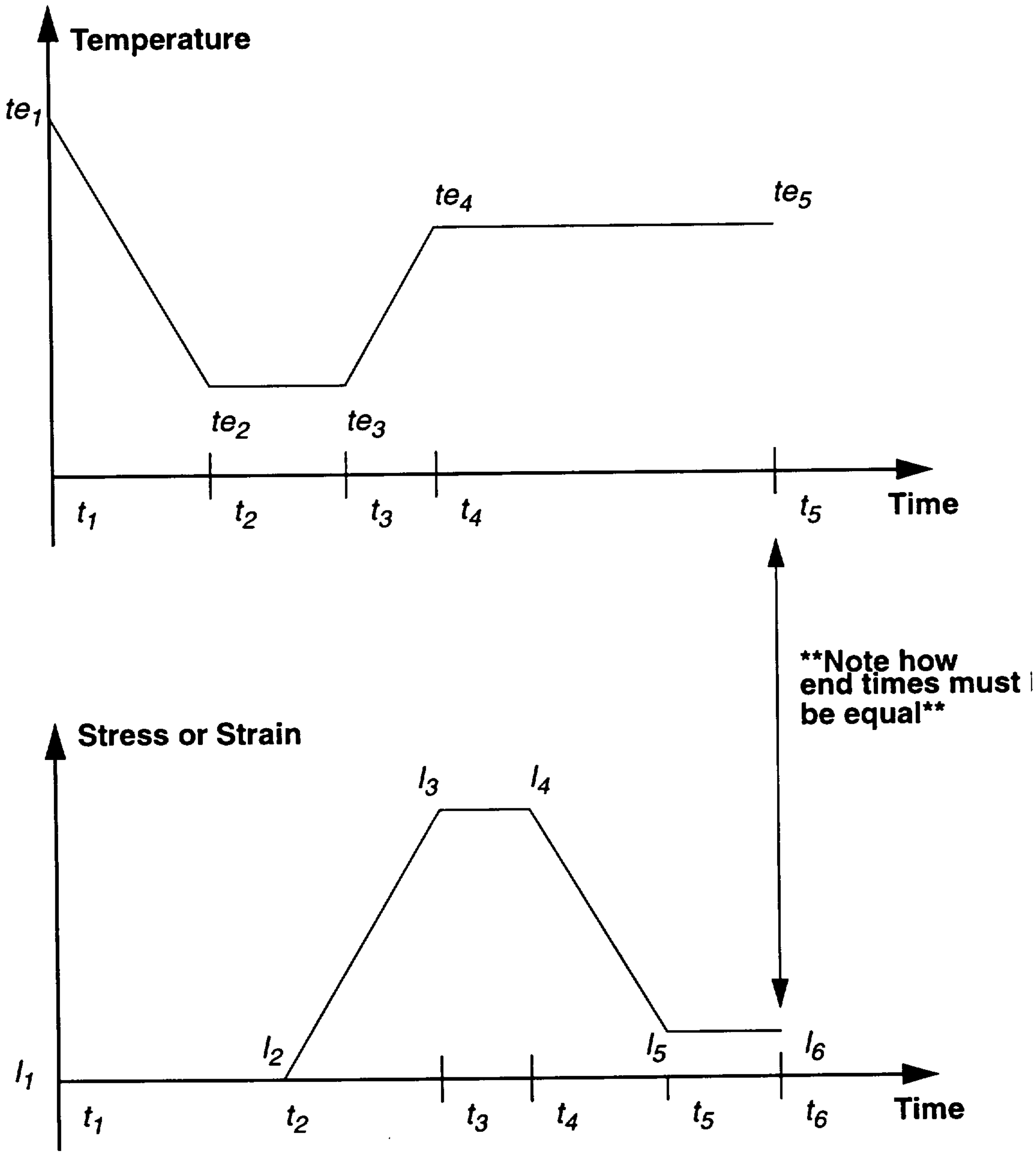


Figure 7: Load History Specification

4.2.6 Fatigue Damage Option:

Purpose: Activate the Fatigue Damage analysis option

☞ **Note:** Currently this option is **only available** for continuous reinforcements, modid=1 see section **4.2.8**

☞ **FEAMAC Note:** This option is currently **not available** from within ***FEAMAC***

***DAMAGE**
NCY=*ncycle* D=*deld* DMAX=*dmax* FG=*fg* FL=*fl*

where:

ncycle - number of "blocks" of cycles to be run (see Fig. 5)

deld - damage increment

dmax - maximum amount of damage allowed $(0 \leq dmax \leq 1)$

fg:

0 - no macro (global) failure check
1 - macro failure check

fl:

0 - no micro failure check
1 - micro failure check

☞ **Note:** For FG=1, the following data is required on a new line;

T=ss IC=comp V=val

Where

ss:
1 - stress
2 - strain

comp:
1 - component 11
2 - component 22
3 - component 33
4 - component 23
5 - component 13
6 - component 12

val - value of failure stress or strain (depending on value of *ss*)

☞ **Note:** For FL=1, see description of additional required data in section **4.2.12**

☞ **Note:** the actual time vs. load data for the "block" of cycles is specified using the previously defined ***MECH** input data (section **4.2.4**).

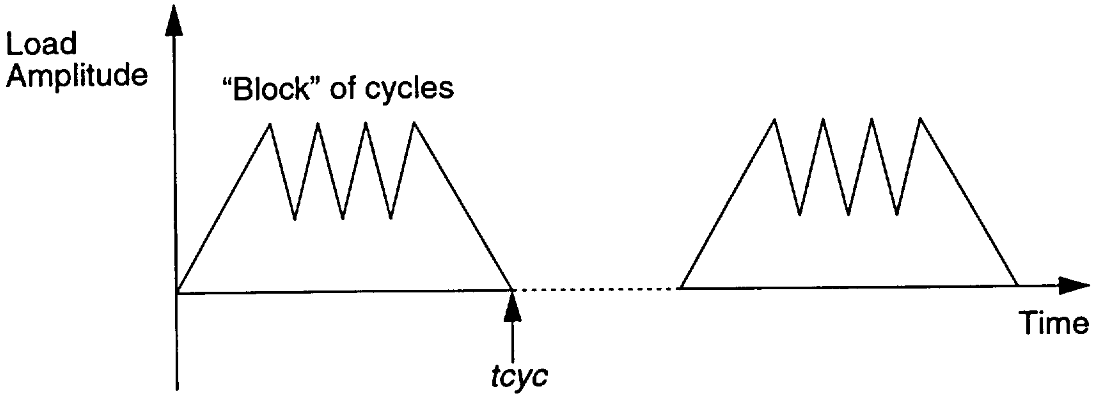


Figure 8

☞ **Note**: When the fatigue damage analysis option is activated the following additional files will be generated:

dam1.data	intermediate damage calculations
dam2.data	print-out of cycles to failure at the end of each load block
dam3.data	print-out of current damage in each subcell at the end of each load block

Example:

```
*DAMAGE
    NCY=2 D=0.25 DMAX=0.95 FG=1 FL=0
    T=1 IC=2 V=100.
```

☞ **Note**: In the above example:

- 2 "blocks" of cycles have been specified via NCY
- the damage increment is 0.25 (25%) with a maximum allowable amount of damage set to 0.95 (95%)
- a Global failure criteria is used: stress component -22 with a failure value of 100.

4.2.7 Yield and Damage Surface Generation Analysis

Purpose: Initiate probing history so as to generate flow/damage surface plots

☞ **Note:** Currently this option is **only available** for continuous reinforcements, modid=1 see section **4.2.8**

☞ **FEAMAC Note: All the data on the SURF card is ignored by FEAMAC; option is meaningless in the context of finite element.**

***SURF**
NPRE=*npre* ISP=*isp* IAN=*ian* C1=*c1* C2=*c2* C3=*c3* C4=*c4*

where:

npre - the number of preloading steps before probing for the yield surface begins (e.g., this is allows one to represent stress-free cooling)

isp - stress space
= 1 transverse-axial $(\sigma_{22} - \sigma_{11})$
= 2 transverse-transverse $(\sigma_{22} - \sigma_{33})$
= 3 shear-axial $(\sigma_{12} - \sigma_{11})$

ian - probe angle increment in degrees

"yield" criteria:

c1 - equivalent plastic strain, $\sqrt{\frac{2}{3}\Delta\varepsilon^{i}_{ij}\Delta\varepsilon^{i}_{ij}}$

c2 - Surface of Constant Dissipation Rate, SCDR, $\Sigma_{ij} \cdot \dot{\varepsilon}^{i}_{ij}$

c3 - Surface of Constant Inelastic Strain Rate, SCISR, $\dot{\varepsilon}^{i}_{ij} \cdot \dot{\varepsilon}^{i}_{ij}$

c4 - Surface of Constant Inelastic Power, SCIP, $\bar{\sigma} \cdot \Delta\bar{\varepsilon}^{i}$

☞ **Note:** Results from application of this option are described in: Lissenden C. J. and Arnold, S. M.; "Theoretical and Experimental Considerations in Representing Macroscale Flow/Damage Surfaces For Metal Matrix Composites", Int. Jnl. of Plasticity, Vol. 13, No. 4, pp. 327-358, 1997.

☞ **Note:** Upon fulfillment of each criteria (i.e., c1, c2, c3, and c4) the probe angle and stress vector are written to **output files** surf1.dat, surf2.dat, surf3.dat, and surf4.dat, respectively, for post processing by the user.

☞ **Note:** Probing continues until all four criteria are satisfied or the specified load history is completed for a given probe angle. Thus, to render a particular criterion inactive, use a value of 0 for the criterion so it is fulfilled immediately, thereby rendering the associated output meaningless. Also, it is suggested that a large load history be imposed to ensure yielding along a given probe angle occurs.

4.2.8 Micromechanics Model Identification:

Purpose: Select desired **GMC** micromechanics model

***MODEL**
MOD=*modid*

☞ **Note:** If modid=3 (laminate option) you must enter the following line of data,
MATSYS=*matsys* NLY=*nly* THK=$thk_1, thk_2, \ldots thk_{nly}$ CON=$c_1, c_2, \ldots c_{nly}$ SYS=$s_1, s_2, \ldots s_{nly}$
ANG=$a_1, a_2, \ldots a_{nly}$

☞ **FEAMAC Note:** Currently laminate option (modid=3) is not available within the finite element implementation.

%

Where

modid:	- defines micromechanics analysis 1 = double periodicity 2 = triple periodicity 3 = laminate option
matsys:	- number of different material systems in the analysis
nly:	*- number of layers in the laminate*
$thk_1, thk_2, \ldots thk_{nly}$:	- thickness of each layer
$c_1, c_2, \ldots c_{nly}$:	- layer material type id: 1 = isotropic 2 = anisotropic
$s_1, s_2, \ldots s_{nly}$:	- material system id number
$a_1, a_2, \ldots a_{nly}$:	- angles of each layer (in degrees)

Example 1: double periodicity model

```
*MODEL
 MOD=1 %
```

Example 2: laminate option with 1 material system

```
*MODEL
 MOD=3 MATSYS=1 NLY=4 THK=0.25,0.25,0.25,0.25 CON=2,2,2,2 &
 SYS=1,1,1,1 ANG=45.,-45.,-45.,45. %
```

4.2.9 PATRAN Output: (Optional)

Purpose: To enable generation of additional output files for use within the **MSC/PATRAN** graphical pre- and post-processing environment - when running the **MACPOST** program. For details on using **MACPOST**, see NASA TM 209062

☞ **Note: This is an optional input line. If output for use in the MACPOST program is not desired, do NOT include this data block. Inclusion of this data block will increase execution time significantly, particularly for complex RVE's, as microfield data for each subcell are written to additional output files.**

```
*PATRAN
   FN=prefix  TPRE=tpre  STP= npstp

%
```

Where:

prefix: - filename prefix that will be assigned to the geometry file for use in **MACPOST**.

tpre: - preloading time after which output to data files begins

npstp: - time step interval for output to the **MACPOST** data files (output occurs every npstp time steps).
CAUTION: a small value for npstp will cause large output files to be generated, which can result in significantly longer execution times.

Example 1: enable the generation of **MACPOST** output files using prefix “run1”

```
*PATRAN
   fn=run1  TPRE=57600  STP=300  %
```

☞ **Note:** If this option is chosen, **MAC/GMC** generates 14 additional output files for use in the **MACPOST** program within **MSC/PATRAN.** After the time reaches 57600 units, data is written every 300 time steps. The following 14 output files are generated:

prefix.macgeo: Contains unit cell (RVE) geometry data.

prefix.total_pat.data: Contains number of time steps and number of subcells data

prefix.macro1_pat.data: Macroscopic strain data at different times

prefix.macro2_pat.data: Macroscopic stress data at different times

prefix.macro3_pat.data: Macroscopic inelastic strain data at different times

prefix.macro4_pat.data: Macroscopic thermal strain, creep time, temperature, and stress invariant data at different times

prefix.micro1_pat.data: Microscopic (subcell) strain data at different times

prefix.micro2_pat.data: Microscopic (subcell) stress data at different times

prefix.micro3_pat.data: Microscopic (subcell) inelastic strain data at different times

prefix.micro4_pat.data: Microscopic (subcell) thermal strain, creep time, temperature, and stress invariant data at different times

prefix.micro1_pat.contour: Microscopic (subcell) strain data for contour plots

prefix.micro2_pat.contour: Microscopic (subcell) stress data for contour plots

prefix.micro3_pat.contour: Microscopic (subcell) inelastic strain data for contour plots

prefix.micro4_pat.contour: Microscopic (subcell) thermal strain, creep time, temperature, and stress invariant data for contour plots

See **Example S** for an example illustrating how to construct the required input deck that utilizes the **PATRAN** option and typical results.

☞ **Note:** All 14 files listed above must be present in the active directory used in **MSC/PATRAN** in order to use the **MACPOST** program. For more details on the content of the various **MACPOST** files, see NASA TM 209062.

☞ **FEAMAC Note:** Invoking this option within ***FEAMAC,*** will cause all **"_pat"** extensions in the above output files to be changed to ***"_feapat"***

4.2.10 Integrator Identification

Purpose: Select type of integration scheme

☞ **FEAMAC Note:** ***All the data on the *SOLVER card is ignored by FEAMAC***

***SOLVER**
NTF=*ntf* ... (additional data as specified below) %

If NTF=1, enter the following:
NTF= 1 NPTS=*npts* TIM= $t_1,t_2,...t_{npts}$ STP=$st_1,st_2,...st_{npts-1}$ %

If NTF=2, enter the following:
NTF=2 ISTM=*istpm* ISTT=*istpt* ERR=*errtol* %

Where:

ntf:
1 = Forward Euler method (specified time step)
2 = Predictor/Corrector method (self-adaptive time step)

npts: = number of time points

t_i : = start and end times for load increments

st_i : = time step for that load increment

istpm: = initial mechanical load time step

istpt: = initial thermal load time step

errtol: = error tolerance for predictor/corrector

☞ **Note: only *istpm* is required for a mechanical load (*nsel*=2, see section 4.2.3) , only *istpt* is required for a thermal load (*nsel*=1) and both *istpm* and *istpt* are required for a thermomechanical load (*nsel*=3)**

☞ **Note: *errtol* is only required when using predictor/corrector (*ntf* = 2) Suggested errtol = 0.1, if one sees oscillations in predicted response curve the errtol should be lowered.**

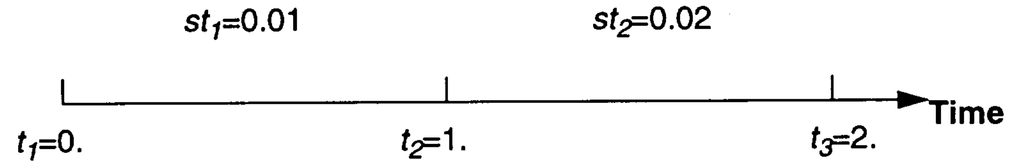
☞ **Note:** ***Laminate option and cases using the second debonding implementation must use forward euler.***

Example 1: for a mechanical load, select predictor/corrector, with initial time step = 0.001 and error tolerance = 0.01

```
*SOLVER
  NTF=2 ISTM=0.001   ERR=0.01%
```

Example 2: select Forward Euler, time step = 0.01 and 0.02

```
*SOLVER
  NTF=1 NPTS=3   TIM=0.,1.,2.   STP=0.01,0.02%
```



4.2.11 Thermal Conductivity Calculation: (Optional)

Purpose: Calculate the coefficients of thermal conductivity for the composite

***COND**

☞ **Note:** If the ***COND** keyword is not used, the thermal conductivity calculations are not performed.

☞ **Note:** To use this option, all materials must be user input material properties (mat=U, see section **4.2.12**) and must be at the same temperatures. **MAC/GMC** will then calculate the global thermal conductivities at the material property input temperatures, and write them to the output file, see **Example T**.

4.2.12 Constituent Material Model Identification:

Purpose: To select the model for the fiber and matrix constituents.

*FIBER
NFIBS=*nfibs*

The following new line is to be repeated for each fiber (nfibs):
NF=nf_f **MS=ms_f** MF=$ncmd_f$ NDPT=*dpt* ***TEMP*=mtemp** MAT=mat_f &
***IFM*=ifm** ***D*=d_1,d_2,d_3**

*MATRIX
NMATX=*nmatx*

The following new line is to be repeated for each matrix (nmatx):
NM=nm_m **MS=ms_m** MM=$ncmd_m$ NDPT=*dpt* ***TEMP*=mtemp** MAT=mat_m &
***IFM*=ifm** ***D*=d_1,d_2,d_3**

*MONOL (available for laminate option only)
NMON=*nmon*

The following new line is to be repeated for each monolithic material (nmon):
NMO=nmo_{iso} MS=ms_{iso} MMO=$ncmd_{iso}$ NDPT=*dpt* ***TEMP*=mtemp** &
MAT=mat_{iso} ***IFM*=ifm**

%

Where:

nfibs: - number of different fibers

nmatx: - number of different matrices

nmon: - number of different monolithic layers

nf_f: - fiber material designation number running from 1 to *nfibs*, sequentially.

nm_m: - matrix material designation number running from 1 to *nmatx*, sequentially.

nmo_{iso}: - monolithic material designation number running from 1 to *nmon*, sequentially.

msf: - fiber material system ID **(required only when using laminate option)**

ms_m: - matrix material system ID. **(required only when using laminate option)**

ms_{iso}: - monolithic material system ID

ncmd: - material model identifier for either fiber, matrix or monolithic layer:
1 = Bodner-Partom Model
2 = Modified Bodner-Partom Model
3 = Robinson Viscoplastic Model
4 = Generalized Viscoplastic Potential Structure (GVIPS) Model
6 = Transversely Isotropic Elastic Model (2-3 isotropic plane)
7 = Transversely Isotropic GVIPS Model (TGVIPS)
9 = Local Transversely Isotropic Elastic Model

99 = User defined model (see note at the end of section for special format instructions)

mat: - material identification letter for either fiber or matrix, selected from material database, see **Table II.**

☞ **Note: By specifying MAT=U allows the user to specify the material constants according to the formats specified below.**

dpt: - flag indicating whether material constants should be temperature independent or temperature dependent
1 = Temperature Independent
2 = Temperature Dependent

mtemp: - the constant temperature at which material properties are to be taken (**only required for dpt= 1 and when using database properties)**

ifm - flag indicating whether material properties will be read from input file or taken from a user defined function (provided in the USRFUN subroutine) **ONLY NEEDED when MAT=U.**
1 = read from input file
2 = functional form taken from USRFUN routine

$\mathbf{d_i}$: - direction vector defining the normal to the plane of local isotropy (**only required for ncmd = 3, 7 or 9)**

☞ **Note:** If **modid=1** (Double periodicity) and one desires transverse isotropy using **ncmd= 3,7 or 9** then the strong material direction **must** be specified in the 1-direction.

☞ **Note:** Additional Input is required if the ***DAMAGE** (section **4.2.6**) option is invoked

The following additional data is entered on a **new line**:

ANG=θ BN=b BP=b' OMU=ω_u OMFL=ω_{fl} OMM=ω_m ETU=η_u &
ETFL=η_{fl} ETM=η_m BE=β A=a SFL=σ_{fl} XML=*M* SU=σ_u SK=*sk* %

where:

sk:

= 1 skip fatigue damage calculations for this material
= 0 perform fatigue damage calculations

The remaining constants are described in section **3.5**

☞ **Note:** In addition, for FL=1 (micro failure criteria) in section **4.2.6**, the additional data is also required:

T=*ss* IC=*icomp* V=*val*

ss:
1 - stress
2 - strain

comp:
1 - component 11
2 - component 22
3 - component 33
4 - component 23
5 - component 13
6 - component 12

val - value of failure stress or strain (depending on value of *ss*)

Example 1: select 1 fiber, SCS-6, and 1 matrix material, TIMETAL 21S, both read from database; i.e., SCS-6/TIMETAL 21S composite system

```
*FIBER
   NFIB=1
   NF=1   MF=6   NDPT=1   TEMP = 23   MAT=D   %

*MATRIX
   NMATX=1
   NM=1   MM=4   NDPT=1   TEMP = 23   MAT=A   %
```

☞ **Note:** See **Section 3.3** for a mathematical description of each material model.

Example 2: (select 2 matrix materials; material 1: Boron, read from database, material 2: user supplied properties)

***MATRIX**
NMATX=2
NM=1 MM=6 NDPT=1 MAT=U IFM=1 EL=E_L, E_T, ν_A, ν_T, &
G_A, α_A, α_T VI= D_0, Z_0, Z_1, m, n,q K= κ_A, κ_T
NM=2 MM=1 NDPT=1 MAT=U IFM=1 EL=E_L, E_T, ν_A, ν_T, &
G_A, α_A, α_T VI= D_0, Z_0, Z_1, m, n,q K= κ_A, κ_T%

☞ **Note:** The κ_A, κ_T data is only required if the **✱COND** keyword has been used; where κ_A, κ_T are the axial and transverse thermal conductivities, respectively.

☞ **Note: Required Material constants for each model is as follows:**

Bodner-Partom: *ncmd* = 1

Elastic:
EL=E_L, E_T, ν_A, ν_T, G_A, α_A, α_T

Inelastic:
VI= D_0, Z_0, Z_1, m, n,q

Modified Bodner-Partom: *ncmd* = 2

Elastic:
EL=E_L, E_T, ν_A, ν_T , G_A, α_A, α_T

Inelastic:
VI= D_0, Z_0, Z_1,Z_2, Z_3, m_1 m_2, n, a_1, a_2, r_1, r_2, Dm_1, Dm_2

Robinson Viscoplastic: *ncmd* = 3

Elastic:
EL=E_L, E_T, ν_A, ν_T, G_A, α_A, α_T

Directions:
D=d_1, d_2, d_3

Inelastic:
VI= n, m, μ, κ_T, β, R, H, $\hat{G}_0$, η, ω

GVIPS: *ncmd* = 4

Elastic:
EL=E, ν,α

Inelastic:
VI= μ, κ, R_α, R_κ, B_0, B_1, L_0, L_1, m, n, p, q, w, z_0

Elastic Model: *ncmd* = 6
EL=E_L, E_T, ν_A, ν_T, G_A, α_A,α_T

TGVIPS: *ncmd* = 7

Elastic:
EL=E_L, E_T, ν_A, ν_T, G_A, α_A, α_T

Directions:
D=d_1,d_2,d_3

Inelastic:
VI= κ, n, μ, m, β, R, H, G_0, ω, η

Transversely Isotropic Elastic Model: *ncmd* = 9

Elastic
EL=E_L, E_T, ν_A, ν_T, G_A, α_A,α_T

Directions:
D=d_1,d_2,d_3

Table II. MAC/GMC Material Constant Database

Model	Material	Temperature Dependent ?	Units	*mat*
Bodner-Partom *ncmd* = 1 all properties taken from ref [1]	Aluminum (2024-T4)	Yes	Pa, sec,^{o}C	A
	Aluminum (2024-0)	Yes	Pa, sec,^{o}C	B
	Aluminum (6061-0a)	Yes	Pa, sec,^{o}C	C
	Aluminum (6061-0b)	Yes	Pa, sec,^{o}C	D
	Aluminum (pure)	Yes	Pa, sec,^{o}C	E
	Titanium (pure)	No	Pa, sec,^{o}C	F
	Copper (pure)	No	Pa, sec,^{o}C	G
Modified Bodner-Partom *ncmd* = 2	TIMETAL 21S	Yes	Pa, sec, ^{o}C	A
Robinson Visco-plastic *ncmd* = 3	Kanthal	No, 600^{o}C	ksi, hr,^{o}C	A
	FeCrAlY	Yes	ksi, hr, ^{o}C	B
	W/Kanthal (vf=35%)	No, 600^{o}C	ksi, hr, ^{o}C	C
GVIPS *ncmd* = 4	TIMETAL 21S	Yes	ksi, sec, ^{o}C	A
Linear Elastic *ncmd* = 6 All properties are assumed isotropic	Boron	No	Pa, ^{o}C	A
	SCS-6	Yes	Pa, ^{o}C	B
	Tungsten (W)	No	Pa, ^{o}C	C
	Boron	No	ksi, ^{o}C	D
	SCS-6	Yes	ksi, ^{o}C	E
	Tungsten (W)	No	ksi, ^{o}C	F
TGVIPS *ncmd* = 7	Ti-6-4	Yes	ksi, sec, ^{o}C	A
Linear Elastic *ncmd* = 9 Transversely Isotropic	T50 Graphite	No	Pa, ^{o}C	A
	T300 Graphite	No	Pa, ^{o}C	B
	P100 Graphite	Yes	Pa, ^{o}C	C
	T50 Graphite	No	ksi, ^{o}C	D
	T300 Graphite	No	ksi, ^{o}C	E
	P100 Graphite	Yes	ksi, ^{o}C	F

☞ **Note: Warning:** It is the user's responsibility to ensure that **consistent material property units** are being employed within a given problem. Particularly, when mixing database and user supplied material properties.

☞ **Note:** Even if a material model is temperature independent, it can still be used in a nonisothermal analysis (ndpt=2). Its properties will just not vary with temperature.

☞ **Note: Required Format for User Supplied Non-Isothermal Material Constants:** each of the following data statements are on separate lines.

NTP= *ntpts*

TEM= $T_1, T_2, \ldots, T_{ntpts}$

EA= $E_{AT_1}, E_{AT_2}, \ldots E_{AT_{ntpts}}$

ET= $E_{TT_1}, E_{TT_2}, \ldots E_{TT_{ntpts}}$

NUA= $\nu_{AT_1}, \nu_{AT_2}, \ldots \nu_{AT_{ntpts}}$

NUT= $\nu_{TT_1}, \nu_{TT_2}, \ldots \nu_{TT_{ntpts}}$

GA= $G_{AT_1}, G_{AT_2}, \ldots G_{AT_{ntpts}}$

ALPA= $\alpha_{AT_1}, \alpha_{AT_2}, \ldots \alpha_{AT_{ntpts}}$

ALPT= $\alpha_{TT_1}, \alpha_{TT_2}, \ldots \alpha_{TT_{ntpts}}$

V1= $V1_{T_1}, V1_{T_2}, \ldots V1_{T_{ntpts}}$

V2= $V2_{T_1}, V2_{T_2}, \ldots V2_{T_{ntpts}}$

⋮

VN= $VN_{T_1}, VN_{T_2}, \ldots VN_{T_{ntpts}}$

D = d_1, d_2, d_3

☞ **Note:** The total number of viscoplastic constants (V1, V2, V3, ... VN) required for each model are described on the bottom of page 71. **For ncmd=4, three extra lines (V15= κ_o V16= B_o' V17= β_κ) must be added**. Also, D= d_1, d_2, d_3 is only required when ncmd= 3, 7 or 9.

☞ Note: Format for User defined material model (ncmd=99):

Given **User Supplied Isothermal Material Constants**, the following special format is required:

***FIBER**
NFIBS=*nfibs*

The following line is to be repeated for each fiber (nfibs):
NF=nf_f MS=ms_f MF=99 *NDPT=dpt* NPE=*npe* EL=$e_1, e_2, \ldots e_{npe}$ &
ALP= α_A, α_T NPV=*npv* VI=$v_1, v_2, \ldots v_{npv}$ K=κ_A, κ_T %

***MATRIX**
NMATX=*nmatx*

The following line is to be repeated for each matrix (nmatx):
NM=nm_m MS=ms_m MM=99 *NDPT=dpt* NPE=*npe* EL=$e_1, e_2, \ldots e_{npe}$ &
ALP= α_A , α_T NPV=*npv* VI=$v_1, v_2, \ldots v_{npv}$ K=κ_A, κ_T %

Where:

npe: - total number of elastic constants (maximum of 9)

npv: - total number of inelastic constants (model specific, max of 19)

$e_1, e_2, ..$ - elastic constants

$v_1, v_2, \ldots$ - inelastic constants (model specific)

$\alpha_A, \alpha_T, \ldots$ - longitudinal and transverse thermal expansion coefficients

$\kappa_A, \kappa_T, \ldots$ - thermal conductivities (if ***COND** only)

OR given **User Supplied Non-Isothermal Material Constants:**

***FIBER**
NFIBS=*nfibs*

The following line is to be repeated for each fiber (nfibs):
NF=nf_f MS=ms_f MF=99 *NDPT=2* MAT= U IFM= 1
NPE=*npe* NPV=*npv*

with the following data statements immediately following each material declaration on a separate line.

NTP= *ntpts*

TEM= $T_1, T_2, \ldots, T_{ntpts}$

E1= $E_{1T_1}, E_{1T_2}, \ldots E_{1T_{ntpts}}$

E2= $E_{2T_.}, E_{2T_.}, \ldots E_{2T}$

⋮

Enpe= $E_{npeT_1}, E_{npeT_2}, \ldots E_{npeT_{ntpts}}$

ALPA= $\alpha_{AT_1}, \alpha_{AT_2}, \ldots \alpha_{AT_{ntpts}}$

ALPT= $\alpha_{TT_1}, \alpha_{TT_2}, \ldots \alpha_{TT_{ntpts}}$

V1= $V1_{T_1}, V1_{T_2}, \ldots V1_{T_{ntpts}}$

V2= $V2_{T_1}, V2_{T_2}, \ldots V2_{T_{ntpts}}$

$\vdots$

Vnpv= $Vnpv_{T_1}, Vnpv_{T_2}, \ldots Vnpv_{T_{ntpts}}$

KA= $\kappa_{AT_1}, \kappa_{AT_2}, \ldots \kappa_{AT_{ntpts}}$

KT= $\kappa_{TT_1}, \kappa_{TT_2}, \ldots \kappa_{TT_{ntpts}}$

***MATRIX**

NMATX=*nmatx*

The following line is to be repeated for each matrix (nmatx):
NM=nm_m MS=ms_m MM=99 *NDPT=2* MAT=U IFM = 1
NPE=*npe* NPV=*npv*

with the following data statements immediately following each material declaration on a separate line.

NTP= *ntpts*

TEM= $T_1, T_2, \ldots, T_{ntpts}$

E1= $E_{1T_1}, E_{1T_2}, \ldots E_{1T_{ntpts}}$

E2= $E_{2T_1}, E_{2T_2}, \ldots E_{2T_{ntpts}}$

$\vdots$

Enpe= $E_{npeT_1}, E_{npeT_2}, \ldots E_{npeT_{ntpts}}$

ALPA= $\alpha_{AT_1}, \alpha_{AT_2}, \ldots \alpha_{AT_{ntpts}}$

ALPT= $\alpha_{TT_1}, \alpha_{TT_2}, \ldots \alpha_{TT_{ntpts}}$

V1= $V1_{T_1}, V1_{T_2}, \ldots V1_{T_{ntpts}}$

V2= $V2_{T_1}, V2_{T_2}, \ldots V2_{T_{ntpts}}$

$\vdots$

Vnpv= $Vnpv_{T_1}, Vnpv_{T_2}, \ldots Vnpv_{T_{ntpts}}$

KA= $\kappa_{AT_1}, \kappa_{AT_2}, \ldots \kappa_{AT_{ntpts}}$

KT= $\kappa_{TT_1}, \kappa_{TT_2}, \ldots \kappa_{TT_{ntpts}}$

☞ **Note:** Required material user subroutines; see **Example K** for details on how they integrate together

USRFUN: - Routine that defines functional form for material properties
Required when: ifm=2 and mat=U

USRFORMDE*:* - Routine that forms the stiffness matrix. Required when ncmd = 99

USRCPEVAL: - Routine that defines the time derivative of the stiffness matrix. Required when:
a) **ifm=2 and mat=U**
b) **ncmd=99 AND ifm=1 AND ndpt=2**

4.2.13 RVE Data:

Purpose: Select RVE representing desired fiber packing arrangement/architecture.

✳MRVE
IDP=*idp*

For Double Periodic RVE (i.e., MOD=1):

☞ **Note:** The following data is entered on the same line as IDP (except where noted)

Without Interface:

- For IDP = 0
  no further data required **(only used for a monolithic layer in a laminate)**
- For IDP = 1, 2,or 3
  VF=*vf*
- For IDP = 4
  VF=*vf* XA=*xa*
- For IDP = 6
  VF=*vf* R=*R*
- For IDP = 7
  VF=*vf* R=*R*
- For IDP = 9
  VF1=*vf1* RAD1=*rad1* VF2=*vf2* RAD2=*rad2* R=*R*
- For IDP = 11
  VF=*vf* RAD=*rad1* R=*R*
- For IDP = 13
  VF=*vf* R=*R*

With Interface:

- For IDP = 1, 2,or 3
  VF=*vf* RAD=*rad1* CPER=*cper*
- For IDP = 4
  ☞ **Currently NOT Available**
- For IDP = 6
  VF=*vf* R= *R* CPER=*cper*
- For IDP = 7
  ☞ **Currently NOT Available**

- For IDP = 9
  VF1=*vf1* RAD1=*rad1* CPER1=*cper1* VF2=*vf2* &
  RAD2=*rad2* CPER2=*cper2* R=*R*
- IDP = 11
  VF=*vf* RAD=*rad1* R=*R* CPER=*cper*
- For IDP = 13
  ☞ **Currently NOT Available**

%

where:

vf, vf1, vf2 = the fiber volume ratios

rad, rad1, rad2 = fiber radii

cper, cper1, cper2 = ratios of interface thickness to fiber radius

xa = length of the cross, see Fig. 10.

R = X/Y which defines the ratio of distances between fiber centers within a "ply" and those between a "ply" (see Fig. 11)

☞ **Note:** For the laminate option the RVE data has the following input format:

✳MRVE
IDP=idp_{l1}, idp_{l2}, ... idp_{nly}
L=1 VF= ... (rve data for layer 1)
L=2 VF= ... (rve data for layer 2)
:
L=*nly* VF= ... (rve data for layer *nly*)

%

Where:

idp:- unique identifying number of each **Double Periodic** RVE as given below

idp	Description
• 1 =	Square Fiber, Square Pack (original 4-cell model) $V_f \le 1/(1+\Delta)^2$ RVE shown in Fig. 9

idp — Description

- 2 = Square Fiber, Triangular (hexagonal) Pack

$$V_f \leq 0.86602/(1+\Delta)^2$$

RVE shown in Fig. 9

- 3 = Square Fiber, Square Diagonal Pack;

$$V_f \leq 0.5/(1+\Delta)^2$$

RVE shown in Fig. 9

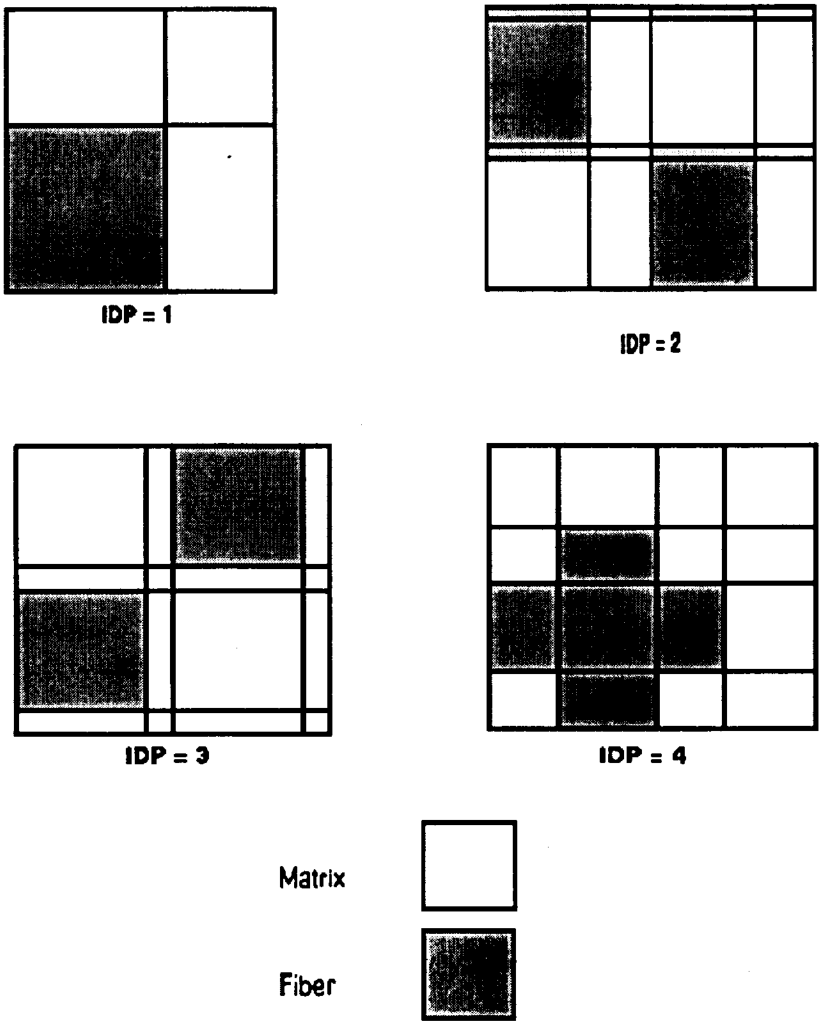


Figure 9: RVE's Available in **MAC/GMC**

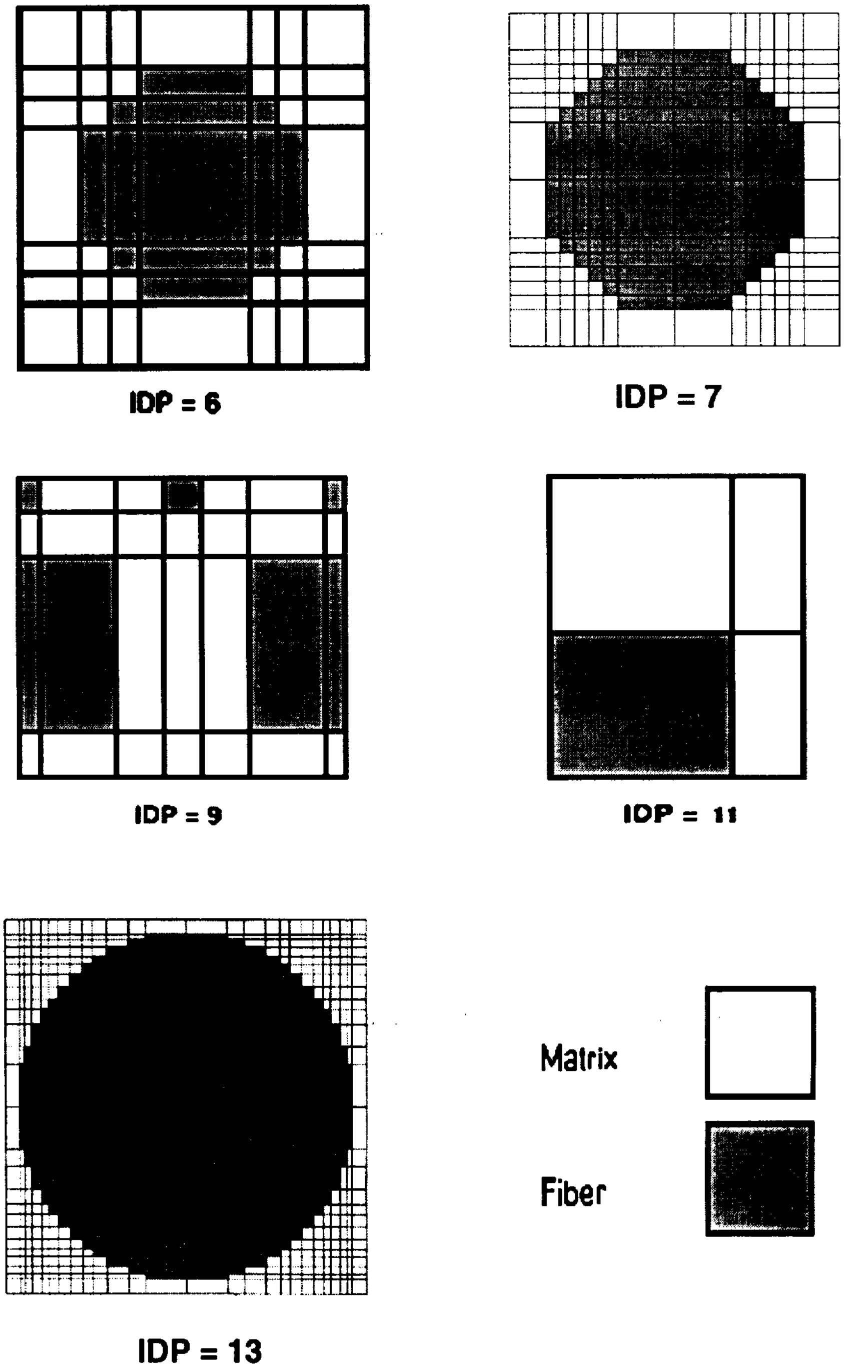


Figure 9 con't: RVE's Available in **MAC/GMC** idp — Description

- 4 = Cross Shaped Fiber, Square Pack

$$V_f \le 1 - 4(xa)^2$$

RVE shown in Fig. 9

- 6 = 7x7 Circular Fiber Approximation Rectangular or Square Pack

$$V_f \le \frac{0.8125}{R(1+\Delta)^2} \qquad if \qquad R > 1.0$$

$$V_f \le \frac{R(0.8125)}{(1+\Delta)^2} \qquad if \qquad R < 1.0$$

RVE shown in Fig. 9

- 7= 14x14 Circular Fiber Approximation Rectangular or Square Pack

$$V_f \le \frac{0.8148}{R} \qquad if \qquad R > 1.0$$

$$V_f \le R(0.8148) \qquad if \qquad R < 1.0$$

RVE shown in Fig. 9

- 9 = Two Different Size Square Fibers, Rectangular or Square Pack

$$V_{f_2} \le \frac{1}{R\left[(1+\Delta_2)+(1+\Delta_1)\frac{R_{f_1}}{R_{f_2}}\right]^2}$$

$$V_{f_1} \le \frac{2\sqrt{RV_{f_2}}}{\left[\left(\frac{R_{f_2}}{R_{f_1}}\right)^2(1+\Delta_2)+\left(\frac{R_{f_2}}{R_{f_1}}\right)(1+\Delta_1)\right]}$$

$$\frac{V_{f_1}}{V_{f_2}}\left[\frac{R_{f_1}}{R_{f_2}}\right]^2 \ge 1$$

RVE shown in Fig. 9

☞ **Note:** Two fibers and two interfaces must be defined in *Fiber and *Interface.

☞ **Note:** V_{f_1} will be altered, if necessary, such that an integer number of small fibers fit within the RVE.

idp	Description
• 11 =	Square Fiber, Rectangular Pack

$$V_f \le \frac{1}{R(1+\Delta)^2} \qquad if \qquad R > 1.0$$

$$V_f \le \frac{R}{(1+\Delta)^2} \qquad if \qquad R < 1.0$$

RVE shown in Fig. 9

• 13 =	26 x 26 Circular Fiber, Rectangular or Square Pack

$$V_f \le \frac{0.80613}{R} \qquad if \qquad R > 1.0$$

$$V_f \le 0.80613R \qquad if \qquad R < 1.0$$

RVE shown in Fig. 9

☞ **Note:** R_{f_1} = rad1, R_{f_2} = rad2 and if no interface is present, Δ = *cper*, Δ_1 = *cper1* and Δ_2 = *cper2* should be taken to be zero, in the above formulas.

• 99 =	User Defined RVE Example of RVE representing random packing shown in Fig. 12. Required input shown in **Example I.**

Example: Triangular packing with interface thickness 1% of fiber radius.

```
*MRVE
   IDP=2   VF=35   RAD=0.07   CPER=0.01 %
```

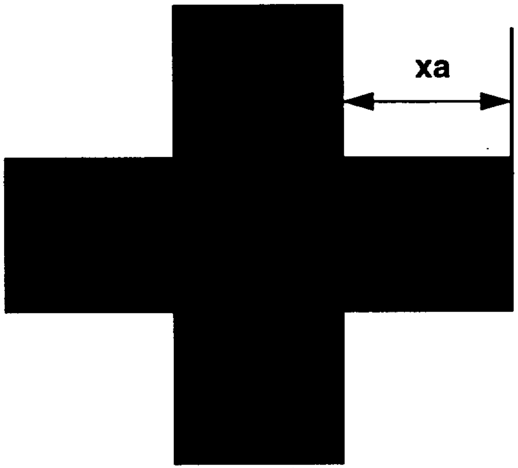


Figure 10: Cross Shaped Fiber Distance xa

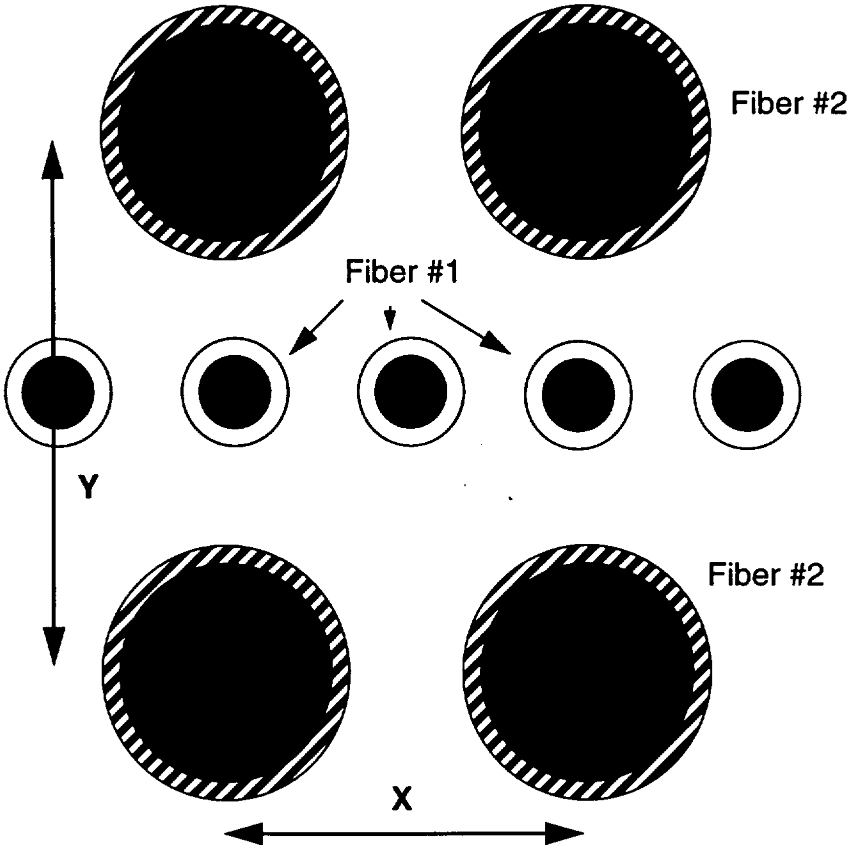


Figure 11: Hybrid Composite RVE; IDP = 9. Large Fiber Spacing Ratio, R=X/Y

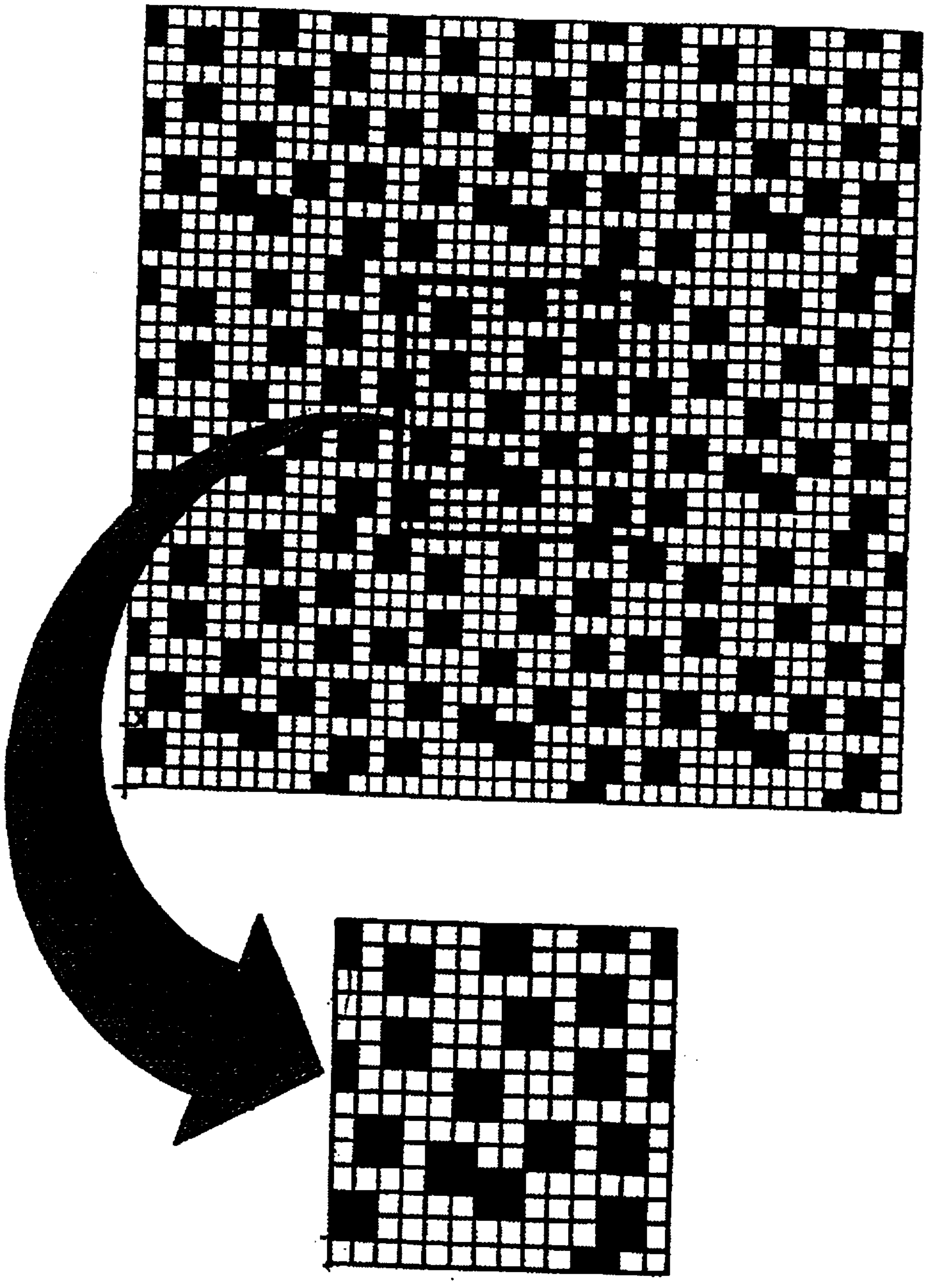


Figure 12: User defined RVE (see **Example I**)

For Triple Periodic RVE (i.e., MOD=2):

☞ **Note:** The following data is entered on the same line as the IDP data, except where noted.

Thus for

- For IDP = 1
  VF=*vf* *ASP=asp*
- For IDP = 2
  VF=*vf* *ASP1=asp1* *ASP2=asp2*
- For IDP = 3
  VF=*vf* *ASP1=asp1* *ASP2=asp2* *DR=dr*

where,

asp,asp1 - *are the aspect ratio of the short fiber (i.e., fiber length/fiber diameter; e.g. in IDP=3 this is* $(d_1 + d_2 + d_3)/h_1$*)*

asp2 - is the aspect ratio of the unit cell , d/h (see Fig. 2)

dr - is the ratio d_1/d_3 which quantifies the fiber off-set (see Fig. 13)

- For IDP = 4
  One of the following lines should be entered after the IDP data depending upon the option desired
  OPT=1 A=a B=b C=c D=d H=h L=l %
  OPT=2 VF=vf A=a B=b C=c RD=rd RL=rl %
  OPT=3 VF=vf RA=ra RC=rc D=d H=h L=l %
  OPT=4 VF=vf A=a B=b C=c RD=rd RLC= rlc %

where,
a,b, and c are the ellipsoid's semi-major axes (see Fig. 13)
d, h and l are the unit cell dimensions (see Fig. 13)
rd = aspect ratio of d/h
rl = aspect ratio of l/h
ra = aspect ratio of a/b
rc = aspect ratio of c/b
rlc = aspect ratio of l/(2c)

- For IDP = 10
  VF=*vf*

Where

idp:- unique identifying number of each **Triple Periodic** RVE given Fig. 13

idp Description

- 1 = Short fibers in Square array, with equal spacing in all directions (i.e., $d_2 = h_2 = l_2$ see Fig. 13)

$$0 \le V_f \le 1$$

RVE shown in Fig. 13

- 2= Short fibers in Diagonal array, with variable inclusion spacing in the x_1 - direction

$$V_f < \frac{1}{2}\left(\frac{Asp1}{Asp2}\right)\Big) \qquad if \qquad Asp1 < Asp2$$

$$V_f < \frac{1}{2}\left(\frac{Asp2}{Asp1}\right)^2 \qquad if \qquad Asp1 > Asp2$$

RVE shown in Fig. 13

- 3= Off-set short fibers in square array, variable fiber spacing in the x_1 - direction --- **NOTE Asp1 > Asp2**

$$V_f < 4\left(\frac{Asp1}{Asp2}\right)^2 \qquad if \qquad \frac{Asp1}{Asp2} > 2$$

$$V_f < \frac{1}{2}\left(\frac{Asp2}{Asp1}\right) \qquad if \qquad \frac{Asp1}{Asp2} < 2$$

$$V_f > \frac{1}{2}\left(\frac{Asp2}{Asp1}\right)^2$$

RVE shown in Fig. 13

- 4= Ellipsoidal Inclusion

☞ **Note:** Analytical expressions for Vf limits are unavailable as forming this unit cell geometry requires solution of non-linear equations.

☞ **Note:** Use of non-physical ellipsoid unit cell dimensions (e.g., 2b > h) will cause execution of **MAC/GMC** to stop.

- 10= Open cell

$$0 \le V_f \le 1$$

☞ **Note:** Solid material (see Fig. 13) is the matrix material specified in *MATRIX. whereas, the "open" material is the fiber material specified in *FIBER. To simulate a truly "open cell" material, specify a fiber material with the properties of air, thus fiber volume fraction becomes the void fraction

- 99= User defined RVE

☞ **Note: The following is the format for the user defined RVE's, IDP = 99 (see Fig. 12 for an example).**

1) **For 2-D RVE**: (Each line of data must be on a separate line)

NB=nb NG=ng
H=$h_1, h_2, ... h_{ng}$
L=$l_1, l_2, ... l_{nb}$
CM=$ss_{nb,1}, ss_{nb,2}, ... ss_{nb,ng}$
repeat data for the (nb x ng) **2-D RVE**
CM=$ss_{1,1}, ss_{1,2}, ... ss_{1,ng}$

2) **For 3-D RVE**: (Each line of data must be on a separate line)

NA=na NB=nb NG=ng
D=$d_1, d_2, ... d_{na}$
H=$h_1, h_2, ... h_{ng}$
L=$l_1, l_2, ... l_{nb}$
CM=$ss_{na,1,1}, ss_{na,2,1}, ... ss_{na,nb,,1}$
CM=$ss_{na-1,1,1}, ss_{na-1,2,1}, ... ss_{na-1,nb,,1}$
repeat data for the (na x nb x ng) **3-D RVE**
CM=$ss_{1,1,ng}, ss_{1,2,ng}, ... ss_{1,nb,ng}$

Where:

$ss_{i,j}$ and $ss_{i,j,k}$ - are the identifying material labels which are given in the following format:

first character: F - for a fiber or M - for matrix
second character: 1, 2, 3, ... for fiber/matrix number 1, 2, 3 ...

na - number of subcells in the alpha direction

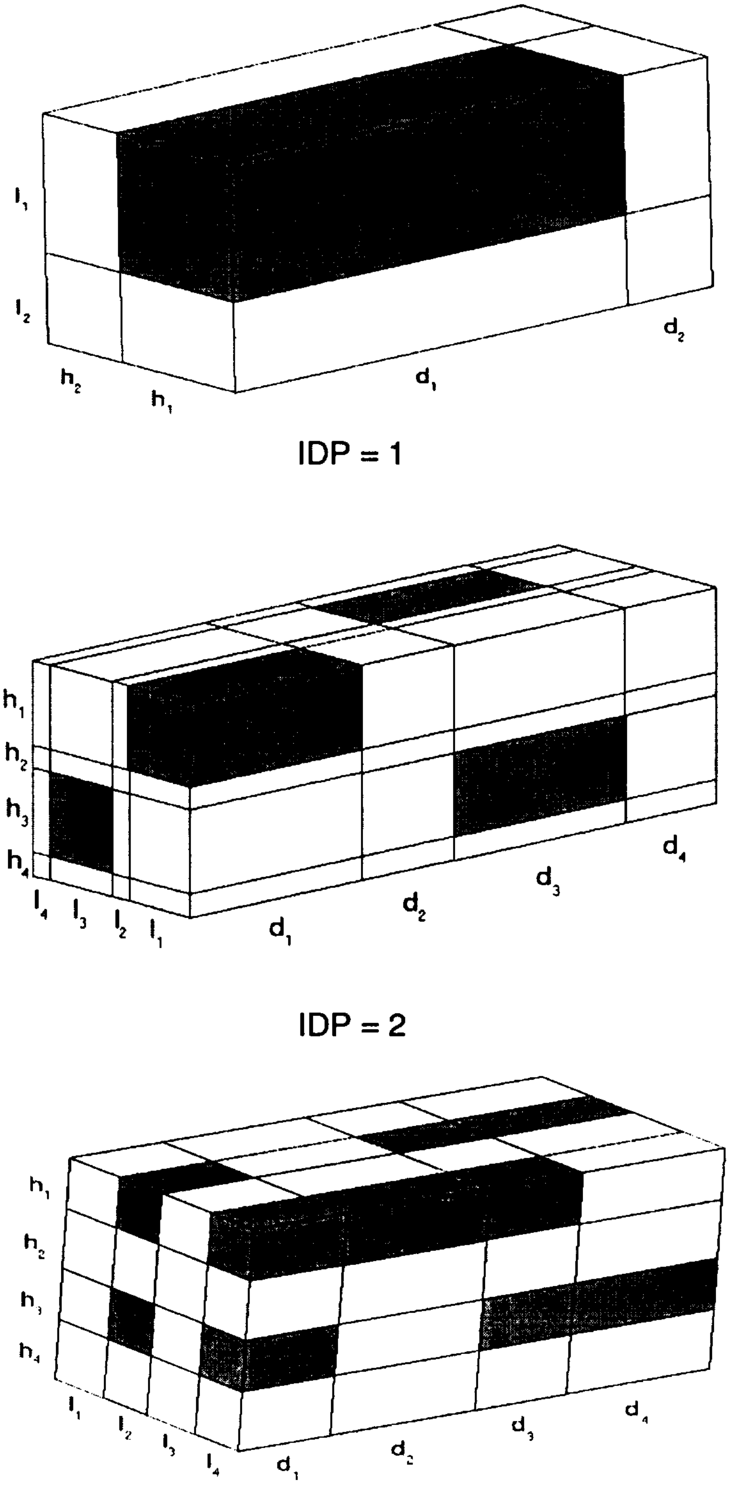
nb - number of subcells in the beta direction

ng - number of subcells in the gamma direction

h - the height of each subcell (x_2-dimension)

l - the length of each subcell (x_3- dimension)

d - the depth of each subcell (x_1- dimension)

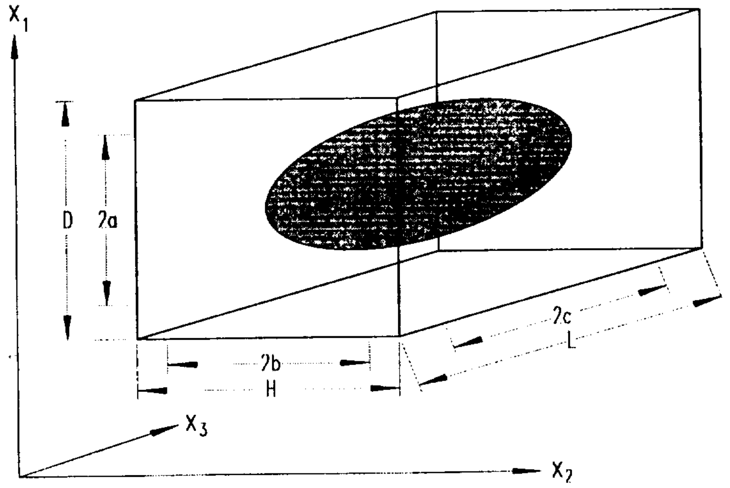


IDP = 1

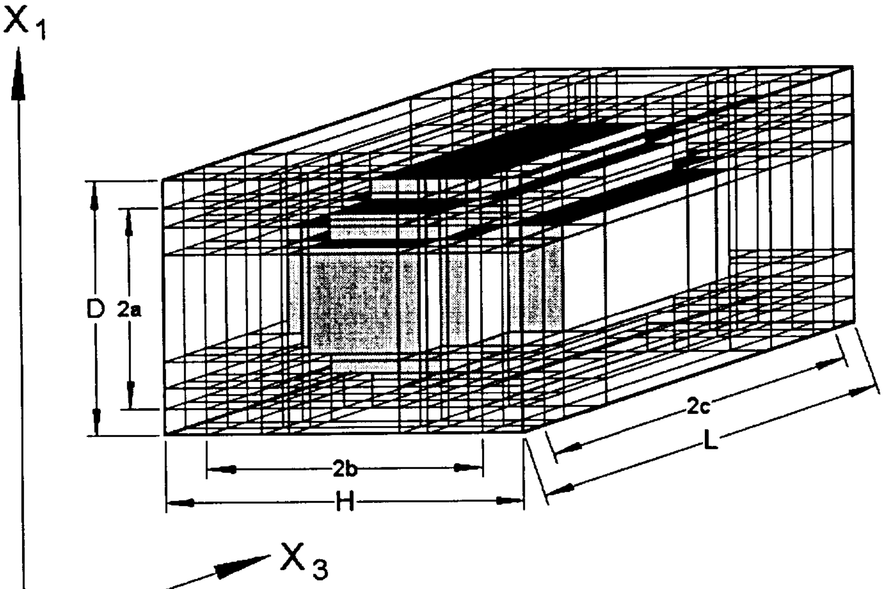
IDP = 2

IDP = 3

Figure 13 Available 3-D RVE's

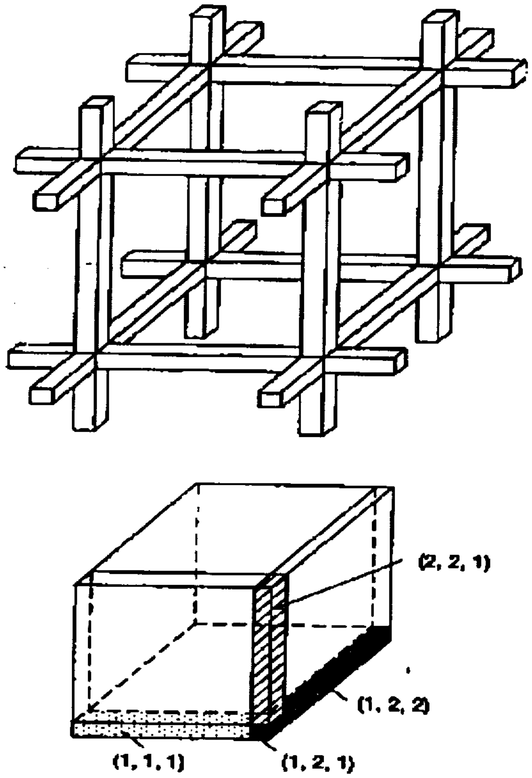


Ellipsoidal Inclusion



Ellipsoidal Inclusion RVE (IDP = 4)

Figure 13 con't: Available 3-D RVE's



IDP=10 - Open cell

Figure 13 con't: Available 3-D RVE's

☞ **Note:** A 2-D (continuous reinforcement) and 3-D (discontinuous reinforcement) example are described next with a detailed example of the user defined inputs being found in **Example I,** 2-D case, and in **Examples H** and **N**, for the 3-D case.

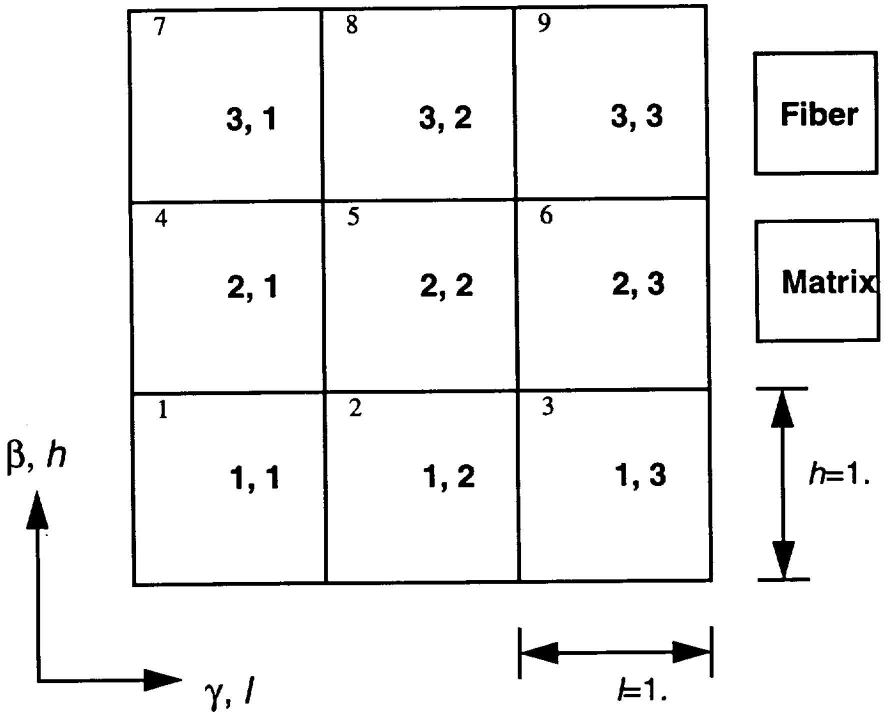


Figure 14

Example: 2-D user defined RVE (as shown above)

```
*MRVE
  IDP=99
  NB=3   NG=3
  H=1.,1.,1.
  L=1.,1.,1.
  CM=M1,M1,M1
  CM=M1,M1,F1
  CM=F1,M1,M1   %
```

☞ **Note:** the fiber ID is F1 and the matrix ID is M1

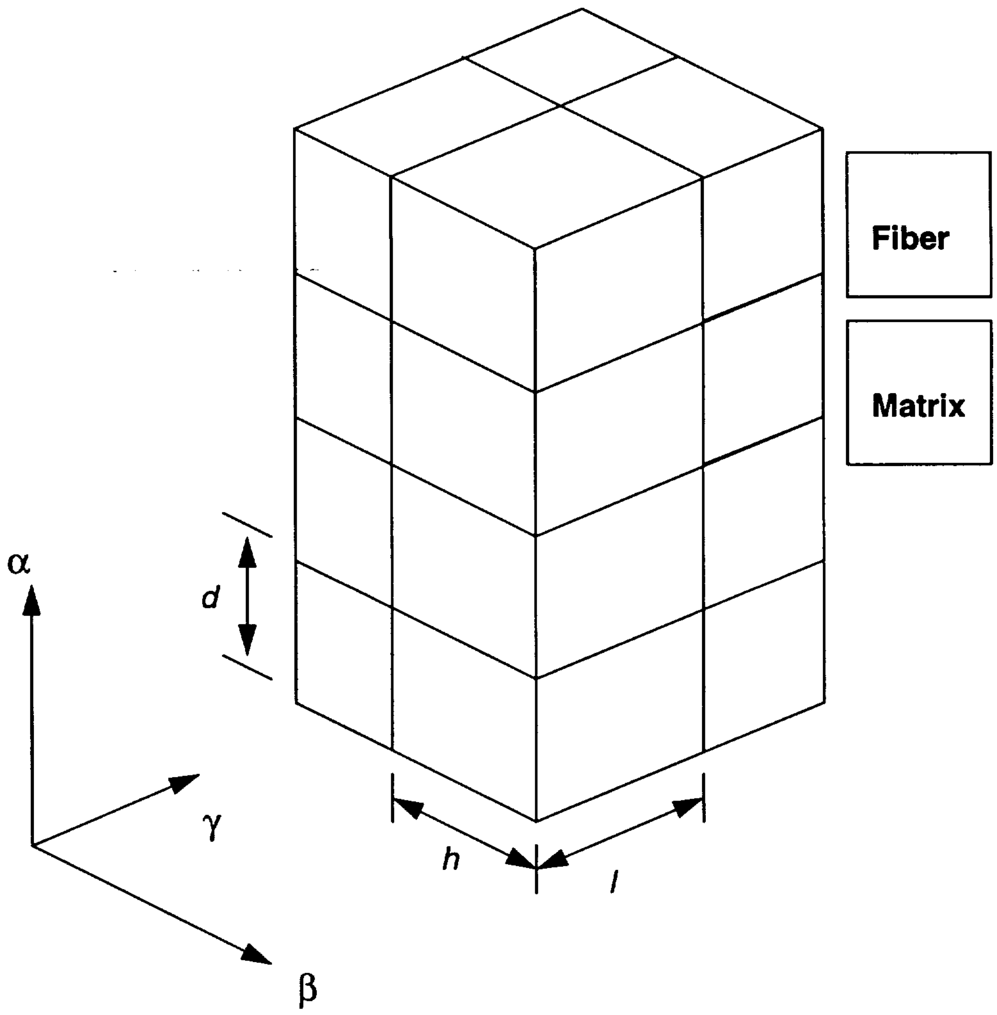
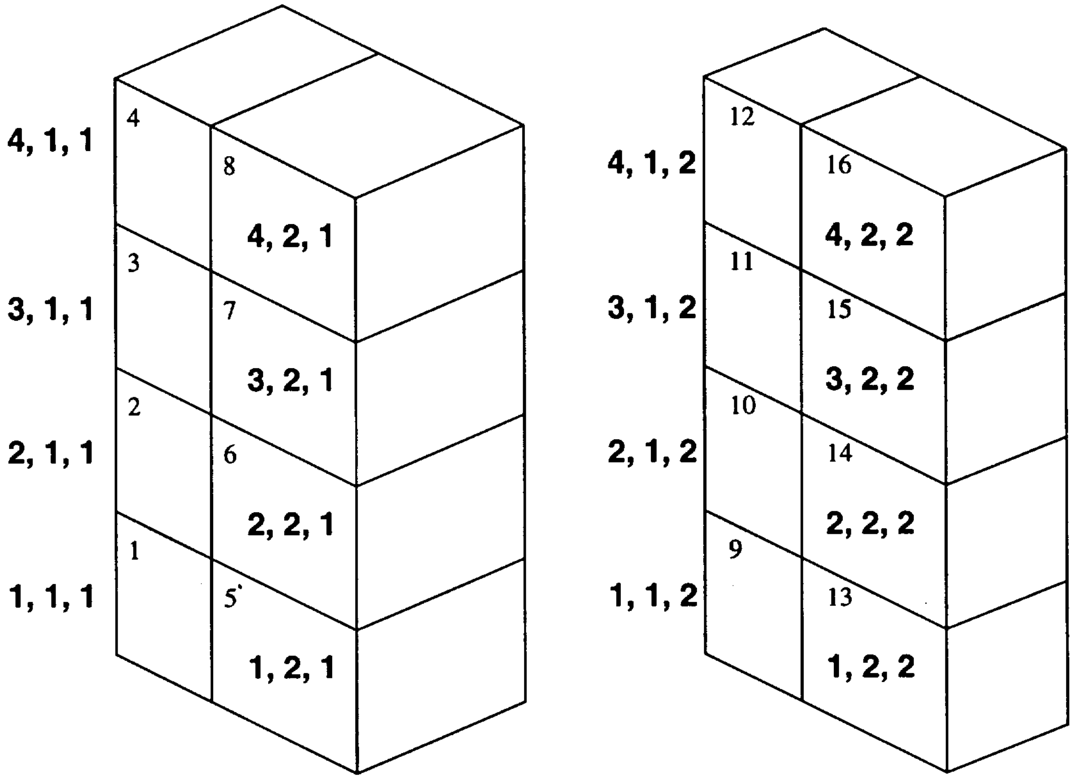


Figure 15

Example: A 3-D User defined RVE (as shown above), representing a [0/90] continuous reinforced laminate composite.

☞ **Note:** The fiber is material number F1 and the matrix is material number M1.



cross-section 1 cross-section 2

Figure 16

```
*MRVE
   IDP=99   ASP=1.
   NA=4  NB=2  NG=2
   D=1.,1.,1.,1.
   H=1.,1.
   L=1.,1.
   CM=M1,M1                    rows 1-4 are for cross-section 1 (γ=1)
   CM=M1,M1
   CM=M1,M1
   CM=F1,M1
   CM=M1,M1                    rows 5-8 are for cross-section 2 (γ=2)
   CM=F1,F1
   CM=M1,M1
   CM=F1,M1   %
```

<u>4.2.14 Interface Data</u>: (Optional, for IDP=1, 2, 3, 6 or 11)

Purpose: Specify interface layer properties

***INTERFACE**
NINT=*nint*

The following line is to be repeated for each interface(nint):
NI=nm_i MS=ms_i MI=$ncmd_i$ NDPT=*dpt TEMP=mtemp* MAT=mat_i *IFM=*ifm_i *D=*d_1,d_2,d_3

%

☞ **Note:** If MAT= U, see section **4.2.12** for format of additional input required.

Where

nint: - number of different interfaces

nm_i: - *i^{th} interface number*

ms_i: - matrix material system ID. **(required only when using laminate option)**

$ncmd_i$: - material model identifier for the i^{th} interface

$mat._i$ - material id letter for the i^{th} interface

dpt: - flag indicating whether material constants should be temperature independent or temperature dependent
1 = Temperature Independent
2 = Temperature Dependent

mtemp: - the constant temperature at which material properties are to be taken **(only required for dpt= 1 and when using database properties)**

ifm - flag indicating whether material properties will be read from input file or taken from a user defined function (provided in the USRFUN subroutine) **ONLY NEEDED when MAT=U.**
1 = read from input file
2 = functional form taken from USRFUN routine

d_i: - direction vector defining the normal to the plane of local isotropy **(only required for ncmd = 3, 7 or 9)**

☞ **Note:** Please refer to section **4.2.12** for *ncmd* and *mat* values

Example: (Assuming one interface with user supplied material constants)

```
*INTERFACE
 NINT=1
 NI=1  MI=4  NDPT=1  MAT=U  IFM=1 &
 EL=11700.,0.365,1.0E-06  &
 VI=0.8E-8,0.1,0.1E-5,0.,0.85E-3,0.05,1.,1.,1.,3.3,1.8,1.35,1.,0.01  %
```

<u>4.2.15 Debond Data:</u> (Optional)

Purpose: Specify the subcell faces where debonding can take place (see section **3.8**)

<u>2-D **GMC** Model Format</u>

∗DEBOND
NII=*nii*

# Option 1

DBCH= 1 NBI=*nbi* NGI=*ngi* FACE=*nfc* RN=*rn* BDN=*bdstrn* RS=*rs* &
BDS=*bdstrs* TI=*tmd*
OR
# Option 2
DBCH= 2 NBI=*nbi* NGI=*ngi* FACE=*nfc* BDN=*bdstrn* GCN=*gamman* &
BCN= *betan* TOLN=*toln* BDS= *bdstrs* GCS=*gammas* BCS=*betas* TI=*tmd*

⋮

repeat nii times

%

<u>3-D **GMC** Model Format</u>

∗DEBOND
NII=*nii*

# Option 1

DBCH= 1 NAI=*nai* NBI=*nbi* NGI=*ngi* FACE=*nfc* RN=*rn* BDN=*bdstrn &*
RS=*rs* BDS=*bdstrs* TI=*tmd*
OR
# Option 2
DBCH= 2 NAI=*nai* NBI=*nbi* NGI=*ngi* FACE=*nfc* BDN=*bdstrn &*
GCN=*gamman* BCN= *betan* TOLN=*toln* BDS= *bdstrs* GCS=*gammas &*
BCS=*betas* TI=*tmd*

%

<u>Laminate Model Format</u>

☞ **Note:** Currently this is **only available** for continuous reinforcements, modid=1 see section **4.2.8**

∗DEBOND
L=1 NII=*nii* (data for layer 1)
see 2-D GMC format above --- for required input here
L=2 Nii=nii (data for layer 2)
see 2-D GMC format above --- for required input here
:
L=*nly* Nii=*nii* (data for layer *nly*)
see 2-D GMC format above --- for required input here
%

Where:

nii: - number of subcell interfaces with debonding

dbch: - debond model implementation choice, =1 (for first) or = 2 (for second), see section **3.8**

nai, nbi and *ngi*: - subcell indices (α, β, γ)

nfc: - used specify which "face" of the subcell is to be debonded (See Fig. 17 and 18)

<u>For 2-D GMC</u>
= 1 Top Face
= 2 Right Face
= 3 Both Faces

<u>For 3-D GMC</u>
= 1 Top Face
= 2 Right Face
= 3 Back Face
= 4 Top & Right Face
= 5 Top & Back Face
= 6 Right & Back Face
= 7 ALL Faces

rn: - R_n value (see section **3.8**)

bdstrn: - debond stress, normal component

rs: - R_s value (see section **3.8**)

bdstrs: - debond stress, shear component

tmd: - time at which debond criteria becomes active (defaults to *tmd*=0. if not specified)

gamman: - Λ_n value (see section **3.8**)

gammas: - Λ_s value (see section **3.8**)

betan: - B_n value (see section **3.8**)

betas: - B_s value (see section **3.8**)

toln: - Normal stress reversal tolerance for unloading (i.e., stress value that is low enough to treat as zero stress - interface has closed)

☞ **Note:** **For the Laminate Model Only:** NII=0 specifies NO debonding for that layer and no additional debond data is required.

☞ **Note:** All debond parameters must be calibrated for a given composite system. **Example M** provides some initial estimates for an SCS-6/Ti system.

Example 1:

allow debonding of subcell 1,1 top face normal component starting at time = 11

```
NII=1
DBCH= 1 NBI=1 NGI=1 FACE=1 RN=1. BDN=100. RS=0. BDS=0. TI= 11.
```

Example 2:

allow debonding of subcell 1,1 right face shear component starting at time = 15

```
NII=1
DBCH= 1 NBI=1 NGI=1 FACE=2 RN=0. BDN=0. RS=1. BDS=100. TI= 15.
```

Example 3: Three Layer Laminate Analysis

debond subcell 1,1 right face shear component for layer 2 only

```
L=1 NII=0                (no debonding specified)
L=2 NII=1
DBCH=1 NBI=1 NGI=1 FACE=2 RS=1. BDN=100.
L=3 NII=0                (no debonding specified)
```

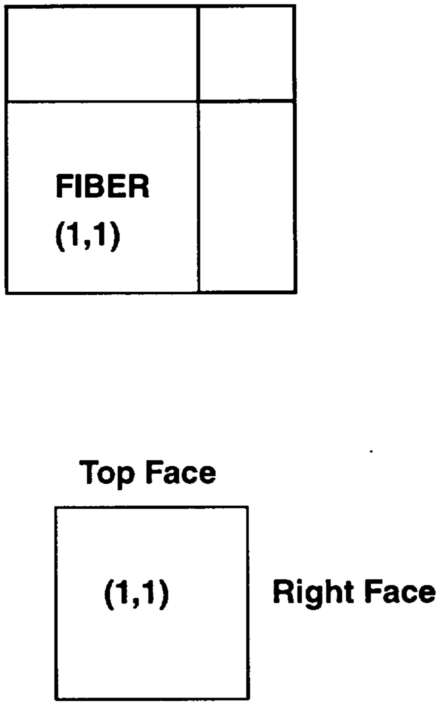


Figure 17: Subcell Faces For 2-D **GMC**

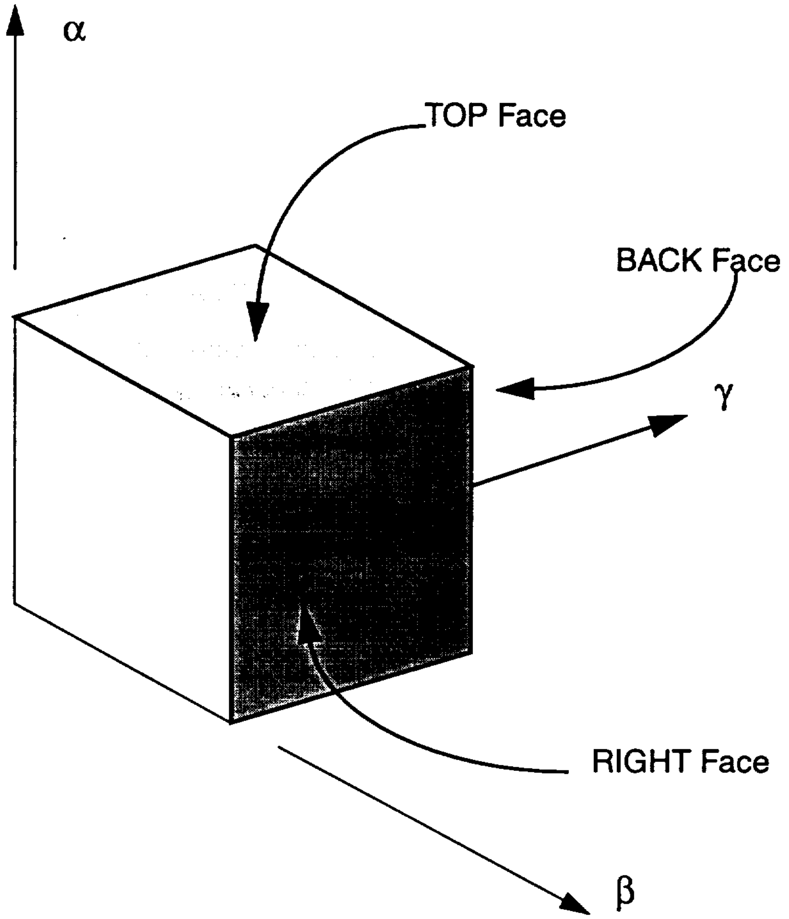


Figure 18: Subcell Faces For 3-D **GMC**

4.2.16 Plot Point Information:

Purpose: Specify the frequency at which data will be written to output files for both X-Y data

```
*CURVE
  NP=npmax

%
```

Where:

npmax: - plot point increment

Example: (print out every fifth data point)

```
*CURVE
  NP=5    %
```

4.2.17 Curve Data:

☞ **Note:** Currently a maximum of 5 macro and 5 micro curves may be specified per problem

Specifying Curve Data For Macro (composite) Quantities:

***MACRO**
NT=*nucuv*

Repeat the following line *nucuv* times:
NC=*nocu* X=*maidx* Y=*maidy* NAM=*tname*
%

Where:

nucuv - total number of curves

nocu: - curve number

maidx and *maidy* variable options are:

1 - e_{11}	7 - σ_{11}	13 - ε^{i}_{11}	19 - 30 n.a.	36 - γ^{th}_{13}
2 - e_{22}	8 - σ_{22}	14 - ε^{i}_{22}	31 - ε^{th}_{11}	37 - Total Time
3 - e_{33}	9 - σ_{33}	15 - ε^{i}_{33}	32 - ε^{th}_{22}	38 - Creep Time
4 - γ_{23}	10 - τ_{23}	16 - γ^{i}_{23}	33 - ε^{th}_{33}	39 - Temperature
5 - γ_{13}	11 - τ_{13}	17 - γ^{i}_{13}	34 - γ^{th}_{23}	
6 - γ_{12}	12 - τ_{12}	18 - γ^{i}_{12}	35 - γ^{th}_{12}	

tname: - name of plot file

☞ **Note:** The file(s) generated will be of the form *tname*_macro.data. If the user desires to use the same *tname* for files 2-5, a double quote, “, is entered for *tname*. Those files then will have the form:

*tname*2_macro.data
*tname*3_macro.data
etc.

(see the example on the next page for more details)

☞ **FEAMAC Note:** The file(s) generated when running **FEAMAC** will be of the form *tname*.element#.integrationpt#.feamacro.data, see section **4.1.2.**

EXAMPLE:

∗MACRO

```
NT=3
NC=1    X=1     Y=7     NAM=stress        file created: stress_macro.data
NC=2    X=37    Y=13    NAM ="            file created:stress2_macro.data
NC=3    X=38    Y=13    NAM= plot  %      file created: plot_macro.data
```

☞ **Note:** For the **Laminate option**, the *maidx* and *maidy* variable options are modified as follows:

1 - $\bar{\varepsilon}_{xx}$
2 - $\bar{\varepsilon}_{yy}$
3 - $\bar{\varepsilon}_{33}$
4 - n.a.
5 - n.a.
6 - $\bar{\gamma}_{xy}$
7 - $\overline{N}_{XX}$
8 - $\overline{N}_{YY}$
9 - 11 n.a.
12 - $\overline{N}_{XY}$
13 - $\overline{N}^I_{XX}$
14 - $\overline{N}^I_{YY}$
15 - $\overline{N}^I_{XY}$
16 - $\overline{M}^I_{XX}$
17 - $\overline{M}^I_{YY}$
18 - $\overline{M}^I_{XY}$
19 - $\bar{\kappa}_{xx}$
20 - $\bar{\kappa}_{yy}$
21 - 23 n.a.
24 - $\bar{\kappa}_{xy}$
25 - $\overline{M}_{XX}$
26 - $\overline{M}_{YY}$
27 - 29 n.a.
30 - $\overline{M}_{XY}$
31 -$\overline{N}^T_{XX}$
32 - $\overline{N}^T_{YY}$
33 - $\overline{N}^T_{XY}$
34 - $\overline{M}^T_{XX}$
35 - $\overline{M}^T_{YY}$
36 - $\overline{M}^T_{XY}$
37 - Total Time
38 - Creep Time
39 - Temperature

☞ **Note:** When using laminate option, resultant forces ($\overline{N}_{XX}$, $\overline{N}_{YY}$, or $\overline{N}_{XY}$) are output in place of stress components. To obtain associated stresses one must merely divide resultant force by overall laminate thickness. Also, the strains are mid-plane strains while $\bar{\varepsilon}_{33}$ represents the average strain through the thickness.

Specifying Curve Data for Micro (subcell) Quantities:

∗MICRO

NT=*nucuv*

Repeat the following line *nucuv* times:

NC=*nocu* LYR=*lyr* CELL=*nssel* X=*maidx* Y=*maidy* NAM=*tname2*

%

Where:

nucuv - total number of curves

nocu: - curve number

lyr: - layer number **(only required when using laminate option)**

nssel: - subcell number (see next page for details on numbering)

tname2: - name of plot file

maidx and *maidy* variable options are:

1 - e_{11}	7 - σ_{11}	13 - ε^{i}_{11}	19 - Φ_{11}	25 - Ψ_{11}	31 - ε^{th}_{11}	37- Total Time
2 - e_{22}	8 - σ_{22}	14 - ε^{i}_{22}	20 - Φ_{22}	26 - Ψ_{22}	32 - ε^{th}_{22}	38 - Creep Time
3 - e_{33}	9 - σ_{33}	15 - ε^{i}_{33}	21 - Φ_{33}	27 - Ψ_{33}	33 - ε^{th}_{33}	39 - Temperature
4 - γ_{23}	10 - τ_{23}	16 - γ^{i}_{23}	22 - Φ_{23}	28 - Ψ_{23}	34 - γ^{th}_{23}	
5 - γ_{13}	11 - τ_{13}	17 - γ^{i}_{13}	23 - Φ_{13}	29 - Ψ_{13}	35 - γ^{th}_{12}	
6 - γ_{12}	12 - τ_{12}	18 - γ^{i}_{12}	24 - Φ_{12}	30 - Ψ_{12}	36 - γ^{th}_{13}	

☞ **Note:** The quantities Φ and Ψ are the possible internal state variables (constitutive model dependent).

☞ **Note:** For the **Laminate option**, the data will be output for both integration points within a layer. Thus 4 rather than 2 columns of data will appear in the associated output file(s), with the first 2 columns representing the x and y quantities at the lower quadrature point and the last 2 columns those associated with the upper quadrature point (see **Example P**).

☞ **Note:** The file(s) generated will be of the form *tname*2_micro.data. If the user desires to use the same *tname*2 for files 2-5, a double quote, “, is entered for *tname*2. Those files then will have the form:

*tname*22_micro.data
*tname*23_micro.data
etc.

☞ **Note:** The file(s) generated when running **the laminate option** will be of the form *tname*.l#.micro.data, where # indicates the layer number, see example below

☞ **FEAMAC Note:** The file(s) generated when running **FEAMAC** will be of the form *tname*.element#.integrationpt#.feamicro.data, again see section **4.1.2.**

☞ **Note: Subcell numbering is assigned according to the following algorithm.**

2-D case

```
Do * IB=1,NB
Do * IG=1,NG
subcell number = NG*(IB-1)+IG
* continue
```

☞ **Note: Refer to numbers in upper left corners of subcells in Fig. 14 for example of 2-D subcell numbering scheme.**

3-D case

```
Do * IA=1,NA
Do * IB=1,NB
Do * IG=1,NG
subcell number = IA+(IB-1)*NA+(IG-1)*NA*NB
* continue
```

☞ **Note: Refer to numbers in upper left corners of subcells in Fig. 16 for example of 3-D subcell numbering scheme.**

EXAMPLE:

```
*MICRO
  NT = 1
  NC=1  CELL=1  X=1  Y=7  NAM=cell %        file created:cell_micro.data
```

EXAMPLE: Assuming laminate comprised of 2 layers

```
*MICRO
  NT = 2
  NC=1  LYR=1 CELL=2  X=3  Y=2  NAM=lam-x2 %
                                   file created:lam-x2.l1.micro.data

  NC=2  LYR=2 CELL=2  X=3  Y=1  NAM=lam-x3 %
                                   file created:lam-x3.l2.micro.data
```

5.0 Conclusion/Future Modifications

A computationally efficient, user-friendly, comprehensive, micromechanics analysis tool, **MAC/GMC**, has been presented that admits physically based viscoplastic deformation and life models, can analyze continuous or discontinuous multiphased materials of interest in advanced propulsion systems, and can assist both the materials scientist and structural analyst in developing, designing and analyzing strategic materials. **MAC/GMC**'s most outstanding feature is its ability to accurately model various laminated fiber architectures (including both shape and packing arrangements) at minimal expense both from a computational and required user input standpoint. However, the development of this tool is far from complete. A number of possible future enhancements that are planned or currently underway include:

- Provision of an implicit integration algorithm to improve computational efficiency.
- The incorporation of woven composite IDP architectures.
- The ability to analyze smart composites, i.e. piezoelectric, etc.
- The addition of debond unit cells, so as to minimize user input.
- The incorporation of additional damage evolution laws and failure criteria (e.g., ultimate longitudinal and transverse tensile strength) so as to automate life estimates.
- The incorporation of viscoelastic behavior models for the analysis of elevated temperature polymeric composites.
- The ability to use the triple-periodicity model in conjunction with laminate theory allowing individual lamina to be reinforced with particulates or weaves.

5.1 Acknowledgment

The authors would like to thank Mr. Robert Goldberg and Mrs. Cheryl Bowman for their diligent efforts and assistance in checking the accuracy and consistency of this manual by running numerous test cases. Also, we would like to thank Prof. J. Aboudi for many fruitful discussions and technical assistance.

6.0 EXAMPLE PROBLEMS

It is recommended that a new user construct a **MAC/GMC** input file using the data given in these Examples and then check to see if the same result plots and/or files are obtained.

6.1 Example A: Pure Mechanical Load

Sample Input File For A Mechanical Load Problem

The following example is used to explain the control blocks in more detail.

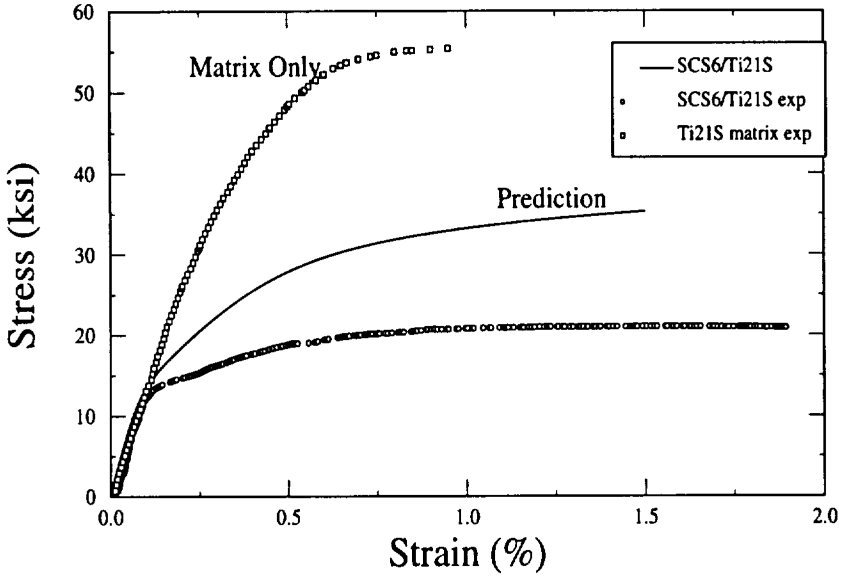
Problem Summary:

Load Type:	Mechanical
Load Component:	22-direction (transverse to fiber)
Load History:	Monotonic
Load Control:	Strain
Load History Data:	$\dot{\varepsilon} = 8.333x10^{-4}$/sec,
	$\varepsilon_{max} = 0.015$,
	$\varepsilon_{min} = 0.$
	$\Delta t_{initial} = 0.0000024$ sec
Micromechanics model:	Double Periodicity
Fiber Packing Arrangement:	Hexagonal Pack at 35% fiber volume ratio
Integration Algorithm:	Predictor/Corrector
Constituent Material Model:	GVIPS - isotropic form
Constituents:	Fiber: SCS-6 Matrix: TIMETAL21S Interface: fictitious weak interface for TIMETAL21S

☞ **Note:** This example **will take an extremely long time to run** due to the elastic-perfectly viscoplastic definition of the fictitious weak interface, as the overall time step is limited to that of the allowable interface time step. **This slow computational response is NOT indicative of GMC.**

```
Mechanical load, no residual, perfectly viscoplastic interface, takes long time
*PRINT
 NPL=0 %
*LOAD
 LCON=2 LOP=2 LSS=1 %
*MECH
 NPTW=2 TI=0.,18. LO=0.,0.015 %
*MODEL
 MOD=1 %
*SOLVER
 NTF=2 ISTM=0.0000024 ERR=0.1E-2 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=1 MAT=D TEMP=650 %
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=1 MAT=A TEMP=650 %
*MRVE
 IDP=2 VF=0.35 RAD=0.07 CPER=0.1 %
*INTERFACE
 NINT=1
 NI=1 MI=4 NDPT=1 MAT=U IFM=1  &
 EL=11700.,0.365,1. &
 VI=0.8E-8,0.1,0.1E-5,0.,0.85E-3,0.05,1.,1.,1.,3.3,1.8,1.35,1.,0.01 %
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=2 Y=8 NAM=apdxa %
*END
```

The following figure was obtained from the x-y plot data file produced by the present example.



☞ **Note:** Using this fictitious interface model the qualitative feature of a weak bond (reduction below that of the matrix only) is obtained, however an accurate fit of the experimental data is still lacking. For a more accurate prediction of transverse behavior see **Examples C** and **D**.

6.2 Example B: Pure Thermal Problem

Sample Input File For A Thermal Load Problem

The following example is used to explain the control blocks in more detail.

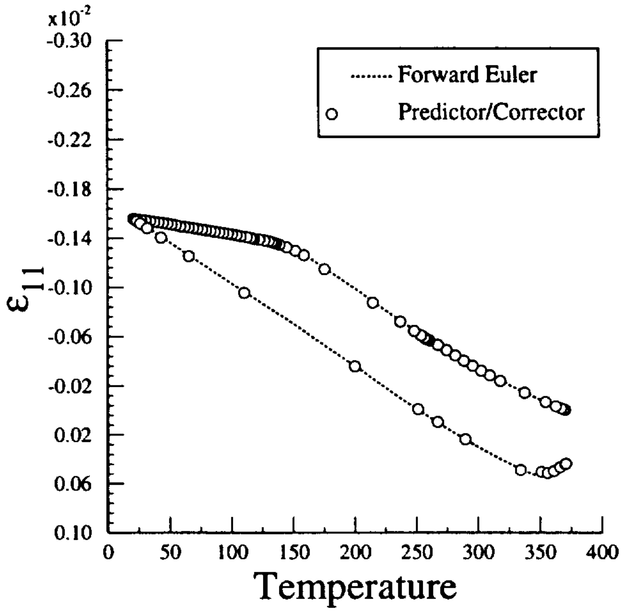
Problem Summary:

Load Type:	Thermal
Load History Data:	$\dot{T} = 0.01$ °C/sec,
	$T_{max} = 371.1^{o}$C
	$T_{min} = 21.1^{o}$C
	$\Delta t_{initial} = 17.505$ sec
Micromechanics model:	Double Periodicity
Fiber Packing Arrangement:	Square Pack at 35% fiber volume ratio
Integration Algorithm:	Predictor/Corrector
Constituent Material Model:	Bodner-Partom
Constituents:	Fiber: T50 Graphite Matrix: Aluminum (2024-T4)

☞ **Note:** This problem is taken from reference 1., pg. .238

```
cooldown heatup bodner model aboudi prob. pre/cor gmc2d
*PRINT
 NPL=0  %
*LOAD
 LCON=1  %
*THERM
 NPTT=3  TI=0.,35010.,70020.  &
 TE=371.1,21.1,371.1  %
*MODEL
 MOD=1  %
*SOLVER
 NTF=2  ISTT=17.505  ERR=0.001  %
*FIBER
 NFIBS=1
 NF=1  MF=9  NDPT=2  MAT=A   D=1.,0.,0.%
*MATRIX
 NMATX=1
 NM=1  MM=1  NDPT=2  MAT=A   %
*MRVE
 IDP=1  VF=0.30  %
*CURVE
 NP=5  %
*MACRO
 NT=1
 NC=1  X=39  Y=1  NAM=apdxb  %
*END
```

The following figure was obtained from the x-y plot data file produced by the present example.



☞ **Note:** It is recommended that a new user construct a mac input file using the data given in this Example and then check to see if the same result plot is obtained.

6.3 Example C: Thermomechanical Load Problem

Sample Input File For A Thermomechanical Load Problem

The following example is used to explain how to impose thermal residual stresses due to manufacturing into mechanical analysis.

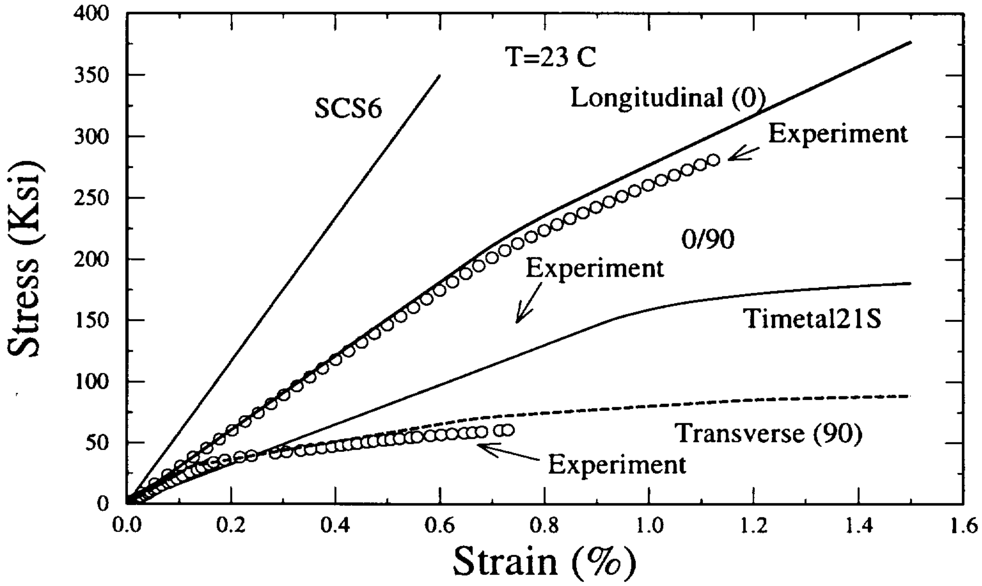
Problem Summary:

Load Type:	Thermomechanical
Load Component:	33-direction (transverse to fiber)
Load History:	Cyclic
Load Control:	Strain
Load History Data:	Cool-down from 900 °C to 23 °C then hold temperature constant during mechanical loading $\dot{T} = 1.52x10^{-2}$ °C/sec $\dot{\varepsilon} = 1.666x10^{-4}$ /sec, $\varepsilon_{max} = 0.015$, $\varepsilon_{min} = 0.$ $\Delta t_{thermal} = 100.$ sec, $\Delta t_{mech} = 0.1$ sec
Micromechanics model:	Double Periodicity
Fiber Packing Arrangement:	Rectangular Pack at 33% fiber volume ratio
Integration Algorithm:	Forward Euler
Constituent Material Model:	GVIPS - isotropic form
Constituents:	Fiber: SCS-6 (properties input manually) Matrix: TIMETAL21S
Debonding:	First implementation of debond model

☞ **Note: This example demonstrates how one would included the effects of residual stresses on the composite behavior. Also, although strain control is prescribe, during a pure cool down MAC/GMC allows Poisson's strains to be incurred.**

```
test of residual stress idp=11 23C apply transverse loading
*PRINT
 NPL=0 %
*LOAD
 LCON=3 LOP=3 LSS=1 %
# LCON=3 LOP=1 LSS=1 %
*MECH
 NPTW=3 TI=0.,57600.,57690. LO=0.,0.,0.015 %
# NPTW=3 TI=0.,57600.,57750. LO=0.,0.,0.015 %
*THERM
 NPTT=3 TI=0.,57600.,57690. TE=900.,23.,23. %
# NPTT=3 TI=0.,57600.,57750. TE=900.,23.,23. %
*MODEL
 MOD=1 %
*SOLVER
 NTF=1 NPTS=3 TIM=0.,57600.,57690. STP=100.,0.1 %
# NTF=1 NPTS=3 TIM=0.,57600.,57750. STP=100.,0.1 %
*FIBER
 NFIBS=1
# Use GVIPS but make it elastic k=1E20, so it represents SCS-6 fiber
 NF=1 MF=4 NDPT=1 MAT=U IFM=1 &
 EL=58.E3, 0.32, 3.5E-06 &
 VI=0.8E-9,1.E20,0.1E-5,0.,0.85E-4,0.05,1.,1.,1.,3.3,1.8,1.35,1.,0.01 %
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=2 MAT=A %
*MRVE
 IDP=11 VF=0.33 RAD=0.07 R=1.1111 %
*DEBOND
 NII=2
 DBCH=1 NBI=1 NGI=1 FACE=2 RN=0.1 BDN=14. TI=57600.
 DBCH=1 NBI=1 NGI=2 FACE=2 RN=0.1 BDN=14. TI=57600. %
*CURVE
 NP=2 %
*MACRO
 NT=2
 NC=1 X=3 Y=9 NAM=apdxc-t
 NC=2 X=1 Y=7 NAM=apdxc-l
*END
```

The following figure was obtained from the x-y plot data file produced by the present example.



☞ **Note:** In order to produce the longitudinal response curve one must change the direction of loading (LOP=1 instead of 3) and the mechanical strain rate imposed, so as to agree with the imposed experimental history by commenting and uncommenting the appropriate lines under *LOAD, *MECH, *THERM, and *SOLVER.

☞ **Note:** It is recommended that a new user construct a **MAC/GMC** input file using the data given in this Example and then check to see if the same resulting plot can be obtained.

6.4 Example D: Transverse Debonding

Sample Input File for Transverse Debonding

Problem Summary

Load Type:	Thermomechanical
Load History Data:	33-direction (transverse to fiber)
Load History:	Cyclic
Load Control:	Strain
Load History Data:	Cool-down from 900°C to 534°C to 23°C, heat-up to 650°C, then hold temperature.during mechanical loading $\dot{T}$=0.0152 °C/sec, 0.0871 °C/sec $\dot{\varepsilon}=1.666\times10^{-4}$ /sec, $\varepsilon_{max}=0.018$, $\varepsilon_{min}=0.$ $\Delta t_{thermal}$=250., 40. sec.; Δt_{mech} =0.4 sec.
Micromechanics Model:	Double Periodicity
Fiber Packing Arrangement:	Rectangular Pack, R = 1.1111, 35% fiber volume ratio
Repeating Unit Cell:	26x26 circular fiber cross-section approximation
Integration Algorithm:	Forward Euler
Constituent Material Model:	Fiber: Elastic, transversely isotropic Matrix: GVIPS - isotropic form
Constituents:	Fiber: SCS-6 (temp. dep. properties input manually) Matrix: TIMETAL21S
Debonding:	Second implementation of debond model
Debond Parameters:	$\sigma_{DB} = \sigma^o_{DB} + C_1/\lvert slope\rvert$, σ^o_{DB}=7.ksi,C_1= 15.ksi $\Lambda = \Lambda_o - C_2/\lvert slope\rvert$, $\Lambda_o = 0.06\frac{in^3}{kip}$, $C_2 = 0.03997\frac{in^3}{kip}$ B= 16.5 1/sec

☞ **Note:** The debond parameters σ_{DB}^{o} and Λ have been weighted with respect to the local slope (i.e., x_2/x_3) of the representative circular fiber at each interface (see IDP=13 Figure 9). This helps correct for the fact that the representative circle has interfaces normal to the loading direction (due to the necessarily rectangular-shaped subcells), while an actual circular fiber does not. For an actual circular fiber, the fiber matrix interface is only normal to the loading direction at two points, and as we trace the fiber's perimeter away from these points, the interface becomes increasingly parallel to the loading direction. In our representative fiber, we thus provide interfaces that are more parallel to the loading direction with larger debond strengths since the component of stress normal to the actual fiber interface (at this point) is lower. Similarly, we provide these more parallel interfaces with lower values of Λ so that, once debonding occurs, the stress at the simulated interface (which is in the global loading direction) unloads more slowly. The weighting has been chosen as linear (with respect to the local interfacial slope) for simplicity and because this introduces only one additional parameter for each of σ_{DB}^{o} and Λ. Note that modeling of interfacial debonding is an area of active research using **MAC/GMC**, and thus the techniques employed in this example are still developing.

The results of this example shown in the following figure have been plotted only to an applied strain level of 0.016 because at this point the slope of the predicted global stress-strain curve becomes negative. We are treating this as a criterion for simulated global failure of the composite.

The interfacial curves shown in the figure are **NOT** simple plots of the ***MICRO** output files. Rather, the microlevel stress from these output files is plotted versus the macro level strain applied to the composite from the first ***MACRO** output file.

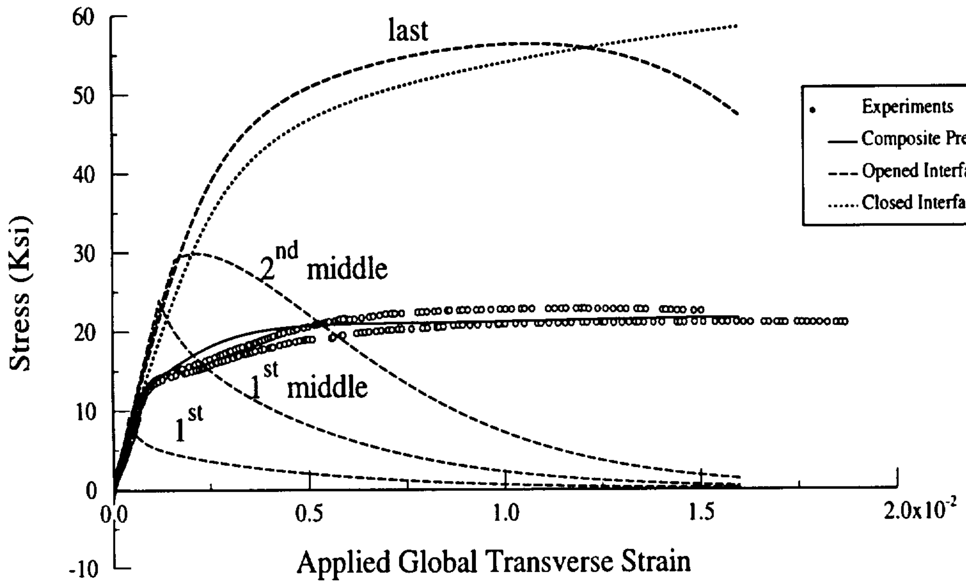
```
test of refined subcell transverse debonding
*PRINT
 NPL=1  %
*LOAD
 LCON=3 LOP=3  LSS=1 %
*MECH
 NPTW=5 TI=0.,24000.,57600.,64800.,64908. LO=0.,0.,0.,0.,0.018  %
*THERM
 NPTT=5 TI=0.,24000.,57600.,64800.,64908. TE=900.,534.583,23.,650.,650. %
*MODEL
 MOD=1  %
*SOLVER
 NTF=1 NPTS=5 TIM=0.,24000.,57600.,64800.,64908. STP=250.,40.,40.,0.4 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=2 MAT=U IFM=1
 NTP=6
 TEM=21.1,204.44,315.56,426.67,537.78,871.11
 EA=57.0E3,55.98E3,55.4E3,54.82E3,54.24E3,53.36E3
 ET=57.0E3,55.98E3,55.4E3,54.82E3,54.24E3,53.36E3
 NUA=0.25,0.25,0.25,0.25,0.25,0.25
 NUT=0.25,0.25,0.25,0.25,0.25,0.25
 GA=22.8E3,22.392E3,22.16E3,21.928E3,21.696E3,21.344E3
 ALPA=3.564E-6,3.618E-6,3.726E-6,3.906E-6,4.068E-6,4.572E-6
 ALPT=3.564E-6,3.618E-6,3.726E-6,3.906E-6,4.068E-6,4.572E-6
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=2 MAT=A %
*MRVE
 IDP=13 VF=0.35 R=1.111111111
*DEBOND
 NII=48
 DBCH=2 NBI=13 NGI= 1 FACE= 2 BDN=7. GCN=0.060 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=12 NGI= 2 FACE= 2 BDN=11.3 GCN=0.0486 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=11 NGI= 3 FACE= 2 BDN=12. GCN=0.0467 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=10 NGI= 4 FACE= 2 BDN=17. GCN=0.0334 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI= 9 NGI= 5 FACE= 2 BDN=18.3 GCN=0.0300 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI= 8 NGI= 6 FACE= 2 BDN=19. GCN=0.0280 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI= 7 NGI= 7 FACE= 2 BDN=22. GCN=0.0200 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
```

```
DBCH=2 NBI= 6 NGI= 8 FACE= 2 BDN=25.8 GCN=0.0100 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 5 NGI= 9 FACE= 2 BDN=27.  GCN=0.00671 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 4 NGI=10 FACE= 2 BDN=29.5 GCN=0.000045 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 3 NGI=11 FACE= 2 BDN=52. GCN=0.0 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 2 NGI=12 FACE= 2 BDN=59.5 GCN=0.0 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 2 NGI=14 FACE= 2 BDN=59.5 GCN=0.0 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 3 NGI=15 FACE= 2 BDN=52. GCN=0.0 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 4 NGI=16 FACE= 2 BDN=29.5 GCN=0.000045 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 5 NGI=17 FACE= 2 BDN=27.  GCN=0.00671 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 6 NGI=18 FACE= 2 BDN=25.8 GCN=0.0100 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 7 NGI=19 FACE= 2 BDN=22. GCN=0.0200 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 8 NGI=20 FACE= 2 BDN=19. GCN=0.0280 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI= 9 NGI=21 FACE= 2 BDN=18.3 GCN=0.0300 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=10 NGI=22 FACE= 2 BDN=17. GCN=0.0334 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=11 NGI=23 FACE= 2 BDN=12. GCN=0.0467 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=12 NGI=24 FACE= 2 BDN=11.3 GCN=0.0486 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=13 NGI=25 FACE= 2 BDN=7. GCN=0.060 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=14 NGI=25 FACE= 2 BDN=7. GCN=0.060 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=15 NGI=24 FACE= 2 BDN=11.3 GCN=0.0486 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=16 NGI=23 FACE= 2 BDN=12. GCN=0.0467 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=17 NGI=22 FACE= 2 BDN=17. GCN=0.0334 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=18 NGI=21 FACE= 2 BDN=18.3 GCN=0.0300 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
DBCH=2 NBI=19 NGI=20 FACE= 2 BDN=19. GCN=0.0280 BCN=16.5 TOLN=1 &
BDS= 40 GCS=0.1 BCS=100 TI=64800
```

```
 DBCH=2 NBI=20 NGI=19 FACE= 2 BDN=22. GCN=0.0200 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=21 NGI=18 FACE= 2 BDN=25.8 GCN=0.0100 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=22 NGI=17 FACE= 2 BDN=27. GCN=0.00671 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=23 NGI=16 FACE= 2 BDN=29.5 GCN=0.000045 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=24 NGI=15 FACE= 2 BDN=52. GCN=0.0 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=25 NGI=14 FACE= 2 BDN=59.5 GCN=0.0 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=25 NGI=12 FACE= 2 BDN=59.5 GCN=0.0 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=24 NGI=11 FACE= 2 BDN=52. GCN=0.0 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=23 NGI=10 FACE= 2 BDN=29.5 GCN=0.000045 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=22 NGI= 9 FACE= 2 BDN=27. GCN=0.00671 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=21 NGI= 8 FACE= 2 BDN=25.8 GCN=0.0100 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=20 NGI= 7 FACE= 2 BDN=22. GCN=0.0200 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=19 NGI= 6 FACE= 2 BDN=19. GCN=0.0280 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=18 NGI= 5 FACE= 2 BDN=18.3 GCN=0.0300 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=17 NGI= 4 FACE= 2 BDN=17. GCN=0.0334 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=16 NGI= 3 FACE= 2 BDN=12. GCN=0.0467 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=15 NGI= 2 FACE= 2 BDN=11.3 GCN=0.0486 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
 DBCH=2 NBI=14 NGI= 1 FACE= 2 BDN=7. GCN=0.060 BCN=16.5 TOLN=1 &
 BDS= 40 GCS=0.1 BCS=100 TI=64800
*CURVE
 NP=3 %
*MACRO
 NT=2
 NC=1 X=3 Y=9 NAM=TRAN-M
 NC=2 X=24 Y=3 NAM=TRAN-T
*MICRO
 NT=5
# 13,1
 NC=1 CELL=313 X=3 Y=9 NAM=TRAN-1
```

```
# 2,14
 NC=2 CELL=40  X=3 Y=9 NAM=TRAN-2
# 4,16
 NC=3 CELL=94  X=3 Y=9 NAM=TRAN-3
# 7,19
 NC=4 CELL=175 X=3 Y=9 NAM=TRAN-4
# 5,17
 NC=5 CELL=121 X=3 Y=9 NAM=TRAN-5 %
*END
```

The following figure was obtained from the x-y plot data file produced by the present example.



6.5 Example E: General Loading

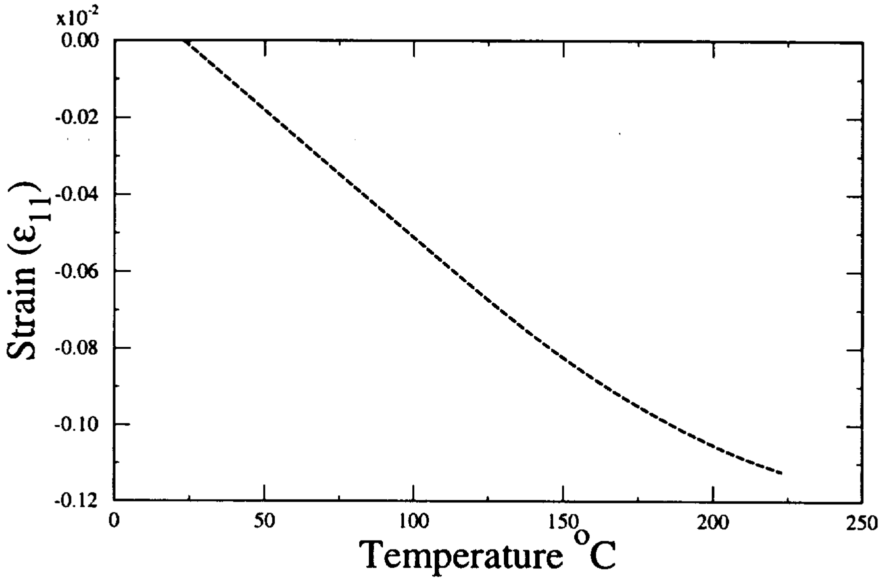
Sample Input File for General Loading

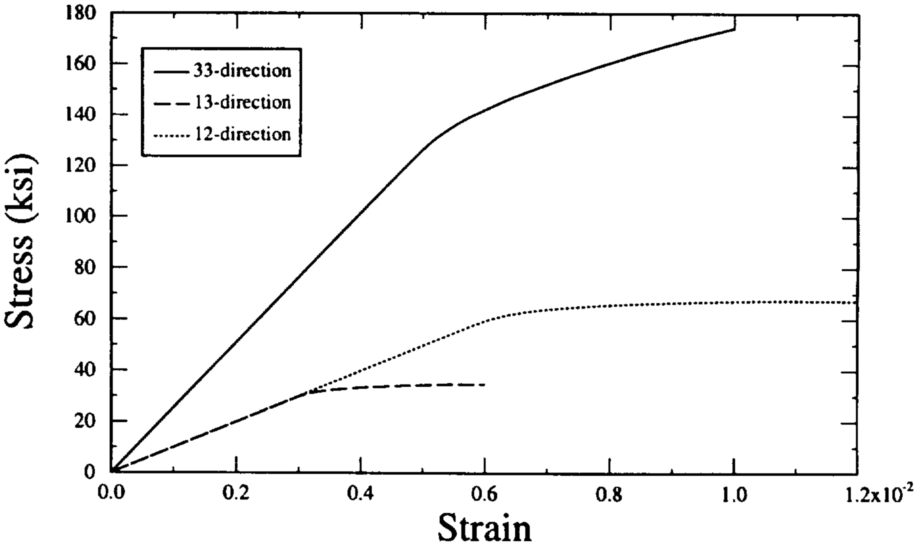
Problem Summary

Load Type:	Thermo-Mechanical
Load History Data:	Combined: 33 - normal strain (transverse to fiber) 13 - shear strain 12 - shear strain 11, 22, and 23 component stress-free Heat-up from 23 °C to 223 ° C
Load History:	Monotonic
Load Control:	Combined stress and strain
Load History Data:	$\dot{T} = 1.85$ °C/sec $\dot{\varepsilon}_{33} = 9.26x10^{-5}$/sec, $\varepsilon_{33max} = 0.01$, $\varepsilon_{33min} = 0$ $\dot{\varepsilon}_{13} = 5.56x10^{-5}$/sec, $\varepsilon_{13max} = 0.006$, $\varepsilon_{13min} = 0$ $\dot{\varepsilon}_{12} = 1.11x10^{-4}$/sec, $\varepsilon_{12max} = 0.012$, $\varepsilon_{12min} = 0$ $\Delta t = 0.4$ sec
Micromechanics Model:	Double Periodicity
Fiber Packing Arrangement:	Square Pack, R = 1., 45% fiber volume ratio
Repeating Unit Cell:	2x2 subcells, square pack using IDP=11
Integration Algorithm:	Forward Euler
Constituent Material Model:	Fiber: Elastic, isotropic Matrix: GVIPS - isotropic form
Constituents:	Fiber: SCS-6 (temp. dep. properties input manually) Matrix: TIMETAL21S

```
example of general loading
*PRINT
 NPL=1  %
*LOAD
 LCON=3 LOP=99 LSS=2,2,1,2,1,1 %
*MECH
 NPTW=2 TI=0.,108. LO=0.,0.
 NPTW=2 TI=0.,108. LO=0.,0.
 NPTW=2 TI=0.,108. LO=0.,0.01
 NPTW=2 TI=0.,108. LO=0.,0.
 NPTW=2 TI=0.,108. LO=0.,0.006
 NPTW=2 TI=0.,108. LO=0.,0.012
*THERM
 NPTT=2 TI=0.,108. TE=23.,223.
*MODEL
 MOD=1  %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,108. STP=0.4 %
*FIBER
 NFIBS=1
 NF=1 MF=9 NDPT=2 IFM= 1 MAT=U
 NTP=6
 TEM=21.1,204.44,315.56,426.67,537.78,871.11
 EA=57.0E3,55.98E3,55.4E3,54.82E3,54.24E3,53.36E3
 ET=57.0E3,55.98E3,55.4E3,54.82E3,54.24E3,53.36E3
 NUA=0.25,0.25,0.25,0.25,0.25,0.25
 NUT=0.25,0.25,0.25,0.25,0.25,0.25
 GA=22.8E3,22.392E3,22.16E3,21.928E3,21.696E3,21.344E3
 ALPA=3.564E-6,3.618E-6,3.726E-6,3.906E-6,4.068E-6,4.572E-6
 ALPT=3.564E-6,3.618E-6,3.726E-6,3.906E-6,4.068E-6,4.572E-6
 D=1.,0.,0.
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=2 MAT=A %
*MRVE
 IDP=11 VF=0.45 RAD=0.07 R=1.0
*CURVE
 NP=5 %
*MACRO
 NT=4
 NC=1 X=3 Y=9 NAM=GENLD-33
 NC=2 X=5 Y=11 NAM=GENLD-13
 NC=3 X=6 Y=12 NAM=GENLD-12
 NC=4 X=39 Y=1 NAM=GENLD-T %
*END
```

The following figures were obtained from the x-y plot data file produced by the present example.





6.6 Example F: Unidirectional Laminate Problem

Sample Input File For A Laminate Problem

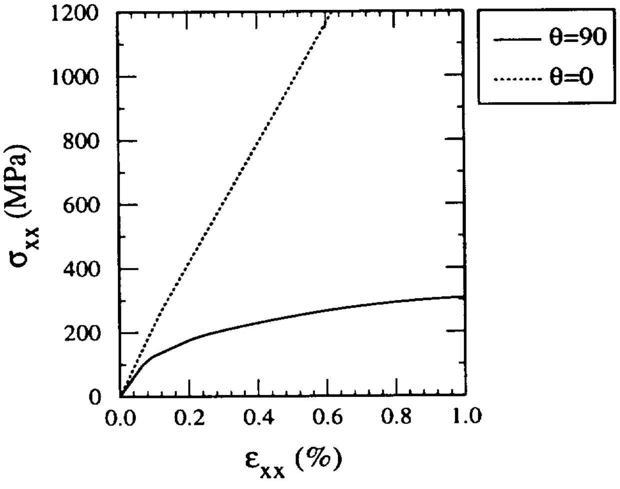
The following example is used to explain the control blocks in more detail.

Problem Summary:

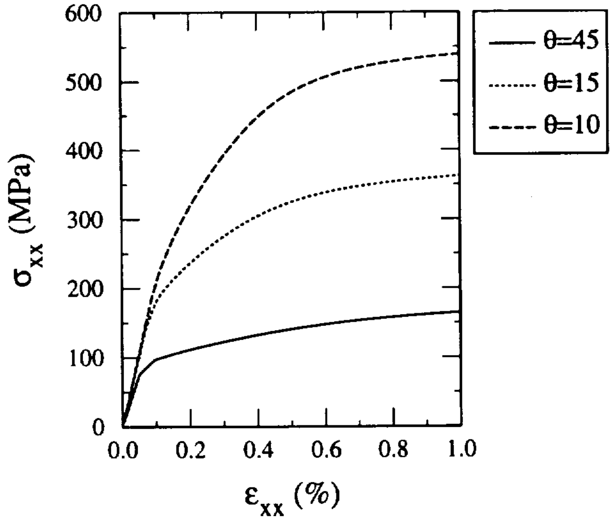
Load Type:	Mechanical
Load History:	Monotonic
Load Control:	Strain
Load History Data:	$\dot{\varepsilon} = 0.01$ /sec, $\varepsilon_{max} = 0.01$, $\Delta t_{constant}$= 0.00025 sec
Micromechanics model:	Laminate Theory
Fiber Packing Arrangement:	Square Pack at 46% fiber volume ratio
Integration Algorithm:	Forward Euler
Constituent Material Model:	Bodner-Partom
Constituents:	Fiber: Boron Matrix: Aluminum (6061-0a)

```
test of laminate strain control
*PRINT
 NPL=0  %
*LOAD
 LCON=2  LOP=1  LSS=1 %
*MECH
 NPTW=2 TI=0.,1. LO=0.,0.01  %
*MODEL
 MOD=3 MATSYS=1 NLY=1 THK=1. CON=2 SYS=1 ANG=45. %
# MOD=3 MATSYS=1 NLY=1 THK=1. CON=2 SYS=1 ANG=0. %
# MOD=3 MATSYS=1 NLY=1 THK=1. CON=2 SYS=1 ANG=90. %
# MOD=3 MATSYS=1 NLY=1 THK=1. CON=2 SYS=1 ANG=10. %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,1. STP=0.00025 %
*FIBER
 NFIBS=1
 NF=1 MS=1 MF=6 NDPT=1 TEMP= 21. MAT=A %
*MATRIX
 NMATX=1
 NM=1 MS=1 MM=1 NDPT=1 TEMP= 21. MAT=C %
*MRVE
 IDP=1
  L=1 VF=0.46 %
*CURVE
 NP=10 %
*MACRO
 NT=1
 NC=1 X=1 Y=7 NAM=apdxf %
*END
```

The following figures were obtained from the x-y plot data file produced by the present example. The validity of these predictions were assessed by comparing these results to those previously obtain from:



reference 1 Fig. 8-5, pg. 235



reference 1 Fig. 8-7, pg. 237

6.7 Example G: A Cross-Ply Laminate Problem

Sample Input File For A Laminate Problem

The following example is used to explain the control blocks in more detail.

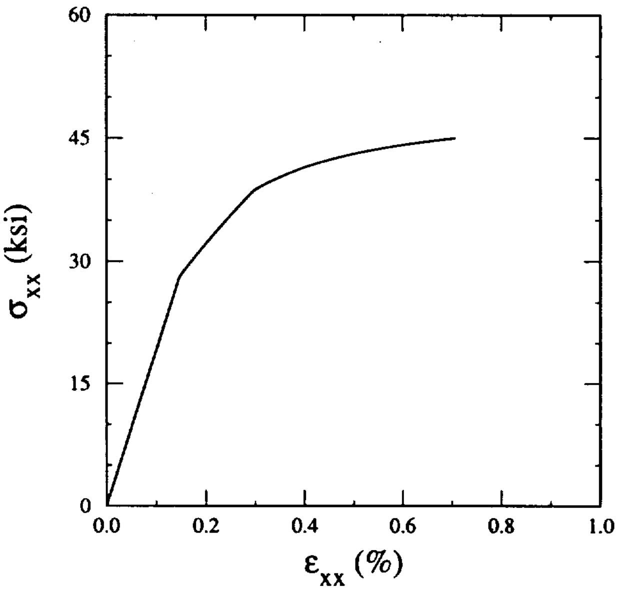
Problem Summary:

Load Type:	Mechanical
Load History:	Monotonic
Load Control:	Stress
Load History Data:	$\dot{\sigma} = 0.01$ ksi/sec,
	$\sigma_{max} = 45.$ ksi,
	$\Delta t_{constant}$= 1.125 sec
Micromechanics model:	$[\pm 45]_s$ Laminate with Double Periodicity
Fiber Packing Arrangement:	Square Pack at 45% fiber volume ratio
Integration Algorithm:	Forward Euler
Constituent Material Model:	Bodner-Partom
Constituents:	Fiber: Boron (properties input manually) Matrix: Aluminum (properties input manually)

☞ **Note:** This problem is taken from the reference: Analysis of Metal-Matrix Composite Structures-II. Laminate Analysis, Arenburg, R. T. and Reddy, J. N., Computers and Structures, Vol. 40, N.6, pp. 1369-1385, 1991.

```
test of [+-45]2s laminate stress control
*PRINT
  NPL=0  %
*LOAD
  LCON=2  LOP=1  LSS=2 %
*MECH
  NPTW=2 TI=0.,4500. LO=0.,45.  %
*MODEL
  MOD=3 MATSYS=1 NLY=4 &
  THK=0.25,0.25,0.25,0.25 &
  CON= 2, 2, 2, 2 &
  SYS= 1, 1, 1, 1 &
  ANG=45.,-45.,-45.,45. %
*SOLVER
  NTF=1 NPTS=2 TIM=0.,4500. STP=1.125 %
*FIBER
  NFIBS=1
  NF=1 MS=1 MF=6 NDPT=1 MAT=U IFM=1 &
  EL=58.E3,58.E3,0.20,0.20,24.17E3,6.3E-6,8.28E-6   %
*MATRIX
  NMATX=1
  NM=1 MS=1 MM=1 NDPT=1 MAT=U IFM=1 &
  EL=9.53E3,9.53E3,0.33,0.33,3.58E3,21.06E-6,21.06E-6  &
  VI=1.E4,49.,63.,300.,4.,1. %
*MRVE
 IDP=1,1,1,1
 L=1 VF=0.45 %
 L=2 VF=0.45 %
 L=3 VF=0.45 %
 L=4 VF=0.45 %
*CURVE
  NP=10 %
*MACRO
  NT=1
  NC=1 X=1 Y=7 NAM=apdxg %
*END
```

The following figure was obtained from the x-y plot data file produced by the present example and for validation was compared to Arenburg and Reddy, Fig. 16, pg. 1382.



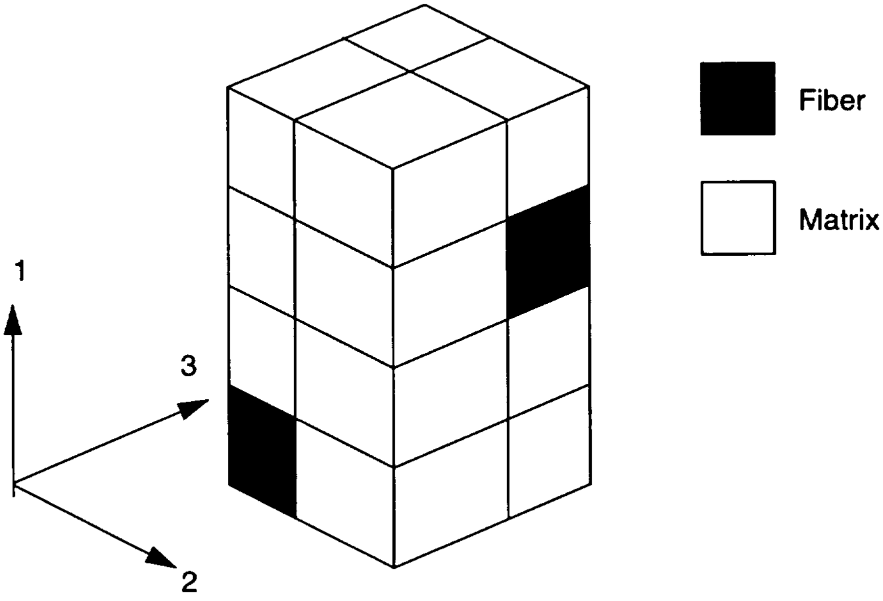
6.8 Example H: Triple Periodic GMC; [0/90]

Sample Input File For Triple Periodic **GMC**

Problem Summary:

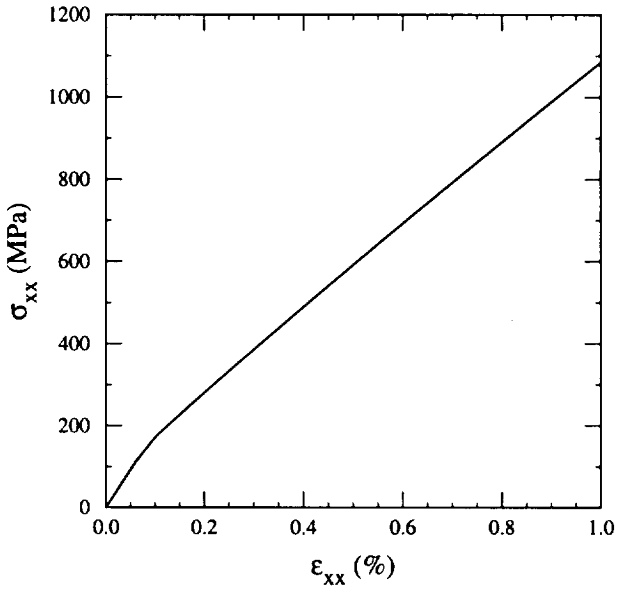
Loading:	Mechanical, Strain control
Load History Data:	$\dot{\varepsilon} = 0.01/\text{sec}$, $\varepsilon_{max} = 0.01$, $\Delta t_{constant}$= 0.00025 sec
Micromechanics model:	Triple Periodic **GMC**
Fiber Packing Arrangement:	Rectangular Pack at 46% fiber volume ratio (input manually, see figure below)
Integration Algorithm:	Forward Euler
Constituent Material Model:	Bodner-Partom
Constituents:	Fiber: Boron Matrix: Aluminum (6061-0a)

This problem uses the 3-D **GMC** model to simulate a [0/90] laminate as shown below.



```
test of gmc3d model 0/90 laminate simulation
*PRINT
 NPL=10  %
*LOAD
 LCON=2  LOP=3 LSS=1 %
*MECH
 NPTW=2 TI=0.,1. LO=0.,0.01  %
*MODEL
 MOD=2  %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,1. STP=0.00025 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=1 TEMP= 21. MAT=A %
*MATRIX
 NMATX=1
 NM=1 MM=1 NDPT=1 TEMP= 21. MAT=C %
*MRVE
 IDP=99
 NA=4 NB=2 NG=2
 D=0.67823,0.32177,0.67823,0.32177
 H=0.67823,0.32177
 L=0.32177,0.67823
 CM=M1,M1
 CM=M1,M1
 CM=M1,M1
 CM=F1,M1
 CM=M1,M1
 CM=F1,F1
 CM=M1,M1
 CM=F1,M1 %
*CURVE
 NP=10 %
*MACRO
 NT=1
 NC=1 X=3 Y=9 NAM=apdxh %
*END
```

The following figure was obtained from the x-y plot data file produced by the present example.



☞ **Note:** It is recommended that a new user construct a **MAC/GMC** input file using the data given in this Example and then check to see if the same result plot is obtained.

6.9 Example I: A User Defined RVE

<u>Sample Input File For A User Defined RVE</u>

The following example is used to explain how to enter a user defined architecture.

<u>Problem Summary</u>:

Load Type:	Mechanical	
Load History: Load Control:	Monotonic Strain	
Load History Data:	$\dot{\varepsilon} = 0.1$ /sec,	
	$\varepsilon_{max} = 0.01$,	
	$\Delta t_{constant}$= 0.1 sec	
Micromechanics model:	Double periodic	
Fiber Packing Arrangement:	Random composite (see figure 13)	
Integration Algorithm:	Forward Euler	
Constituent Material Model:	Bodner-Partom	
Constituents:	Fiber: Matrix:	Boron Aluminum (6061-0a)

```
random composite rve using 2-d gmc
*PRINT
 NPL=-1 %
*LOAD
 LCON=2 LOP=2 LSS=1 %
*MECH
 NPTW=2 TI=0.,0.1 LO=0.,0.01  %
*MODEL
 MOD=1  %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,0.1 STP=0.00005 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=1 TEMP=21. MAT=A %
*MATRIX
 NMATX=1
 NM=1 MM=1 NDPT=1 TEMP=21. MAT=C %
*MRVE
 IDP=99
 NB=14 NG=14
 H=1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.
 L=1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.
 CM=F1,M1,M1,M1,M1,M1,F1,F1,M1,M1,F1,F1,M1,F1
 CM=F1,M1,F1,F1,M1,M1,F1,F1,M1,M1,M1,M1,M1,F1
 CM=M1,M1,F1,F1,M1,M1,M1,M1,M1,M1,F1,F1,M1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,F1,F1,M1,F1,F1,M1,M1
 CM=M1,M1,F1,F1,M1,M1,M1,F1,F1,M1,M1,M1,M1,M1
 CM=F1,M1,F1,F1,M1,M1,M1,M1,M1,M1,F1,F1,M1,F1
 CM=F1,M1,M1,M1,M1,F1,F1,M1,M1,M1,F1,F1,M1,F1
 CM=M1,M1,M1,M1,M1,F1,F1,M1,M1,M1,M1,M1,M1,M1
 CM=M1,F1,F1,M1,M1,M1,M1,M1,F1,F1,M1,F1,F1,M1
 CM=M1,F1,F1,M1,F1,F1,M1,M1,F1,F1,M1,F1,F1,M1
 CM=M1,M1,M1,M1,F1,F1,F1,F1,M1,M1,M1,M1,M1,M1
 CM=F1,F1,M1,M1,M1,M1,F1,F1,M1,M1,M1,F1,F1,M1
 CM=F1,F1,M1,M1,M1,M1,M1,M1,M1,M1,M1,F1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1,M1,M1,F1,F1,M1,M1   %
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=2 Y=8 NAM=apdxi %
*END
```

The corresponding output produced:

```
MATERIAL NO.= 1     VOLUME RATIO= 0.347E+00
          MATERIAL NO.= 2     VOLUME RATIO= 0.653E+00
 CG - Effective/Macro Stiffness Matrix
    0.221E+12    0.610E+11    0.608E+11
    0.610E+11    0.148E+12    0.675E+11
    0.608E+11    0.675E+11    0.147E+12
                                   0.384E+11
                                          0.392E+11
                                                0.398E+11
 CI - Inverse of Effective/Macro Stiffness
    0.537E-11   -0.152E-11   -0.153E-11
   -0.152E-11    0.897E-11   -0.350E-11
   -0.153E-11   -0.350E-11    0.906E-11
                                   0.260E-10
                                         0.255E-10
                                               0.251E-10
 Effective Engineering Moduli
     E11S=  0.186E+12
     N12S=  0.282E+00
     E22S=  0.111E+12
     N23S=  0.390E+00
     E33S=  0.110E+12
     G23S=  0.384E+11
     G13S=  0.392E+11
     G12S=  0.398E+11

 Effective Thermal Expansion Coefficients

    0.120E-04    0.192E-04    0.196E-04

 STOPPING: NPLVL == -1

 ENDENDENDENDENDENDENDENDENDENDENDENDENDENDENDENDEND

              _______    ______
             / _____/ | / / __ \\
            / __/ /   |/ / / / /
           / /___/ /|   / /_/ /
          /______/_/ |_/_____/

 ENDENDENDENDENDENDENDENDENDENDENDENDENDENDENDENDEND
```

6.10 Example J: A Biaxial Load

Sample Input File For A Biaxial Load

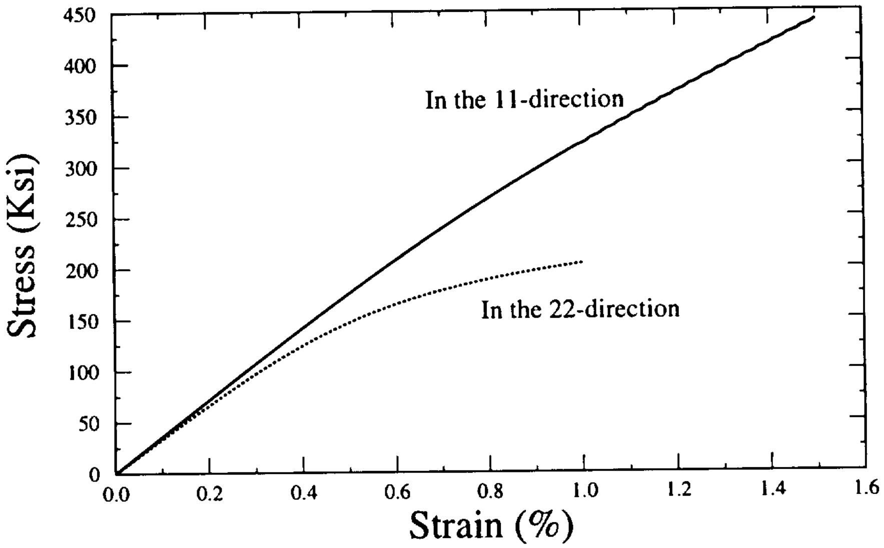
The following example is used to explain how to impose a biaxial load state.

Problem Summary:

Load Type:	Mechanical
Load History:	Monotonic
Load Control:	Strain
Load History Data:	$\dot{\varepsilon}_{11} = 0.01$ /sec ,$\dot{\varepsilon}_{22} = 0.0067$/sec $\varepsilon_{11_{max}} = 0.015$ $\varepsilon_{22max} = 0.01$, $\Delta t_{constant}$= 0.015 sec
Micromechanics model:	Double Periodic
Fiber Packing Arrangement:	Square Pack at 35% fiber volume ratio
Integration Algorithm:	Forward Euler
Constituent Material Model:	GVIPS
Constituents:	Fiber: SCS-6 (GVIPS forced to be elastic) Matrix: TIMETAL 21S

```
test of biaxial load
*PRINT
 NPL=7 %
*LOAD
 LCON=2 LOP=7 LSS=1 %
*MECH
 NPTW=2 TI=0.,1.5 LO=0.,0.015
 NPTW=2 TI=0.,1.5 LO=0.,0.010 %
*MODEL
 MOD=1 %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,1.5 STP=0.015 %
*FIBER
 NFIBS=1
 NF=1 MF=4 NDPT=1 MAT=U IFM=1&
 EL=58.E3,0.32,3.5E-06 &
 VI=0.8E-9,1.E20,0.1E-5,0.,0.85E-4,0.05,1.,1.,1.,3.3,1.8,1.35,1.,0.01 %
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=1 TEMP=650. MAT=A %
*MRVE
 IDP=1 VF=0.35 %
*CURVE
 NP=1 %
*MACRO
 NT=2
 NC=1 X=1 Y=7 NAM=biaxl
 NC=2 X=2 Y=8 NAM=biaxt  %
*END
```

The following figure was obtained from the x-y plot data file produced by the present example.



6.11 Example K: User Defined Inelastic Material Model

Sample Input File For A User Defined Material Model

Problem Summary:

Load Type:	Thermomechanical
Load History:	Cyclic
Load Control:	Strain
Load History Data:	Cool-down from 400^oC to 23^oC , hold temperature during mechanical loading $\dot{T}$=0.419 ^{o}C/sec $\dot{\varepsilon}$ =1.667x10^{-4} /sec, ε_{max}=0.015, ε_{min}=0. $\Delta t_{thermal}$= 0.5 sec.; Δt_{mech} =0.05 sec.
Micromechanics model:	Double Periodicity
Fiber Packing Arrangement:	User Input 4x2 RVE with 2 fibers
Integration Algorithm:	Forward Euler
Constituent Material Model:	Fiber 1: User Model (elastic) Fiber 2: User Model (elastic) Matrix: User Model (Bodner-Partom via USRMAT)
Constituents:	Fiber 1: Imaginary material, user function material properties. Elastic modulus is a function of longitudinal strain: E_{new} = 700- 80.(ε_{11})($E_{previous}$) with units of MPa
	Fiber 2: Imaginary material, temperature dependent properties input manually.
	Matrix: Imaginary material, user function temperature dependent material properties. (see USRFUN subroutine)

☞ **Note:** The purpose here is to demonstrate how to use the various user definable subroutines, i.e., **USRMAT, USRFUN, USRFORMDE, USRCPEVAL** see section **4.2.12** for more information.

```
test of user subroutines/input
*PRINT
 NPL=10 %
*LOAD
 LCON=3 LOP=1 LSS=1 %
*MECH
 NPTW=3 TI=0.,900.,990. LO=0.,0.,0.015 %
*THERM
 NPTT=3 TI=0.,900.,990. TE=400.,23.,23. %
*MODEL
 MOD=1 %
*SOLVER
 NTF=1 NPTS=3 TIM=0.,900.,990 STP=0.5,0.05 %
*FIBER
 NFIBS=2
 NF=1 MF=99 NDPT=2 MAT=U IFM=2
 NF=2 MF=99 NDPT=2 MAT=U IFM=1 NPE=2 NPV=0
 NTP=3
 TEM=21.,200.,400.
 E1=314.1E9,293.2E9,253.0E9
 E2=0.33,0.33,0.33
 ALPA=4.5E-6,5.3E-6,6.1E-6
 ALPT=4.5E-6,5.3E-6,6.1E-6
*MATRIX
 NMATX=1
 NM=1 MM=99 NDPT=2 MAT=U IFM=2 %
*MRVE
 IDP=99
 NB=4 NG=2
 H=1.,1.,1.,1.
 L=1.,1.
 CM=M1,M1
 CM=F2,M1
 CM=M1,M1
 CM=F1,M1
*CURVE
 NP=2 %
*MACRO
 NT=1
 NC=1 X=1 Y=7 NAM=apdx-k
*END
```

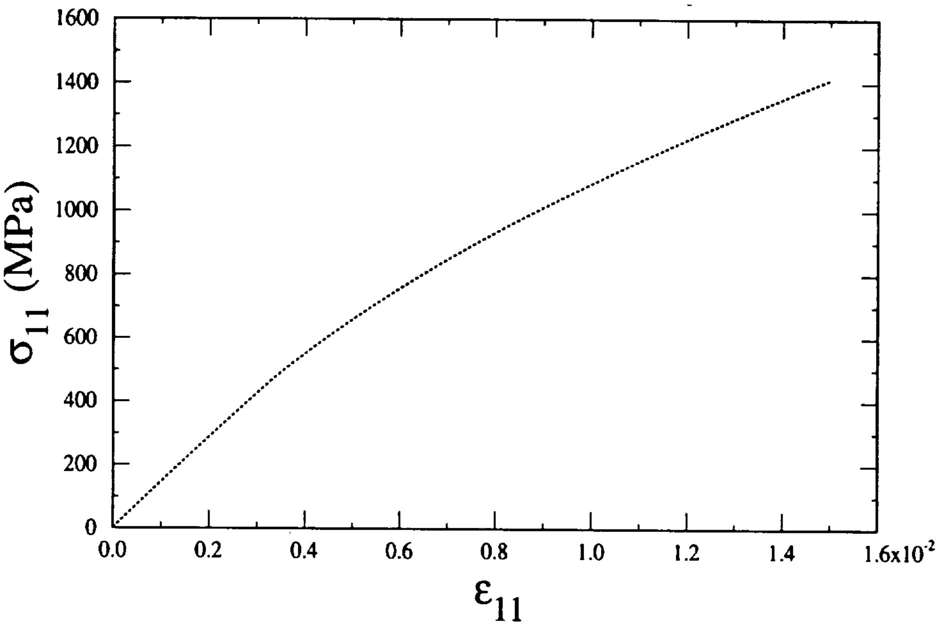


Figure was obtained from the x-y plot data file produced by the present example.

The USRMAT subroutine

The USRMAT subroutine is used here to implement the Bodner-Partom Visco-plastic model currently available in **MAC/GMC**

```
c#######################################################################
      SUBROUTINE USRMAT(DSA, SA, PE, PV, D, LOCTISO, TIME, TSTEP,
     &     CTEMP, DTEMPR, NIO, NE, NV, NS, MN, CDUM, DMGF,
     &     NEP, NVP, NSASIZE)

c     purpose:  user material constitutive model for determination of
c               the inelastic strain and state variable rates
c               (used when ncmd = 99)

      IMPLICIT DOUBLE PRECISION (A - H, O - Z)


      CHARACTER*2 CDUM
      DIMENSION SS(6), S(6), R(6)
      DIMENSION DSA(NSASIZE), SA(NSASIZE)
      DIMENSION PV(NVP), PE(NEP), D(3)

************************************************************************
c  note: 1) in this subroutine, [SA] and [DSA] contain the
c           micro (subcell) quantities for aboudi's micromechanics model
c
c        2) arrangement of [dsa] & [sa]  arrays:
c           variable             location
c         +-------------------------------------------------------------
c         | strain rate          (1-6)  (contains ENGINEERING shears)
c         |-------------------------------------------------------------
c         | stress rate          (7-12)
c         |-------------------------------------------------------------
c         | inelastic
c         | strain rate          (13-18)  (contains ENGINEERING shears)
c         |-------------------------------------------------------------
c         | 12 "slots"           (19-30)
c         | for state variables
c         |-------------------------------------------------------------
c         | thermal strain rate  (31-36)
c         +-------------------------------------------------------------
************************************************************************
c   on entry:
c        SA        - vector of total (integrated) quantities (see above)
c        PE(NE)    - vector of elastic constants for material MN
c                    (where NE = # of elastic constants --> 9 MAX)
c        PV(NV)    - vector of viscoplastic constants for material MN
c                   (where NV = # of viscoplastic constants --> 19 MAX)
c        D(3)      - vector of direction cosines (for models 3, 7, & 9)
c        LOCTISO   - flag indicating if ANY material exhibits local
C                    transverse isotropy (and global anisotropy)
c                  = 0 - all materials are at most globally transversely
c                        isotropic (D not used)
```

```
c                   = 1 - at least one material is locally transversely
c                         isotropic (D used)
c        TIME       - current time
c        TSTEP      - current time step
c        CTEMP      - current temperature
c        DTEMPR     - time rate of change of temperature
c        NIO        - unit number of .out file
c        NE         - # of elastic constants --> 9 MAX
c        NV         - # of viscoplastic constants --> 19 MAX
c        NS         - subcell number
c        MN         - material number
c        CDUM       - material character/number designation
c                     (i.e. F1 = fiber #1)
c        DMGF       - damage factor - if damage is included the user
c                     should multiply material stiffness terms by DMGF
c                     when using such terms in his inelastic model.
c
c   expected on exit:
c        DSA        - vector of rate quantities (see above)
************************************************************************


      IF( NE .GT. NEP ) THEN
         CALL FATALERROR(NIO)
         WRITE(NIO, *) 'TO MANY ELASTIC PROPERTIES FOR MATERIAL # ', MN
         WRITE(NIO, *) '    # USED      = ', NE
         WRITE(NIO, *) '    # ALLOCATED = ', NEP
         STOP
      ENDIF

      IF( NV .GT. NVP ) THEN
         CALL FATALERROR(NIO)
         WRITE(NIO, *) 'TO MANY NON-LINEAR PROPERTIES FOR MATERIAL # ',
     &         MN
         WRITE(NIO, *) '    # USED      = ', NV
         WRITE(NIO, *) '    # ALLOCATED = ', NVP
         STOP
      ENDIF


************************************************************************
*                           BEGIN USER EDITS                           *
************************************************************************
C      WRITE(6, *) ' PV=', PV
C      WRITE(6, *) ' PE=', PE

c-----------------------------------------------------------------------
c   MATERIALS #1 & #2 ARE ELASTIC
c    (set inelastic strain rates to zero)
c-----------------------------------------------------------------------
      IF ((MN .EQ. 1) .OR. (MN .EQ. 2)) THEN
         DSA(13) = 0
         DSA(14) = 0
         DSA(15) = 0
         DSA(16) = 0
         DSA(17) = 0
         DSA(18) = 0
         RETURN
      ENDIF

c------------------------------------------------------------------------
c   MATERIAL #3 --> USE BODNER-PARTOM
c------------------------------------------------------------------------

c------------------------------------------------------------------------
c  copy appropriate viscoplastic material constants
c------------------------------------------------------------------------

      IF( NV .LT. 6 ) THEN
         CALL FATALERROR(NIO)
         WRITE(NIO, *) ' NOT ENOUGH PV SPACE: NV =', NV
         STOP
      ENDIF
      D0 = PV(1)
      Z0 = PV(2)
      Z1 = PV(3)
      BM = PV(4)
      AN = PV(5)
      Q = PV(6)
c------------------------------------------------------------------------
c  copy stress from [sa] to [ss]
c------------------------------------------------------------------------
      SS(1) = SA(7)
      SS(2) = SA(8)
      SS(3) = SA(9)
      SS(4) = SA(10)
      SS(5) = SA(11)
      SS(6) = SA(12)
c------------------------------------------------------------------------
c  compute the deviatoric stress [s] in the subcell
c------------------------------------------------------------------------
      TEMP = (SS(1) + SS(2) + SS(3)) / 3.0
      S(1) = SS(1) - TEMP
      S(2) = SS(2) - TEMP
      S(3) = SS(3) - TEMP
      S(4) = SS(4)
      S(5) = SS(5)
      S(6) = SS(6)
c------------------------------------------------------------------------
c------------------------------------------------------------------------
      AJ2 = 0.5 * (S(1)**2 + S(2)**2 + S(3)**2) + S(4)**2 + S(5)**2 +
     &     S(6)**2
      SQ3AJ = DSQRT( SS(1)**2 + SS(2)**2 + SS(3)**2 + 2 * (SS(4)**2 +
     &     SS(5)**2 + SS(6)**2) )
      SQ2 = 1.414215
      IF (SQ3AJ .EQ. 0.0) THEN
         CALL ZEROR(R, 6)
      ELSE
```

```
         R(1) = SS(1) / SQ3AJ
         R(2) = SS(2) / SQ3AJ
         R(3) = SS(3) / SQ3AJ
         R(4) = SQ2 * SS(4) / SQ3AJ
         R(5) = SQ2 * SS(5) / SQ3AJ
         R(6) = SQ2 * SS(6) / SQ3AJ
      ENDIF
c-----------------------------------------------------------------------
c  if d0=0 then assume elastic and zero-out
c  [dsa(13-30)] (inelastic strain rate and
c  internal variable rates), then return
c-----------------------------------------------------------------------
      IF (D0 .EQ. 0) THEN
         DO 100 JJ = 13, 30
            DSA(JJ) = 0.0
  100    CONTINUE
         RETURN
c-----------------------------------------------------------------------
c  inelastic
c-----------------------------------------------------------------------
      ELSE
         ZEF = Z0 + Q * SA(20) + (1 - Q) * (R(1) * SA(21) + R(2) *
     &           SA(22) + R(3) * SA(23) + R(4) * SA(24) + R(5)
     &           * SA(25) + R(6) * SA(26))

         IF (AJ2 .EQ. 0.0) THEN
            AL = 0.0
         ELSE
            ARG1 = ZEF**2.0 / (3.0 * AJ2)

            IF (ARG1 .GT. 1E6) ARG1 = 1E6
            CON = .5 * (AN + 1.0) / AN
            ARG = CON * (ARG1)**AN

            IF (ARG .GT. 50.0) ARG = 50.0
            AL = D0 / (DEXP(ARG) * DSQRT(AJ2))
         ENDIF
c-----------------------------------------------------------------------
c  inelastic strain rates
c-----------------------------------------------------------------------
         DSA(13) = AL * S(1)
         DSA(14) = AL * S(2)
         DSA(15) = AL * S(3)
         DSA(16) = 2 * AL * S(4)
         DSA(17) = 2 * AL * S(5)
         DSA(18) = 2 * AL * S(6)

c-----------------------------------------------------------------------
c  plastic work rate
c-----------------------------------------------------------------------
         WPD = S(1) * DSA(13) + S(2) * DSA(14) + S(3) * DSA(15) + S(4) *
     &           DSA(16) + S(5) * DSA(17) + S(6) * DSA(18)
c-----------------------------------------------------------------------
c  state variable rates
```

```
c-----------------------------------------------------------------------
         DSA(19) = WPD
         Z0M = BM / Z0
         ZD = Z0M * (Z1 - ZEF) * WPD
         DSA(20) = ZD

         DSA(21) = ZD * R(1)
         DSA(22) = ZD * R(2)
         DSA(23) = ZD * R(3)
         DSA(24) = ZD * R(4)
         DSA(25) = ZD * R(5)
         DSA(26) = ZD * R(6)
      ENDIF

************************************************************************
*                         END USER EDITS                               *
************************************************************************

      RETURN
      END
```

The USRFUN subroutine

The USRFUN subroutine is used here to allow material properties to be entered in functional form instead of having to be linearly interpolated within **MAC/GMC**

```
c#######################################################################
      SUBROUTINE USRFUN(MN, TIME, TSTEP, CTEMP, DTEMPR, SA, DSA,
     &     DOLD, PEM, PVM, D, LOCTISO, ALPA, ALPT,
     &     NE, NV, NMTS, NEP, NVP, NSASIZE)

c     purpose:  user subroutine to allow elastic and viscoplastic
c               material properties to be functions of TEMP or
c               field variables.  Used for user defined functional
c               form material properties, that is, when:
c               (mat .eq. ‘U’) .and. (ifm .eq. 2)
c
c     note:     can be used in conjunction with a provided material
c               constitutive model, or a constitutive model input
c               by the user in USRMAT

      IMPLICIT DOUBLE PRECISION (A - H, O - Z)

      DIMENSION DOLD(6, 6), D(3)

      DIMENSION PEM(NEP, NMTS), PVM(NVP, NMTS)
      DIMENSION ALPA(NMTS), ALPT(NMTS)
      DIMENSION DSA(NSASIZE), SA(NSASIZE)
```

```
************************************************************************
c  note: 1) in this subroutine, [SA] and [DSA] contain the
c           micro (subcell) quantities for aboudi's micromechanics model
c
c        2) arrangement of [dsa] & [sa]  arrays:
c           variable             location
c          +-----------------------------------------------------------
c          | strain rate          (1-6)  (contains ENGINEERING shears)
c          |-----------------------------------------------------------
c          | stress rate          (7-12)
c          |-----------------------------------------------------------
c          | inelastic
c          | strain rate          (13-18)  (contains ENGINEERING shears)
c          |-----------------------------------------------------------
c          | 12 "slots"           (19-30)
c          | for state variables
c          |-----------------------------------------------------------
c          | thermal strain rate  (31-36)
c          +-----------------------------------------------------------
************************************************************************
c NOTE:  quantities in [SA] and [DSA] are SUBCELL quantities - the
c        values on entry are for the first subcell containing material
c        # MN - the values on exit of this subroutine will be applied to
c        ALL SUBCELLS containing material # MN.  It is thus recommended
c        that, if using the field variables, you assign the appropriate
c        material # to ONE SUBCELL ONLY.  Use of [SA] and [DSA] in this
c        context in conjunction with bending in laminate theory will
c        result in erroneous results as field variables become dependent
c        on through-thickness position (while the material # does not).
************************************************************************
c   on entry:
c      MN           - material number
c      TIME         - current time
c      TSTEP        - current time step
c      CTEMP        - current temperature
c      DTEMPR       - time rate of change of temperature
c      SA           - vector of total (integrated) quantities (see above)
c      DSA          - vector of rate quantities (see above)
c      DOLD(6, 6)   - previous elastic material stiffness matrix
c      PEM(NE, MN)  - vector of previous elastic constants for material
c                     # MN (where NE = # of elastic constants --> 9 MAX)
c      PVM(NV, MN)  - vector of previous viscoplastic constants for
c                     material # MN
c                     (where NV = # of viscoplastic constants --> 19 MAX)
c
c   expected on exit:
c      PEM(NE, MN)  - vector of current elastic constants for material MN
c      PVM(NV, MN)  - vector of current viscoplastic constants for
c                     material MN
c      D(3)         - vector of direction cosines
c                    (required for models 3, 7, & 9)
c      LOCTISO      - flag indicating if ANY material exhibits local
c                     transverse isotropy (and global anisotropy)
c                   = 0 - all materials are at most globally transversely
```

```
c                        isotropic (D not used)
c                  = 1 - at least one material is locally transversely
c                        isotropic (D used)
c      ALPA(MN)    - longitudinal cte for material MN
c      ALPT(MN)    - transverse cte for material MN
c           NE     - NUMBER OF ELASTIC PROPERTIES USED
c           NV     - NUMBER OF VISCOPLASTIC PROPERTIES USED
c
************************************************************************


************************************************************************
*                      BEGIN USER EDITS                                *
************************************************************************
C      WRITE(6, *) 'TOP OF USRFUN'

c-----------------------------------------------------------------------
c  MATERIAL # 1:  E = FUNCTION OF STRAIN & PREVIOUS E
c                 FOR USE WITH USER CONSTITUTIVE MODEL
c-----------------------------------------------------------------------
      IF (MN .EQ. 1) THEN

         IF (SA(1) .GT. 0) THEN
            EAOLD = PEM(1, MN)
            EA = 700.E9 - EAOLD * SA(1) * 80.0
         ELSE
            EA = 700.E9
         ENDIF

         ET = EA
         FNA = 0.41
         FNT = 0.41
         GA = EA / (2.0 * (1.0 + FNA))
         ALPA(MN) = 4.5E-6
         ALPT(MN) = 4.5E-6

         NE = 5
         PEM(1, MN) = EA
         PEM(2, MN) = ET
         PEM(3, MN) = FNA
         PEM(4, MN) = FNT
         PEM(5, MN) = GA

c-----------------------------------------------------------------------
c  MATERIAL #  2:  PROPERTIES = INTERPOLATED AT INPUT TEMPERATURES
c                  THIS SUBROUTINE IS NOT CALLED FOR MN #2 SINCE THE
c                  MATERIAL PROPERTIES ARE NOT FUNCTIONAL FORM
c-----------------------------------------------------------------------


c-----------------------------------------------------------------------
c  MATERIAL # 3:  PROPERTIES = LINEAR FUNCTION OF TEMPERATURE
c                 FOR USE WITH USER CONSTITUTIVE MODEL
c-----------------------------------------------------------------------
```

```
      ELSEIF (MN .EQ. 3) THEN

         IF (CTEMP .LT. 21.0) CTEMP = 21.0
         IF (CTEMP .GT. 400.) CTEMP = 400.

C -- ELASTIC
         E = 72.4E9 - 81.53E6 * (CTEMP - 21.)
         FN = 0.33 + 7.916E-5 * (CTEMP - 21.)
         ALP = 22.5E-6 + 3.958E-9 * (CTEMP - 21.)

         NE = 5
         PEM(1, MN) = E
         PEM(2, MN) = E
         PEM(3, MN) = FN
         PEM(4, MN) = FN
         PEM(5, MN) = E / (2. * (1. + FN))
         ALPA(MN) = ALP
         ALPT(MN) = ALP

C -- VISCOPLASTIC
         D0 = 1.E4
         Z0 = 340.E6
         Z1 = 435.E6
         BM = 300.0
         AN = 10.0 - 0.02493 * (CTEMP - 21.)
         Q  = 1.0

         NV = 6
         PVM(1, MN) = D0
         PVM(2, MN) = Z0
         PVM(3, MN) = Z1
         PVM(4, MN) = BM
         PVM(5, MN) = AN
         PVM(6, MN) = Q


      ENDIF
************************************************************************
*                        END USER EDITS                                *
************************************************************************

      RETURN
      END
```

The USRFORMDE subroutine

The USRFORMDE subroutine is used here to form the elastic material stiffness matrix in **MAC/GMC** when a user defined constitutive model is used.

```
c#######################################################################
      SUBROUTINE USRFORMDE(MN, PEM, PVM, D, LOCTISO, DNEW,
     &     NE, NV, NEP, NVP, NMTS)

c     purpose:  user subroutine to allow formation of material stiffness
c               matrices based on a user constitutive model, which may use
c               arbitrary material properties (used when ncmd = 99)
c

      IMPLICIT DOUBLE PRECISION (A - H, O - Z)

      DIMENSION DNEW(6, 6)
      DIMENSION PEM(NEP, NMTS), PVM(NVP, NMTS)
      DIMENSION D(3)

************************************************************************
c   on entry:
c      MN          - material number
c      PEM(NE, MN) - vector of elastic constants for material
c                     # MN (where NE = # of elastic constants --> 9 MAX)
c      PVM(NV, MN) - vector of viscoplastic constants for material # MN
c                     (where NV = # of viscoplastic constants --> 19 MAX)
c      D(3)        - vector of direction cosines
c                    (required for models 3, 7, & 9)
c      LOCTISO     - flag indicating if ANY material exhibits local
C                     transverse isotropy (and global anisotropy)
c                  = 0 - all materials are at most globally transversely
c                         isotropic (D not used)
c                  = 1 - at least one material is locally transversely
c                         isotropic (D used)
c
c   expected on exit:
c      DNEW(6, 6)  - current elastic material stiffness matrix

************************************************************************
*                      BEGIN USER EDITS                                *
************************************************************************
C      WRITE(6, *) 'TOP OF USRFORMDE'

C NOTE:  In the examples shown here, standard engineering material
C        elastic constants (E, nu) are used.  However, the user is
C        free to employ any material elastic constants with his
C        constitutive model (i.e., bulk modulus, etc.).  Thus, the
C        user must provide the equations to determine the stiffness
C        components required by GMC from his elastic constants.

c-----------------------------------------------------------------------
c  MATERIAL # 1:  E = FUNCTION OF STRAIN & PREVIOUS E
```

```
c                  FOR USE WITH USER CONSTITUTIVE MODEL
c-----------------------------------------------------------------------
      IF (MN .EQ. 1) THEN

         EA = PEM(1, MN)
         ET = PEM(2, MN)
         FNA = PEM(3, MN)
         FNT = PEM(4, MN)
         GA = PEM(5, MN)

         DO 100 I = 1, 6
            DO 100 J = 1, 6
               DNEW(I, J) = 0
 100     CONTINUE

         GT = 0.5 * ET / (1 + FNT)
         FK = 0.25 * EA / (0.5 * (1 - FNT) * (EA / ET) - FNA**2)

         DNEW(1, 1) = EA + 4.0 * FK * FNA**2
         DNEW(2, 1) = 2.0 * FK * FNA
         DNEW(3, 1) = 2.0 * FK * FNA

         DNEW(1, 2) = 2.0 * FK * FNA
         DNEW(2, 2) = FK + GT
         DNEW(3, 2) = FK - GT

         DNEW(1, 3) = 2.0 * FK * FNA
         DNEW(2, 3) = FK - GT
         DNEW(3, 3) = FK + GT

         DNEW(4, 4) = GT
         DNEW(5, 5) = GA
         DNEW(6, 6) = GA

c-----------------------------------------------------------------------
c  MATERIAL #   2:  PROPERTIES = INTERPOLATED AT TEMPERATURES
c                   FOR USE WITH USER CONSTITUTIVE MODEL
c-----------------------------------------------------------------------
      ELSEIF (MN .EQ. 2) THEN

         E = PEM(1, MN)
         FN = PEM(2, MN)
         GA = E / (2. * (1. + FN))

         DO 320 I = 1, 6
            DO 320 J = 1, 6
               DNEW(I, J) = 0
 320     CONTINUE

         GT = 0.5 * E / (1 + FN)
         FK = 0.25 * E / (0.5 * (1 - FN) - FN**2)

         DNEW(1, 1) = E + 4.0 * FK * FN**2
         DNEW(2, 1) = 2.0 * FK * FN
         DNEW(3, 1) = 2.0 * FK * FN

         DNEW(1, 2) = 2.0 * FK * FN
         DNEW(2, 2) = FK + GT
         DNEW(3, 2) = FK - GT

         DNEW(1, 3) = 2.0 * FK * FN
         DNEW(2, 3) = FK - GT
         DNEW(3, 3) = FK + GT

         DNEW(4, 4) = GT
         DNEW(5, 5) = GA
         DNEW(6, 6) = GA

c-----------------------------------------------------------------------
c  MATERIAL # 3:  PROPERTIES = LINEAR FUNCTION OF TEMPERATURE
c                 FOR USE WITH USER CONSTITUTIVE MODEL
c-----------------------------------------------------------------------
      ELSEIF (MN .EQ. 3) THEN

         E = PEM(1, MN)
         FN = PEM(3, MN)
         GA = PEM(5, MN)

         DO 300 I = 1, 6
            DO 300 J = 1, 6
               DNEW(I, J) = 0
 300     CONTINUE

         GT = 0.5 * E / (1 + FN)
         FK = 0.25 * E / (0.5 * (1 - FN) - FN**2)

         DNEW(1, 1) = E + 4.0 * FK * FN**2
         DNEW(2, 1) = 2.0 * FK * FN
         DNEW(3, 1) = 2.0 * FK * FN

         DNEW(1, 2) = 2.0 * FK * FN
         DNEW(2, 2) = FK + GT
         DNEW(3, 2) = FK - GT

         DNEW(1, 3) = 2.0 * FK * FN
         DNEW(2, 3) = FK - GT
         DNEW(3, 3) = FK + GT

         DNEW(4, 4) = GT
         DNEW(5, 5) = GA
         DNEW(6, 6) = GA

      ENDIF
************************************************************************
*                        END USER EDITS                                *
************************************************************************
      RETURN
      END
```

The USRCPEVAL subroutine

The USRCPEVAL subroutine is used here to form the time derivative of the material stiffness matrix in **MAC/GMC**

```
c#########################################################################
      SUBROUTINE USRCPEVAL(DSA, SA, MN, TIME, TSTEP, CTEMP, DTEMPR,
     &     DNEW, DOLD, PEM, PVM, D, LOCTISO, ALPA, ALPT, DDOT,
     &     NE, NV, NMTS, NEP, NVP, NSASIZE)


c     purpose:  user subroutine to allow formation of the TIME
c               derivative of the material stiffness matrix.
c               this subroutine is used when:
c               a) material properties are user defined and functional
c                  form.  That is:  (mat .eq. 'U') .and. (ifm .eq. 2)
c               b) the constitutive model is user-defined, and the
c                  material properties are not functional form, and
c                  the material properties are temperature-dependent
c                  that is: (ncmd .eq. 99) .and. (ifm .ne. 2) .and.
c                  (ndpt .eq. 2)

      IMPLICIT DOUBLE PRECISION (A - H, O - Z)


      DIMENSION DNEW(6, 6), DOLD(6, 6)
      DIMENSION DDOT(6, 6)

      DIMENSION PEM(NEP, NMTS), PVM(NVP, NMTS)
      DIMENSION ALPA(NMTS), ALPT(NMTS)
      DIMENSION DSA(NSASIZE), SA(NSASIZE)

************************************************************************
c  note: 1) in this subroutine, [SA] and [DSA] contain the
c           micro (subcell) quantities for aboudi's micromechanics model
c
c        2) arrangement of [dsa] & [sa]  arrays:
c           variable               location
c         +-----------------------------------------------------------
c         | strain rate            (1-6)  (contains ENGINEERING shears)
c         |-----------------------------------------------------------
c         | stress rate            (7-12)
c         |-----------------------------------------------------------
c         | inelastic
c         | strain rate            (13-18)  (contains ENGINEERING shears)
c         |-----------------------------------------------------------
c         | 12 "slots"             (19-30)
c         | for state variables
c         |-----------------------------------------------------------
c         | thermal strain rate  (31-36)
c         +-----------------------------------------------------------
************************************************************************
c   on entry:
c      SA           - vector of total (integrated) quantities (see above)
```

```
c       DSA           - vector of rate quantities (see above)
c       MN            - material number
c       TIME          - current time
c       TSTEP         - current time step
c       CTEMP         - current temperature
c       DTEMPR        - time rate of change of temperature
c       DNEW(6, 6)    - current elastic material stiffness matrix
c       DOLD(6, 6)    - previous elastic material stiffness matrix
c       PEM(NE, MN)   - vector of current elastic constants for material MN
c                       (where NE = # of elastic constants --> 9 MAX)
c       PVM(NV, MN)   - vector of current viscoplastic constants for
c                       material MN
c                       (where NV = # of elastic constants --> 19 MAX)
c       D(3)          - vector of direction cosines
c                       (required for models 3, 7, & 9) local
c       LOCTISO       - flag indicating if ANY material exhibits
c                       transverse isotropy (and global anisotropy)
c                     = 0 - all materials are at most globally transversely
c                           isotropic (D not used)
c                     = 1 - at least one material is locally transversely
c                           isotropic (D used)
c       ALPA(MN)      - longitudinal cte for material MN
c       ALPT(MN)      - transverse cte for material MN
c
c    expected on exit:
c       DDOT(6, 6)   - derivative with respect to TIME of stiffness matrix
************************************************************************


************************************************************************
*                          BEGIN USER EDITS                            *
************************************************************************

      DIMENSION DHOLD(6, 6, 3), DNEW1(6, 6), PEM1(9, 10), PVM1(19, 10)
      DIMENSION ALPA1(10), ALPT1(10)

C      WRITE(6, *) 'TOP OF USRCPEVAL'


c-----------------------------------------------------------------------
c  MATERIAL # 1:  E = FUNCTION OF STRAIN & PREVIOUS E
c                 FOR USE WITH USER CONSTITUTIVE MODEL
c-----------------------------------------------------------------------
      IF (MN .EQ. 1) THEN

c --- method one - simply divide change in stiffness components by TSTEP
         DO 200 I = 1, 6
            DO 200 J = 1, 6
               DDOT(I, J) = (DNEW(I, J) - DOLD(I, J)) / TSTEP
  200    CONTINUE

c-----------------------------------------------------------------------
c  MATERIAL #  2:  PROPERTIES = INTERPOLATED AT TEMPERATURES
c                  FOR USE WITH USER CONSTITUTIVE MODEL
```

```
c-----------------------------------------------------------------------
      ELSEIF (MN .EQ. 2) THEN

c --- method 2 - if material props are only function of temperature,
c --- calculate stiffness at  CTEMP - 0.5 & CTEMP + 0.5, difference
c --- equals change per temp, multiply by DTEMPR equals change per time
         IF (CTEMP .LT. 21.0) CTEMP = 21.0
         IF (CTEMP .GT. 400.) CTEMP = 400.

         DO 300 I = 1, 6
            DO 300 J = 1, 6
               DO 300 K = 1, 2
                  DHOLD(I, J, K) = 0
 300     CONTINUE

         DO 420 K = 1, 2

            IF (K .EQ. 1) CTEMP1 = CTEMP - 0.5
            IF (K .EQ. 2) CTEMP1 = CTEMP + 0.5

c --- this subroutine interpolates properties for material MN at
c --- the temperature CTEMP1
            CALL INTTMP(MN, CTEMP1, PEM1, ALPA1, ALPT1, PVM1)

c --- calculate the stiffness components
            CALL USRFORMDE(MN, PEM1, PVM, D, LOCTISO, DNEW1,
     &           NE, NV, NEP, NVP, NMTS)
c --- note: for internal constitutive models (ncmd .ne. 99), use
c --- subroutine FORMDE to calculate stiffness components.
c --- syntax: CALL FORMDE(MN, PEM1, D)

            DO 480 I = 1, 6
               DO 480 J = 1, 6
                  DHOLD (I, J, K) = DNEW1(I, J)
  480       CONTINUE
  420    CONTINUE

         DO 520 I = 1, 6
            DO 520 J = 1, 6
               DDOT(I, J) = (DHOLD(I, J, 2) - DHOLD(I, J, 1)) * DTEMPR
  520    CONTINUE

c-----------------------------------------------------------------------
c  MATERIAL # 3:  PROPERTIES = LINEAR FUNCTION OF TEMPERATURE
c                 FOR USE WITH USER CONSTITUTIVE MODEL
c-----------------------------------------------------------------------
      ELSEIF (MN .EQ. 3) THEN

c --- method 2 - if material props are only function of temperature,
c --- calculate stiffness at  CTEMP - 0.5 & CTEMP + 0.5, difference
c --- equals change per temp, multiply by DTEMPR equals change per time
         IF (CTEMP .LT. 21.0) CTEMP = 21.0
         IF (CTEMP .GT. 400.) CTEMP = 400.
         DO 330 I = 1, 6
            DO 330 J = 1, 6
               DO 330 K = 1, 2
                  DHOLD(I, J, K) = 0
 330     CONTINUE

         DO 400 K = 1, 2

            IF (K .EQ. 1) CTEMP1 = CTEMP - 0.5
            IF (K .EQ. 2) CTEMP1 = CTEMP + 0.5

c --- determine material props at the appropriate temperature
            CALL USRFUN(MN, TIME, TSTEP, CTEMP1, DTEMPR, SA, DSA,
     &           DOLD, PEM1, PVM1, D, LOCTISO, ALPA1, ALPT1,
     &           NE, NV, NMTS, NEP, NVP, NSASIZE)

c --- calculate the stiffness components
            CALL USRFORMDE(MN, PEM1, PVM, D, LOCTISO, DNEW1,
     &           NE, NV, NEP, NVP, NMTS)
c --- note: for internal constitutive models (ncmd .ne. 99), use
c --- subroutine FORMDE to calculate stiffness components.
c --- syntax: CALL FORMDE(MN, PEM1, D)

            DO 450 I = 1, 6
               DO 450 J = 1, 6
                  DHOLD (I, J, K) = DNEW1(I, J)
  450       CONTINUE
  400    CONTINUE

         DO 500 I = 1, 6
            DO 500 J = 1, 6
               DDOT(I, J) = (DHOLD(I, J, 2) - DHOLD(I, J, 1)) * DTEMPR
  500    CONTINUE


      ENDIF
*************************************************************************
*                        END USER EDITS                                 *
*************************************************************************

      RETURN
      END
```

6.12 Example L: Fatigue Damage Analysis

Sample Input File For A Fatigue Damage Analysis

The following example is used to explain how to conduct fatigue damage analysis within **MAC/GMC**.

Problem Summary:

Load Type:	Mechanical
Load History:	Cyclic
Load Control:	Stress
Load Component:	11 - direction (in the fiber direction)
Load History Data: $\Delta D = 0.15$	$\sigma_{11_{max}} = 115.$ ksi, $\dot{\sigma}_{11} = 128.$ ksi/sec,
	$\Delta t_{constant}$= 0.01797 sec
Micromechanics model:	Double Periodic
Fiber Packing Arrangement:	Square Pack at 35% fiber volume ratio
Integration Algorithm:	Forward Euler
Constituent Material Model:	Elastic (fiber) and Bodner-Partom (matrix)
Constituents:	Fiber: SCS-6 (input manually) Matrix: Ti - 15 - 3 (at 800^{o}F, input manually)

☞ **Note:** A similar example problem is described in more detail in:
Wilt, T.E., Arnold, S. M., and Saleeb, A.F.: A Coupled/Uncoupled Computational Scheme for Deformation and Fatigue Damage Analysis of Unidirectional Metal Matrix Composites, Applications of Continuum Damage Mechanics to Fatigue and Fracture, ASTM STP 1315, D.L. McDowell, Ed., American Society for Testing and Materials, pp. 65-82, 1997.

The purpose here is to demonstrate how to use the ***DAMAGE** option.

```
Longitudinal fatigue analysis of square packed SCS6/Ti-15-3 system
*PRINT
 NPL=0%
*LOAD
 LCON=2 LOP=1 LSS=2 %
*MECH
 NPTW=3 TI=0.,0.89844,1.79688 LO=0.,115.,0. %
*MODEL
 MOD=1 %
*DAMAGE
 NCY=12 D=0.15 DMAX=1.0 FG=0 FL=1%
*SOLVER
 NTF=1 NPTS=3 TIM=0.,0.89844,1.79688 STP=0.01797,0.01797 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=1 MAT=U IFM=1 &
 EL=58.E3,58.E3,0.25,0.25,23.2E3,3.5E-6,3.5E-6
 ANG=1. BN=2. BP=4. OMU=1. OMFL=1. OMM=1. ETU=1. &
 ETFL=1. ETM=1. BE=1. A=1. SFL=1. XML=1.   &
 SU=1. SK=1
# Note sk=1 means skip fatigue damage for fiber, so damage properties are
# meaningless - but failure criteria is still applicable.
 T=1 IC=1 V=320. %
*MATRIX
 NMATX=1
 NM=1 MM=1 NDPT=1 MAT=U IFM=1 &
 EL=12770.,12770.,0.32,0.32,4837.12,21.06E-6,21.06E-6  &
 VI=1000.,120.,120.,10.,3.,1.
 ANG=0. BN=0. BP=0. OMU=1. OMFL=1. OMM=1. ETU=1. &
 ETFL=1. ETM=1. BE=2.27 A=0.2302 SFL=20.3 XML=900. &
 SU=128. SK=0
 T=2 IC=1 V=0.01 %
*MRVE
 IDP=1 VF=0.35 %
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=1 Y=7 NAM=apdxl %
*END
```

The following results is contained in the output file: **dam2.data**

```
 Calculated Cycles To Failure At Increment 100 Applied Load Cycle   1

Subcell #       Nf
    1      0.100000E+10
    2      0.613603E+05
    3      0.613603E+05
    4      0.664385E+05
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL            2 *****
 NF =>    60494.2250830768
 NF =>    60494.0000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =    60494.0000000000
 *** AFTER APPLIED LOAD CYCLE            1
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 200 Applied Load Cycle   2

Subcell #       Nf
    1      0.100000E+10
    2      0.116736E+04
    3      0.116736E+04
    4      0.545641E+04
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL            2 *****
 NF =>    674.890587838351
 NF =>    674.000000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =    61168.0000000000
 *** AFTER APPLIED LOAD CYCLE            2
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 300 Applied Load Cycle   3

Subcell #       Nf
    1      0.100000E+10
    2      0.709630E+03
    3      0.709630E+03
    4      0.438086E+04
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL            2 *****
 NF =>    419.320819753184
 NF =>    419.000000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =    61587.0000000000
 *** AFTER APPLIED LOAD CYCLE            3
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 400 Applied Load Cycle    4

Subcell #       Nf
   1       0.100000E+10
   2       0.463247E+03
   3       0.463247E+03
   4       0.361975E+04
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL             3 *****
 NF =>     307.516313605484
 NF =>     307.000000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =     61894.0000000000
 *** AFTER APPLIED LOAD CYCLE            4
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 500 Applied Load Cycle    5

Subcell #       Nf
   1       0.100000E+10
   2       0.295356E+03
   3       0.295356E+03
   4       0.301874E+04
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL             2 *****
 NF =>     232.831665300220
 NF =>     232.000000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =     62126.0000000000
 *** AFTER APPLIED LOAD CYCLE            5
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 600 Applied Load Cycle    6

Subcell #       Nf
   1       0.100000E+10
   2       0.166907E+03
   3       0.166907E+03
   4       0.253239E+04
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL             2 *****
 NF =>     158.292840626614
 NF =>     158.000000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =    62284.0000000000
 *** AFTER APPLIED LOAD CYCLE            6
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 700 Applied Load Cycle   7

Subcell #       Nf
    1      0.100000E+10
    2      0.620698E+02
    3      0.620698E+02
    4      0.214841E+04
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL           3 *****
 NF =>    62.0697834887934
 NF =>    62.0000000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =    62346.0000000000
 *** AFTER APPLIED LOAD CYCLE            7
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 800 Applied Load Cycle   8

Subcell #       Nf
    1      0.100000E+10
    2      0.735691E+01
    3      0.735691E+01
    4      0.195843E+04
 *****************************************
 *****************************************
 *****************************************
 ***** CONTROLLING SUBCELL           2 *****
 NF =>    7.35691370876420
 NF =>    7.00000000000000
 *****************************************
 *****************************************
 **** CURRENT TOTAL # CYCLES =    62353.0000000000
 *** AFTER APPLIED LOAD CYCLE            8
 *****************************************
 *****************************************
 Calculated Cycles To Failure At Increment 900 Applied Load Cycle   9

Subcell #       Nf
    1      0.100000E+10
    2      0.290197E+01
    3      0.290197E+01
    4      0.194075E+04
 *****************************************
 *****************************************
 *****************************************
```

```
***** CONTROLLING SUBCELL          2 *****
NF =>    2.90196867892969
NF =>    2.00000000000000
******************************************
******************************************
**** CURRENT TOTAL # CYCLES =    62355.0000000000
*** AFTER APPLIED LOAD CYCLE            9
******************************************
******************************************
Calculated Cycles To Failure At Increment *** Applied Load Cycle  10

Subcell #       Nf
    1      0.100000E+10
    2      0.202802E+01
    3      0.202802E+01
    4      0.193669E+04
******************************************
******************************************
******************************************
***** CONTROLLING SUBCELL          2 *****
NF =>    2.02801986635817
NF =>    2.00000000000000
******************************************
******************************************
**** CURRENT TOTAL # CYCLES =    62357.0000000000
*** AFTER APPLIED LOAD CYCLE           10
******************************************
******************************************
Calculated Cycles To Failure At Increment *** Applied Load Cycle  11

Subcell #       Nf
    1      0.100000E+10
    2     -0.999000E+03
    3     -0.999000E+03
    4      0.193110E+04
SUBCELL              2 FAILED ... SKIPPED
**** WARNING ****  DELN < 1 FOR CELL           2
SUBCELL              3 FAILED ... SKIPPED
**** WARNING ****  DELN < 1 FOR CELL           3
**** WARNING ****  DELN < 1 FOR CELL           4
SORTN: **** STOP ****  DELN < 1
```

At this point the matrix subcells have all failed and the analysis stops.

Thus the life of the composite is 62357 cycles.

The following results is contained in the output file: **dam3.data**

```
****************************
SUBCELL #           1
IDAM  F
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #           2
IDAM  F
CYCLES TO FAILURE XNF    61360.2956047440
****************************
****************************
SUBCELL #           3
IDAM  F
CYCLES TO FAILURE XNF    61360.2956047440
****************************
****************************
SUBCELL #           4
IDAM  F
CYCLES TO FAILURE XNF    66438.5151084627
****************************
****************************************
** Current Damage At Increment 100 Applied Load Cycle    1
SUBCELL #     D         1-D
   1  0.000000E+00   0.100000E+01
   2  0.149958E+00   0.850042E+00
   3  0.149958E+00   0.850042E+00
   4  0.632021E-03   0.999368E+00
****************************
SUBCELL #           1
IDAM  F
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #           2
IDAM  T
CYCLES TO FAILURE XNF    1167.36343322662
****************************
****************************
SUBCELL #           3
IDAM  T
CYCLES TO FAILURE XNF    1167.36343322663
****************************
****************************
SUBCELL #           4
IDAM  T
CYCLES TO FAILURE XNF    5456.41016344678
****************************
****************************************
** Current Damage At Increment 200 Applied Load Cycle    2
SUBCELL #     D         1-D
   1  0.000000E+00   0.100000E+01
   2  0.299638E+00   0.700362E+00
   3  0.299638E+00    0.700362E+00
   4  0.141115E-02    0.998589E+00
 ****************************
 SUBCELL #             1
 IDAM  T
 CYCLES TO FAILURE XNF    1000000000.00000
 ****************************
 ****************************
 SUBCELL #             2
 IDAM  T
 CYCLES TO FAILURE XNF    709.630361973297
 ****************************
 ****************************
 SUBCELL #             3
 IDAM  T
 CYCLES TO FAILURE XNF    709.630361973300
 ****************************
 ****************************
 SUBCELL #             4
 IDAM  T
 CYCLES TO FAILURE XNF    4380.85967672225
 ****************************
 ****************************************
 ** Current Damage At Increment 300 Applied Load Cycle   3
 SUBCELL #     D         1-D
   1  0.000000E+00    0.100000E+01
   2  0.449466E+00    0.550534E+00
   3  0.449466E+00    0.550534E+00
   4  0.243012E-02    0.997570E+00
 ****************************
 SUBCELL #             1
 IDAM  T
 CYCLES TO FAILURE XNF    1000000000.00000
 ****************************
 ****************************
 SUBCELL #             2
 IDAM  T
 CYCLES TO FAILURE XNF    463.247417312908
 ****************************
 ****************************
 SUBCELL #             3
 IDAM  T
 CYCLES TO FAILURE XNF    463.247417312908
 ****************************
 ****************************
 SUBCELL #             4
 IDAM  T
 CYCLES TO FAILURE XNF    3619.74747832020
 ****************************
 ****************************************
 ** Current Damage At Increment 400 Applied Load Cycle   4
 SUBCELL #     D         1-D
   1  0.000000E+00    0.100000E+01
   2  0.599071E+00    0.400929E+00
   3  0.599071E+00   0.400929E+00
   4  0.375666E-02   0.996243E+00
****************************
SUBCELL #            1
IDAM  T
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #            2
IDAM  T
CYCLES TO FAILURE XNF    295.355917088531
****************************
****************************
SUBCELL #            3
IDAM  T
CYCLES TO FAILURE XNF    295.355917088531
****************************
****************************
SUBCELL #            4
IDAM  T
CYCLES TO FAILURE XNF    3018.74014755274
****************************
***************************************
** Current Damage At Increment 500 Applied Load Cycle    5
SUBCELL #     D        1-D
   1  0.000000E+00   0.100000E+01
   2  0.748061E+00   0.251939E+00
   3  0.748061E+00   0.251939E+00
   4  0.539214E-02   0.994608E+00
****************************
SUBCELL #            1
IDAM  T
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #            2
IDAM  T
CYCLES TO FAILURE XNF    166.907472034450
****************************
****************************
SUBCELL #            3
IDAM  T
CYCLES TO FAILURE XNF    166.907472034450
****************************
****************************
SUBCELL #            4
IDAM  T
CYCLES TO FAILURE XNF    2532.39380892976
****************************
***************************************
** Current Damage At Increment 600 Applied Load Cycle    6
SUBCELL #     D        1-D
   1  0.000000E+00   0.100000E+01
   2  0.897013E+00   0.102987E+00
   3  0.897013E+00   0.102987E+00
   4  0.707338E-02   0.992927E+00
```

```
****************************
SUBCELL #            1
IDAM  T
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #            2
IDAM  T
CYCLES TO FAILURE XNF    62.0697834887973
****************************
****************************
SUBCELL #            3
IDAM  T
CYCLES TO FAILURE XNF    62.0697834887934
****************************
****************************
SUBCELL #            4
IDAM  T
CYCLES TO FAILURE XNF    2148.40603399676
****************************
***************************************
** Current Damage At Increment 700 Applied Load Cycle   7
SUBCELL #     D        1-D
   1  0.000000E+00   0.100000E+01
   2  0.987089E+00   0.129106E-01
   3  0.987089E+00   0.129106E-01
   4  0.796197E-02   0.992038E+00
****************************
SUBCELL #            1
IDAM  T
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #            2
IDAM  T
CYCLES TO FAILURE XNF    7.35691370876420
****************************
****************************
SUBCELL #            3
IDAM  T
CYCLES TO FAILURE XNF    7.35691370914608
****************************
****************************
SUBCELL #            4
IDAM  T
CYCLES TO FAILURE XNF    1958.42907053144
****************************
***************************************
** Current Damage At Increment 800 Applied Load Cycle   8
SUBCELL #     D        1-D
   1  0.000000E+00   0.100000E+01
   2  0.994882E+00   0.511765E-02
   3  0.994882E+00   0.511765E-02
   4  0.807672E-02   0.991923E+00
```

```
****************************
SUBCELL #            1
IDAM  T
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #            2
IDAM  T
CYCLES TO FAILURE XNF    2.90196867892969
****************************
****************************
SUBCELL #            3
IDAM  T
CYCLES TO FAILURE XNF    2.90196868203595
****************************
****************************
SUBCELL #            4
IDAM  T
CYCLES TO FAILURE XNF    1940.74760195980
****************************
***************************************
** Current Damage At Increment 900 Applied Load Cycle   9
SUBCELL #     D        1-D
   1  0.000000E+00   0.100000E+01
   2  0.996420E+00   0.357991E-02
   3  0.996420E+00   0.357991E-02
   4  0.811000E-02   0.991890E+00
****************************
SUBCELL #            1
IDAM  T
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #            2
IDAM  T
CYCLES TO FAILURE XNF    2.02801986635817
****************************
****************************
SUBCELL #            3
IDAM  T
CYCLES TO FAILURE XNF    2.02801987334260
****************************
****************************
SUBCELL #            4
IDAM  T
CYCLES TO FAILURE XNF    1936.69355062971
****************************
***************************************
** Current Damage At Increment *** Applied Load Cycle  10
SUBCELL #     D        1-D
   1  0.000000E+00   0.100000E+01
   2  0.999034E+00   0.966462E-03
   3  0.999034E+00   0.966462E-03
   4  0.814345E-02   0.991857E+00
```

```
****************************
SUBCELL #             1
IDAM  T
CYCLES TO FAILURE XNF    1000000000.00000
****************************
****************************
SUBCELL #             2
IDAM  T
CYCLES TO FAILURE XNF   -999.000000000000
****************************
****************************
SUBCELL #             3
IDAM  T
CYCLES TO FAILURE XNF   -999.000000000000
****************************
****************************
SUBCELL #             4
IDAM  T
CYCLES TO FAILURE XNF    1931.10439061903
*
```

6.13 Example M: Longitudinal Discrete Fiber Breakage

Sample Input File for Longitudinal Discrete Fiber Breakage

Problem Summary

Load Type:	Thermomechanical
Load History:	Cyclic
Load Control:	Strain
Load Component:	11-direction (in fiber direction)
Load History Data:	Cool-down from 900°C to 23°C, heat-up to 650°C, then hold temperature during mechanical loading $\dot{T}$=0.0152 °C/sec, 0.0871 °C/sec $\dot{\varepsilon}$=1.0x10^{-4}/sec , ε_{max}=0.012, ε_{min}=0. $\Delta t_{thermal}$= 500., 40. sec.; Δt_{mech} =0.4 sec.
Micromechanics Model:	Triple Periodicity
Fiber Packing Arrangement:	Square Pack, 35% fiber volume ratio
Repeating Unit Cell:	1x8x14 unit cell with 28 square fibers (input manually)
Integration Algorithm:	Forward Euler
Constituent Material Model:	Fiber: Elastic, transversely isotropic Matrix: GVIPS - isotropic form
Constituents:	Fiber: SCS-6 (temp. dep. properties input manually) Matrix: TIMETAL21S
Fiber Breakage:	Second implementation of debond model
Fiber Breakage Parameters:	taken from room temperature SCS-6 fiber strength histogram with strengths decreased by 5.8 % to account for elevated temperature effect.

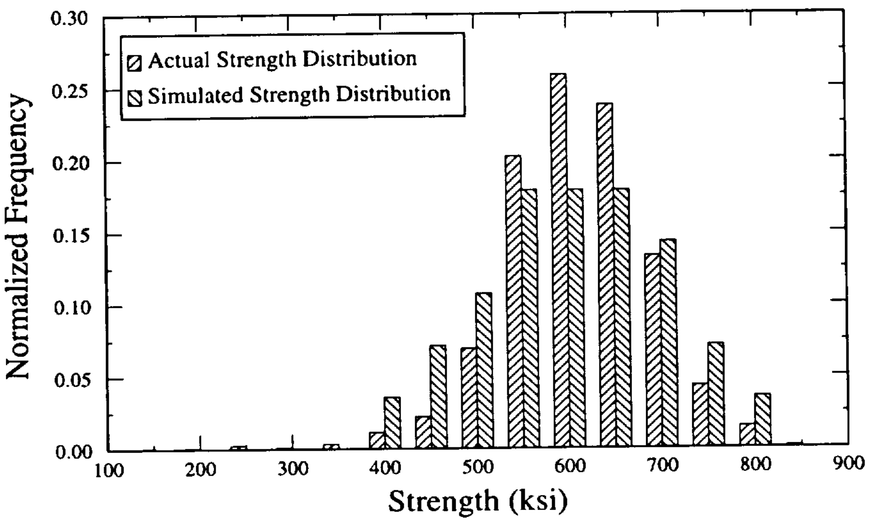
$$\Lambda = 1.0x10^{-5}\frac{in^3}{kip}, \quad B = 10.\frac{1}{s}$$

☞ **Note:** In the case of longitudinal discrete fiber breakage, each fiber is modeled with an internal weak interface oriented normal to the fiber direction encompassing the entire fiber cross-section. This interface is then given a strength, σ_{DB}, which corresponds to the fiber ultimate strength. Thus, during simulated loading, when the longitudinal stress in the fiber reaches σ_{DB}, the fiber's internal interface debonds, and the longitudinal stress in the fiber begins to unload. This simulates a local fiber failure in a real composite.

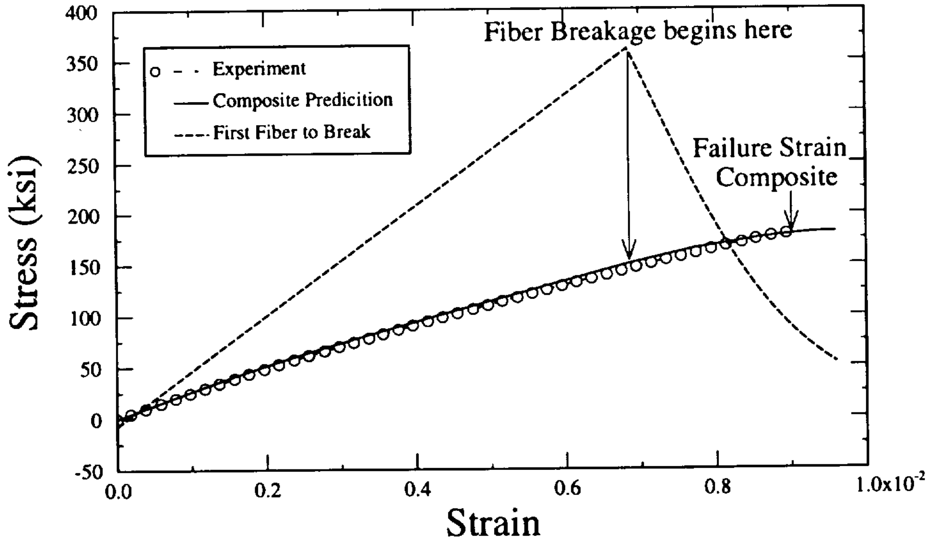
In order to model longitudinal discrete fiber breakage as realistically as possible, a subcell containing 28 fibers has been used. Square-shaped fibers have been employed because longitudinal behavior is insensitive to fiber shape, and square fibers require the smallest number of subcells. Strength data for the SCS-6 fiber was taken from a vendor-supplied histogram. The room-temperature simulated distribution, as well as the actual strength distribution are shown in the subsequent figure. Note that the strengths employed in the example (in which the tensile simulation is performed at 650°C) were reduced by 5.8 % from the room-temperature values to account for the effect of the elevated temperature. In addition, the fiber strengths have been distributed over the strength ranges indicated by the histogram, rather than bunched up. That is, rather than providing 5 fibers with a strength of 600 ksi, the strengths of those 5 fibers were distributed between 575 ksi and 625 ksi (prior to the 5.8 % reduction).

The results of this example shown in the subsequent figure have been plotted only to an applied strain level of 0.0096 because at this point the slope of the predicted global stress-strain curve becomes negative. We are treating this as a criterion for simulated global failure of the composite.

```
test of longitudinal discrete fiber breakage
*PRINT
 NPL=1  %
*LOAD
 LCON=3 LOP=1  LSS=1 %
*MECH
 NPTW=5 TI=0.,24000.,57600.,64800.,64920. LO=0.0,0.0,0.,0.,0.012  %
*THERM
 NPTT=5 TI=0.,24000.,57600.,64800.,64920. TE=900.,534.583,23.,650.,650. %
*MODEL
 MOD=2  %
*SOLVER
 NTF=1 NPTS=5 TIM=0.,24000.,57600.,64800.,64920. STP=500.,40.,40.,0.4 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=2 MAT=U IFM=1
 NTP=6
 TEM=21.1,204.44,315.56,426.67,537.78,871.11
 EA=57.0E3,55.98E3,55.4E3,54.82E3,54.24E3,53.36E3
 ET=57.0E3,55.98E3,55.4E3,54.82E3,54.24E3,53.36E3
 NUA=0.25,0.25,0.25,0.25,0.25,0.25
 NUT=0.25,0.25,0.25,0.25,0.25,0.25
 GA=22.8E3,22.392E3,22.16E3,21.928E3,21.696E3,21.344E3
 ALPA=3.564E-6,3.618E-6,3.726E-6,3.906E-6,4.068E-6,4.572E-6
 ALPT=3.564E-6,3.618E-6,3.726E-6,3.906E-6,4.068E-6,4.572E-6
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=2 MAT=A %
*MRVE
 IDP=99
 NA=1 NB=8 NG=14
 D=1.0
#NOTE: VF = 35%
 H=0.5916,0.4084,0.5916,0.4084,0.5916,0.4084,0.5916,0.4084
 L=0.5916,0.4084,0.5916,0.4084,0.5916,0.4084,0.5916,0.4084 &
   0.5916,0.4084,0.5916,0.4084,0.5916,0.4084
 CM=F1,M1,F1,M1,F1,M1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1
 CM=F1,M1,F1,M1,F1,M1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1
 CM=F1,M1,F1,M1,F1,M1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1
 CM=F1,M1,F1,M1,F1,M1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1
 CM=F1,M1,F1,M1,F1,M1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1
 CM=F1,M1,F1,M1,F1,M1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1
 CM=F1,M1,F1,M1,F1,M1,F1,M1
 CM=M1,M1,M1,M1,M1,M1,M1,M1
*DEBOND
 NII=28
 DBCH=2 NAI=1 NBI=1 NGI=1  FACE=1 BDN=357 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=1 NGI=3  FACE=1 BDN=403 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=1 NGI=5  FACE=1 BDN=469 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=1 NGI=7  FACE=1 BDN=565 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=1 NGI=9  FACE=1 BDN=496 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=1 NGI=11 FACE=1 BDN=503 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=1 NGI=13 FACE=1 BDN=612 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=3 NGI=1  FACE=1 BDN=450 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=3 NGI=3  FACE=1 BDN=655 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=3 NGI=5  FACE=1 BDN=603 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=3 NGI=7  FACE=1 BDN=512 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=3 NGI=9  FACE=1 BDN=407 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=3 NGI=11 FACE=1 BDN=517 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=3 NGI=13 FACE=1 BDN=556 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=5 NGI=1  FACE=1 BDN=716 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=5 NGI=3  FACE=1 BDN=593 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=5 NGI=5  FACE=1 BDN=631 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=5 NGI=7  FACE=1 BDN=640 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=5 NGI=9  FACE=1 BDN=678 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=5 NGI=11 FACE=1 BDN=584 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=5 NGI=13 FACE=1 BDN=488 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=7 NGI=1  FACE=1 BDN=575 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=7 NGI=3  FACE=1 BDN=697 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=7 NGI=5  FACE=1 BDN=754 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=7 NGI=7  FACE=1 BDN=546 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=7 NGI=9  FACE=1 BDN=537 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=7 NGI=11 FACE=1 BDN=664 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
 DBCH=2 NAI=1 NBI=7 NGI=13 FACE=1 BDN=622 GCN=0.00001 BCN=10. TOLN=1 &
 BDS=100 GCS=0.1 BCS=100 TI=0
*CURVE
 NP=3 %
*MACRO
 NT=2
 NC=1 X=1  Y=7 NAM=LONG-M
 NC=2 X=39 Y=1 NAM=LONG-T
*MICRO
 NT=3
 NC=1 CELL=1  X=1 Y=7 NAM=LONG-1
 NC=2 CELL=17 X=1 Y=7 NAM=LONG-2
 NC=3 CELL=33 X=1 Y=7 NAM=LONG-3 %
*END
```



Room Temperature SCS-6 Fiber Strength Histogram



Longitudinal Prediction of SCS-6/Timetal 21S with 35% fiber volume content at 650°C

6.14 Example N: Plain Weave Composite

<u>Sample Input File for Plain Weave Composite:</u> Step 1 - Determine Yarn Properties

<u>Problem Summary</u>

Load Type:	Mechanical	
Load Component:	33-direction (transverse to fiber)	
Load History:	Monotonic	
Load Control:	Strain	
Load History Data:	$\dot{\varepsilon}=1.0\text{x}10^{-4}$/sec, $\varepsilon_{max}=0.018$, $\varepsilon_{min}=0.$ Δt_{mech} = 4. sec.	
Micromechanics Model:	Double Periodicity	
Fiber Packing Arrangement:	Square Pack, R = 1., 65% fiber volume ratio	
Repeating Unit Cell:	26x26 circular fiber cross-section approx.	
Integration Algorithm:	Forward Euler	
Constituent Material Model:	Fiber:	Elastic, transversely isotropic
	Matrix:	Elastic, isotropic
Constituents:	Fiber:	AS-4 Graphite Fiber (properties input manually)
	Matrix:	PMR-15 Epoxy (properties input manually)

☞ **Note:** Step 1 of analyzing a woven composite with **MAC/GMC** involves determining the effective properties of the fiber/matrix yarns which reinforce the woven composites. For the elastic example shown, this is quite simple and involves determining only the five transversely isotropic elastic properties and the two transversely isotropic CTEs for the unidirectional fiber/matrix yarn. These effective properties will then be employed in step 2 (using ncmd = 9, transversely isotropic model) to analyze the actual woven composite unit cell. Inelastic analysis of woven composites is possible with **MAC/GMC**, but the inelastic characterization of the unidirectional fiber/matrix yarns (for implementation with one of the transversely isotropic inelastic constitutive models) becomes more complex. Future versions of **MAC/GMC** may include automated routines to aid in the analysis of thermo-inelastic woven composites.

```
analysis of a graphite/epoxy yarn
*PRINT
 NPL=3 %
*LOAD
 LCON=2 LOP=3 LSS=1 %
*MECH
 NPTW=2 TI=0.,108. LO=0.,0.018  %
*MODEL
 MOD=1  %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,108. STP=4. %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=1 MAT=U IFM=1&
 EL=31.E6,2.E6,0.2,0.25,2.E6,-0.55E-6,5.6E-6 %
*MATRIX
 NMATX=1
 NM=1 MM=6 NDPT=1 MAT=U IFM=1 &
 EL=0.5E6,0.5E6,0.41,0.41,0.1773E6,57.E-6,57.E-6 %
*MRVE
 IDP=13 VF=0.65 R=1.0
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=3 Y=9 NAM=YARN
*END
---------------------------------------------------------------------------------------------------
                                   Results for fiber/matrix yarn:

                             EFFECTIVE ENGINEERING MODULI
                                    E11S=  .203E+08
                                    N12S=  .272E+00
                                    E22S=  .134E+07
                                    N23S=  .432E+00
                                    E33S=  .134E+07
                                    G23S=  .359E+06
                                    G13S=  .596E+06
                                    G12S=  .596E+06

                        EFFECTIVE THERMAL EXPANSION COEFFICIENTS
                               .493E-07    .310E-04    .310E-04
```

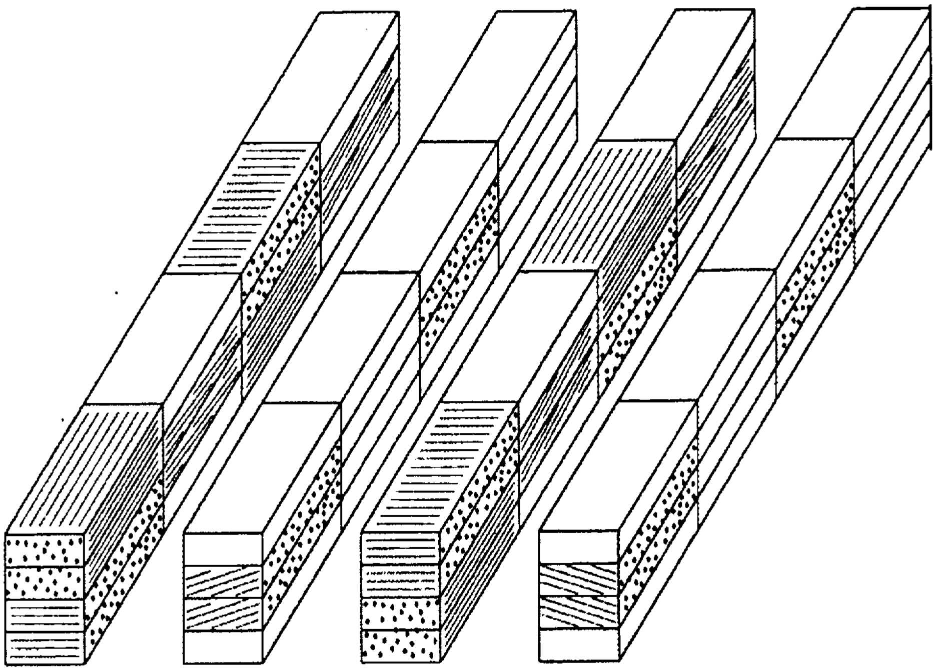
Sample Input File for Plain Weave Composite: Step 2 - Composite Behavior

Problem Summary

Load Type:	Mechanical
Load Component:	33-direction (in the plane of the woven reinforcement)
Load History:	Monotonic
Load Control:	Strain
Load History Data:	$\dot{\varepsilon}=1.0\text{x}10^{-4}$/sec , $\varepsilon_{max}=0.018$, $\varepsilon_{min}=0.$ $\Delta t_{mech} = 4.$ sec.
Micromechanics Model:	Triple Periodicity
Repeating Unit Cell:	Input manually - approximates a plain weave reinforced composite
Integration Algorithm:	Forward Euler
Constituent Material Model:	Fiber/Matrix Yarns: Elastic, locally transversely isotropic Pure Matrix: Elastic, isotropic
Constituents:	Fibers 1 - 4: V_f = 65% AS-4/PMR-15 yarns with different fiber orientations (properties input manually) Matrix 1: PMR-15 Epoxy (properties input manually) Matrix 2, 3: V_f = 65% AS-4/PMR-15 yarns with different fiber orientations (properties input manually)

☞ **Note**: **MAC/GMC** currently limits the number of fiber and matrix constituents to 4 each.

☞ **Note:** Step 2 of analyzing a woven composite with **MAC/GMC** involves assembling a repeating unit cell that represents the woven composite using the effective fiber/matrix yarn properties determined in step 1. The repeating unit cell employed for a plain weave composite in this example is shown in the figure below. The heterogeneous subcells in this figure are represented by effective fiber/matrix yarn properties, while the local fiber direction of these subcells is accounted for by the D vector in the locally transversely isotropic constitutive model.



User defined IDP for woven architecture (not drawn to scale)

```
plain weave reinforced composite
*PRINT
 NPL= 3%
*LOAD
 LCON=2 LOP=3 LSS=1 %
*MECH
 NPTW=2 TI=0.,108. LO=0.,0.018  %
*THERM
 NPTT=2 TI=0.,108. TE=23.,23.
*MODEL
 MOD=2  %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,108. STP=4. %
*FIBER
 NFIBS=4
 NF=1 MF=9 NDPT=1 MAT=U IFM=1 &
 EL=20.3E6,1.34E6,0.272,0.432,0.596E6,0.0493E-6,31.E-6 D=0.,0.,1. %
 NF=2 MF=9 NDPT=1 MAT=U IFM=1&
 EL=20.3E6,1.34E6,0.272,0.432,0.596E6,0.0493E-6,31.E-6 D=0.,1.,0. %
 NF=3 MF=9 NDPT=1 MAT=U IFM=1&
 EL=20.3E6,1.34E6,0.272,0.432,0.596E6,0.0493E-6,31.E-6 D=0.25,1.,0. %
 NF=4 MF=9 NDPT=1 MAT=U IFM=1&
 EL=20.3E6,1.34E6,0.272,0.432,0.596E6,0.0493E-6,31.E-6 D=-0.25,1.,0. %
*MATRIX
 NMATX=3
 NM=1 MM=6 NDPT=1 MAT=U IFM=1 &
 EL=0.5E6,0.5E6,0.41,0.41,0.1773E6,57.E-6,57.E-6 %
 NM=2 MM=9 NDPT=1 MAT=U IFM=1 &
 EL=20.3E6,1.34E6,0.272,0.432,0.596E6,0.0493E-6,31.E-6 D=0.25,0.,1. %
 NM=3 MM=9 NDPT=1 MAT=U IFM=1 &
 EL=20.3E6,1.34E6,0.272,0.432,0.596E6,0.0493E-6,31.E-6 D=-0.25,0.,1. %
*MRVE
 IDP=99
 NA=4 NB=4 NG=4
 D=0.25,0.25,0.25,0.25
 H=1.,1.,1.,1.
 L=1.,1.,1.,1.
 CM=F1,M1,F2,M1
 CM=F1,F3,F2,F4
 CM=F2,F3,F1,F4
 CM=F2,M1,F1,M1
 CM=M1,M1,M1,M1
 CM=M3,M1,M2,M1
 CM=M3,M1,M2,M1
 CM=M1,M1,M1,M1
 CM=F2,M1,F1,M1
 CM=F2,F4,F1,F3
 CM=F1,F4,F2,F3
 CM=F1,M1,F2,M1
 CM=M1,M1,M1,M1
 CM=M2,M1,M3,M1
 CM=M2,M1,M3,M1
 CM=M1,M1,M1,M1
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=3 Y=9 NAM=WEAVE
*END
```

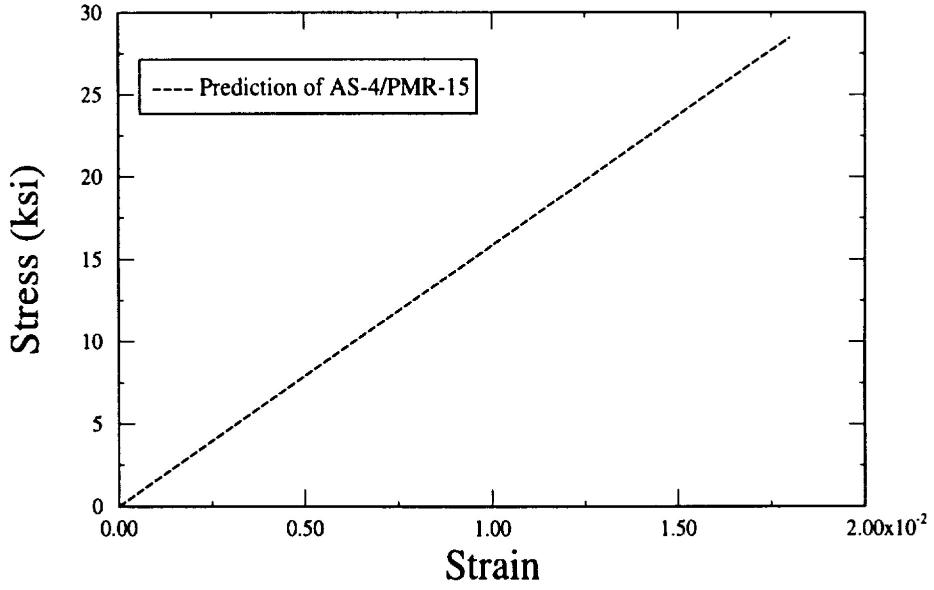
Results for woven composite:

```
EFFECTIVE ENGINEERING MODULI

E11S= .920E+06
N12S= .288E+00
E22S= .158E+07
N23S= .151E+00
E33S= .158E+07
G23S= .357E+06
G13S= .351E+06
G12S= .351E+06

EFFECTIVE THERMAL EXPANSION COEFFICIENTS

.548E-04  .261E-04  .261E-04
.000E+00  .318E-22 -.567E-23
```



☞ **Note**: The overall volume fraction is 32.5% while that of the yarn itself is 65%.

6.15 Example O: FEAMAC Analysis

Sample Input Files For **FEAMAC** Analysis

Problem Summary:

Load Type:	Thermomechanical; Load Control (**ABAQUS**)
Load History Data:	From **ABAQUS**: STEP 1: Cool down Temperature: 860.0 to 427.0°C Strain: 0.0 - 0.0 STEP 2: Transverse loading Temperature held fixed at 427.0°C Strain: 0.0 to 0.015
Micromechanics model:	Double Periodic
Fiber Packing Arrangement:	Square Fiber (idp=2) Triangular Pack at 33% fiber volume ratio Circular Fiber (idp=6) Rectangular Pack at 33% fiber volume ratio
Integration Algorithm:	Forward Euler (**ABAQUS**)
Constituent Material Model:	Fiber: Elastic (ncmd = 7) Matrix: TGVIPS (ncmd = 7)
Constituents:	Fiber: SCS-6 Matrix: Ti-6-4 (isotropic)
Debonding:	First implementation of debond model
Finite Element Model:	Consists of 2 (8-noded brick) elements each with a different material. In this way the example demonstrates the use of multiple material groups.

☞ **FEAMAC Note:** All data that is struck-out in the listing is not required by **FEAMAC** and is ignored if present. Warnings are generated informing the user that the data provided is being ignored. **FEAMAC** can be employed with any size of a finite element mesh. However, **FEAMAC** requires that an appropriate amount of state variable space be allocated for each user material found in the finite element model. The amount of state variable space to be allocated for a given user material is based on the size of the RVE requested in the corresponding **MAC/**

GMC input deck. The number of state variables must be less than 1000. The ***DEPVAR** option under each of the ***USER MATERIAL** sections should be used to define the size of this space. The number of CONSTANTS should be set to one on the ***USER MATERIAL** line and on the following line the value should be set to one. If the material model is linear elastic and no micro curve data files have been requested then no state variable space is required for that material.

To determine the amount of space to request use the following equation:

NG = Number of cells in the gamma direction

NB = Number of cells in the beta direction

NDEPVAR = Number of state variables to be to be provided to the *DEPVAR option.

$$\text{NDEPVAR} = 46 + (42 * \text{NG} * \text{NB}) \qquad < 1000$$

Therefore, based on the predefined RVE's in **section 4.2.13**:

IDP	NDEPVAR	
1	214	
2	718	
3	718	
4	718	
6	2104	Too Large: Must use alternate method described below
9	2104	Too Large: Must use alternate method described below
11	214	
13	28,438	Too Large: Must use alternate method described below

EXAMPLE:

```
...
*MATERIAL,NAME=MACDECK1
*USER MATERIAL, CONSTANTS=1
1
*DEPVAR
214
...
```

Where MACDECK1 references an IDP of size 2x2.

ALTERNATE METHOD:

If a chosen IDP requires more than 1000 state variables then the following alternate procedure must be followed:

On the *USER MATERIAL line set the number of <u>CONSTANTS</u> equal to 1and on the following line set the value of this constant to be <u>two</u>. Also set the number of state variables on the *DEPVAR option to zero.

```
EXAMPLE:
...
*MATERIAL,NAME=MACDECK2
*USER MATERIAL, CONSTANTS=1
2
*DEPVAR
0
...
```

Where MACDECK2 references an IDP that would require more than 1000 state variables (e.g. IDP = 9).

☞ **FEAMAC Note:** This alternate procedure will cause execution times to increase greatly.

☞ **FEAMAC Note:** It is highly recommended that linear elastic materials with no requests for micro curve data be used in all areas of the model except in areas of suspected material nonlinearity, irrespective of memory approach.

MAC/GMC Input Listings: MACMAN1:

```
FEAMAC DEMO - Material group No. 1, identified in ABAQUS deck as MACMAN1
*PRINT
 NPL=1 %
*LOAD
 LCON=3 ~~LOP=3 LSS=1~~ %
*~~MECH~~
~~ NPTW=3 TI=0.0, 58000.0, 58150.0 LO=0.0, 0.0, 0.015 %~~
~~*THERM~~
~~ NPTT=3 TI=0.0, 58000.0, 58150.0 TE=860.0, 427.0, 427.0 %~~
*MODEL
 MOD=1 %
~~*SOLVER~~
~~ NTF=1 &~~
~~ NPTS=3 TIM=0.0, 58000.0, 58150.0 STP=2000.0, 1.0 %~~
*FIBER
 NFIBS=1
 NF=1 MF=4 NDPT=1 MAT=U IFM=1 &
 EL=58.E3, 0.32, 3.5E-06 &
 VI=0.8E-9,1.E20,0.1E-5,0.,0.85E-4,0.05,1.,1.,1.,3.3,1.8,1.35,1.,0.01 %
*MATRIX
 NMATX=1
 NM=1 MM=7 NDPT=2 MAT=A D=1.0, 1.0, 1.0 %
*MRVE
 IDP=2 VF=0.33 %
*DEBOND
 NII=2
 DBCH=1 NBI=1 NGI=2 FACE=2 RN=0.1 BDN=35.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=1 NGI=3 FACE=2 RN=0.1 BDN=35.0 RS=0.0 BDS=0.0 TI=58000.0
 %
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=3 Y=9 NAM=macman1_s3_e3n %
*END
```

MAC/GMC Input Listings: MACMAN2:

```
FEAMAC DEMO - Material group No. 2, identified in ABAQUS deck as MACMAN2
*PRINT
 NPL=1 %
*LOAD
 LCON=3 ~~LOP=3 LSS=1~~ %
~~*MECH~~
~~NPTW=3 TI=0.0, 58000.0, 58150.0 LO=0.0, 0.0, 0.015 %~~
~~*THERM~~
~~NPTT=3 TI=0.0, 58000.0, 58150.0 TE=860.0, 427.0, 427.0 %~~
*MODEL
 MOD=1 %
~~*SOLVER~~
~~NTF=1 &~~
~~NPTS=3 TIM=0.0, 58000.0, 58150.0 STP=2000.0, 1.0 %~~
*FIBER
 NFIBS=1
 NF=1 MF=9 NDPT=2 MAT=A D=1.0, 0.0, 0.0 %
*MATRIX
 NMATX=1
 NM=1 MM=7 NDPT=2 MAT=A D=1.0, 1.0, 1.0 %
*MRVE
 IDP=6 VF=0.33 R=0.75%
*DEBOND
 NII=10
 DBCH=1 NBI=4 NGI=1 FACE=2 RN=0.1 BDN=14.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=4 NGI=6 FACE=2 RN=0.1 BDN=14.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=2 NGI=3 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=2 NGI=4 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=3 NGI=2 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=3 NGI=5 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=6 NGI=3 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=6 NGI=4 FACE=2 RN=0.1 BDN=22.0 RS=0.0 BDS=0.0 TI=58000.0
#
 DBCH=1 NBI=5 NGI=2 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
 DBCH=1 NBI=5 NGI=5 FACE=2 RN=0.1 BDN=18.0 RS=0.0 BDS=0.0 TI=58000.0
#
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=3 Y=9 NAM=macman2_s3_e3n %
*END
```

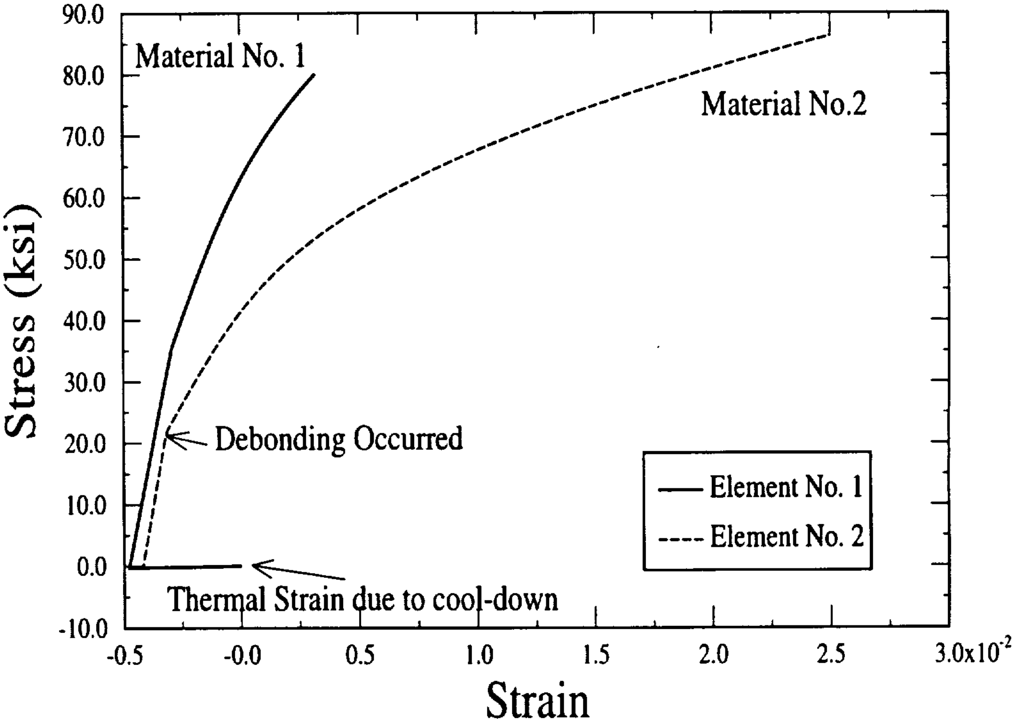
ABAQUS 5.8 Input Listing:

```
*HEADING
 VERIFICATION OF FEAMAC
*NODE,NSET=ALL
1, 0.0, 0.0, 0.0
2, 0.0, 1.0, 0.0
3, 0.0, 1.0, 1.0
4, 0.0, 0.0, 1.0
5, 1.0, 0.0, 0.0
6, 1.0, 1.0, 0.0
7, 1.0, 1.0, 1.0
8, 1.0, 0.0, 1.0
9, 2.0, 0.0, 0.0
10, 2.0, 1.0, 0.0
11, 2.0, 1.0, 1.0
12, 2.0, 0.0, 1.0
*ORIENTATION, NAME=TRANS
0.0, 0.0, 1.0, 0.0, 1.0, 0.0
*ELEMENT, TYPE=C3D8, ELSET=EL1
1, 1, 2, 3, 4, 5, 6, 7, 8
*ELEMENT, TYPE=C3D8, ELSET=EL2
2, 5, 6, 7, 8, 9, 10, 11, 12
*SOLID SECTION, MATERIAL=MACMAN1, ELSET=EL1, ORIENTATION=TRANS
1.0
*SOLID SECTION, MATERIAL=MACMAN2, ELSET=EL2, ORIENTATION=TRANS
1.0
*MATERIAL,NAME=MACMAN1
*USER MATERIAL, CONSTANTS=1
1
*DEPVAR
718
*USER DEFINED FIELD
*EXPANSION, TYPE=ANISO, USER
*MATERIAL,NAME=MACMAN2
*USER MATERIAL, CONSTANTS=1
2
*USER DEFINED FIELD
*EXPANSION, TYPE=ANISO, USER
*USER SUBROUTINE
*INCLUDE, INPUT=/home/smdant/macdev/feabench/man_demo/umacman.f
*INITIAL CONDITIONS, TYPE=TEMPERATURE
ALL, 860.0
*INITIAL CONDITIONS, TYPE=FIELD, VARIABLE=1
ALL, 1.0
**------------------------------------------
*STEP, AMPLITUDE=RAMP, INC=2000000
```

```
 STEP ONE - COOL DOWN
*STATIC, DIRECT
2000.0, 58000.0
*TEMPERATURE
ALL, 427.0
*BOUNDARY
1, 1, 3
2, 1
2, 3
3, 1
4, 1, 2
5, 2, 3
6, 3
8, 2
9, 2, 3
10, 3,
12, 2
*NSET, NSET=ALL
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
*EL PRINT, FREQ=1000
S
*EL PRINT, FREQ=1000
E
*EL PRINT, FREQ=1000
THE, TEMP
*NODE PRINT, FREQ=1000, NSET=ALL
U
*NODE PRINT, FREQ=1000, NSET=ALL
CF
*NODE PRINT, FREQ=1000, NSET=ALL
RF
*END STEP
**------------------------------------------
*STEP, AMPLITUDE=RAMP, INC=2000000
 STEP TWO - LOAD UP
*STATIC, DIRECT
1.0, 150.0
*BOUNDARY
1, 1, 3
2, 1
2, 3
3, 1
4, 1, 2
5, 2, 3
6, 3
8, 2
```

```
9, 2, 3
10, 3,
12, 2
9, 1, 1, 0.015
10, 1, 1, 0.015
11, 1, 1, 0.015
12, 1, 1, 0.015
*TEMPERATURE
ALL, 427.0
*END STEP
**-------------------------------------------
```

The following figure was obtained from the macro x-y plot data files produced by the present example.



ABAQUS UMAT Subroutine Listing:

```
C COMMON TO SOLIDS AND SHELLS
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678 2
C    __  _______ ____  __    _________
C   / / / / ___// __ \/ /   / ____/ __ \
C  / / / /\__ \/ / / / /   / /_  / / / /
C / /_/ /___/ / /_/ / /___/ __/ / /_/ /
C \____//____/_____/_____/_/   /_____/
C
C
  1111111111222222222233333333334444444444555555555566666666667777
C23456789012345678901234567890123456789012345678901234567890123456789
C


C ORDER OF ABAQUS FUNCTION CALLS
C
C                        - UMAT  (SOLID)
C        (if needed)   /
C USDFLD -=> UEXPAN -=<     or
C                      \
C                        - UGENS (SHELL) <-= TBD

C -----------------

      SUBROUTINE USDFLD(FIELD, STATEV, PNEWDT, DIRECT, T, CELENT,
     &     TIME, DTIME, CMNAME, ORNAME, NFIELD, NSTATV, NOEL, NPT,
     &     LAYER, KSPT, KSTEP, KINC, NDI, NSHR)


      INCLUDE 'ABA_PARAM.INC'

      PARAMETER (ZERO = 0.0D0, HALF = 0.5D0)
      PARAMETER (ONE = 1.0D0, TWO = 2.0D0, THREE = 3.0D0)
      PARAMETER (NFROM_UTHERM = 1, NFROM_UMECH = 2)

      CHARACTER*8 CMNAME, ORNAME
      DIMENSION FIELD(NFIELD), STATEV(NSTATV), DIRECT(3,3)
      DIMENSION T(3,3), TIME(2)

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH
      CHARACTER*80 TSTRING

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2

      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP

      LOGICAL SIGAL_STATV


      CHARACTER*8 FLGRAY(15)
      DIMENSION ARRAY(15), JARRAY(15)

      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION

      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR

      COMMON / KTHERMONLY / DTHERMSTRESS(6), DTHERMSTRAIN(6),
     &     DDTHERMSTRAIN_DT(6), THERMSTRESS(6), THERMSTRAIN(6)


      DATA NFIRSTELEM / 0 /
      DATA NFIRSTIP / 0 /
      DATA NFIRSTTIME / 1 /


      NFEA_FROM = 0

C*********************************
C INITIALIZE
C*********************************
      NNOEL = NOEL
      NNPT = NPT
      NNDI = NDI
      NNSHR = NSHR

      NCALL = 0

      IF (NFIRSTTIME .EQ. 1) THEN
         CALL FEAMAC_INIT(NOEL, NPT)
      ENDIF

      AB_PRINT = GAB_PRINT

C*********************************
C DEFINE MAC INPUT DECK NAME
C*********************************
      CALL FILL_STRING(TSTRING, ' ')
      CALL FILL_STRING(FNAME, ' ')

      CALL STRING_COPY(FNAME, CMNAME)
      CALL STRING_APPEND(FNAME, EXTSION)

      TSTRING = PATH
      CALL STRING_APPEND(TSTRING, FNAME)
      FNAME = TSTRING

C*********************************
C DEFINE STEP INFORMATION
C*********************************
      IF ( ( NOEL .EQ. NFIRSTELEM )
     &     .AND. ( NPT .EQ. 1 )
     &     .AND. ( (KINC - 1) .NE. INC) ) THEN
         NCYCLE = -1
      ENDIF

      IF ( ( NOEL .EQ. NFIRSTELEM )
     &     .AND. ( NPT .EQ. 1 ) ) THEN
         NCYCLE = NCYCLE + 1
      ENDIF

      AB_STEP = KSTEP
      GDT = DTIME
      STEP_TIME = TIME(1)
      TOT_TIME = TIME(2)
      INC = KINC - 1

C WE CAN MAKE CALLS TO GETVRM FROM HERE TO GET ADDTIONAL INFO IF NEEDED
      CALL GETVRM('THE', ARRAY, JARRAY, FLGRAY, JRDC)

      DO J = 1, 6
         THERMSTRAIN(J) = ARRAY(J)
         THERMSTRESS(J) = 0.0
         DTHERMSTRESS(J) = 0.0
         DTHERMSTRAIN(J) = 0.0
         DDTHERMSTRAIN_DT(J) = 0.0
      ENDDO


      IF ( AB_PRINT .GE. 1 ) THEN
         WRITE(NDU, *) ' '
         WRITE(NDU, *) '=========================================',
     &        '==============================================='
         WRITE(NDU, *) '-=> START OF USDFLD <=-'
         WRITE(NDU, *) '-=> VALUE OF STEP TIME =', TIME(1), STEP_TIME
         WRITE(NDU, *) '-=> TOTAL TIME         =', TIME(2), TOT_TIME
         WRITE(NDU, *) '-=> DTIME              =', DTIME, GDT
         WRITE(NDU, *) '-=> KSTEP              =', KSTEP, AB_STEP
         WRITE(NDU, *) '-=> KINC               =', KINC, INC+1
         WRITE(NDU, *) '-=> NOEL               =', NOEL, NNOEL
         WRITE(NDU, *) '-=> NPT                =', NPT, NNPT
         WRITE(NDU, *) '-=> NDI                =', NDI
         WRITE(NDU, *) '-=> NSHR               =', NSHR
         WRITE(NDU, *) '-=> NCYCLE             =', NCYCLE
         WRITE(NDU, *) '-=> NFIRSTELEM         =', NFIRSTELEM
         WRITE(NDU, *) '-=> NFIRSTIP           =', NFIRSTIP
         WRITE(NDU, *) '-=> NCALL              =', NCALL
         WRITE(NDU, *) '-=> FNAME =|', FNAME, '|'
      ENDIF

      RETURN
      END
```

```
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
  012
C      ______________    __  ______   ______      ____   ______________
C     / ____/ ____/   |  /  |/  /   |  / ____/     /  _/ | / /  _/_  __/
C    / /_  / __/ / /| | / /|_/ / /| | / /          / //  |/ // /  / /
C   / __/ / /___/ ___ |/ /  / / ___ |/ /___      _/ // /|  // /  / /
C  /_/   /_____/_/  |_/_/  /_/_/  |_\____/_____/___/_/ |_/___/ /_/
C                                        /_____/
C
C
  1111111111222222222233333333334444444444555555555566666666667777
C2345678901234567890123456789012345678901234567890123456789012345678901234567890
  012

      SUBROUTINE FEAMAC_INIT(NOEL, NPT)

      INCLUDE 'ABA_PARAM.INC'

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH
      CHARACTER*80 TSTRING

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2


      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP


      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION

      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR

      LOGICAL SIGAL_STATV

      CALL FILL_STRING(PATH, ' ')
      CALL FILL_STRING(FNAME, ' ')
      CALL FILL_STRING(EXTSION, ' ')
      CALL FILL_STRING(DFNAME, ' ')
      CALL FILL_STRING(TSTRING, ' ')
      CALL FILL_STRING(TNAME1, ' ')
      CALL FILL_STRING(TNAME2, ' ')

C ****************************************
C     ____  ________________   __
C    / __ )/ ____/ ____/  _/ | / /
C   / __  / __/ / / __ / //  |/ /
C  / /_/ / /___/ /_/ // // /|  /
C /_____/_____/\____/___/_/ |_/
```

```
C     __  _____ __________
C    / / / / ___// ____/ __ \
C   / / / /\__ \/ __/ / /_/ /
C  / /_/ /___/ / /___/ _, _/
C  \____//____/_____/_/ |_|
C
C      __________  __________________
C     / ____/ __ \/  _/_  __/ ___/
C    / __/ / / / // /  / /  \__ \
C   / /___/ /_/ // /  / /  ___/ /
C  /_____/_____/___/ /_/  /____/
C
C ******************************************
C *** BEGIN USER EDITS ***
C *** USER NEEDS TO CHANGE THE NEXT LINES AS NEEDED...
C PATH TO WORKING DIRECTORY
      PATH = '/home/smdant/macdev/feabench/man_demo'

C NOTE: ABAQUS WILL CONVERT THE MATERIAL NAME TO UPPER CASE.
C MAC INPUT DECK NAME EXTENSION
      EXTSION = '.MAC'


C THE FOLLOWING IS ONLY USED TO RESOLVE VERSION DIFFERENCES BETWEEN
  SUN AND SGI
C NAME OF DIAGNOSTIC FILE
      DFNAME = 'MACDIAG'


C     MAC PLOT FILES:
C     - # OF POINTS TO PLOT, I.E # OF PAIRS TO FOLLOW
      NPEL = 2

C     - ELEMENT # AND INTEGRATION POINT # PAIRS TO PLOT
C           N_PEN - ELEMENT NUMBERS
C           N_PEN - INTEGRATION POINT NUMBERS

      J = 1
      N_PEN(J) = 1
      N_PIN(J) = 1

      J = J + 1
      N_PEN(J) = 2
      N_PIN(J) = 1


C     PATRAN PLOT FILES:
C     - # OF POINTS TO PLOT PATRAN DATA, I.E # OF PAIRS TO FOLLOW
      NPATPEL = 0

C     - ELEMENT # AND INTEGRATION POINT # PAIRS TO PLOT PATRAN DATA
C           N_PATPEN - ELEMENT NUMBERS
C           N_PATPEN - INTEGRATION POINT NUMBERS

C      J = 1
C      N_PATPEN(J) = 1
C      N_PATPIN(J) = 1

C      J = J + 1
C      N_PATPEN(J) = 2
C      N_PATPIN(J) = 1


C THE FOLLOWING IS DEPRECIATED OPTION AND MAY NOT BE SUPPORTED
C IN THE NEXT RELEASE OF FEAMAC.  IT'S USE IS NOT RECOMMENDED.
```

```
C
C SET THE AL_STORE VARIABLE TO 1 IF YOU WANT TO POPULATE THE STATEV
C  ARRAY PER ALLISON ENGINE COMPANY'S SPECS. (IDP 1 & 11 ONLY)
C
C  NOTES: 1) THIS OPTION REQUIRES THE USE OF THE ALTERNATE METHOD OF
C            USER MATERIAL DEFINTION AS DESCRIBED IN THE MAC MANUAL.
C
C         2) A STATE VARIABLE SPACE OF 900 IS REQUIRED.
C
C                      ---------------
C                     |       |       |
C                     |  M1   |  M2   |
C                     |       |       |
C                     |  (3)  |  (4)  |
C                     |       |       |
C                      ---------------
C            BETA     |       |       |
C             ^       |   F   |  M3   |
C             |       |       |       |
C             |       |  (1)  |  (2)  |
C             |       |       |       |
C             |        ---------------
C             |
C             ----------> GAMMA
C    INDICES
C     1 -  12  -=> MATRIX STRESS AND STRAIN IN SUBCELL M1
C    23 -  34  -=> FIBER STRESS AND STRAIN
C   132 - 137  -=> MATRIX MECHANICAL STRAIN COMPONENTS IN SUBCELL M1
C   301 - 312  -=> MATRIX STRESS AND STRAIN IN SUBCELL M2
C   432 - 437  -=> MATRIX MECHANICAL STRAIN COMPONENTS IN SUBCELL M2
C   601 - 612  -=> MATRIX STRESS AND STRAIN IN SUBCELL M3
C   732 - 737  -=> MATRIX MECHANICAL STRAIN COMPONENTS IN SUBCELL M3

      AL_STORE = 0


      AB_PRINT = 0


C *****************************************
C *** TEMP FILE LOCATIONS
C *** SYS ADMIN OR USER NEEDS TO CHANGE THE NEXT LINES AS NEEDED...
      TNAME1 = '/var/tmp/feamac1'
      TNAME2 = '/var/tmp/feamac2'


C *****************************************
C *** END OF USER EDITS ***
C     _______   ______
C    / ____/ | / / __ \
C   / __/ /  |/ / / / /
C  / /___/ /|  / /_/ /
C /_____/_/ |_/_____/
C    __  _______ __________
C   / / / / ___// ____/ __ \
C  / / / /\__ \/ __/ / /_/ /
C / /_/ /___/ / /___/ _, _/
C \____//____/_____/_/ |_|
C     __________  __________________
C    / ____/ __ \/  _/_  __/ ___/
C   / __/ / / / // /  / /  \__ \
C  / /___/ /_/ // /  / /  ___/ /
C /_____/_____/___/ /_/  /____/
C
```

```
C ******************************************
      TSTRING = PATH
      CALL STRING_APPEND(TSTRING, '/ ')
      PATH = TSTRING

      TSTRING = PATH
      CALL STRING_APPEND(TSTRING, DFNAME)
      DFNAME = TSTRING

      DO 100 J = 1, NPEL
         IF ( (N_PIN(J) .LT. 1) .OR. (N_PIN(J) .GT. 8) ) THEN
            N_PIN(J) = 1
         ENDIF
 100  CONTINUE

      NDU = 7
C UN-COMMENT NEXT 2 LINES IF YOUR ON A SGI
C         NDU = 18
C         OPEN(UNIT = NDU, FILE = DFNAME, STATUS = 'UNKNOWN')

      WRITE(NDU, *) 'BEGIN'

      NFIRSTELEM = NOEL
      NFIRSTIP = NPT

      NCYCLE = -1

      GAB_PRINT = AB_PRINT

      RETURN
      END


C
  1111111111222222222233333333334444444444555555555566666666667777
C23456789012345678901234567890123456789012345678901234567890123456789
  012
C    __  ____  ______  ______
C   / / / /  |/  /   |/_  __/
C  / / / / /|_/ / /| | / /
C / /_/ / /  / / ___ |/ /
C \____/_/  /_/_/  |_/_/
C
C
  1111111111222222222233333333334444444444555555555566666666667777
C23456789012345678901234567890123456789012345678901234567890123456789
  012

       SUBROUTINE UMAT(STRESS, STATEV, DDSDDE, SSE, SPD, SCD, RPL,
     &     DDSDDT, DRPLDE, DRPLDT, STRAN, DSTRAN, TIME, DTIME, TEMP,
     &     DTEMP, PREDEF, DPRED, CMNAME, NDI, NSHR, NTENS, NSTATV,
     &     PROPS, NPROPS, COORDS, DROT, PNEWDT, CELENT, DFGRD0,
     &     DFGRD1, NOEL, NPT, LAYER, KSPT, KSTEP, KINC)


C
  1111111111222222222233333333334444444444555555555566666666667777
C23456789012345678901234567890123456789012345678901234567890123456789
  012
C
C     STRESS(NTENS) - PASSED IN AS THE STRESSES TENSOR AT THE BEGINNING
C           OF THE INCREMENT, SIGMA_I_J.  (THIS SHOULD BE UPDATED IN
C           THIS ROUTINE.)
C           -IN/OUT
```

```
C     STATEV(NSTATV) - SOLUTION DEPENDANT STATE VARIABLES.
C          -IN/OUT
C            Format of state variable space...
C
C
C
C                      total strain      (1-6)
C                                                  "MACRO" quantities when
C                      cauchy stress     (7-12)     micromechanics model
C                                                   or continuum model
C                      inelastic strain  (13-18)    is used.
C
C                      state variable(s) (19-30)
C                      (space for 2 (6x1) vectors)
C
C                      thermal strain    (31-36)
C
C                      temperature       (37)
C
C                      future use        (38-40)
C
C                      future use        (41-43)
C
C                      future use        (44-46)
C
C                     -----------------------------------------------------
C
C                      total strain      (47-52)
C
C                      cauchy stress     (53-58)     "MICRO" quantities:
C                                                     this block of NSVARS = 42
C                     inelastic strain  (59-64)     is repeated for each subcell
C
C                      internal state variable(s)
C                                        (65-76)
C                      (space for 2 [6x1] vectors)
C
C                      thermal strain    (77-82)
C
C                      rn & rd debond    (83-88)
C                      parameters
C
C                     .........................
C                                :
C                               etc.     (89-...)
C
C
C
C     DDSDDE(NTENS, NTENS) - JACOBIAN MATRIX Of THE CONSTITUTIVE
C          MODEL.
C          -OUT
C     SSE - SPECIFIC ELASTIC STRAIN ENERGY.
C          -OUT
C     SPD - PLASTIC DISSIPATION.
C          -OUT
C     SCD - CREEP DISSIPATION.
C          -OUT
C     RPL - VOLUMETRIC HEAT GENERATION PER UNIT TIME AT THE END
C          OF THE INC.
C          -OUT(COUPLED)
C     DDSDDT(NTENS) - VARIATION OF STRESS INCREMENTS WITH
C          RESPECT TO TEMPERATURE.
C          -OUT(COUPLED)
C     DRPLDE(NTENS) - VARIATION OF RPL WRT STRAIN INCREMENTS.
```

```
C           -OUT(COUPLED)
C       DRPLDT - VARIATION OF RPL WRT THE TEMPERATURE INCREMENT.
C           -OUT(COUPLED)
C       STRAN(NTENS) - TOTAL STRAINS AT BEGINNING OF THE INCREMENT.
C           -IN
C       DSTRAN(NTENS) - STRAIN INCREMENTS
C           -IN
C       TIME(1) - VALUE OF STEP TIME AT BEGINNING OF CURRENT INCREMENT
C           -IN
C       TIME(2) - TOTAL TIME AT BEGINNING OF CURRENT INCREMENT
C           -IN
C       DTIME - TIME INCREMENT
C           -IN
C       TEMP - TEMPERATURE AT START OF INCREMENT
C           -IN
C       DTEMP - INCREMENT OF TEMPERATURE
C           -IN
C       PREDEF - ARRAY OF INTERPOLATED VALUES OF PREDEFINED FIELD
C           VARIABLES.
C           -IN
C       DPRED - ARRAY OF INCREMENTS OF PREDEFINED FIELD VARIABLES.
C           -IN
C       CMNAME - NAME GIVEN ON *MATERIAL OPTION, LEFT JUSTIFIED.
C           -IN
C       NDI - NUMBER OF DIRECT STRESS COMPONENTS AT THIS POINT.
C           -IN
C       NSHR - NUMBER OF ENGINEERING SHEAR STRESS COMPONENTS AT
C           THIS POINT.
C           -IN
C       NTENS - SIZE OF THE STRESS OR STRAIN COMPONENT ARRAY
C           (NDI + NSHR)
C           -IN
C       NSTATV - NUMBER OF SOLUTION DEPENDANT STATE VARIABLES.
C           -IN
C       PROPS(NPROPS) - ARRAY OF MATERIAL CONSTANTS.
C           -IN
C       NPROPS - NUMBER OF MATERIAL CONSTANTS.
C           -IN
C       COORDS - ARRAY CONTAINING THE COORDINATES OF THIS POINT.
C           -IN
C       DROT(3,3) - ROTAION INCREMENT MATRIX.
C           -IN
C       PNEWDT - RATIO OF SUGGESTED NEW TIME INCREMENT TO THE
C           TIME INCREMENT BEING USED.
C           -UPDATE
C       CELENT - CHARACTERISTIC ELEMENT LENGTH.
C           -IN
C       DFGRD0(3,3) - ARRAY CONTAINING DEFORMATION GRADIENT
C           AT THE BEGINNING OF THE INCREMENT
C           -IN
C       DFGRD1(3,3) - ARRAY CONTAINING DEFORMATION GRADIENT
C           AT THE END OF THE INCREMENT
C           -IN
C       NOEL - ELEMENT NUMBER
C           -IN
C       NPT - INTEGRATION POINT NUMBER
C           -IN
C       LAYER - LAYER NUMBER (FOR COMPOSITE SHELLS
C           AND LAYERED SOLIDS).
C           -IN
C       KSPT - SECTION POINT NUMBER WITHIN THE CURRENT LAYER
C           -IN
C       KSTEP - STEP NUMBER
C           -IN
```

```
C     KINC - INCREMENT NUMBER
C          -IN
C
C
  111111111122222222223333333333444444444455555555556666666666777
C23456789012345678901234567890123456789012345678901234567890123456789
  012

      INCLUDE 'ABA_PARAM.INC'
      PARAMETER (ZERO = 0.0D0, HALF = 0.5D0)
      PARAMETER (ONE = 1.0D0, TWO = 2.0D0, THREE = 3.0D0)
      PARAMETER (NFROM_UTHERM = 1, NFROM_UMECH = 2)

      CHARACTER*8 CMNAME

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2


      DIMENSION STRESS(NTENS)
      DIMENSION STATEV(NSTATV)
      DIMENSION DDSDDE(NTENS, NTENS)
      DIMENSION DDSDDT(NTENS)
      DIMENSION DRPLDE(NTENS)
      DIMENSION STRAN(NTENS)
      DIMENSION DSTRAN(NTENS)
      DIMENSION TIME(2)
      DIMENSION PREDEF(1)
      DIMENSION DPRED(1)
      DIMENSION PROPS(NPROPS)
      DIMENSION COORDS(3)
      DIMENSION DROT(3, 3)
      DIMENSION DFGRD0(3, 3)
      DIMENSION DFGRD1(3, 3)


      DIMENSION PLSTRAN(6)

      DIMENSION OSTRESS(6)
      DIMENSION OSTRAIN(6)
      DIMENSION ODSTRAIN(6)

      INTEGER N(2)
      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP


      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION
      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR
      COMMON / KTHERMONLY / DTHERMSTRESS(6), DTHERMSTRAIN(6),
     &     DDTHERMSTRAIN_DT(6), THERMSTRESS(6), THERMSTRAIN(6)


      LOGICAL SIGAL_STATV

C =========================================

      NFEA_FROM = NFROM_UMECH

C********************************
C KEEP TRACK OF THE NUMBER OF CALLS TO FEAMAC
C********************************
C IF NCALL == 1->2 : WE HAVE ALREADY BEEN TO THERMAL
C IF NCALL == 0->1 : NO THERMAL WAS REQUESTED OR DTEMP IS ZERO
      NCALL = NCALL + 1


C********************************
C FLAG TO DETERMINE HOW DATA WILL BE STORED
C  IF SIGAL_STATV == TRUE  - DATA STORED IN STATE VARIABLE
C                                 SPACE PROVIDED BY ABAQUS
C  IF SIGAL_STATV == FALSE - DATA STORED IN FILES
C********************************
      SIGAL_STATV = .TRUE.
      AB_PRINT = GAB_PRINT
      IF( NPROPS .GE. 1 ) THEN
         IF( PROPS(1) .EQ. 2 ) SIGAL_STATV = .FALSE.
         IF( NPROPS .GE. 2 ) AB_PRINT = INT(PROPS(2))
      ELSE
         SIGAL_STATV = .FALSE.
      ENDIF


      N(1) = NOEL
      N(2) = NOEL

      IF ( AB_PRINT .GE. 1 ) THEN
         WRITE(NDU, *) ' '
         WRITE(NDU, *) '======================================',
     &        '============================================'
         WRITE(NDU, *) '-=> START OF UMAT <=-'
         WRITE(NDU, *) '-=> STORE IN STATEV ?  =', SIGAL_STATV
         WRITE(NDU, *) '-=> VALUE OF STEP TIME =', TIME(1), STEP_TIME
         WRITE(NDU, *) '-=> TOTAL TIME         =', TIME(2), TOT_TIME
         WRITE(NDU, *) '-=> DTIME              =', DTIME, GDT
         WRITE(NDU, *) '-=> KSTEP              =', KSTEP, AB_STEP
         WRITE(NDU, *) '-=> KINC               =', KINC, INC+1
         WRITE(NDU, *) '-=> NOEL               =', NOEL, NNOEL
         WRITE(NDU, *) '-=> NPT                =', NPT, NNPT
         WRITE(NDU, *) '-=> NCYCLE             =', NCYCLE
         WRITE(NDU, *) '-=> NFIRSTELEM         =', NFIRSTELEM
         WRITE(NDU, *) '-=> NFIRSTIP           =', NFIRSTIP
         WRITE(NDU, *) '-=> NCALL              =', NCALL
         WRITE(NDU, *) '-=> FNAME =|', FNAME, '|'
         WRITE(NDU, *) '-=> TEMP =', TEMP
         WRITE(NDU, *) '-=> DTEMP =', DTEMP
         WRITE(NDU, *) 'STRAIN IN:'
         WRITE(NDU, *) '|', STRAN(1), STRAN(2), STRAN(3)
         WRITE(NDU, *) '|', STRAN(4), STRAN(5), STRAN(6)
         WRITE(NDU, *) 'DSTRAIN IN:'
         WRITE(NDU, *) '|', DSTRAN(1), DSTRAN(2), DSTRAN(3)
         WRITE(NDU, *) '|', DSTRAN(4), DSTRAN(5), DSTRAN(6)
         WRITE(NDU, *) 'STRESS IN:'
         WRITE(NDU, *) '|', STRESS(1), STRESS(2), STRESS(3)
         WRITE(NDU, *) '|', STRESS(4), STRESS(5), STRESS(6)

         WRITE(NDU, *) ' CMNAME =|', CMNAME, '|'
         WRITE(NDU, *) ' NDI = ', NDI
         WRITE(NDU, *) ' NSHR = ', NSHR
         WRITE(NDU, *) ' NTENS = ', NTENS
         WRITE(NDU, *) ' NSTATV = ', NSTATV
         WRITE(NDU, *) ' NPROPS = ', NPROPS
         WRITE(NDU, *) ' CELENT = ', CELENT
         WRITE(NDU, *) ' NOEL = ', NOEL
         WRITE(NDU, *) ' NPT = ', NPT
         WRITE(NDU, *) ' LAYER = ', LAYER
         WRITE(NDU, *) ' KSPT = ', KSPT
         WRITE(NDU, *) ' KSTEP = ', KSTEP
         WRITE(NDU, *) ' KINC = ', KINC

         WRITE(NDU, *) '-=> N=', N, '|'

         WRITE(NDU, *) '-=> INC    =', INC
         WRITE(NDU, *) '-=> NCYCLE =', NCYCLE

         WRITE(NDU, *) '-=> TEMP =', TEMP
         WRITE(NDU, *) '-=> DTEMP =', DTEMP

         WRITE(NDU, *) '-=> BEFORE FEAMAC_PRE <=-'
      ENDIF



      DO J = 1, 6
         OSTRESS(J) = STRESS(J)

         STRAN(J) = STRAN(J) + THERMSTRAIN(J)
         DSTRAN(J) = DSTRAN(J) + DTHERMSTRAIN(J)

         OSTRAIN(J) = STRAN(J)
         ODSTRAIN(J) = DSTRAN(J)

      ENDDO


      CALL FEAMAC_PRE(DDSDDE, PLSTRAN, STRAN, DSTRAN, STRESS, STATEV,
     &     NSTATV, NTENS, N, NPT, LAYER, NDI, NSHR,
     &     TEMP, DTEMP, THERMSTRAIN)


      IF( (NCALL .EQ. 2) .AND. (NCYCLE .NE. 0) ) THEN

         DO I = 1, 6
            STRESS(I) = OSTRESS(I)
            DO J = 1, 6
               STRESS(I) = STRESS(I)
     &              + DDSDDE(I, J) * (ODSTRAIN(J) - DTHERMSTRAIN(J))
            ENDDO
         ENDDO


         DO J = 1, 6
```

```
            IF( DTEMP .NE. 0.0 ) THEN
               DDSDDT(J) = DTHERMSTRESS(J) / DTEMP
            ELSE
               DDSDDT(J) = 0.0
            ENDIF
            DRPLDE(J) = 0.0
         ENDDO
         RPL = 0.0
         DRPLDT = 0.0
      ENDIF


      IF ( AB_PRINT .GE. 1 ) THEN
         WRITE(NDU, *) '-=> AFTER FEAMAC_PRE <=-'

         WRITE(NDU, *) 'DTHERMAL STRESS:'
         WRITE(NDU, *) '|', DTHERMSTRESS(1), DTHERMSTRESS(2),
     &         DTHERMSTRESS(3)
         WRITE(NDU, *) '|', DTHERMSTRESS(4), DTHERMSTRESS(5),
     &         DTHERMSTRESS(6)

         WRITE(NDU, *) 'DTHERMAL STRAIN:'
         WRITE(NDU, *) '|', DTHERMSTRAIN(1), DTHERMSTRAIN(2),
     &         DTHERMSTRAIN(3)
         WRITE(NDU, *) '|', DTHERMSTRAIN(4), DTHERMSTRAIN(5),
     &         DTHERMSTRAIN(6)

         WRITE(NDU, *) 'DDSDDT:'
         WRITE(NDU, *) '|', DDSDDT(1), DDSDDT(2),
     &         DDSDDT(3)
         WRITE(NDU, *) '|', DDSDDT(4), DDSDDT(5),
     &         DDSDDT(6)

         WRITE(NDU, *) '-=> END OF UMAT <=-'
      ENDIF

C WHATEVER IS IN STRAN/DSTRAN WILL SHOW UP AS MECHANICAL STRAIN
C MECHANICAL STRAIN + THERMAL STRAIN WILL GIVE GLOBAL STRAIN
C WHATEVER IS IN STRESS WILL BE GIVEN AS GLOBAL STRESS
      DO I = 1, 6
         STRAN(I) = OSTRAIN(I) - THERMSTRAIN(I)
         DSTRAN(I) = ODSTRAIN(I) - DTHERMSTRAIN(I)
      ENDDO

      NFIRSTTIME = 0

      RETURN
      END

C
   1111111111222222222233333333334444444444555555555566666666667777777
   7778
C23456789012345678901234567890123456789012345678901234567890123456789
   01234
C    __  __________  __ ____  ____  _    __
C    / / / / ____/ |/ // __ \/   |  /  | / /
C  / / / /  __/  |   // /_/ / /| | /   |/ /
C / /_/ / /____ /   |/ ____/ ___ |/ / | /
C \_____/______//_/|_/_/    /_/   |_/_/ |_/
C
C
   1111111111222222222233333333334444444444555555555566666666667777777
   7778
```

```
C23456789012345678901234567890123456789012345678901234567890123456789012345678901234

      SUBROUTINE UEXPAN(EXPAN, DEXPANDT, TEMP, TIME, DTIME, PREDEF,
     &     DPRED, STATEV, CMNAME, NSTATV)

      INCLUDE 'ABA_PARAM.INC'

      PARAMETER (ZERO = 0.0D0, HALF = 0.5D0)
      PARAMETER (ONE = 1.0D0, TWO = 2.0D0, THREE = 3.0D0)
      PARAMETER (NFROM_UTHERM = 1, NFROM_UMECH = 2)


      CHARACTER*8 CMNAME

      DIMENSION EXPAN(*), DEXPANDT(*), TEMP(2), TIME(2), PREDEF(*)
      DIMENSION DPRED(*), STATEV(NSTATV)

      CHARACTER*80 FNAME
      CHARACTER*80 EXTSION
      CHARACTER*80 DFNAME
      CHARACTER*80 PATH

      CHARACTER*80 TNAME1
      CHARACTER*80 TNAME2

      DIMENSION STRAN(6), DSTRAN(6)
      DIMENSION STRESS(6)
      DIMENSION PLSTRAN(6)
      DIMENSION DDEDDT(6, 6)

      INTEGER N(2)
      INTEGER AB_PRINT
      INTEGER GAB_PRINT
      INTEGER AL_STORE
      INTEGER AB_STEP


      COMMON / KINCINFO / AB_STEP, INC, NCYCLE
      COMMON / KTIME / STEP_TIME, TOT_TIME, GDT

      COMMON / KMACFLAGS / NFIRSTTIME, NFIRSTELEM, NFIRSTIP,
     &     AL_STORE, N_PEN(100), N_PIN(100), NPEL,
     &     N_PATPEN(100), N_PATPIN(100), NPATPEL,
     &     AB_PRINT, GAB_PRINT, SIGAL_STATV, NCALL, NFEA_FROM

      COMMON / KABNAMES / PATH, FNAME, DFNAME, TNAME1, TNAME2, EXTSION
      COMMON / KPREMAC / NDU, NNOEL, NNPT, NNDI, NNSHR
      COMMON / KTHERMONLY / DTHERMSTRESS(6), DTHERMSTRAIN(6),
     &     DDTHERMSTRAIN_DT(6), THERMSTRESS(6), THERMSTRAIN(6)


      LOGICAL SIGAL_STATV

C ===================================================
      NFEA_FROM = NFROM_UTHERM

      AB_PRINT = GAB_PRINT

      DO J = 1, 6
         PLSTRAN(J) = 0.0
         STRAN(J) = 0.0
         DSTRAN(J) = 0.0
         STRESS(J) = 0.0
         PLSTRAN(J) = 0.0
```

```
            DTHERMSTRESS(J) = 0.0
            DTHERMSTRAIN(J) = 0.0
            DDTHERMSTRAIN_DT(J) = 0.0
            DO I = 1, 6
               DDEDDT(I, J) = 0.0
            ENDDO
            EXPAN(J) = 0.0
            DEXPANDT(J) = 0.0
         ENDDO

C NOTE THE "TIME" ARRAY FOR THIS ROUTINE IS PROVIDED AT THE  -E N D-
   OF THE INCREMENT
      TIME(1) = TIME(1) - DTIME
      TIME(2) = TIME(2) - DTIME

C NOTE THE "TEMP" ARRAY FOR THIS ROUTINE IS PROVIDED AT THE  -E N D-
   OF THE INCREMENT
      DTEMP = TEMP(2)
      TEMP(1) = TEMP(1) - DTEMP
      TEMPERATURE = TEMP(1)


      IF ( DTEMP .EQ. 0.0 ) THEN
C          WRITE(NDU, *) 'UEXPAN EARLY EXIT - DTEMP = ', DTEMP,
C     &         ' TEMP = ', TEMPERATURE
         RETURN
      ENDIF

      NCALL = NCALL + 1
      SIGAL_STATV = .FALSE.

      IF ( AB_PRINT .GE. 1 ) THEN
         WRITE(NDU, *) ' '
         WRITE(NDU, *) '========================================',
     &        '==============================================='
         WRITE(NDU, *) '-=> START OF UEXPAN <=-'
         WRITE(NDU, *) '-=> VALUE OF STEP TIME =', TIME(1), STEP_TIME
         WRITE(NDU, *) '-=> TOTAL TIME         =', TIME(2), TOT_TIME
         WRITE(NDU, *) '-=> DTIME              =', DTIME, GDT
         WRITE(NDU, *) '-=> KSTEP              =', AB_STEP
         WRITE(NDU, *) '-=> KINC               =', INC+1
         WRITE(NDU, *) '-=> NOEL               =', NNOEL
         WRITE(NDU, *) '-=> NPT                =', NNPT
         WRITE(NDU, *) '-=> NNDI               =', NNDI
         WRITE(NDU, *) '-=> NMSHR              =', NNSHR
         WRITE(NDU, *) '-=> NCYCLE             =', NCYCLE
         WRITE(NDU, *) '-=> NFIRSTELEM         =', NFIRSTELEM
         WRITE(NDU, *) '-=> NFIRSTIP           =', NFIRSTIP
         WRITE(NDU, *) '-=> NCALL              =', NCALL
         WRITE(NDU, *) '-=> FNAME =|', FNAME, '|'
         WRITE(NDU, *) '-=> TEMPERATURE =', TEMPERATURE
         WRITE(NDU, *) '-=> DTEMP =', DTEMP

         WRITE(NDU, *) '-=> BEFORE FEAMAC_PRE <=-'
      ENDIF


      N(1) = NNOEL
      N(2) = NNOEL
      LAYER = 0
      NTENS = 6
```

```
C NOTE: HERE DSTRAN IS NOT A RATE BUT THE ALPHAS INSTEAD - I.E. THE
  VARIATION OF THERMAL
C            STRAINS WITH RESPECT TO TEMPERATURE (NOT TIME)

      CALL FEAMAC_PRE(DDEDDT, PLSTRAN, STRAN, DSTRAN, STRESS,
  STATEV,
     &     NSTATV, NTENS, N, NNPT, LAYER, NNDI, NNSHR,
     &     TEMPERATURE, DTEMP, THERMSTRAIN)

C NOTE: THESE ARE INCREMENTS IN THERMAL STRAIN/STRESS
C NEED TO FIGURE OUT A GOOD WAY TO GET THE SIZES OF THESE
      DO J = 1, 6
         EXPAN(J) = STRAN(J)
         DTHERMSTRAIN(J) = STRAN(J)
         DDTHERMSTRAIN_DT(J) = DSTRAN(J)
         DEXPANDT(J) = DSTRAN(J)
      ENDDO

      DO I = 1, 6
         DTHERMSTRESS(I) = 0.0
         DO J = 1, 6
            DTHERMSTRESS(I) = DTHERMSTRESS(I)
     &            +  DDEDDT(I, J) * DTHERMSTRAIN(J)
         ENDDO
         STRESS(I) = DTHERMSTRESS(I)
      ENDDO

      RETURN
      END


C
  11111111112222222222333333333344444444445555555555666666666677 7
C234567890123456789012345678901234567890123456789012345678901234 56
```

6.16 Example P: Non-Symmetric Laminate

Sample Input File For Non-Symmetric Laminate Problem

The following example is used to explain the use of monolithic layer in more detail.

Problem Summary:

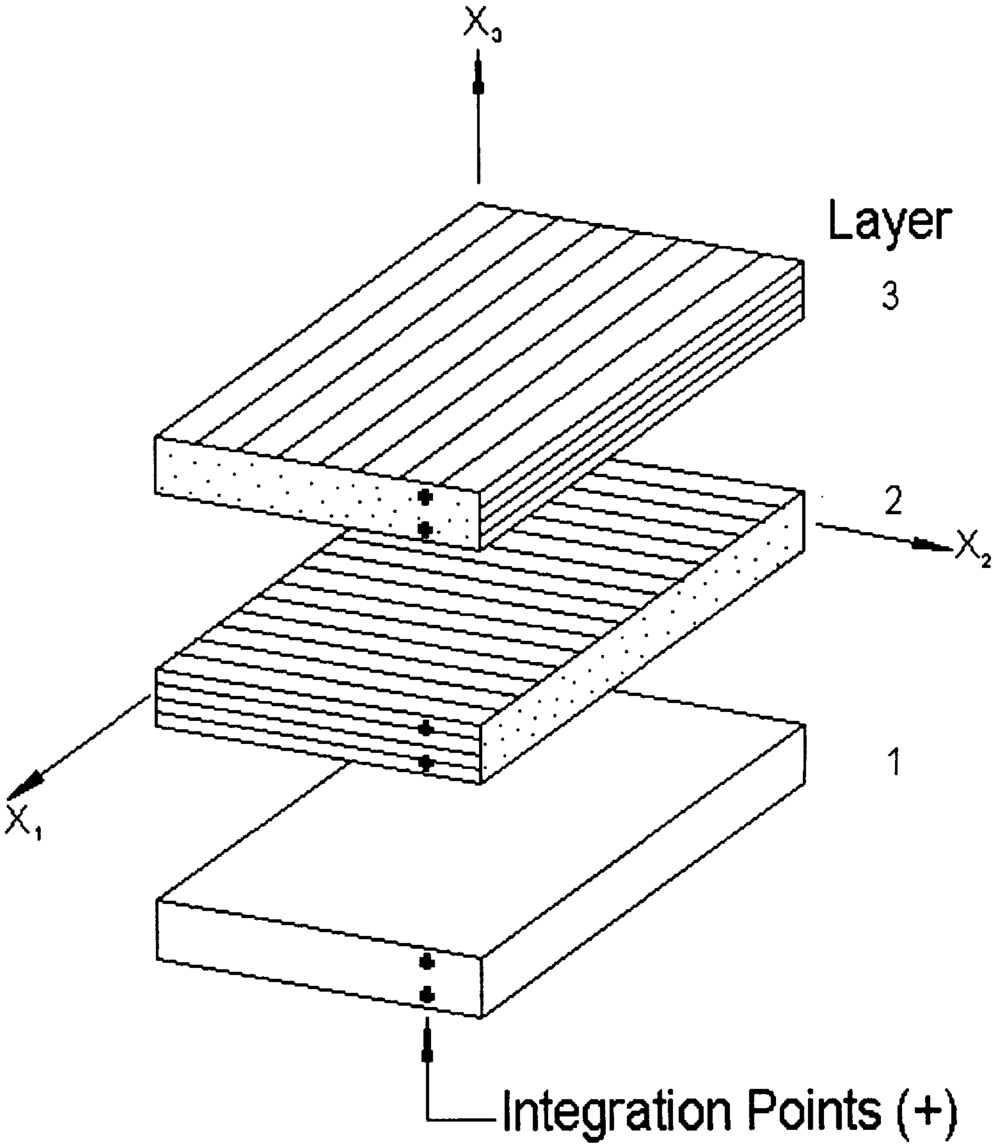
Load Type:	Thermomechanical
Load Component:	11-direction (applied mid-plane strain)
Load History:	Monotonic
Load Control:	Strain
Load History Data:	Cool-down from 371.1°C to 21.1 °C then hold temperature constant during mechanical loading $\dot{T} = 35.$ °C/sec $\dot{\varepsilon} = 1.x10^{-3}$/sec, $\varepsilon_{max} = 0.01$, $\varepsilon_{min} = 0.$ $\Delta t_{thermal} = 0.05$ sec, $\Delta t_{mech} = 0.01$ sec
Micromechanics model:	Laminate Option

Laminate Details:

Layer	Thickness (mm)	Fiber	Matrix	Fiber Volume Fraction	Unit Cell	Packing Arrangement	Angle
1	1	monolithic	Aluminum (2024-T4)	-	-	-	-
2	1	Graphite T-50	Aluminum (2024-T4)	30%	2x2 subcells	square	90
3	1	Graphite T-50	Aluminum (2024-T4)	30%	2x2 subcells	square	0

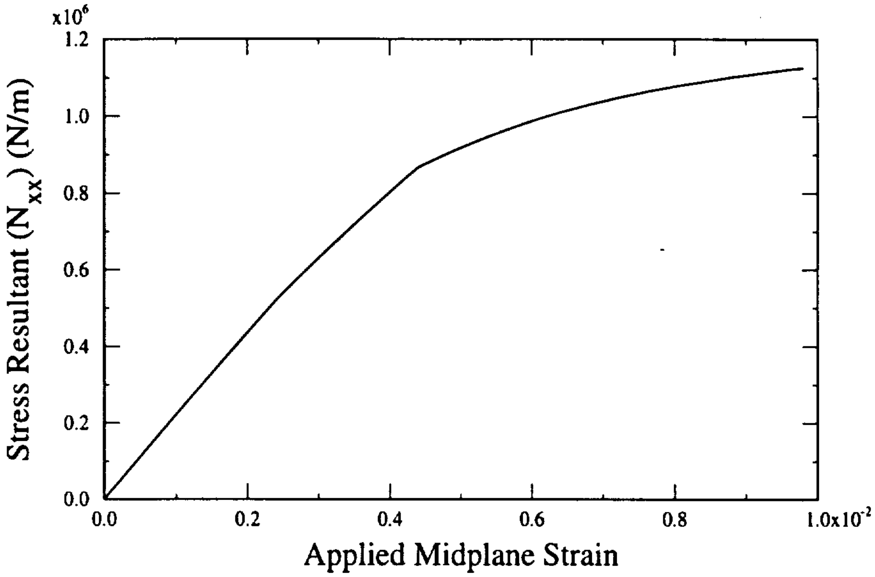
Integration Algorithm:	Forward Euler
Constituent Material Model:	Bodner-Partom

☞ **Note: The laminate analyzed in this example is shown in following figure. The macro output for the laminate option consists of stress resultants and mid-plane strains. The micro output is always in the local coordinates of the layer in question. Micro output is given for both integration points within the layer in question.**

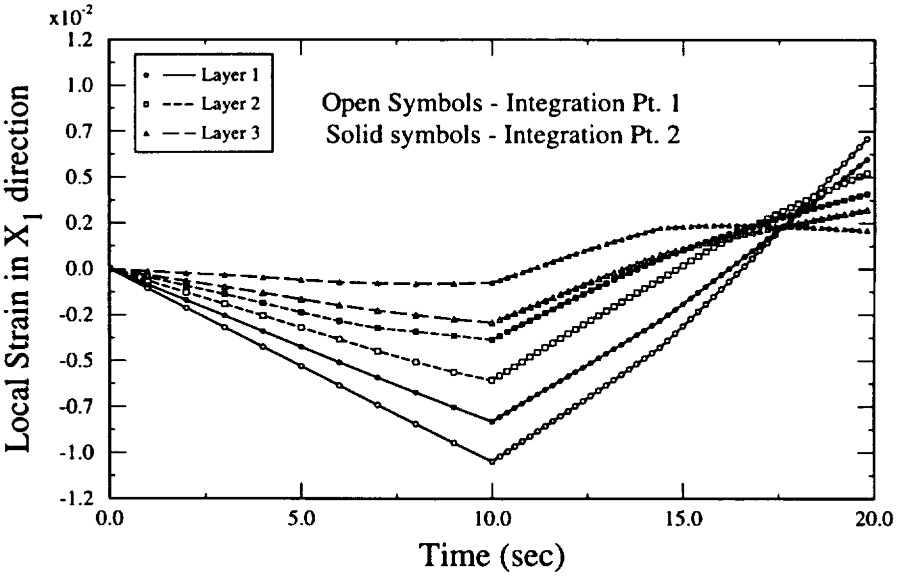


```
example of a non-symmetric laminate with monolithic and heterogeneous layers
*PRINT
 NPL=0 %
*LOAD
 LCON=3 LOP=1 LSS=1
*MECH
 NPTW=3 TI=0.0, 10.0, 20.0 LO=0.0, 0.0, 0.01
*THERM
 NPTT=3 TI=0.0, 10.0, 20.0  &
 TE=371.1, 21.1, 21.1 %
*MODEL
 MOD=3 MATSYS=2 NLY=3 THK=0.001,0.001,0.001 CON=1,2,2 &
 SYS=1,2,2 ANG=0.,90.,0. %
*SOLVER
 NTF=1 NPTS=3 TIM=0.0, 10.0, 20.0 STP=0.05, 0.01 %
*FIBER
 NFIBS=1
 NF=1 MS=2 MF=9 NDPT=2 MAT=A D=1.,0.,0.%
*MATRIX
 NMATX=1
 NM=1 MS=2 MM=1 NDPT=2 MAT=A %
*MONOL
 NMON=1
 NMO=1 MS=1 MMO=1 NDPT=2 MAT=A %
*MRVE
 IDP=0,1,1
 L=1
 L=2 VF=0.3
 L=3 VF=0.3 %
*CURVE
 NP=20 %
*MACRO
 NT=1
 NC=1 X=1 Y=7 NAM=lam-x %
*MICRO
 NT=3
 NC=1 LYR=1 X=37 Y=1 NAM=lam-x1
 NC=2 LYR=2 CELL=2 X=37 Y=2 NAM=lam-x2
 NC=3 LYR=3 CELL=2 X=37 Y=1 NAM=lam-x3 %
*END
```

The following figures were obtained from the macro and micro x-y plot data files produced by the present example.



Global Level Results



Local Level Results

☞ **Note: The increasing magnitude of the strain towards the lower surface of the laminate is due to the increased bending resulting from the nonsymmetric lay-up.**

6.17 Example Q: Yield Surface Plot

Sample Input File for a Yield Surface Plot

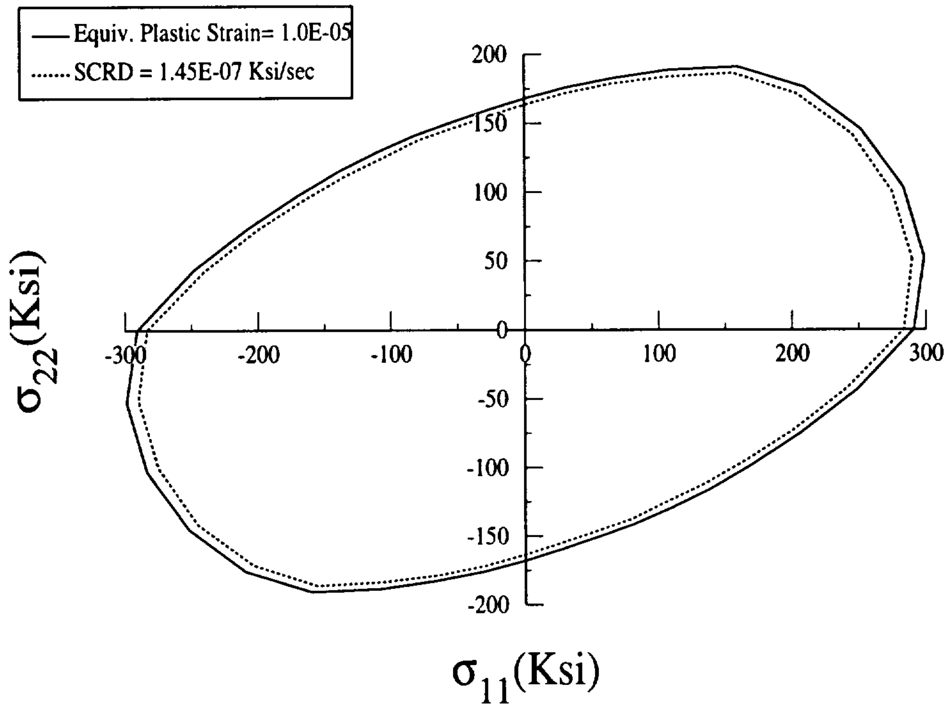
The following example is used to explain the yield surface capabilities in more detail.

Problem Summary:

Load Type:	Themomechanical
Load Component:	Combined 11-component and 22-component
Load History:	Yield surface probing
Load Control:	Strain
Load History Data:	Constant temp. = 23 ^{o} C $\dot{\varepsilon}$=1.0x10^{-5}/sec , ε_{max}=0.05, ε_{min}=0. (along each probing angle) Δt = 5. sec.
Micromechanics model:	Double Periodicity
Fiber Packing Arrangement:	Square pack at 35% fiber volume ratio
Integration Algorithm:	Forward Euler
Constituent Material Model:	Fiber: Elastic Matrix: GVIPS – isotropic form
Constituents:	Fiber: SCS-6 (properties input manually) Matrix: TIMETAL21S

```
test of yield surface option
*PRINT
 NPL=0  %
*LOAD
 LCON=3  LOP=1  LSS=1 %
*MECH
 NPTW=2 TI=0.,5000. LO=0.,0.05  %
*THERM
 NPTT=2 TI=0.,5000. TE=23.,23. %
*SURF
 NPRE=0 ISP=1 IAN=10
 C1=10.E-6  C2=1.45E-7  C3=0.0  C4=0.0
*MODEL
 MOD=1 %
*SOLVER
 NTF=1 NPTS=2 TIM=0.,5000. STP=5. %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=1 MAT=U IFM=1&
 EL=58.E3,58.E3,0.20,0.20,24.167E3,4.5E-6,4.5E-6 %
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=2 MAT=A %
*MRVE
 IDP=1 VF=0.35 %
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=2 Y=8 NAM=surf %
*END
```

☞ **Note:** A zero value for a given criteria means that specific criteria will be immediately satisfied, consequently the output associated with that criteria is meaningless. The output associated with the nonzero criteria are correct.



Resulting yield surface

6.18 Example R: Ellipsoidal Inclusions

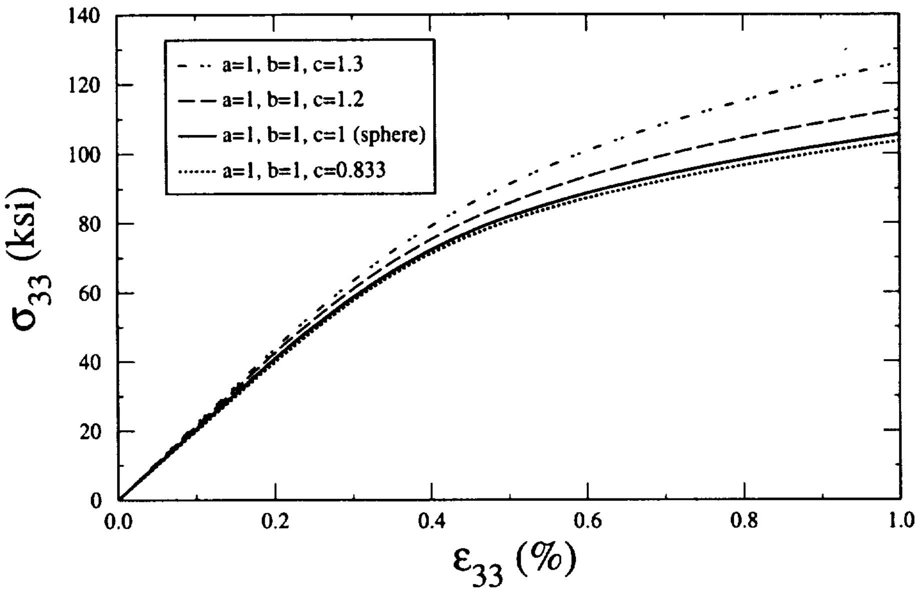
Sample Input File for Ellipsoidal Inclusions

Problem Summary

Load Type:	Thermomechanical
Load Direction:	33
Load History:	Monotonic
Load Control:	Strain
Load History Data:	Hold temperature during mechanical loading $\dot{\varepsilon}$=6.667x10^{-5} /sec, ε_{max}=0.01, ε_{min}=0. T = 427 °C; $\Delta t_{initial}$ =0.01 sec.
Micromechanics Model:	Triple Periodicity
Repeating Unit Cell:	Ellipsoidal Inclusion RVE (IDP=4) a = 1, b = 1, c = 1.3, d/h = 1, l/h = 1
Integration Algorithm:	Predictor/Corrector
Constituent Material Model:	Fiber: Elastic Matrix:TGVIPS
Constituents:	Fiber: SCS-6 Matrix:Ti-6-4

```
*PRINT
 NPL=1 %
*LOAD
 LCON=3 LOP=3 LSS=1 %
*MECH
 NPTW=2 TI=0.0, 150.0 LO=0.0, 0.01 %
*THERM
 NPTT=2 TI=0.0, 150.0 TE=427.0, 427.0 %
*MODEL
 MOD=2 %
*SOLVER
 NTF=2 ISTM=0.01 ISTT=5. ERR=0.01 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=2 MAT=E %
*MATRIX
 NMATX=1
 NM=1 MM=7 NDPT=2 MAT=A D=1.0, 1.0, 1.0 %
*MRVE
 IDP=4
 OPT=2 VF=0.3 A=1 B=1 C=1.3 RD=1 RL=1 %
#  OPT=2 VF=0.3 A=1 B=1 C=1.2 RD=1 RL=1 %
#  OPT=2 VF=0.3 A=1 B=1 C=1. RD=1 RL=1 %
#  OPT=2 VF=0.3 A=1 B=1 C=0.83333 RD=1 RL=1 %
*CURVE
 NP=1 %
*MACRO
 NT=1
 NC=1 X=3 Y=9 NAM=ellipsoid %
*END
```

☞ **Note:** To generate the four curves shown in the plot, change which line does not start with "#" under *MRVE



Results of Discontinuously reinforced composite with 30% volume fraction of ellipsoidal inclusions

☞ **Note:** As the ellipsoidal inclusion becomes longer and thinner (i.e., increasing c) and thus more fiber-like, the response becomes stiffer.

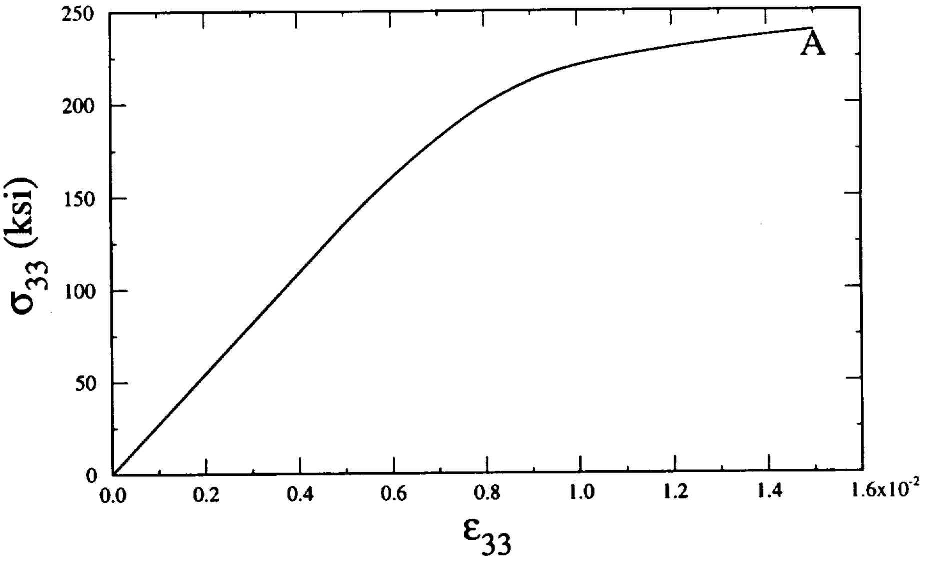
6.19 Example S: PATRAN/MSC MACPOST Output

Sample Input File for PATRAN/MSC MACPOST Output

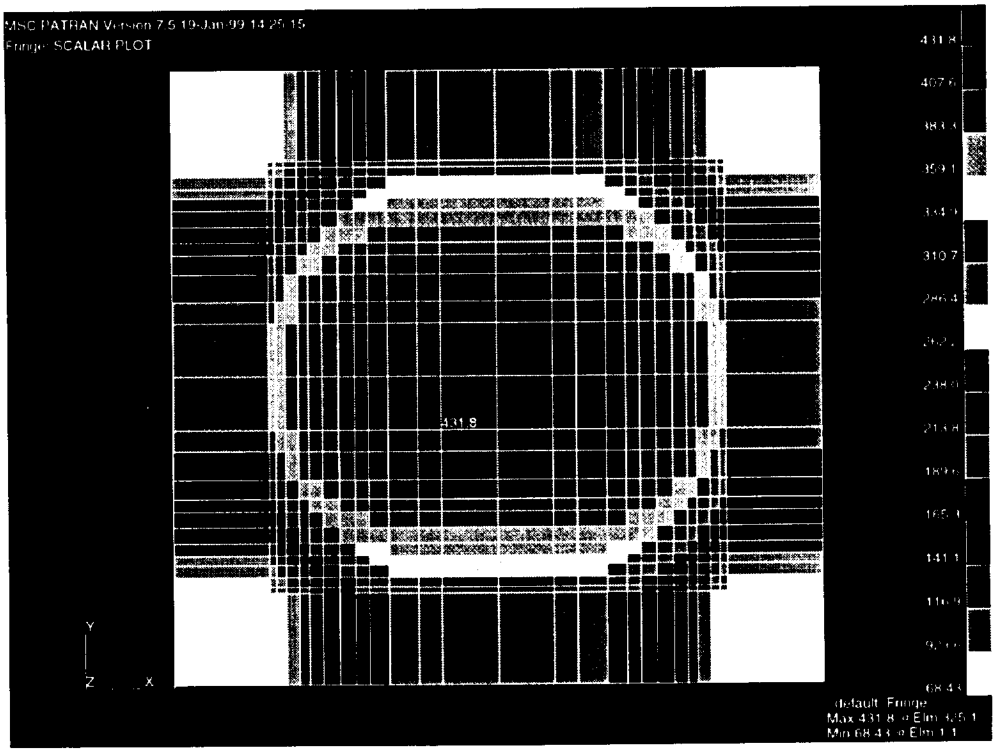
Problem Summary

Load Type:	Thermomechanical
Load Direction:	33
Load History:	Monotonic
Load Control:	Strain
Load History Data:	Cool from 900°C to 23°C, then hold temperature during mechanical loading $\dot{T}$=0.0152 °C/sec $\dot{\varepsilon}$ =1.667x10^{-4} /sec, ε_{max}=0.015, ε_{min}=0. $\Delta t_{thermal}$= 100 sec.; Δt_{mech} =0.1 sec.
Micromechanics Model:	Double Periodicity
Fiber Packing Arrangement:	Square Pack, 40% fiber volume ratio
Repeating Unit Cell:	26x26 circular fiber cross-section approximation
Integration Algorithm:	Forward Euler
Constituent Material Model:	Fiber: Elastic Matrix: GVIPS – isotropic form
Constituents:	Fiber: SCS-6 Matrix: TIMETAL21S
PATRAN Output:	Start output after cooldown: *tpre* = 57600 sec. Write to output files every 300 time steps

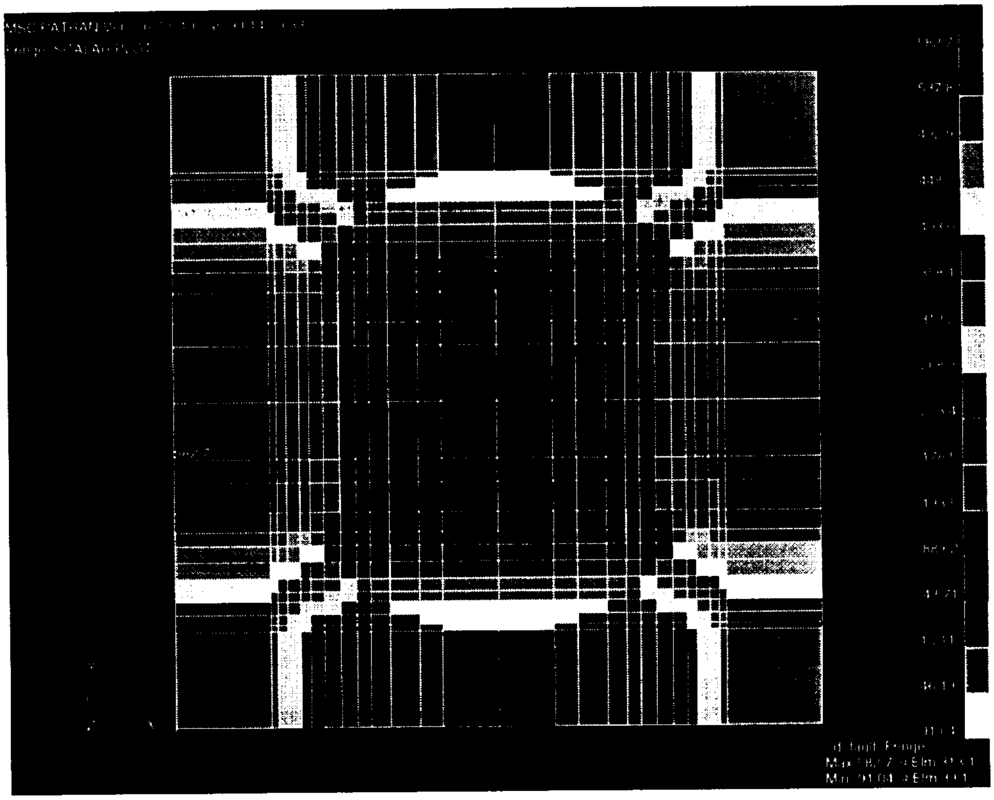
```
test of PATRAN output
*PRINT
 NPL=1 %
*LOAD
 LCON=3 LOP=3 LSS=1 %
*MECH
 NPTW=3 TI=0.,57600.,57690. LO=0.,0.,0.015 %
*THERM
 NPTT=3 TI=0.,57600.,57690. TE=900.,23.,23. %
*MODEL
 MOD=1 %
*PATRAN
 FN=apdxs TPRE=57600 STP=300  %
*SOLVER
 NTF=1 NPTS=3 TIM=0.,57600.,57690. STP=100,0.1 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=2 MAT=E %
*MATRIX
 NMATX=1
 NM=1 MM=4 NDPT=2 MAT=A %
*MRVE
 IDP=13 VF=0.4 R=1
*CURVE
 NP=2 %
*MACRO
 NT=1
 NC=1 X=3 Y=9 NAM=apdxs
*END
```



Macro stress-strain behavior results



Contour of the microlevel J_2 invariant at 1.5% macrostrain (@ pt. A)



Contour of the microlevel I_1 invariant at 1.5% macrostrain (@ pt. A)

6.20 Example T: Effective Thermal Conductivity Calculation

Sample Input File For Effective Thermal Conductivity Calculation

The following example is used to explain the calculation of effective thermal conductivities.

Problem Summary:

Loading:	Not Used (nplvl = -1 used to determine effective properties only)	
Micromechanics Model:	Double Periodicity	
Fiber Packing Arrangement:	Square Pack, R = 1., 50% fiber volume ratio	
Repeating Unit Cell:	7x7 circular fiber cross-section approximation	
Integration Algorithm:	Not Used	
Constituent Material Model:	Fiber:	Elastic, isotropic
	Matrix:	Bodner-Partom (viscoplastic properties not used)
Constituents:	Fiber:	Fictitious temperature dependent material (based loosely on tungsten)
	Matrix:	Fictitious temperature dependent material (based loosely on aluminum)

```
Test of user input properties with thermal conductivity
*PRINT
 NPL=-1 %
*LOAD
 LCON=3 LOP=2 LSS=1 %
*MECH
 NPTW=3 TI=0.,200.,218. LO=0.,0.,0.015 %
*THERM
 NPTT=3 TI=0.,200.,218. TE=600.,21.,21. %
*MODEL
 MOD=1 %
*COND
*SOLVER
 NTF=1 NPTS=3 TIM=0.,200.,218 STP=0.2,0.01 %
*FIBER
 NFIBS=1
 NF=1 MF=6 NDPT=2 MAT=U IFM=1
 NTP=4
 TEM=18.,200.,400.,600.
 EA=314.1E9,300.E9,280.E9,200.E9
 ET=314.1E9,300.E9,280.E9,200.E9
 NUA=0.41,0.41,0.41,0.41
 NUT=0.41,0.41,0.41,0.41
 GA=111.38E9,106.38E9,99.29E9,70.92E9
 ALPA=4.5E-6,4.8E-6,5.1E-6,5.5E-6
 ALPT=4.5E-6,4.8E-6,5.1E-6,5.5E-6
 KA=0.2,0.25,0.31,0.44
 KT=0.2,0.25,0.31,0.44
*MATRIX
 NMATX=1
 NM=1 MM=1 NDPT=2 MAT=U IFM=1
 NTP=4
 TEM=18.,200.,400.,600.
 EA=72.E9,67.E9,55.E9,33.E9
 ET=72.E9,67.E9,55.E9,33.E9
 NUA=0.33,0.33,0.33,0.33
 NUT=0.33,0.33,0.33,0.33
 GA=27.07E9,25.19E9,20.68E9,12.41E9
 ALPA=22.E-6,24.E-6,28.E-6,33.E-6
 ALPT=22.E-6,24.E-6,28.E-6,33.E-6
 V1=1.E4, 1.E4, 1.E4, 1.E4
 V2=340.E6, 340.E6, 340.E6, 340.E6
 V3=435.E6, 435.E6, 435.E6, 435.E6
 V4=300.0, 300.0, 300.0, 300.0
 V5=10.0, 4.0, 1.6, 0.55
 V6=1.0, 1.0, 1.0, 1.0
 KA=40.,38.,35.,29.
 KT=40.,38.,35.,29.
*MRVE
 IDP=6 VF=0.5 R=1. %
*CURVE
 NP=10 %
*MACRO
 NT=1
 NC=1 X=2 Y=8 NAM=apdxt %
*END
```

Thermal conductivity results from outfile:

```
-----------------------------------------------
 Effective Thermal Conductivities

  At Temperature =   18.0
   K Axial      =   0.201E+02
   K Transverse =   0.889E+01

  At Temperature =  200.0
   K Axial      =   0.191E+02
   K Transverse =   0.852E+01

  At Temperature =  400.0
   K Axial      =   0.177E+02
   K Transverse =   0.795E+01

  At Temperature =  600.0
   K Axial      =   0.147E+02
   K Transverse =   0.683E+01

-----------------------------------------------
```

This Page Intentionally Left Blank

INDEX

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
March 1999

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
Micromechanics Analysis Code With Generalized Method of Cells
(MAC/GMC)
User Guide: Version 3.0

5. FUNDING NUMBERS
WU-505-23-2L-00

6. AUTHOR(S)
S.M. Arnold, B.A. Bednarcyk, T.E. Wilt, and D. Trowbridge

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E-11626

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM—1999-209070

11. SUPPLEMENTARY NOTES

S.M. Arnold, NASA Glenn Research Center; B.A. Bednarcyk, Ohio Aerospace Institute, 22800 Cedar Point Road, Cleveland, Ohio 44142; T.E. Wilt and D. Trowbridge, The University of Akron, Akron, Ohio 44325-0001. Responsible person, S.M. Arnold, organization code 5920, (216) 433-3334.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Categories: 24 and 39 Distribution: Nonstandard

This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The ability to accurately predict the thermomechanical deformation response of advanced composite materials continues to play an important role in the development of these strategic materials. Analytical models that predict the effective behavior of composites are used not only by engineers performing structural analysis of large-scale composite components but also by material scientists in developing new material systems. For an analytical model to fulfill these two distinct functions it must be based on a micromechanics approach which utilizes physically based deformation and life constitutive models and allows one to generate the average (macro) response of a composite material given the properties of the individual constituents and their geometric arrangement. Here the user guide for the recently developed, computationally efficient and comprehensive micromechanics analysis code, MAC, who's predictive capability rests entirely upon the fully analytical generalized method of cells, GMC, micromechanics model is described. MAC/GMC is a versatile form of research software that "drives" the double or triply periodic micromechanics constitutive models based upon GMC. MAC/GMC enhances the basic capabilities of GMC by providing a modular framework wherein 1) various thermal, mechanical (stress or strain control) and thermomechanical load histories can be imposed, 2) different integration algorithms may be selected, 3) a variety of material constitutive models (both deformation and life) may be utilized and/or implemented, and 4) a variety of fiber architectures (both unidirectional, laminate and woven) may be easily accessed through their corresponding representative volume elements contained within the supplied library of RVEs or input directly by the user, and 5) graphical post processing of the macro and/or micro field quantities is made available.

14. SUBJECT TERMS
Composites; Computer code; Elasticity; Micromechanics; Viscoplasticity; Deformation; Life

15. NUMBER OF PAGES
231

16. PRICE CODE
A11

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT