Micromechanical Modeling of Composite Materials in Finite Element Analysis Using an Embedded Cell Approach

by

Jeffrey P. Gardner

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degrees of

Master of Science

and

Bachelor of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1994

Author . .....
Department of Mechanical Engineering
May 13, 1994

Certified by .....
Richard W. Macek
Company Supervisor
Thesis Supervisor

Certified by .....
Mary C. Boyce
Associate Professor of Mechanical Engineering
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairman, Departmental Committee on Graduate Students

Micromechanical Modeling of Composite Materials in Finite Element Analysis Using an Embedded Cell Approach

by

Jeffrey P. Gardner

Submitted to the Department of Mechanical Engineering
on May 13, 1994, in partial fulfillment of the requirements for the degrees of
Master of Science
and
Bachelor of Science in Mechanical Engineering

Abstract

The material properties of composites can be heavily dependent on localized phenomena. As a result, micromechanical models have been introduced to account for these phenomena. In this thesis, the micromechanical method of cells model by Aboudi is cast into a finite element framework. The model is first implemented for linear-elastic, continuous fiber composites. During the implementation, additional interface elements are introduced into the unit cell to later provide for damage evolution in the composite. The resulting finite element user material is compared with the original Aboudi model equations and standard finite element solutions. The model is also used to approximate a statistical representation of the composite geometry by introducing variability into the volume fraction.

A Newton iteration scheme on the displacements is introduced into the material model to allow for nonlinear material behavior. The interface elements are given a failure criterion to model debonding between the fiber and matrix in addition to brittle fracture of the matrix and fibers.

A series of problems (loadings include a temperature change, a thermal gradient, distributed pressure, and beam bending) are analyzed demonstrating the prediction of local fiber and matrix stress states in addition to the macroscopic stress state of the composite. It is shown that a statistical representation of the fiber volume fraction increases the predicted maximum constituent stresses. Debonding and fiber breakage are examined to demonstrate the resulting degradation of the composite stiffness.

The use of the method of cells material model is found to have a large effect on the computational expense of finite element analysis, especially in nonlinear analyses. However, this effect decreases with increasing problem size and depends upon computer architecture. Due to the continually improving power of even desktop workstations, the use of micromechanical material models in finite element analysis, and the method of cells in particular, is found to be a viable and powerful option.

Thesis Supervisor: Richard W. Macek
Title: Company Supervisor

Thesis Supervisor: Mary C. Boyce
Title: Associate Professor of Mechanical Engineering

Acknowledgments

I wish to thank Richard W. Macek for his tireless patience and guidance during the time I spent working at Los Alamos National Laboratory. Even though he may not believe it, his wisdom was recognized and greatly appreciated. I would also like to thank Professor Robert M. Hackett for his contributions to enlightening me and Professor Mary C. Boyce for her time and patience in helping me to write this thesis.

I must also recognize Stefan, Jim, Jason, Troy, and Monte, my friends, for their neverending lightheartedness and good humor. I owe you one, or maybe a couple...

I wish to thank Paul Smith, Richard Browning, Ronald Flury, North Carey, and all the rest of the gang at Los Alamos for their advice and assistance. I am especially indebted to Brian and Mary, my officemates at Los Alamos, for their humor and ramblings and Elizabeth, Kay, Russ, and Andrew for helping me to endure.

Thanks to the ESA-11 group of Los Alamos National Laboratory for providing the resources and financial support during the time I was performing my research there.

Last, and most importantly, I wish to thank my family; without whom I would never have had the opportunity to complete this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The discipline of composite materials is constantly providing engineers with stiffer and stronger, yet lighter materials. The design of composite materials provides great flexibility in choosing a material. In fact, many times materials can be custom tailored to meet the design needs of a particular engineering task. This flexibility has in the end led to vastly improved products. However, not everything about composite materials make life easier for the design engineer. Composite materials are generally anisotropic or at best, transversely isotropic. This fact greatly complicates the analysis of their behavior necessary to the design process. In addition, not only are most composites anisotropic, but often times the reinforcement material, the matrix material, or both may be non-elastic or even nonlinear in their behavior. This complicates the analysis even further. Finally, the properties of the composite itself are often not known, particularly if it is a new layup of materials or if the constituent materials themselves have been changed. As a result, extensive testing must often times be performed before the composite will be usable. In short, the analysis of composite materials requires knowledge of not only anisotropy, but also appropriate structural theory to derive the laminate properties. In addition, if the composite is to truly be pushed to its limits, failure criteria must also be included. [25, 47]

Many composite analyses are performed using a macroscopic approach. In this approach, the properties of the composite are homogenized to produce an anisotropic, yet homogeneous continuum before the analysis is conducted [15]. The true nature

of the composite is generally one of a randomly spaced anisotropic reinforcement material in an isotropic medium. In contrast to the macroscopic approach, the micromechanical approach to analyzing composites instead considers the properties of the fiber and matrix separately and applies the loading and boundary conditions at the individual fiber and matrix level. The overall properties of the composite are developed by relating the average stresses and strains. In doing so, the micromechanical approach may provide much more detail into the true interactions between the fiber and matrix, potentially leading to a more accurate model of the composite behavior.

One of the advantages of a micromechanical approach to deriving the effective material properties arises from the fact that many composites are formed of layers in addition to being anisotropic. A micromechanical approach can be performed on the composite provided that the individual phase properties are known; the effective material properties for the composite are a result of the analysis. A macroscopic analysis on the other hand requires that the effective material properties be known before the analysis may be performed. As the effective properties are a function of the configuration of the individual layers, in a macroscopic analysis a different layup is a completely different composite whereas the micromechanical analysis may still be performed by simply changing the orientation of the layers. A macroscopic analysis is however usually less costly in terms of computation time due to the fact that the properties are calculated off-line.

Another advantage of micromechanical analysis falls in the area of failure. Failure in composites usually occurs at the micromechanical level and is difficult to capture in a macroscopic model using macroscopic failure criteria. Failure at the microscopic level can take many forms including fiber breakage, matrix cracking, and matrix-fiber interface debonding, or damage. Failure at the interface between phases is of particular interest due to the fact that it is this type of damage that is most common in composites. Modeling the interface between the matrix and the fiber becomes very involved and only a cursory model of localized damage is introduced in the work of this thesis.

Other benefits of micromechanical analysis include the ability to study the ef-

fects of reinforcement volume fraction and thermal stresses at the matrix-fiber interface [11].

1.1 Finite Element Analysis

With the advent of computers, finite element analysis has become one of the most important tools available to an engineer for use in design analysis. The finite element method is one of the most general procedures for attacking complex analysis problems. The aim of this work is to increase its generality even more by expanding the material model library. This was done by casting a micromechanical composite model into the finite element framework. The micromechanical model is then applied by the finite element program at every material calculation point in the finite element mesh. By selecting a model with the capability to analyze a number of different composite types, it should greatly increase the flexibility of composite analysis. As always though, the most important steps in using the finite element method still reside with the engineer in making an appropriate choice for the idealization of the problem and correctly interpreting the results. [19]

The micromechanical material model was developed to be used with ABAQUS, a large commercially available finite element code. ABAQUS provides the analyst with the ability to add to the material and element libraries through the use of user subroutines coded in FORTRAN. These subroutines are entirely the responsibility of the developer; the only requirements on them are that they provide the information needed by ABAQUS for the solution.

1.2 Micromechanical Composite Models

It must be pointed out that micromechanics models are still only approximate models of the behavior of composite materials. This begins with the approximation used for the geometry. It is practically impossible, and also generally undesirable, to use a model based on the actual spatial distribution of the reinforcing material within

the specific composite which is to be used in a design. Instead, two approaches are commonly used to arrive at an approximation for the geometry. The first of these is the use of a statistical distribution for the fiber within the matrix material. The fiber spacing is hence a random variable. In the other geometry approximation, a periodic structure is assumed in which the fiber is evenly spaced throughout the matrix continuum. This approach is generally simpler and allows the analysis of a single unit cell of the material. The use of a periodic distribution is typically justified when the volume fraction of fibers is high.

Many micromechanical models have been proposed over the years for use in computing the effective material properties of composites. A very brief review of some of the ideas behind these models will be presented here. A more complete review can be found in Chapter 2 of Aboudi [16].

1.2.1 The Voigt Approximation

The first model, introduced by Voigt, is probably the simplest. It finds the effective material stiffness as the combination of the individual material stiffnesses weighted by the appropriate volume fractions, corresponding to the assumption that the strain is constant throughout the composite. That is,

$$[C^*] = v_f[C_1] + (1 - v_f)[C_2] \tag{1.1}$$

where $[C^*]$ is the effective material stiffness matrix of the composite, $[C_1]$ is the stiffness matrix of the fiber, $[C_2]$ is the stiffness matrix of the matrix material, and v_f is the fiber volume fraction.

1.2.2 The Reuss Approximation

Another very simplistic model is that proposed by Reuss. The assumption here is that the stress is constant throughout the composite. In this case it is then the effective

compliance which is a weighted combination of the individual material compliances,

$$[S^*] = v_f[S_1] + (1 - v_f)[S_2] \tag{1.2}$$

where $[S^*]$ is the effective compliance matrix of the composite, $[S_1]$ is the compliance matrix of the fiber, and $[S_2]$ is the compliance matrix of the matrix.

It was shown by Hill [37] that the Voigt and Reuss approximations bound the actual overall moduli. The Voigt approximation provides the upper bound while the Reuss approximation provides the lower bound [16].

1.2.3 The Self-Consistent Scheme

The version of the self-consistent scheme discussed here is that proposed by Hill [38]. In this model it is assumed that a single fiber exists in an infinite homogeneous medium as shown in Figure 1-1(a). This medium has the properties of the composite that are to be developed by the model itself. A uniform strain in the fiber can be produced by applying a uniform force on the boundary of the continuum. The uniform strain is then assumed to be the average over all the fibers in the composite. This assumption is the basic tenet of the self-consistent scheme from which the effective moduli can then be calculated. The self-consistent model has a physically sound base and has been found to provide reliable results. One criticism of self-consistent models to be kept in mind is that they often do not work well for composites with intermediate and high volume fractions of fibers.

The self-consistent method has been extended to applications besides simple elasticity. For example, Dvorak and Bahei-El-Din extend it to allow for elastic-plastic matrix materials in [28]. In doing so, it was necessary for them to change the geometry of the representative cell. A composite cylinder inclusion was substituted for the fiber in the original representative cell of the self-consistent scheme. This composite cylinder consists of the fiber surrounded by a thick layer of matrix material. The modified model then assumes that the composite cylinder is contained within an elastic-plastic medium which has the same properties as the composite. This model

is often referred to as the vanishing fiber diameter model because the fiber diameter, while finite, is assumed to be small enough to have no effect on the matrix behavior in the plane transverse to the fiber's axis. See Figure 1-1(b).

1.2.4 The Method of Cells

The method of cells, developed by Aboudi [1, 2, 5, 4, 6, 12], makes use of a periodic rectangular array for the inclusion geometry, as shown in Figure 2-1(a). The unit cell used to construct the regular array consists of four subcells, one for the fiber and three for the matrix as shown in Figure 2-1(b). The effective stiffness matrix is derived by relating the average stresses to the average strains inside the subcells, and then averaged over the volume of the unit cell.

The continuous fiber case of the method of cells was the micromechanical model selected for use in this thesis. This decision was made based on the following issues:

- Computational expense, generally measured in computation time. Perhaps the most important factor in the decision. The use of a complex model would most certainly have been too computationally expensive for actual use in finite element solutions of large problems[1]. The method of cells as used here is really a first order application of a higher-order theory developed by Aboudi [1].

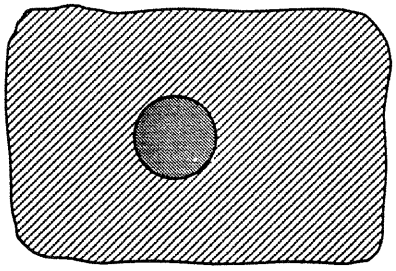
- Capability to analyze nonelastic constituents. Many of the other models do not generalize easily to nonelastic material models for the matrix and reinforcing material while maintaining the same representative geometry.

- Ability to perform a full three-dimensional analysis. This is particularly important when the materials are allowed to become non-elastic.

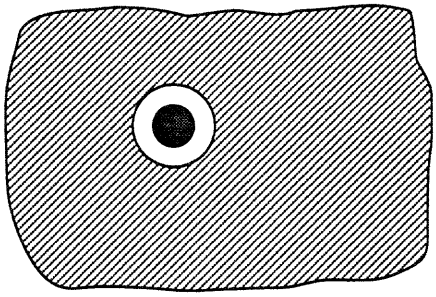
- Ease of adapting to a finite element framework. The method of cells follows a method very similar to finite elements to begin with.

- Provides results which agree well with experimental data and other micromechanical models. In all of the papers researched for this thesis, the results for

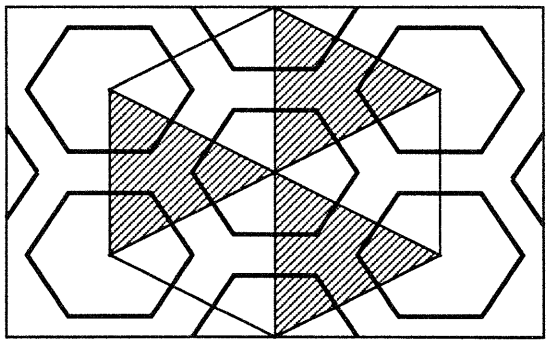
[1] measuring size in terms of numbers of degrees of freedom



(a)



(b)



(c)

Figure 1-1: Different Unit Cells Used in Micromechanical Analysis
(a) The Self-Consistent Scheme (b) The Modified Self-Consistent Scheme
(c) Teply-Dvorak Homogenization Scheme

the method of cells were always found to be within both the scatter of the experimental data and the Hashin-Shtrikman bounds [35].

A complete description of the method of cells is left for Chapter 2 since it will be presented in far more detail than the other models outlined here.

1.2.5 The Teply-Dvorak Homogenization Model

Teply and Dvorak use minimum principles of plasticity in [52] to eliminate some of the limitations of the previous models in analyzing behavior when an elastic-plastic material undergoes plastic deformation. Similar to the approach of Aboudi, they use a periodic model to approximate the composite geometry. However, the fibers in this model are assumed to have a hexagonal cross-section in contrast to the square cross-section used in the method of cells. The unit cell Teply and Dvorak chose is a triangle linking the centers of three adjacent hexagonal fibers. Each fiber is then part of six different unit cells, as shown in Figure 1-1(c). Teply and Dvorak refer to the microstructure as a periodic hexagonal array, abbreviated PHA. The homogenization to derive the overall properties is based on a comparison of unit cell energies in the PHA and the resulting homogeneous medium.

Some additional micromechanical models based on a unit cell approach can be found in [39, 31, 23, 46, 56, 49, 33].

1.3 Comparison of Models

The natural questions to ask at this point are which model provides better results and what limits are there to those results. To get a better understanding of the answers to those questions, comparisons are generally made between the results of the different models.

One such comparison is made by Teply and Reddy in [53]. Teply and Reddy attempt to establish a "unified formulation for micromechanics models" using a finite element formulation. Using this finite element formulation they are prepared to make comparisons between the models on the issues of relative convergence and accuracy of

the overall properties developed. The Aboudi method of cells model and the Teply-Dvorak model are discussed in depth [16, 52]. In order to make the comparison, Teply and Reddy cast the Aboudi model into a finite element model. The formulation is essentially that of a hybrid element, with independent approximations for the displacements and stresses. Consistent with the method of cells, a linear displacement interpolation is used while the stresses are interpolated using a piece-wise constant approximation. Using the homogenization procedure developed by Teply and Dvorak in formulating their model into finite elements [52], it is shown mathematically that the method of cells solution for the overall properties is equivalent to the homogenized method of cells model developed here. The main result Teply and Reddy find is that the method of cells provides stiffness and compliance moduli that constitute lower and upper bounds, respectively, for the actual moduli of the composite.

Another evaluation of the results of the method of cells was performed by Bigelow, Johnson, and Naik [22]. In it the method of cells is compared with three other micromechanical models for metal matrix composites. The three other models used are the vanishing fiber diameter model [28], the multi-cell model [39], and the discrete fiber-matrix model [31]. The four models are very similar in their basic setup; for example, all four of the models assume a square periodic array of continuous fibers. This facilitates direct comparison rather than necessitating a new formulation for each model as was seen in Teply and Reddy [53]. The results of the models for the overall laminate properties and the stress-strain behavior are compared to each other and to experimental data. In addition, the stresses inside the constituents are also compared. The results of the comparison find that all four models did reasonably well in predicting the overall laminate properties and stress-strain behavior. The differences between the models were generally found to be smaller than the variation in the experimental results, making it hard to claim one model performed better than another. On the other hand, when it comes to the area of constituent stresses it is clear that the discrete fiber-matrix model performs better than the other models. This is to be expected though since it is designed to provide accurate values for the fiber and matrix stresses whereas the remaining three are designed more for the

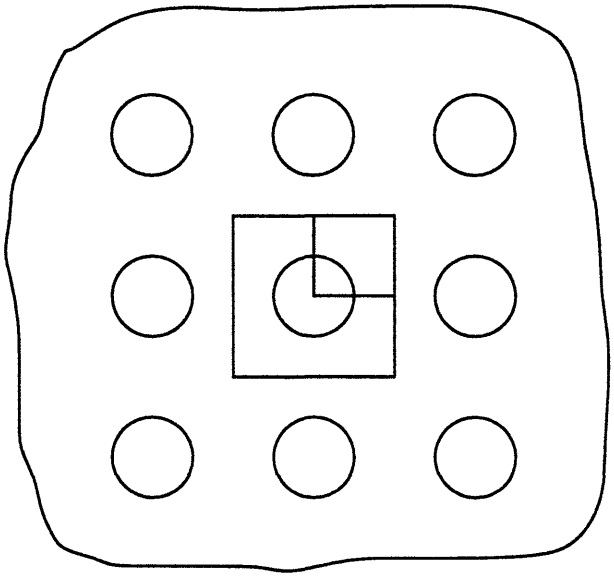
determination of overall laminate properties.

Robertson and Mall have developed a modified version of the method of cells [49]. This model maintains nearly all the tenets of the method of cells but combines it with the vanishing fiber diameter model and multi-cell model by using the assumption that composite normal stresses will not produce shear stresses in either the fiber or matrix. The unit cell used is slightly altered from that of Aboudi. The rectangular periodic array is still used but it is sectioned differently than in the method of cells, as shown in Figure 1-2. The representative volume element is shown in Figure 1-2(a) as the box completely containing a single fiber. The unit cell is then a quarter of this representative volume element. The unit cell may then be sectioned further into matrix and fiber subcells. Figure 1-2(b) shows the eight region model used by Robertson and Mall. Their aim was to simplify the approach used by Aboudi so as to reduce the expense of performing a full three dimensional analysis using nonlinear constituent materials. The results presented show that the free transverse shear approach, as it has been named, provides results that agree quite well with that of Aboudi and finite element solutions for the effective moduli.

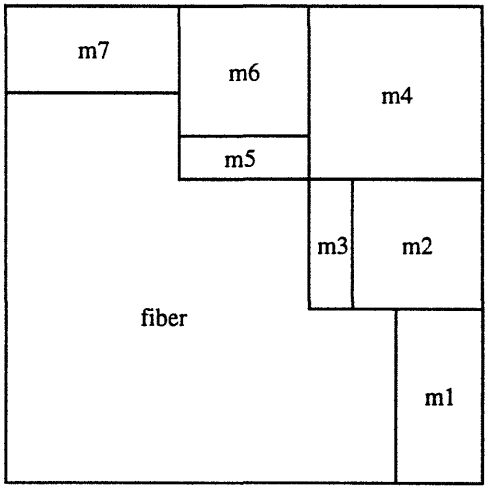
1.4 Literature Review

The use of averaging techniques, or homogenization, as used in the method of cells to arrive at the overall properties of an inhomogeneous material has received a lot of attention for use in composite analysis.

Micromechanical analysis of composites has other applications besides simply calculating the overall stiffness properties. As previously mentioned, it may be used to study the effect of interfacial properties, interfacial debonding, and even the individual constituent stresses. Divakar and Fafitis [27] have used it to study the effect of interface shear in concrete, while King et al. [42] have used it to study the effect of the matrix and interfacial bond strength on the shear strength of carbon fiber composites. In addition, micromechanical models are well suited to studying continuum damage in composites as shown by Bazant [20], Yang and Boehler [55], Ju [40], and



(a)



(b)

Figure 1-2: Unit Cell for the Free Transverse Shear Approach
(a) Representative Volume Element and Unit Cell (b) Further Division of the Unit Cell into Matrix and Fiber Subcells

Lene [43].

Bendsoe and Kikuchi have used homogenization techniques in optimizing the shape design of structural elements [21]. They use the method to turn the shape optimization problem into one of finding the optimal distribution of material. This is done by introducing a composite framework made up of substance and void. The method of homogenization is then used to determine the effective macroscopic material properties. Like the method of cells, the material model is based on a micromechanical model to derive these macroscopic properties. A unit cell consisting of the actual material plus one or more holes is used to construct the composite by repeating the cell so as to create a periodic array. The use of voids in the place of a reinforcement material provides the effective material properties as a function of the density of the material; this relationship may then be used to optimize the shape of the design for the given loads and design requirements. More information on this application of homogenization can be found in [50, 30, 24, 32].

The history of the method of cells itself has seen it applied to many different types of analyses. Aboudi himself has developed many of these applications (refer to Chapter 2 for a list of these applications), but he is not alone. Some examples have already been given in the form of the work of Teply and Reddy [53] and Robertson and Mall [49]. In addition to these examples, Yancey and Pindera [54] have used the method of cells to analyze the creep response of composites with viscoelastic matrix materials and elastic fibers. Pindera has also applied the method of cells to elastoplastic models for metal matrix materials, working with Lin [48]. Similarly, Arenburg and Reddy [18] have also studied the behavior of metal matrix composite structures with the method of cells. Perhaps the most interesting use of the method of cells is that used by Engelstad and Reddy in [29]. Engelstad and Reddy develop a nonlinear probabilistic finite element technique for the analysis of composite shell laminates in an attempt to study the effect of variability in composites. They use a first-order second-moment method to create the probabilistic finite element model. In the analysis all the material properties act as random variables along with the ply thickness and ply angle. The method of cells is then used to calculate the ply-level

properties based on the randomly varying constituent material inputs.

1.5 Scope of This Work

It is shown in this thesis that the method of cells developed by Aboudi can be cast into a general user material routine for use in finite element analysis. The main scope of this thesis has been to establish this user material routine as a framework to which modification can be done easily in extending the model to include more complicated material models for the constituents. The work for this thesis was performed in conjunction with the ESA-11 group of Los Alamos National Laboratory located in Los Alamos, NM. The end product is intended to be a general analysis tool for their use. Their desire was to have a simple working model to allow them to perform composite analysis. The intention was that in the future, after the framework for micromechanical analysis had been put in place, higher order micromechanical methods and more complicated material models may then be added as computing resources permit.

A detailed description of the method of cells is given in Chapter 2. This chapter is intended to familiarize the reader with the specifics of the method of cells as developed by Aboudi. The description is given for a continuous fiber composite whose constituents are strictly elastic as it is simplest. The method is detailed only for the derivation of the elastic properties. The reader interested in the derivation of thermoelastic properties and extensions of the model is referred to [16], Aboudi's numerous papers, and the applications described above.

The finite element formulation used for the method of cells is outlined in Chapter 3. The method is cast into the form of an user material using the continuous fiber version of the method of cells outlined in Chapter 2. In the development of the user material, an extension of the model is introduced to allow the capability to model damage evolution over time in the composite.

The testing of the user material routine is discussed in Chapter 4. The results obtained from finite element analysis are compared with the analytical results of the Aboudi model. Some examples of composite analysis using the user material

are also presented demonstrating some of the advantages of the method of cells and micromechanical analysis in general. Damage is not allowed to occur in the composite for the analyses of this chapter.

Nonlinearity is introduced into the finite element user material in Chapter 5. This is done by allowing the composite to debond over time as a function of the loading history. The function used to represent the failure of the bond is very approximate with the emphasis placed on setting up the nonlinear iteration scheme rather than implementing a detailed model of the behavior at the interface. A simple finite element analysis is performed to demonstrate the degradation of the overall moduli as damage evolves in the composite. The matrix and reinforcement materials remain perfectly elastic in this analysis even though the composite is allowed to debond.

Chapter 2

The Method of Cells

Aboudi has written numerous papers outlining the use of the method of cells to derive the properties for different composite applications. These applications include:

- Calculation of the elastic moduli and thermoelastic properties for continuous fiber, short fiber and particulate composites [2, 4, 5, 12].
- Calculation of the instantaneous properties of elastoplastic, i.e. metal-matrix, composites [6, 7, 10, 3].
- Calculation of the average properties for viscoelastic and elastic-viscoelastic composites [14, 1, 17].
- Prediction of strength properties [11, 13].
- The effects of damage and imperfect bonding on the effective properties of a composite [10, 36, 8, 9].
- Prediction of the behavior of composites with nonlinear constituents [15].

A condensed and consolidated review of Aboudi's work with the method of cells up until 1991 can be found in [16].

In the interest of clarifying and keeping the terminology consistent, the description here of the method of cells uses a slightly different definition of terms than that used by Aboudi. The representative volume element described by Aboudi will here be

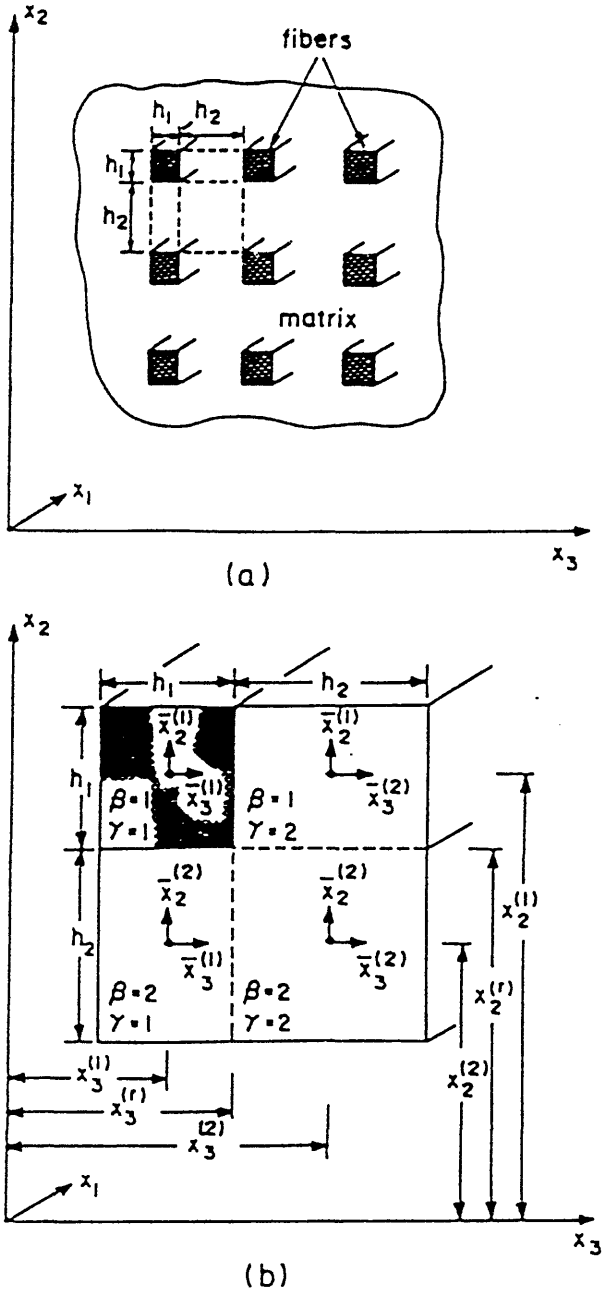
designated a representative volume cell and the cells inside the representative volume element will be called elements, or subcell elements. In effect, the use of the terms has been interchanged for reasons that will become apparent when the finite element adaptation is discussed.

The method of cells will be discussed here for the case of elastic continuous fibers. The derivation of thermoelastic properties as well as the derivation of properties for other material states and geometries is left to the references cited above. The following sections are based on the derivation of the constitutive equations described by Aboudi in [16]. The notation adopted is that proposed by Aboudi so as to not introduce confusion should the reader choose to study some of the extensions to the method of cells described above.

2.1 Assumptions and Geometry

As mentioned previously, the method of cells is based upon the assumption that the composite can be approximated by a periodic array. In using this periodicity, it is possible to analyze a single representative volume element of the continuum rather than the whole continuum. The representative volume element is then used as the building block from which the continuum is constructed, as shown in Figure 2-1(a). As Aboudi himself describes it, the representative volume element must meet two criteria [16]. First, the element must include enough information to correctly represent the continuum, i.e. it must include all the phases present in the continuum. Secondly, the element must be structurally similar to the composite on the whole. These conditions are met by the cell structure shown in Figure 2-1(b).

The microstructure of the composite is modeled within each representative volume element, attempting to better represent the interactions between the matrix and fiber. The matrix is represented by a number of elements inside of each representative volume cell while the reinforcing material is allotted a single element. For the continuous fiber case pursued here, the matrix is assigned three elements in the cell. The coordinate system is set up so that the fibers are assumed to extend into the



J. Aboudi

Figure 2-1: Geometry and Unit Cell for the Method of Cells
(a) Composite Arranged as a Periodic Array of Fibers
(b) Unit Cell for the Method of Cells

global x_1 direction. The periodic array can then be seen in the x_2, x_3 plane, with a cross-sectional view of the element shown in Figure 2-1(b). Following Aboudi's notation for numbering the elements, the fiber element is designated $\beta = 1$ and $\gamma = 1$. The remaining elements, (β, γ) = (1,2), (2,1), and (2,2) are matrix elements. The length of one side of the cell is assumed to be $h_1 + h_2$, where h_1 is the width of the fiber. Since the fiber is transversely isotropic (isotropic in the h_2, h_3 plane), the cross-sectional area of the fiber is then h_1^2. The remaining length, h_2 can be calculated based on the fiber volume fraction of the composite. As shown in Figure 2-1(b), local coordinate systems are defined for each element, the origin of each centered in the element. These local coordinates are designated as $\overline{x}_2^{\beta}$ and $\overline{x}_3^{\gamma}$.

Using these local coordinate systems, the displacements within each element are interpolated linearly from the center. It is possible to use a linear displacement interpolation here since it is the average properties of the composite that are being calculated. Again following Aboudi's notation, the displacement interpolations inside each element may be written:

$$u_i^{(\beta\gamma)} = w_i^{(\beta\gamma)} + \overline{x}_2^{(\beta)} \phi_i^{(\beta\gamma)} + \overline{x}_3^{(\gamma)} \psi_i^{(\beta\gamma)} \tag{2.1}$$

where $i = 1, 2, 3$ and $w_i^{(\beta\gamma)}$ is the displacement of the center of the element. As the displacement interpolation is linear, $\phi_i^{(\beta\gamma)}$ and $\psi_i^{(\beta\gamma)}$ represent the constant coefficients of the linear dependence on the subcell coordinates.

Based on this displacement interpolation, the strains are then calculated as:

$$\{\epsilon_{ij}^{(\beta\gamma)}\} = \frac{1}{2}[\partial_i u_j^{(\beta\gamma)} + \partial_j u_i^{(\beta\gamma)}] \tag{2.2}$$

where ∂ represents partial differentiation with respect to the coordinate noted in the subscript and $i, j = 1, 2, 3$. The strain tensor is ordered here as

$$\{\epsilon^{(\beta\gamma)}\} = [\epsilon_{11}^{(\beta\gamma)}, \epsilon_{22}^{(\beta\gamma)}, \epsilon_{33}^{(\beta\gamma)}, 2\epsilon_{12}^{(\beta\gamma)}, 2\epsilon_{13}^{(\beta\gamma)}, 2\epsilon_{23}^{(\beta\gamma)}] \tag{2.3}$$

The stresses may then be calculated from the strains and the coefficients of thermal

expansion:

$$\{\sigma^{(\beta\gamma)}\} = [C^{(\beta\gamma)}]\{\epsilon^{(\beta\gamma)}\} - \{\Gamma^{(\beta\gamma)}\}\Delta T \tag{2.4}$$

where the stiffness matrix is

$$[C^{(\beta\gamma)}] = \begin{bmatrix} c_{11}^{(\beta\gamma)} & c_{12}^{(\beta\gamma)} & c_{13}^{(\beta\gamma)} & 0 & 0 & 0 \\ & c_{22}^{(\beta\gamma)} & c_{23}^{(\beta\gamma)} & 0 & 0 & 0 \\ & & c_{33}^{(\beta\gamma)} & 0 & 0 & 0 \\ & & & c_{44}^{(\beta\gamma)} & 0 & 0 \\ & symm. & & & c_{44}^{(\beta\gamma)} & 0 \\ & & & & & c_{66}^{(\beta\gamma)} \end{bmatrix}$$

and the vector of coefficients of thermal expansion for the element is

$$\{\Gamma^{(\beta\gamma)}\} = \begin{Bmatrix} c_{11}^{(\beta\gamma)}\alpha_A^{(\beta\gamma)} + 2c_{12}^{(\beta\gamma)}\alpha_T^{(\beta\gamma)} \\ c_{12}^{(\beta\gamma)}\alpha_A^{(\beta\gamma)} + (c_{22}^{(\beta\gamma)} + c_{23}^{(\beta\gamma)})\alpha_T^{(\beta\gamma)} \\ c_{12}^{(\beta\gamma)}\alpha_A^{(\beta\gamma)} + (c_{22}^{(\beta\gamma)} + c_{23}^{(\beta\gamma)})\alpha_T^{(\beta\gamma)} \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{2.5}$$

In this equation, $\alpha_A^{(\beta\gamma)}$ and $\alpha_T^{(\beta\gamma)}$ are the axial and transverse coefficients of thermal expansion for the material of the element $(\beta\gamma)$. The stress tensor in equation 2.4 is ordered in the same manner as the strains, and ΔT is the difference between the actual temperature of the material and the reference temperature at which there are no thermal strains.

2.2 Imposition of Continuity Conditions

The interactions between the elements within a representative volume cell and between the cells themselves are expressed in terms of displacement and traction continuity conditions. In the homogenization procedure these conditions are then used

to derive conditions applicable to the whole continuum. The average properties of the composite result from this homogenization. It is important to note that since it is the average behavior of the composite being derived, the continuity conditions are imposed on an average basis. The stresses and strains which are computed using this behavior are then actually the averages over the volume. In the framework of the method of cells, this implys that the average stress and strain in the composite are computed from the average stresses and strains in the elements by taking yet another average. Thus the average stress and strain are:

$$\overline{\sigma}_{ij} = \frac{1}{V} \sum_{\beta,\gamma=1}^{2} v_{\beta\gamma} \overline{\sigma}_{ij}^{(\beta\gamma)} \tag{2.6}$$

$$\overline{\epsilon}_{ij} = \frac{1}{V} \sum_{\beta,\gamma=1}^{2} v_{\beta\gamma} \overline{\epsilon}_{ij}^{(\beta\gamma)} \tag{2.7}$$

where $v_{\beta\gamma}$ is the volume of the element $(\beta\gamma)$ and V is the volume of the representative cell. The average strains in subcell $(\beta\gamma)$ are obtained from equation 2.1 using 2.2:

$$\overline{\epsilon}_{11}^{(\beta\gamma)} = \frac{\partial}{\partial x_1} w_1 \tag{2.8}$$

$$\overline{\epsilon}_{22}^{(\beta\gamma)} = \phi_2^{(\beta\gamma)} \tag{2.9}$$

$$\overline{\epsilon}_{33}^{(\beta\gamma)} = \psi_3^{(\beta\gamma)} \tag{2.10}$$

$$2\overline{\epsilon}_{12}^{(\beta\gamma)} = \phi_1^{(\beta\gamma)} + \frac{\partial}{\partial x_1} w_2 \tag{2.11}$$

$$2\overline{\epsilon}_{13}^{(\beta\gamma)} = \psi_1^{(\beta\gamma)} + \frac{\partial}{\partial x_1} w_3 \tag{2.12}$$

$$2\overline{\epsilon}_{23}^{(\beta\gamma)} = \phi_3^{(\beta\gamma)} + \psi_2^{(\beta\gamma)} \tag{2.13}$$

The average stresses in the subcell $(\beta\gamma)$ are then calculated from 2.4. Equivalently,

$$\overline{\sigma}_{ij}^{(\beta\gamma)} = \frac{1}{v_{\beta\gamma}} \int_{-h_\gamma/2}^{h_\gamma/2} \int_{-h_\beta/2}^{h_\beta/2} \sigma_{ij}^{(\beta\gamma)} d\overline{x}_2^{(\beta)} d\overline{x}_3^{(\gamma)} \tag{2.14}$$

2.2.1 Traction Continuity

Traction continuity is imposed by simply equating the average stress components between elements:

$$\overline{\sigma}_{2i}^{(1\gamma)} = \overline{\sigma}_{2i}^{(2\gamma)} \tag{2.15}$$

and

$$\overline{\sigma}_{3i}^{(\beta 1)} = \overline{\sigma}_{3i}^{(\beta 2)} \tag{2.16}$$

2.2.2 Displacement Continuity

In order to ensure displacement continuity, it must be true that the normal and tangential displacements are equal at the interfaces between elements as shown in equations 2.17-2.18.

$$u_i^{(1\gamma)} \mid_{\overline{x}_2^{(1)}=-h_1/2} = u_i^{(2\gamma)} \mid_{\overline{x}_2^{(2)}=h_2/2} \tag{2.17}$$

$$u_i^{(\beta 1)} \mid_{\overline{x}_3^{(1)}=h_1/2} = u_i^{(\beta 2)} \mid_{\overline{x}_3^{(2)}=-h_2/2} \tag{2.18}$$

These conditions are expressed for two elements within the same representative cell. The conditions for two elements in adjacent cells are obtained by interchanging the signs of the distances at which the displacements are interpreted. In order to apply these conditions in an average sense, equations 2.17 and 2.18 must be integrated over the length of the boundary. For example, continuity between elements (1γ) and (2γ) (where $\overline{x}_2^{(1)} = \pm h_1/2$ respectively) would require that

$$\int_{-h_\gamma/2}^{h_\gamma/2} u_i^{(1\gamma)} \mid_{\overline{x}_2^{(1)}=-h_1/2} d\overline{x}_3^{(\gamma)} = \int_{-h_\gamma/2}^{h_\gamma/2} u_i^{(2\gamma)} \mid_{\overline{x}_2^{(2)}=h_2/2} d\overline{x}_3^{(\gamma)} \tag{2.19}$$

Substituting in the displacement interpolation of equation 2.1, equation 2.19 becomes

$$w_i^{(1\gamma)} - \frac{h_1}{2}\phi_i^{(1\gamma)} = w_i^{(2\gamma)} + \frac{h_2}{2}\phi_i^{(2\gamma)} \tag{2.20}$$

In order to transform these discrete equations into equations for the whole continuum, equation 2.20 must be applied throughout the whole composite. It is necessary to note first that equation 2.20 is written for the centerline $x_2^{(\beta)}$, and the distance from

the centerline to the interface between elements is $-h_1/2$ for $x_2^{(1)}$ and $h_2/2$ for $x_2^{(2)}$. Using this information, it is possible to make the transformation to the continuous case with a first order expansion of equation 2.20. The result is:

$$w_i^{1\gamma} \pm \frac{h_1}{2}\frac{\partial}{\partial x_2} w_i^{(1\gamma)} \mp \frac{h_1}{2}\phi_i^{(1\gamma)} = w_i^{2\gamma} \mp \frac{h_2}{2}\frac{\partial}{\partial x_2} w_i^{(2\gamma)} \pm \frac{h_2}{2}\phi_i^{(2\gamma)} \tag{2.21}$$

where the $\pm$ and $\mp$ denote the fact that two forms of the equation are obtained depending on whether the starting point is two elements within the same representative cell or two adjacent elements in different cells. By adding the two different relations expressed in equation 2.21, it is found that

$$w_i^{(1\gamma)} = w_i^{(2\gamma)} \tag{2.22}$$

Similarly, by subtracting the two and using equation 2.22 it is found that

$$h_1\phi_i^{(1\gamma)} + h_2\phi_i^{(2\gamma)} = (h_1 + h_2)\frac{\partial}{\partial x_2} w_i^{(1\gamma)} \tag{2.23}$$

Following the same methodology, the continuity condition of equation 2.18 provides

$$w_i^{(\beta 1)} = w_i^{(\beta 2)} \tag{2.24}$$

and

$$h_1\psi_i^{(\beta 1)} + h_2\psi_i^{(\beta 2)} = (h_1 + h_2)\frac{\partial}{\partial x_3} w_i^{(\beta 1)} \tag{2.25}$$

It can be deduced from equations 2.22 and 2.24 that

$$w_i^{(11)} = w_i^{(12)} = w_i^{(21)} = w_i^{(22)} \equiv w_i \tag{2.26}$$

The continuity of the displacements is then described by the twelve expressions which can be formed from equations 2.23, 2.25, and 2.26.

2.3 Derivation of Constitutive Relations

Using the above traction and displacement conditions, it is now possible to derive the constitutive relations for the overall composite behavior. For this derivation, both the fiber and matrix are assumed to be transversely isotropic. The method which follows is broken into two steps. The first step involves deriving the constitutive equations for an orthotropic material with square symmetry, that is, instead of transverse isotropy, the relations are for a material which is equivalent in the x_2 and x_3 directions. To obtain transverse isotropy, these relations must be rotated through 2π around the x_1 axis.

2.3.1 Square Symmetry

Using equation 2.23 for $i = 2$, the following relations are obtained for the coefficients of the displacement interpolation:

$$\phi_2^{(12)} = (h\bar{\epsilon}_{22} - h_2\phi_2^{(22)})/h_1 \tag{2.27}$$

$$\phi_2^{(21)} = (h\bar{\epsilon}_{22} - h_1\phi_2^{(11)})/h_2 \tag{2.28}$$

Likewise, substituting $i = 3$ into equation 2.25 gives

$$\psi_3^{(12)} = (h\bar{\epsilon}_{33} - h_1\psi_3^{(11)})/h_2 \tag{2.29}$$

$$\psi_3^{(21)} = (h\bar{\epsilon}_{33} - h_2\psi_3^{(22)})/h_1 \tag{2.30}$$

where the combination $h_1 + h_2$ has been defined as h.

Substituting these relations for the coefficients into the traction continuity conditions, equations 2.15 and 2.16, and using the relations for the stresses given by

equation 2.4 yields

$$\begin{aligned} A_1\phi_2^{(22)} + A_2\psi_3^{(11)} + A_3\psi_3^{(33)} &= J_1 \\ A_4\phi_2^{(11)} + A_5\psi_3^{(11)} + A_6\psi_3^{(22)} &= J_2 \\ A_7\phi_2^{(11)} + A_8\phi_2^{(22)} + A_9\psi_3^{(11)} &= J_3 \\ A_{10}\phi_2^{(11)} + A_{11}\phi_2^{(22)} + A_{12}\psi_3^{(22)} &= J_4 \end{aligned} \tag{2.31}$$

The coefficients used here are defined as:

$$\begin{aligned} A_1 = c_{22}^m(1 + h_2/h_1) \quad & A_2 = c_{23}^m(h_1/h_2) \quad & A_3 = c_{23}^m \\ A_4 = c_{22}^m(h_1/h_2) + c_{22}^f \quad & A_5 = c_{23}^f \quad & A_6 = c_{23}^m(h_2/h_1) \\ A_7 = c_{23}^f \quad & A_8 = A_6 \quad & A_9 = A_4 \\ A_{10} = c_{23}^m(h_1/h_2) \quad & A_{11} = A_3 \quad & A_{12} = A_1 \end{aligned} \tag{2.32}$$

$$\begin{aligned} J_1 &= c_{22}^m\bar{\epsilon}_{22}(h/h_1) + c_{23}^m\bar{\epsilon}_{33}(h/h_2) \\ J_2 &= (c_{12}^m - c_{12}^f)\bar{\epsilon}_{11} + c_{22}^m\bar{\epsilon}_{22}(h/h_2) + c_{23}^m\bar{\epsilon}_{33}(h/h_1) + (\Gamma_2^f - \Gamma_2^m)\Delta T \\ J_3 &= (c_{12}^m - c_{12}^f)\bar{\epsilon}_{11} + c_{23}^m\bar{\epsilon}_{22}(h/h_1) + c_{22}^m\bar{\epsilon}_{33}(h/h_2) + (\Gamma_2^f - \Gamma_2^m)\Delta T \\ J_4 &= c_{23}^m\bar{\epsilon}_{22}(h/h_2) + c_{22}^m\bar{\epsilon}_{33}(h/h_1) \end{aligned} \tag{2.33}$$

Equations 2.31 can then be solved for the coefficients of the displacement interpolation:

$$\begin{aligned} \phi_2^{(11)} &= T_1J_1 + T_2J_2 + T_3J_3 + T_4J_4 \\ \phi_2^{(22)} &= T_5J_1 + T_6J_2 + T_7J_3 + T_8J_4 \\ \psi_3^{(11)} &= T_9J_1 + T_{10}J_2 + T_{11}J_3 + T_{12}J_4 \\ \psi_3^{(22)} &= T_{13}J_1 + T_{14}J_2 + T_{15}J_3 + T_{16}J_4 \end{aligned} \tag{2.34}$$

The T_i here are defined as

$$
\begin{aligned}
DT_1 &= -(A_5A_8A_{12} + A_6A_9A_{11}) \\
DT_2 &= A_2A_8A_{12} + A_3A_9A_{11} - A_1A_9A_{12} \\
DT_3 &= A_1A_5A_{12} + A_2A_6A_{11} - A_3A_5A_{11} \\
DT_4 &= A_1A_6A_9 + A_8(A_3A_5 - A_2A_6) \\
DT_5 &= A_6A_9A_{10} + A_{12}(A_5A_7 - A_4A_9) \\
DT_6 &= -(A_2A_7A_{12} + A_3A_9A_{10}) \\
DT_7 &= A_3A_5A_{10} + A_2(A_4A_{12} - A_6A_{10}) \\
DT_8 &= A_2A_6A_7 + A_3(A_4A_9 - A_5A_7) \\
DT_9 &= A_4A_8A_{12} + A_6(A_7A_{11} - A_8A_{10}) \\
DT_{10} &= A_1A_7A_{12} + A_3(A_8A_{10} - A_7A_{11}) \\
DT_{11} &= A_3A_4A_{11} + A_1(A_6A_{10} - A_4A_{12}) \\
DT_{12} &= -(A_1A_6A_7 + A_3A_4A_8) \\
DT_{13} &= A_4A_9A_{11} + A_5(A_8A_{10} - A_7A_{11}) \\
DT_{14} &= A_1A_9A_{10} + A_2(A_7A_{11} - A_8A_{10}) \\
DT_{15} &= -(A_1A_5A_{10} + A_2A_4A_{11}) \\
DT_{16} &= A_2A_4A_8 + A_1(A_5A_7 - A_4A_9)
\end{aligned}
\tag{2.35}
$$

where

$$
\begin{aligned}
D = {} & A_1[A_6A_9A_{10} + A_{12}(A_5A_7 - A_4A_9)] \\
& + A_2[A_4A_8A_{12} + A_6(A_7A_{11} - A_8A_{10})] \\
& + A_3[A_4A_9A_{11} + A_5(A_8A_{10} - A_7A_{11})]
\end{aligned}
$$

Now that the coefficients of the displacement interpolation are known, it is possible to solve for the the normal stresses of equation 2.4. They become:

$$
\begin{aligned}
\overline{\sigma}_{11} &= b_{11}\overline{\epsilon}_{11} + b_{12}\overline{\epsilon}_{22} + b_{13}\overline{\epsilon}_{33} - \Gamma_1\Delta T \\
\overline{\sigma}_{22} &= b_{12}\overline{\epsilon}_{11} + b_{22}\overline{\epsilon}_{22} + b_{23}\overline{\epsilon}_{33} - \Gamma_2\Delta T \\
\overline{\sigma}_{33} &= b_{13}\overline{\epsilon}_{11} + b_{23}\overline{\epsilon}_{22} + b_{33}\overline{\epsilon}_{33} - \Gamma_3\Delta T
\end{aligned}
\tag{2.36}
$$

The b_{ij} here are the entries in the constitutive matrix relating the average stress to the average strain, written as the $[B]$ matrix here. They may be solved for as:

$$\begin{aligned}
Vb_{11} &= v_{11}c_{11}^f + c_{11}^m(v_{12}+v_{21}+v_{22}) + (c_{12}^m - c_{12}^f)(Q_2+Q_3) \\
Vb_{12} &= \tfrac{h}{h_1}(c_{12}^m v_{12} + Q_1 c_{22}^m + Q_3 c_{23}^m) + \tfrac{h}{h_2}(c_{12}^m v_{21} + Q_2 c_{22}^m + Q_4 c_{23}^m) \\
b_{13} &= b_{12} \\
Vb_{22} &= \tfrac{h}{h_1}[c_{22}^m(v_{12}+Q_1') + Q_3' c_{23}^m] + \tfrac{h}{h_2}[c_{22}^m(v_{21}+Q_2') + Q_4' c_{23}^m] \\
Vb_{23} &= \tfrac{h}{h_1}[c_{23}^m(v_{21}+Q_2') + Q_4' c_{22}^m] + \tfrac{h}{h_2}[c_{23}^m(v_{12}+Q_1') + Q_3' c_{22}^m] \\
b_{33} &= b_{22}
\end{aligned} \tag{2.37}$$

Also the effective coefficients of thermal expansion are:

$$\begin{aligned}
V\Gamma_1 &= (\Gamma_2^m - \Gamma_2^f)(Q_2+Q_3) + v_{11}\Gamma_1^f + (v_{12}+v_{21}+v_{22})\Gamma_1^m \\
V\Gamma_2 &= (\Gamma_2^m - \Gamma_2^f)(Q_2'+Q_3') + v_{11}\Gamma_2^f + (v_{12}+v_{21}+v_{22})\Gamma_2^m \\
\Gamma_3 &= \Gamma_2
\end{aligned} \tag{2.38}$$

The Q coefficients are defined as:

$$\begin{aligned}
Q_1 &= v_{11}c_{12}^f(T_1+T_9) - v_{12}c_{12}^m(T_5(h_2/h_1) + T_9(h_1/h_2)) \\
&\quad -v_{21}c_{12}^m(T_1(h_1/h_2) + T_{13}(h_2/h_1)) + v_{22}c_{12}^m(T_5+T_{13})
\end{aligned} \tag{2.39}$$

The remaining $Q_i, i = 2,3,4$ are found by replacing the T_j by T_{j+1}, T_{j+2}, and T_{j+3} respectively in Q_1. Similarly,

$$\begin{aligned}
Q_1' &= v_{11}(c_{22}^f T_1 + c_{23}^f T_9) - v_{12}(c_{22}^m T_5(h_2/h_1) + c_{23}^m T_9(h_1/h_2)) \\
&\quad -v_{21}(c_{22}^m T_1(h_1/h_2) + c_{23}^m T_{13}(h_2/h_1)) + v_{22}(c_{22}^m T_5 + c_{23}^m T_{13})
\end{aligned} \tag{2.40}$$

and as before the remaining $Q_i', i = 2,3,4$ are found by replacing the T_j by T_{j+1}, T_{j+2}, and T_{j+3} respectively in Q_1'.

The remaining coefficients to be determined for the constitutive matrix are the shear coefficients. The b_{44} coefficient will be determined first. To begin, $i = 1$ is

substituted into equation 2.23, resulting in:

$$\phi_1^{(21)} = (h\frac{\partial w_1}{\partial x_2} - h_1\phi_1^{(11)})/h_2 \tag{2.41}$$

$$\phi_1^{(12)} = (h\frac{\partial w_1}{\partial x_2} - h_2\phi_1^{(22)})/h_1 \tag{2.42}$$

Following the same method as for the derivation of the normal components, these relations are substituted into the traction continuity equation, equation 2.15, and using the stress relations from equation 2.4 the result can be solved for the coefficients of the linear displacement interpolation. After some lengthy algebra, it is found that:

$$\overline{\sigma}_{13} = 2b_{44}\overline{\epsilon}_{12} \tag{2.43}$$

where

$$b_{44} = c_{44}^m \left[c_{44}^f[h(v_{11} + v_{21}) + h_2(v_{12} + v_{22})] + h_1 c_{44}^m(v_{12} + v_{22})\right]/(V\Delta) \tag{2.44}$$

The term Δ is defined as $\Delta = h_1 c_{44}^m + h_2 c_{44}^f$.

From the square symmetry it follows that

$$\overline{\sigma}_{13} = 2b_{44}\overline{\epsilon}_{13} \tag{2.45}$$

There is now only one remaining coefficient to be determined, b_{66}. The derivation begins by again substituting $i = 3$ into equation 2.23 to obtain:

$$h_1\phi_3^{(1\gamma)} + h_2\phi_3^{(2\gamma)} = h\frac{\partial w_3}{\partial x_2} \tag{2.46}$$

Similarly, $i = 2$ is substituted into equation 2.25 to obtain:

$$h_1\psi_2^{(\beta 1)} + h_2\psi_2^{(\beta 2)} = h\frac{\partial w_2}{\partial x_3} \tag{2.47}$$

Continuing, equation 2.46 is multiplied by h_1 with $\gamma = 1$ and then added to the result

of multiplying equation 2.47 by h_2 for $\beta = 1$ to provide:

$$h_1^2 N^{(11)} + h_2 h_1 \phi_3^{(21)} + h_1 h_2 \psi_2^{(12)} = M_1 \tag{2.48}$$

where $N^{(11)}$ and M_1 are defined to be:

$$N^{(\beta\gamma)} = \phi_3^{(\beta\gamma)} + \psi_2^{(\beta\gamma)} \tag{2.49}$$

$$M_\beta = h h_\beta \left(\frac{\partial w_3}{\partial x_2} + \frac{\partial w_2}{\partial x_3} \right) \tag{2.50}$$

Alternatively, multiplying equation 2.46 by h_2 with $\gamma = 2$ and adding it to the result of multiplying equation 2.47 by h_2 with $\beta = 2$ yields:

$$h_2^2 N^{(22)} + h_1 h_2 \phi_3^{(12)} + h_1 h_2 \psi_2^{(21)} = M_2 \tag{2.51}$$

By adding equations 2.48 and 2.51, we obtain:

$$v_{11} N^{(11)} + v_{12} N^{(12)} + v_{21} N^{(21)} + v_{22} N^{(22)} = 2h^2 \bar{\epsilon}_{23} \tag{2.52}$$

Combining this relation with the traction continuity equations, 2.15 with $i = 3$ and 2.16 with $i = 2$, yields four equations in the four coefficients $N^{(\beta\gamma)}$. Using the fact that

$$\bar{\sigma}_{23}^{(\beta\gamma)} = c_{66}^{(\beta\gamma)} N^{(\beta\gamma)} \tag{2.53}$$

these coefficients are then:

$$\begin{aligned} N^{(11)} &= 2h^2 c_{66}^m \bar{\epsilon}_{23} / \delta \\ N^{(12)} &= 2h^2 c_{66}^f \bar{\epsilon}_{23} / \delta \\ N^{(21)} &= N^{(12)} \\ N^{(22)} &= N^{(12)} \end{aligned} \tag{2.54}$$

where

$$\delta = h_1^2 c_{66}^m + (2h_1 h_2 + h_2^2) c_{66}^f \tag{2.55}$$

It must be noted that in equation 2.53, $c_{66}^{(\beta\gamma)}$ is defined as c_{66}^f for $c_{66}^{(11)}$ and c_{66}^m for

$c_{66}^{(\beta\gamma)}, (\beta + \gamma \neq 2)$. It follows that

$$\overline{\sigma}_{23} = 2b_{66}\bar{\epsilon}_{23} \tag{2.56}$$

where

$$b_{66} = c_{66}^{f} c_{66}^{m} h^2/\delta \tag{2.57}$$

The constitutive equations for the average stresses and strains may then be written as

$$\{\overline{\sigma}\} = [B]\{\bar{\epsilon}\} - \{\Gamma\}\Delta T \tag{2.58}$$

where the stiffness matrix is

$$[B] = \begin{bmatrix} b_{11} & b_{12} & b_{12} & 0 & 0 & 0 \\ & b_{22} & b_{23} & 0 & 0 & 0 \\ & & b_{22} & 0 & 0 & 0 \\ & & & b_{44} & 0 & 0 \\ & symm. & & & b_{44} & 0 \\ & & & & & b_{66} \end{bmatrix}$$

and the vector of effective coefficients of thermal expansion is

$$\{\Gamma\} = [\Gamma_1, \Gamma_2, \Gamma_3, 0, 0, 0] \tag{2.59}$$

The order of the stress and strain tensors follow the convention set by equation 2.3.

2.3.2 Transverse Isotropy

As mentioned previously, the constitutive relation defined for equation 2.58 is for a material exhibiting square symmetry and not transverse isotropy. In order to transform these equations to transverse isotropy, all three coordinates are rotated around the x_1 axis through the angle ξ. The transformation results in a new $[B']$ which has

the components

$$\begin{aligned}
b'_{11} &= b_{11} \\
b'_{12} &= b_{12} \\
b'_{22} &= b_{22}(\cos^4\xi + \sin^4\xi) + 2(b_{23} + 2b_{66})\sin^2\xi\cos^2\xi \\
b'_{23} &= b_{23}(\cos^4\xi + \sin^4\xi) + 2(b_{22} - 2b_{66})\sin^2\xi\cos^2\xi \\
b'_{44} &= b_{44} \\
b'_{66} &= b_{66}(\cos^4\xi + \sin^4\xi) + 2(b_{22} - b_{23} - b_{66})\sin^2\xi\cos^2\xi
\end{aligned} \tag{2.60}$$

The effective stiffness constants are derived from this transformation by integrating through a full period, $\xi = [0, 2\pi]$, as follows

$$[E] = \frac{1}{\pi}\int_0^{\pi}[B'(\xi)]d\xi \tag{2.61}$$

The components of the the effective stiffness matrix are hence

$$\begin{aligned}
e_{11} &= b_{11} \\
e_{12} &= b_{12} \\
e_{22} &= \tfrac{3}{4}b_{22} + \tfrac{1}{4}b_{23} + \tfrac{1}{2}b_{66} \\
e_{23} &= \tfrac{1}{4}b_{22} + \tfrac{3}{4}b_{23} - \tfrac{1}{2}b_{66} \\
e_{44} &= b_{44} \\
e_{66} &= \tfrac{1}{2}(e_{22} - e_{23})
\end{aligned} \tag{2.62}$$

and

$$[E] = \begin{bmatrix}
e_{11} & e_{12} & e_{12} & 0 & 0 & 0 \\
 & e_{22} & e_{23} & 0 & 0 & 0 \\
 & & e_{22} & 0 & 0 & 0 \\
 & & & e_{44} & 0 & 0 \\
 & symm. & & & e_{44} & 0 \\
 & & & & & e_{66}
\end{bmatrix} \tag{2.63}$$

The new vector of effective coefficients of thermal expansion is

$$\{\Gamma'\} = [E][B^{-1}]\{\Gamma\} \tag{2.64}$$

and the constitutive equation for the transversely isotropic in its final form is thus

$$\{\overline{\sigma}\} = [E]\{\overline{\epsilon}\} - \{\Gamma'\}\Delta T \tag{2.65}$$

Chapter 3

Finite Element Adaptation

In adapting the method of cells to a finite element framework, the first step was to write an user element that would be the equivalent of the element in the representative volume cell. Another element was introduced in the process of adapting the method of cells to finite elements. This element, designated here as the interface user element[1], is not part of the method of cells framework and was introduced to add flexibility to the model through the eventual goal of modeling debonding in the composite. The interface element is used to connect the cell user elements in making up a representative volume cell and hence represents the interface between the fiber and matrix materials.

The second step in casting the method of cells into the finite element framework was to create an user material routine which would combine the user elements and create a representative volume cell. This user material routine[2] is essentially a small finite element routine. It sets up a mesh of the representative volume cell at each material point and then performs the necessary operations to derive the stiffness matrix and force vector used by the finite element program ABAQUS. It is in this user material that the homogenization techniques of the method of cells are used.

As mentioned previously, the user material subroutine and user element routines

[1]The subroutine names used in the implementation for the two user elements are ABOUDI and DAMAGE for the subcell element and the interface element, respectively.

[2]ABAQUS uses the subroutine name UMAT for its material subroutines.

were intended from the start to be used as building blocks for future modifications. As a result of this, it was attempted to write them in a modular fashion allowing later parts to be added without changing the whole.

3.1 Subcell User Element

As will be shown later in this work, the expense of using a micromechanical model of the type implemented here can be extreme in terms of computation time. The linear displacement interpolation used in the Method of Cells, while basic, helps to keep the increase in expense from becoming inhibitive.

3.1.1 Geometry

In translating the element of the representative volume cell into a working user element for ABAQUS, a six-noded, three-dimensional finite element was chosen, see Figure 3-1. The nodes are placed in the center of each face of the rectangular element. Three degrees of freedom are allowed at each node. Once again, a local coordinate system is defined with $\overline{x}_1$ running along the axis of the fiber and $\overline{x}_2$, $\overline{x}_3$ defining the plane of isotropy in the fiber. The relative dimensions of the element are designated d_1, d_2, and d_3. Since in the end the properties are averaged over the volume, the actual size of the dimensions is irrelevant. To simplify the computation, the total volume of the element is consequently chosen to be unity, as is d_1, the length of the fiber inside of the representative volume cell. The remaining dimensions may then be calculated from the fiber volume fraction. The stiffness and force vectors are integrated in one point located at the center of the element.

Since the stresses and strains in the composite are to be averaged over the volume, the actual dimensions of the fiber and matrix are unimportant. Hence the volume of each representative volume element is assumed to be one. The depth of the fiber, that is the length of the fiber in the x_1 direction, is also assumed to be one and then the corresponding dimensions h_1 and h_2 are calculated based on the volume fraction of the reinforcing material.

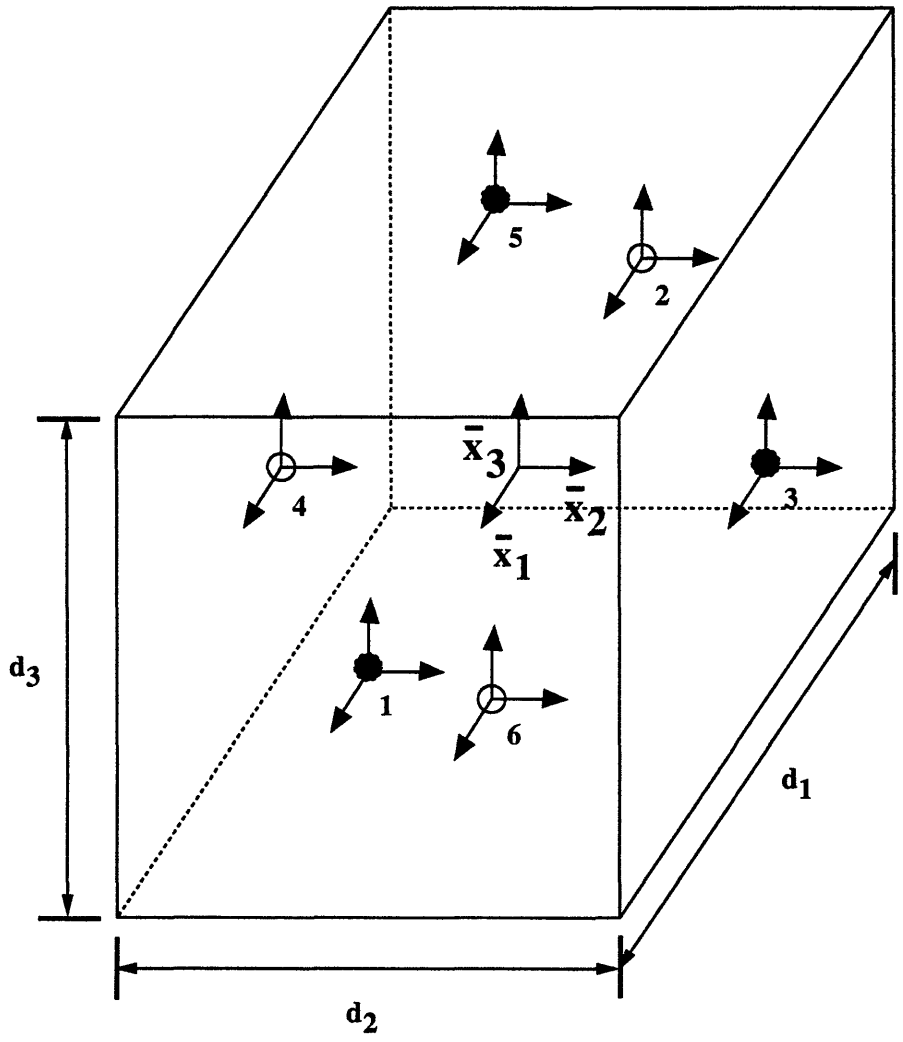


Figure 3-1: Subcell User Element

3.1.2 Derivation of the Stiffness Matrix

As in the method of cells, a linear displacement interpolation is used from the center of each cell. The displacement interpolation may be written here as:

$$u_i^{(n)}(\overline{x}_1, \overline{x}_2, \overline{x}_3) = U_i + \overline{x}_1\phi_i + \overline{x}_2\chi_i + \overline{x}_3\psi_i \tag{3.1}$$

where $i = 1, 2, 3$ and n is the node number. The coefficients of the interpolation in $\overline{x}_1$, $\overline{x}_2$, and $\overline{x}_3$ are then:

$$\phi_i = du_i/d\overline{x}_1 \tag{3.2}$$

$$\chi_i = du_i/d\overline{x}_2 \tag{3.3}$$

and

$$\psi_i = du_i/d\overline{x}_3 \tag{3.4}$$

Since the interpolation of the displacement is assumed to be linear within the element, an approximation of these derivatives is made. One such approximation that is consistent with the compatibility requirements of Aboudi is:

$$\phi_i = (u_i^{(1)} - u_i^{(2)})/d_1 \tag{3.5}$$

$$\chi_i = (u_i^{(3)} - u_i^{(4)})/d_2 \tag{3.6}$$

$$\psi_i = (u_i^{(5)} - u_i^{(6)})/d_3 \tag{3.7}$$

From the displacement interpolation, the element strains are as follows:

$$\epsilon_{11} = du_1/d\overline{x}_1 = \phi_1 \tag{3.8}$$

$$\epsilon_{22} = du_2/d\overline{x}_2 = \chi_2 \tag{3.9}$$

$$\epsilon_{33} = du_3/d\overline{x}_3 = \psi_3 \tag{3.10}$$

$$\gamma_{12} = du_1/d\overline{x}_2 + du_2/d\overline{x}_1 = \phi_2 + \chi_1 \tag{3.11}$$

$$\gamma_{13} = du_1/d\overline{x}_3 + du_3/d\overline{x}_1 = \phi_3 + \psi_1 \tag{3.12}$$

$$\gamma_{23} = du_2/d\overline{x}_3 + du_3/d\overline{x}_2 = \chi_3 + \psi_2 \tag{3.13}$$

Substituting in the derivative approximations, the strains become:

$$\epsilon_{11} = (u_1^{(1)} - u_1^{(2)})/d_1 \tag{3.14}$$

$$\epsilon_{22} = (u_2^{(3)} - u_2^{(4)})/d_2 \tag{3.15}$$

$$\epsilon_{33} = (u_3^{(5)} - u_3^{(6)})/d_3 \tag{3.16}$$

$$\gamma_{12} = (u_2^{(1)} - u_2^{(2)})/d_1 + (u_1^{(3)} - u_1^{(4)})/d_2 \tag{3.17}$$

$$\gamma_{13} = (u_3^{(1)} - u_3^{(2)})/d_1 + (u_1^{(5)} - u_1^{(6)})/d_3 \tag{3.18}$$

$$\gamma_{23} = (u_3^{(3)} - u_3^{(4)})/d_2 + (u_2^{(5)} - u_2^{(6)})/d_3 \tag{3.19}$$

Arranging in matrix form:

$$\{\epsilon\} = [B]\{U\} \tag{3.20}$$

The vectors in equation 3.20 are ordered in the following manner:

$$\{\epsilon\} = \{\epsilon_{11}\ \epsilon_{22}\ \epsilon_{33}\ \gamma_{12}\ \gamma_{13}\ \gamma_{23}\} \tag{3.21}$$

$$\{U\} = \{u_1^{(1)}\ u_2^{(1)}\ u_3^{(1)}\ u_1^{(2)}\ u_2^{(2)}\ u_3^{(2)} \ldots\ u_1^{(6)}\ u_2^{(6)}\ u_3^{(6)}\} \tag{3.22}$$

The $[B]$ matrix is then:

$$[B] = \begin{bmatrix}
\frac{1}{d_1} & 0 & 0 & \frac{-1}{d_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{d_2} & 0 & 0 & \frac{-1}{d_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{d_3} & 0 & 0 & \frac{-1}{d_3} \\
0 & \frac{1}{d_1} & 0 & 0 & \frac{-1}{d_1} & 0 & \frac{1}{d_2} & 0 & 0 & \frac{-1}{d_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{1}{d_1} & 0 & 0 & \frac{-1}{d_1} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{d_3} & 0 & 0 & \frac{-1}{d_3} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{d_2} & 0 & 0 & \frac{-1}{d_2} & 0 & \frac{1}{d_3} & 0 & 0 & \frac{-1}{d_3} & 0
\end{bmatrix}$$

Note that this $[B]$ matrix is constant for each element. Using a one point integration, the stiffness matrix, $[K_{se}]$, is then:

$$[K_{se}] = (d_1 d_2 d_3)[B]^T[D][B] \tag{3.23}$$

where $[D]$ is the constitutive matrix for the material of the element. Similarly, the force, or right-hand-side, vector is:

$$\{R_{se}\} = (d_1 d_2 d_3)[B]^T\{\sigma\} \tag{3.24}$$

It should be noted that because of the coefficients of thermal expansion are used to calculate the thermal strain within each element, the overall coefficients of thermal expansion are a priori contained in the resulting constitutive matrix for the composite.

3.2 Interface User Element

The interface element eventually used to model damage in the composite was implemented in the form of a three-dimensional spring to connect adjoining nodes between elements within each cell. The interface user element has a normal component and two tangential components, the tangential components representing shearing at the interface between the matrix and reinforcing material, as shown in Figure 3-2. The coordinate system adopted for the interface element is such that the one direction is assumed to always be the normal direction. As a result, the two and three directions thereby define the plane of shear, and the two entries in the stiffness matrix from these shearing contributions are equal by symmetry arguments. The $[K_{ie}]$ matrix is

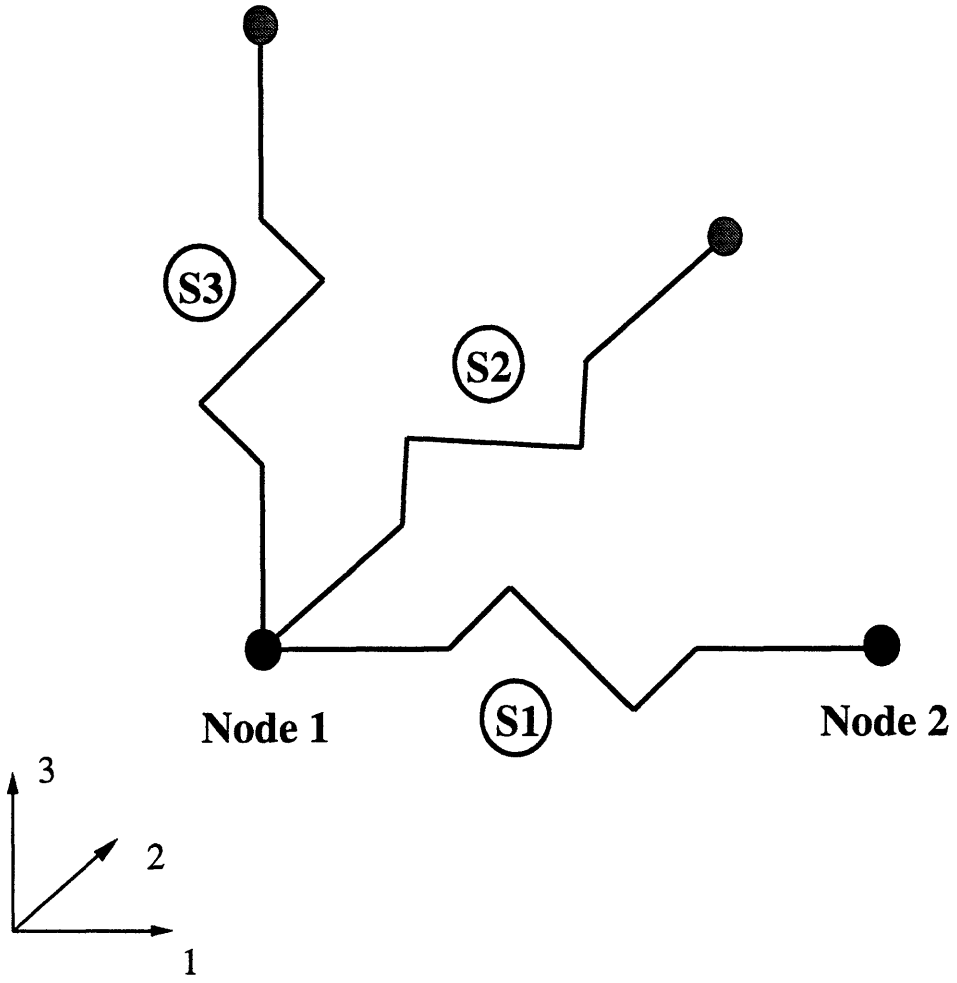


Figure 3-2: The Interface Element: A 3-Dimensional Spring
The S1 Spring Represents the Normal Component of the Interface while the S2 and S3 Springs are the Shear Components. The S2 and S3 Springs Connect Nodes 1 and 2, and Represent the Relative Displacement of the Nodes in the Shear Plane.

quite simple, and may be written directly as:

$$[K_{ie}] = \begin{bmatrix} k_1 & 0 & 0 & -k_1 & 0 & 0 \\ 0 & k_2 & 0 & 0 & -k_2 & 0 \\ 0 & 0 & k_3 & 0 & 0 & -k_3 \\ -k_1 & 0 & 0 & k_1 & 0 & 0 \\ 0 & -k_2 & 0 & 0 & k_2 & 0 \\ 0 & 0 & -k_3 & 0 & 0 & k_3 \end{bmatrix}.$$

with the vector of displacements arranged as follows:

$$\{U\} = \{u_1^{(\beta_1\gamma_1)}\ u_2^{(\beta_1\gamma_1)}\ u_3^{(\beta_1\gamma_1)}\ u_1^{(\beta_2\gamma_2)}\ u_2^{(\beta_2\gamma_2)}\ u_3^{(\beta_2\gamma_2)}\} \tag{3.25}$$

The superscripts $(\beta_1\gamma_1)$ and $(\beta_2\gamma_2)$ above denote the two subcell elements which are to be connected within the cell[3] by the interface user elements. The entries of the $[K_{ie}]$ matrix are properties of the interface itself and as such, are not well documented. To avoid numerical problems in the initial implementation, it was assumed that no debonding occurred during the analysis. This condition is relaxed in Chapter 5 to include a simple debonding criteria.

3.3 ABAQUS User Material Subroutine

As in many advanced nonlinear finite element programs, the user material subroutine option in ABAQUS allows for the development of material models which are not included in the standard ABAQUS library. The material model is coded in FORTRAN as a subroutine which is then included in the ABAQUS input deck when used in an analysis. The subroutine is called by ABAQUS at each material point in the mesh of the problem. At each point, the subroutine is provided with the temperature and the volume-averaged strains along with the material properties for the matrix and fiber. From this information, the material model calculates the stiffness matrix, $[C]$,

[3]Attempting to keep the notation somewhat consistent with that of Aboudi

and the volume-averaged stresses, $\{\overline{\sigma}\}$, and returns these to ABAQUS for use in the solution of the problem. A flow chart of the operation of the user material subroutine is presented in Figure 3-3.

In order to make the initial development of the user material easier, an orthotropic constitutive model was chosen for the matrix and fiber constituents. In addition, the interface elements were chosen to be much stiffer than the fiber or matrix so that the adjoining elements in the cell were kinematically constrained together.

3.3.1 Meshing the Representative Cell

When the user material routine is called by ABAQUS, a small submesh[4] of the representative volume cell is set up at each material point, as shown in Figure 3-4 for the continuous fiber case of four subcell elements connected by four interface elements. It should be noted that in the submesh used here, the fiber element is chosen to be the element corresponding to $(\beta\gamma) = (21)$ of the Aboudi framework shown in Figure 2-1(b). This change has no effect on the results provided by the model; hence, for all future discussions, the fiber will be assumed to be element $(\beta\gamma) = (11)$ in keeping with the notation of Aboudi. The remaining three subcells of the submesh have the properties of the matrix material. This submesh is the same at every point throughout the global mesh and thus it is hardcoded into the routine to reduce the computation time required at each material point.

To extend the model to handle short-fiber composites, another four cells would be added directly behind the four in the current mesh. All four of these added cells would then be matrix cells.

3.3.2 Substructuring and Solution

The individual stiffness matrices from the elements, $[K_{se}]$ and $[K_{ie}]$, are assembled into the global stiffness matrix of the representative cell, $[K]$. It is this global stiffness

[4]The term submesh is used here to differentiate between the mesh of the problem to be solved and the small subcell mesh of the representative cell used to derive the material properties for use in the solution of the mesh of the problem.

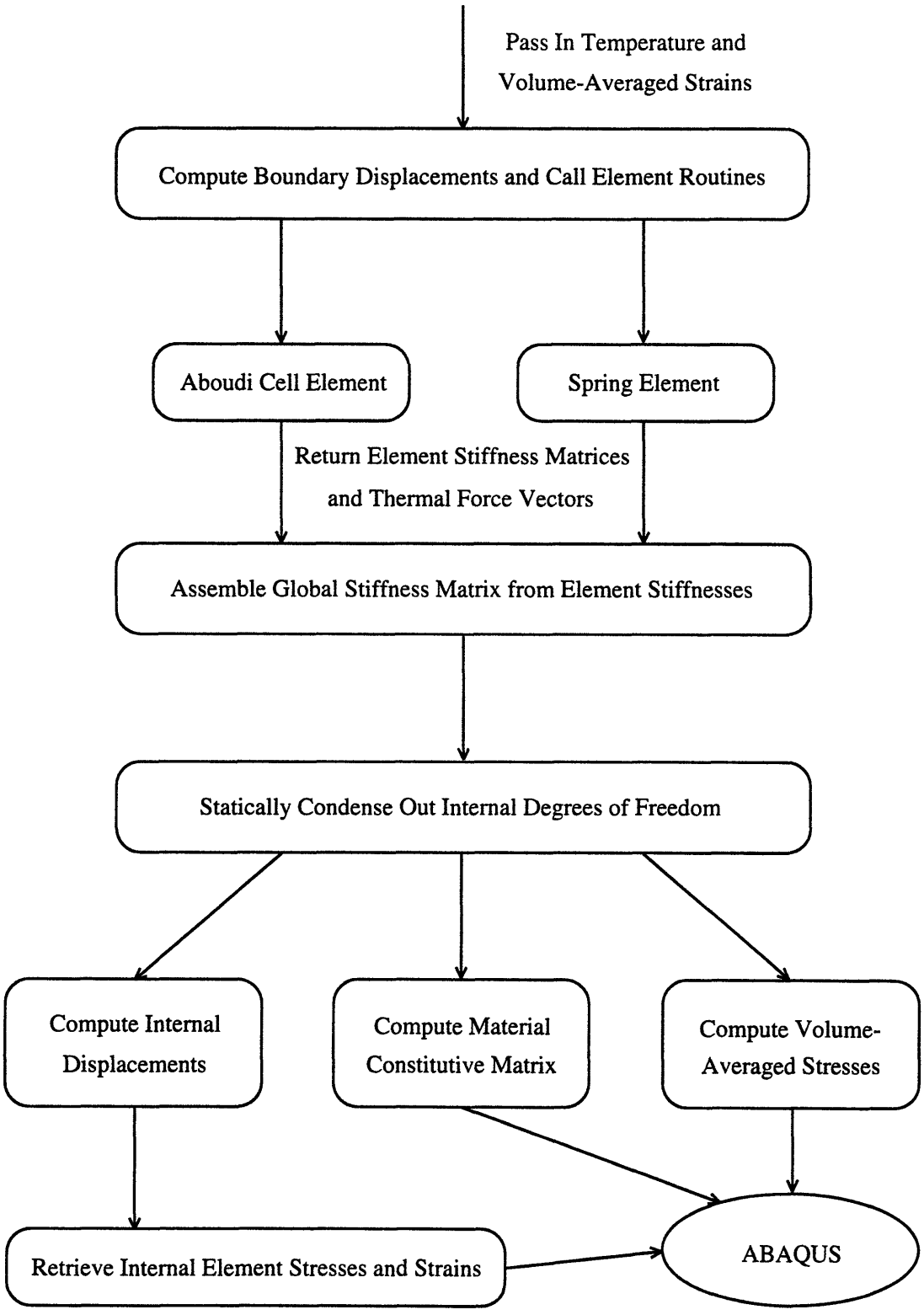


Figure 3-3: Flow Chart of the User Material Subroutine

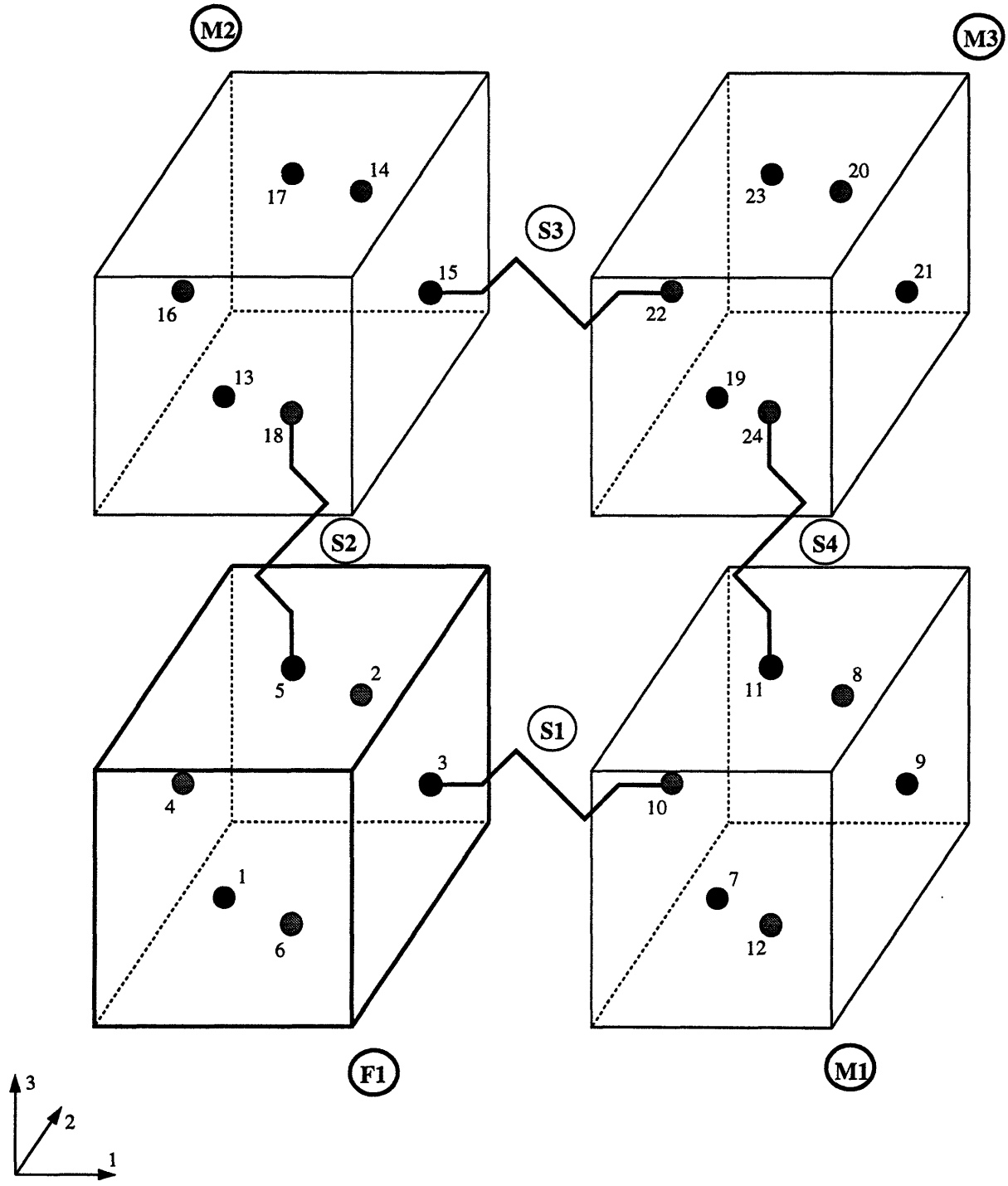


Figure 3-4: Mesh of the Representative Cell
F1 is the Fiber Element, M1–M3 are Matrix Elements,
and S1–S4 are the Interface Elements.

matrix that is used in the calculation of the material constitutive matrix. In preparing the global stiffness matrix, $[K]$, the displacements are ordered such that the internal degrees of freedom are separated from the degrees of freedom on the boundary of the representative cell. The techniques of substructuring, also known as static condensation, are then used to eliminate the internal degrees of freedom from the stiffness matrix for the cell [57]. By doing so, it is then possible to solve for the external force vector, R_b, since the boundary displacements are known from the strains provided by ABAQUS. Beginning by blocking the $[K]$ matrix of the equation $Ku = R$, this may be written in matrix form as:

$$\begin{bmatrix} K_{ii} & K_{ib} \\ k_{bi} & K_{bb} \end{bmatrix} \begin{Bmatrix} U_i \\ U_b \end{Bmatrix} = \begin{Bmatrix} R_i \\ R_b \end{Bmatrix} \tag{3.26}$$

The vector R_i has only contributions from the thermal strains which may be calculated from the information provided by ABAQUS. Consequently, the first row of 3.26 can be solved for U_i:

$$\{U_i\} = K_{ii}^{-1}(R_i - K_{ib}U_b) \tag{3.27}$$

Substituting into the second row of 3.26:

$$(K_{bb} - K_{bi}K_{ii}^{-1}K_{ib})\{U_b\} = R_b - K_{bi}K_{ii}^{-1}R_i \tag{3.28}$$

for which the only unknown is R_b. It must be remembered that the vector R_b can be attributed to both thermal and mechanical strains.

$$R_b = R_{b_{mech}} + R_{b_{ther}} \tag{3.29}$$

The thermal strains may be calculated using the temperature provided by ABAQUS so that the remaining unknown is simply $R_{b_{mech}}$.

The static condensation to eliminate the internal degrees of freedom is performed through Gaussian elimination on the representative cell stiffness matrix, $[K]$. The elimination is performed on the internal degrees of freedom, of which there are 24.

The blocked $[K]$ matrix then looks like:

$$\begin{bmatrix} \overline{K}_{ii} & \overline{K}_{ib} \\ 0 & \overline{K}_{bb} \end{bmatrix} \begin{Bmatrix} U_i \\ U_b \end{Bmatrix} = \begin{Bmatrix} R_i \\ R_b \end{Bmatrix} \tag{3.30}$$

where $\overline{K}_{ii}$ is now an upper triangular matrix. The previous K_{bi} becomes zero during the elimination. Also during the elimination, $\overline{K}_{bb}$ has become the left hand side of equation 3.28 so that:

$$\overline{K}_{bb} = K_{bb} - K_{bi}K_{ii}^{-1}K_{ib} \tag{3.31}$$

The right hand side of equation 3.28 contains the unknown R_b, but before the gaussian elimination is performed this vector contains only the thermal strains. Substituting in 3.29 and solving 3.28 for $R_{b_{mech}}$:

$$R_{b_{mech}} = (K_{bb} - K_{bi}K_{ii}^{-1}K_{ib})\{U_b\} + K_{bi}K_{ii}^{-1}R_i - R_{b_{ther}} \tag{3.32}$$

From this, equation 3.27 may be now be solved for the internal displacements and the solution is complete.

3.3.3 Conversion from Displacement to Strain

The solution outlined in the previous section yields a constitutive matrix for the representative cell in terms of force and displacement. ABAQUS works in terms of stress and strain. Thus, the force vector must be converted to a vector of volume-averaged stresses while the $[K]$ matrix must be converted from being based on displacements to strains. The stiffness matrix after the static condensation is a 48 x 48 matrix since there were 48 boundary displacements. This must be shrunk down to a 6 x 6 matrix. Using energy considerations it will be shown that the matrix which transforms the strains into displacements is simply the transpose of the matrix which transforms the forces into stresses. To begin, the matrix $[A]$ is defined to be the transformation

between the displacements and the strains:

$$\{U_b\} = [A]\{\bar{\epsilon}\} \tag{3.33}$$

where $\{\bar{\epsilon}\}$ is the vector of volume-averaged strains. Using the energy balance:

$$\{\bar{\sigma}\}^T\{\bar{\epsilon}\} = \{R_b\}^T\{U_b\} \tag{3.34}$$

where $\{\bar{\sigma}\}$ is the vector of volume-averaged stresses. Substituting equation 3.33 in 3.34 and canceling $\{\bar{\epsilon}\}$ from both sides completes the proof:

$$\{\bar{\sigma}\} = [A]^T\{R_b\} \tag{3.35}$$

The $[A]$ matrix itself is developed by writing the displacement interpolation as:

$$\begin{Bmatrix} u_1(x,y,z) \\ u_2(x,y,z) \\ u_3(x,y,z) \end{Bmatrix} = \begin{matrix} a_1x + b_1y + c_1z \\ a_2x + b_2y + c_2z \\ a_3x + b_3y + c_3z \end{matrix} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \tag{3.36}$$

By taking the derivatives of the displacements we are able to relate the coefficients to the strains:

$$\frac{\partial u_1}{\partial x} = a_1 = \epsilon_{11} \tag{3.37}$$

$$\frac{\partial u_2}{\partial y} = b_2 = \epsilon_{22} \tag{3.38}$$

$$\frac{\partial u_3}{\partial z} = c_3 = \epsilon_{33} \tag{3.39}$$

The shear strains are:

$$\epsilon_{12} = \frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} = b_1 + a_2 \tag{3.40}$$

$$\epsilon_{13} = \frac{\partial u_1}{\partial z} + \frac{\partial u_3}{\partial x} = c_1 + a_3 \tag{3.41}$$

$$\epsilon_{23} = \frac{\partial u_2}{\partial z} + \frac{\partial u_3}{\partial y} = c_2 + b_3 \tag{3.42}$$

The final conditions necessary to calculate the coefficients come from the rotations:

$$w_x = 0 = \frac{1}{2}\left(\frac{\partial u_3}{\partial y} - \frac{\partial u_2}{\partial z}\right) = \frac{1}{2}(b_3 - c_2) \tag{3.43}$$

$$w_y = 0 = \frac{1}{2}\left(\frac{\partial u_1}{\partial z} - \frac{\partial u_3}{\partial x}\right) = \frac{1}{2}(c_1 - a_3) \tag{3.44}$$

$$w_z = 0 = \frac{1}{2}\left(\frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y}\right) = \frac{1}{2}(a_2 - b_1) \tag{3.45}$$

Rewriting equation 3.36 using relations 3.37 through 3.45 and reorganizing, the strain-displacement relation is:

$$\begin{Bmatrix} u_1(x,y,z) \\ u_2(x,y,z) \\ u_3(x,y,z) \end{Bmatrix} = \begin{bmatrix} x & 0 & 0 & \frac{1}{2}y & \frac{1}{2}z & 0 \\ 0 & y & 0 & \frac{1}{2}x & 0 & \frac{1}{2}z \\ 0 & 0 & z & 0 & \frac{1}{2}x & \frac{1}{2}y \end{bmatrix} \begin{Bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{23} \end{Bmatrix} \tag{3.46}$$

This relation provides the connection between the three displacements at each node of the representative cell and the strains at that node. The complete $[A]$ matrix used to transform the global stiffness matrix into the material constitutive matrix is derived by plugging in the coordinates of each node[5] on the boundary. The constitutive matrix is then:

$$[C] = [A]^T[K][A] \tag{3.47}$$

and the volume-averaged stresses are given by equation 3.35.

3.3.4 Postprocessing Operations

The user material subroutine is actually called twice by ABAQUS during the analysis of the problem. The routine is first called during the assembly of the global stiffness

[5]Of which there are 18. The coordinate system used is a local one defined for the whole representative volume cell, centered at the point where the four subcells come together.

matrix. The strains passed in by ABAQUS at this time are zero as the stiffness matrix is independent of the strain-state (for linear statics.) The second call of the material user routine occurs after ABAQUS has completed the solution of the problem. For this call the strains from the solution are passed into the subroutine so that the actual element stresses and strains may be calculated for use in post-processing.

The user material subroutine option in ABAQUS also allows the user to save his own variables for use in post-processing[6]. The individual subcell elements stresses and strains were output using this option so that behavior inside the cell could be studied.

Similarly, several variables were included to look at possible failure inside of the representative cell. Some simple uniaxial failure criteria, one each for the fiber and matrix, were adopted from Aboudi [11, 13]. The fiber criterion used was

$$\frac{|\sigma_{11}|}{S_f} \leq 1 \tag{3.48}$$

where σ_{11} is the axial stress in the fiber and S_f is either the tensile strength or the compressive strength of the fiber depending on the state of stress present in the fiber. This expression will be less than one if the fiber is not in failure. Similarly, the matrix failure criterion was

$$\frac{\sigma_{pr}^2}{X_m^2} + \frac{\sigma_{12}^2 + \sigma_{13}^2}{S_m^2} \leq 1 \tag{3.49}$$

where σ_{pr} is the maximum principal stress in the plane perpendicular to the axis of the fiber, X_m is the ultimate matrix tensile strength, S_m is the ultimate matrix shear strength, and σ_{12} and σ_{13} are the axial shear stresses. In addition, the von Mises equivalent stress was also calculated for each subcell element.

A complete listing of the user material subroutine can be found in Appendix A.

[6]These variables are called STATEV in ABAQUS.

3.4 Note on the Specifics of Implementation

The development of the user routines for ABAQUS was performed on a VAXstation 4000-60, manufactured by Digital Equipment Corporation (DEC), working in the VMS operating system. It is important to note that the input decks, and consequently all subroutines used in the input decks, to ABAQUS are required to be expressed in capital letters. In addition, a double precision floating point variable on that platform is declared as a REAL*8. The default single precision floating point variable for a Cray Y-MP 8/64, however, is the equivalent of a double precision variable on the VAX. Thus, functions such as SQRT and DSQRT[7] must be used carefully when the routines are transferred for use on platforms other than the original one, with the double precision form used on the VAX and the single precision form used on the Cray for example.

[7]DSQRT is the double precision form of the square root function.

Chapter 4

Testing the Finite Element Model

Once the finite elements for use in the representative cell had been written and the material routine completed, it was then necessary to verify that the results provided were in agreement with both the original method of cells model and results obtained from elasticity solutions. This is obviously of great importance as a model which does not produce good results is useless, but in addition, it was very important to gain an estimate of the cost of using a micromechanical analysis of this type inside a finite element program. To that end, timing studies were performed to gauge the increase in the computation time when the micromechanical model is introduced versus macroscopic methods of composite analysis.

4.1 Verification of the Subcell User Element

The subcell user element described in Section 3.1 represents only a single subcell of the representative volume cell. The user material routine uses four of these subcell elements to construct a representative volume cell for each material point within the mesh of the finite element problem. But before the subcell user element could be included into the user material routine, it was first necessary to verify that the element performed correctly by itself under different loading situations. Towards this end, analysis was performed using first a single element and then a mesh of four elements, effectively setting up a representative cell using ABAQUS. At this point it

is important to make clear the distinction between the use of a four element method of cells model to derive the effective material properties of a composite and the use of the method of cells as a user material to derive the effective material properties of the composite. The use of only four elements in a finite element analysis will inevitably lead to incorrect results in all but the simplest of cases. It is only through the use of a very large number of elements that the finite element approximation approaches the true solution of the problem. On the other hand, in using the method of cells at each material integration point of a larger mesh (for example, at each integration point of the mesh shown in Figure 4-2), a unit cell is analyzed at each integration point. The process of homogenizing to derive the effective overall composite properties, as described in Chapter 2, is then accomplished by the finite element program itself. This homogenization process, in which the discrete rectangular array composite model of the method of cells is transformed into a continuous medium, is the step that is omitted by using a four element mesh in ABAQUS to derive the overall composite properties.

4.1.1 Single Element Case

A mesh consisting of a single subcell user element[1] was set up using the isotropic properties of the fiber material listed in Table 4.1. The model was tested by imposing unit strain states and checking for the correct stresses and internal displacements. The strain states were imposed by prescribing all of the boundary displacements on the element. Six unit strain states were examined, one for each of the normal and pure shear states. For these strain states the resulting stress vector then simply contains the appropriate components of the material stiffness matrix. The element was found to perform correctly for all the situations tested.

[1]This is the user element described in detail previously in Section 3.1. To use it, the elements must be defined from the nodes with the *USER ELEMENT option in ABAQUS.

Table 4.1: Properties Used in Isotropic Test Run

Volume Fraction = 0.5	Fiber	Matrix
Young's Modulus (GPa)	1000	100
Shear Modulus (GPa)	416.67	38.46
Poisson's Ratio	0.2	0.3
Coeff. of Ther. Expansion (cm/cm-$^\circ C$)	5.22e-5	7.0e-5

4.1.2 Multi-Element Case

The next step in verifying the user element was to create a mesh of four subcell user elements. These four elements were arranged as shown in Figure 3-4 to insure that the elements would perform correctly when combined in the setup of the representative cell. The elements were kinematically constrained together to prevent separation during the analysis[2]. The same six strain states as used in the single element case were then imposed on the mesh; once again this was performed by prescribing the boundary displacements. It should be noted that the prescription of the boundary nodes in this case is identical to the way boundary conditions are applied inside the user material routine.

Unlike the single element case, the prescription of the boundary displacements in the four element case does not eliminate all of the zero-energy modes from the four subcell element mesh. An eigenvalue analysis was performed to study the mode shapes and eigenvalues of the mesh. It was found that this combination of four subcell elements had a "gear-shape" zero-energy mode, as shown in Figure 4-1, in addition to the usual rigid-body modes. This mode is a result of the fact that the subcell element is not really a complete element. In order to eliminate the mode, it would be necessary to use a higher order displacement interpolation than the linear one used in the method of cells. It was found that this mode can be canceled by constraining one of the interior nodes in the tangential direction. This corresponds to constraining the vertical displacement at point 1 in Figure 4-1 to be zero. It was necessary that this

[2]This is performed by defining multi-point constraints, the *MPC option in ABAQUS, at adjacent nodes. The nodes are then bound together to move as one.

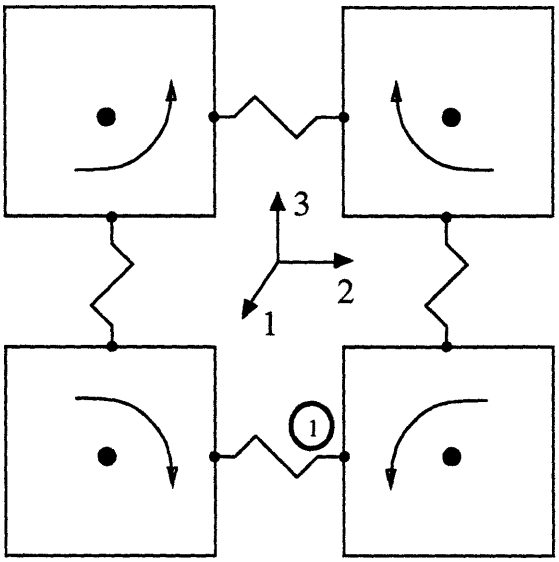


Figure 4-1: Zero Energy Mode for the Four Subcell Element Mesh
Eliminated by setting $u_3 = 0$ at Point 1

constraint also be implemented into the method of cells user material routine. When the interface elements were used to connect the internal nodes of the subcell elements instead of rigidly constraining them to move together, this tangential constraint had to be applied to each element within the representative cell.

The problems were later re-run with four interface elements introduced to connect the cells. The springs of the interface element were made very stiff to mimic the constraint imposed when the interface elements were not used. The results obtained for this case were found to be identical between the cases.

4.1.3 Results for Multi-Element Case

Once the gear-shape zero-energy mode had been eliminated, the multi-element mesh was tested against results obtained from a direct solution of the method of cells equations. This solution was obtained from a program written previously at Los Alamos National Laboratory by R. M. Hackett [34]. As in the method of cells, the element corresponding to $(\beta\gamma) = (11)$ in Figure 2-1(b) was chosen to be the fiber element while the remaining three elements were composed of the matrix material.

The mesh was first tested using fiber and matrix constituents which were isotropic. The properties used for this case are given in Table 4.1. The effective stiffness matrix was then derived by applying the six simple strain states discussed above. For example, the c_{11} component was obtained by imposing a unit strain in the one direction and then averaging the stresses in the four elements over the volume. The c_{22} component was likewise obtained by imposing a unit strain in the two direction, and so forth. The effective coefficients of thermal expansion were also calculated. These values were obtained by applying a 1 $°C$ temperature increase to the unloaded mesh. By using the constitutive relation,

$$\{\overline{\sigma}\} = [C]\{\overline{\epsilon}_{tot} - \overline{\epsilon}_{ther}\}, \tag{4.1}$$

the effective coefficients of thermal expansion can be calculated from the stresses, $\{\overline{\sigma}\}$ (where the overline denotes volume-averaged values), obtained in the analysis. Since $\overline{\epsilon}_{tot} = 0$ and $\overline{\epsilon}_{ther} = \overline{\alpha}\Delta T$, it follows that the solution is

$$\{\overline{\alpha}\} = \frac{-1}{\Delta T}[C]^{-1}\{\overline{\sigma}\} \tag{4.2}$$

The results of the analysis are summarized in Table 4.2. As can be seen, excellent agreement is achieved between the method of cells solution and that derived from using the subcell user element mesh.

Another comparison was made using transversely isotropic constituents for the fiber and matrix. The method followed to back out the components of the constitutive matrix and the coefficients of thermal expansion was exactly the same as described above. Table 4.3 gives the properties used in this analysis and the results of the analysis are summarized in Table 4.4. As can be seen, the agreement is once again very good.

Table 4.2: Comparison of the Method of Cells and User Element Solutions for a Composite with Isotropic Constituents

	Method of Cells[a]			Subcell User Element[a]		
Effective Stiffness Matrix	594.38	90.30	90.30	594.27	90.31	90.31
	90.30	286.14	83.45	90.30	286.13	83.45
	90.30	83.45	286.14	90.30	83.45	286.13
Shear Modulus—12 Dir.	87.20			87.21		
Shear Modulus—13 Dir.	87.20			87.21		
Shear Modulus—23 Dir.	70.42			70.42		
Coeff. of Ther. Exp.—11 Dir.	5.39e-5			5.4e-5		
Coeff. of Ther. Exp.—22 Dir.	6.20e-5			6.2e-5		
Coeff. of Ther. Exp.—33 Dir.	6.20e-5			6.2e-5		

[a] All stiffness moduli are given in terms of GPa; the coefficients of thermal expansion are given in terms of (cm/cm-$^\circ C$)

Table 4.3: Properties Used in Transversely Isotropic Test Run

Volume Fraction = 0.5	Fiber	Matrix
Young's Modulus—11 Dir. (GPa)	1000	100
Young's Modulus—22 & 33 Dirs. (GPa)	500	100
Shear Modulus—12 & 13 Dirs. (GPa)	300	38.46
Shear Modulus—23 Dir. (GPa)	200	38.46
Poisson's Ratio—12 & 13 Dirs.	0.2	0.3
Poisson's Ratio—23 Dir.	0.25	0.3
Coeff. of Ther. Expansion—11 Dir. (cm/cm-$^\circ C$)	1.0e-7	7.0e-5
Coeff. of Ther. Expansion—22 & 33 Dirs. (cm/cm-$^\circ C$)	1.0e-5	7.0e-5

Table 4.4: Comparison of the Method of Cells and User Element Solutions for a Composite with Transversely Isotropic Constituents

	Method of Cells[a]			Subcell User Element[a]		
Effective Stiffness Matrix	588.67	78.32	78.32	588.52	78.32	78.32
	78.32	241.58	77.69	78.33	241.56	77.68
	78.32	77.69	241.58	78.33	77.68	241.56
Shear Modulus—12 Dir.	82.17			82.19		
Shear Modulus—13 Dir.	82.17			82.19		
Shear Modulus—23 Dir.	64.52			64.51		
Coeff. of Ther. Exp.—11 Dir.	6.82e-6			7.0e-6		
Coeff. of Ther. Exp.—22 Dir.	4.50e-5			4.5e-5		
Coeff. of Ther. Exp.—33 Dir.	4.50e-5			4.5e-5		

[a]All stiffness moduli are given in terms of GPa; the coefficients of thermal expansion are given in terms of (cm/cm-$^\circ C$)

4.2 Verification of the User Material

Once the subcell user element had been tested out completely, the element was used to develop the user material routine as described in Chapter 3. In order to test the entire user material, the results of a simple one element analysis were compared to the results obtained from the method of cells in the same manner as described for the multi-element mesh of subcell elements. The results from this analysis showed that the user material routine was indeed providing the same results as the method of cells.

Once this had been established, the user material was ready to be tested on realistic finite element analysis problems. Several problems were set up in ABAQUS. The tests were performed by running the problems first with the standard orthotropic material model provided by ABAQUS and using the effective material properties obtained from the method of cells solution. The same problems were then run once again, this time using the user material routine. These tests, in addition to checking the accuracy of the results, provided information about the increase in computation time required when the micromechanical model is introduced. They also highlight some of the advantages of using a micromechanical material model. For these analyses

Table 4.5: Properties of the Fiber Material, AS

	AS
Longitudinal Modulus (GPa)	213.74
Transverse Modulus (GPa)	13.79
Longitudinal Shear Modulus (GPa)	13.79
Transverse Shear Modulus (GPa)	6.89
Longitudinal Poisson's Ratio	0.20
Transverse Poisson's Ratio	0.25
Long. Coeff. of Ther. Expansion (cm/cm-$°C$)	-0.99e-6
Trans. Coeff. of Ther. Expansion (cm/cm-$°C$)	10.08e-6
Longitudinal Tensile Strength (GPa)	2.07
Longitudinal Compressive Strength (GPa)	1.79

AS, a carbon fiber by Hercules, and LM, a low modulus epoxy, were used as the constituent materials for a fiber volume fraction of 0.6. The use of these properties was intended to be for numerical comparison. The properties are given in Tables 4.5 and 4.6 as found in [45].

4.2.1 Plate Model With Thermal Loading

The first problem chosen for study was a simple plate problem. The plate was meshed with 256 C3D20[3] elements, four elements through the thickness of the plate and eight elements along the width and length as shown in Figure 4-2. Two different composite layups were created. The first was set up as a unidirectional composite, with the fibers running in the one direction for all element layers. The second layup represented a crossply laminate; the fibers in the top two element layers ran in the one direction while the fibers in the bottom two element layers ran in the two direction. Pin and roller boundary conditions were imposed on the bottom of the plates at three of the four corner nodes, as shown in Figure 4-3.

The plates were then subjected to two types of thermal loads: an uniform temperature increase and a temperature gradient through the thickness. In the uniform

[3]The C3D20 element is a 3-dimensional, 20-noded full integration quadratic element.

Table 4.6: Properties of the Matrix Material, LM

	LM
Modulus (GPa)	2.21
Shear Modulus (GPa)	0.77
Poisson's Ratio	0.43
Coeff. of Ther. Expansion (cm/cm-$°C$)	102.6e-6
Tensile Strength[a] (MPa)	55.16
Compressive Strength[a] (MPa)	103.4
Shear Strength[a] (MPa)	55.16
Tensile Fracture Strain (%)	8.1
Compressive Fracture Strain (%)	15
Shear Fracture Strain (%)	10

[a]It is not completely clear how these properties were obtained as the testing was not described in [45].

loading, the temperature throughout the two laminates was increased by $100\,°C$ over the reference temperature[4]. In applying the thermal gradient to the unidirectional laminate, the temperature was increased linearly through the thickness of the plate so that the top of the plate was $100\,°C$ hotter than the bottom. The temperature at the bottom of the plate was kept at the reference temperature. When the thermal gradient was applied to the crossply laminate, the midplane of the composite rather than the bottom was maintained at the reference temperature. The temperature in the composite was then increased linearly to be $50\,°C$ hotter at both the top and bottom of the plate.

The results for the problem of the unidirectional composite under a constant temperature increase are presented in Table 4.7. As can be seen from the data, while the global stresses are zero, the stresses inside the individual fiber and matrix cells are nonzero. This is as expected since the coefficients of thermal expansion are different between the fiber and matrix. The micromechanical model is able to capture this fact, whereas the model based on effective global properties shows only that the global stress state is zero. The composite has simply expanded uniformly

[4]The reference temperature refers to the temperature at which there are no thermal stresses.

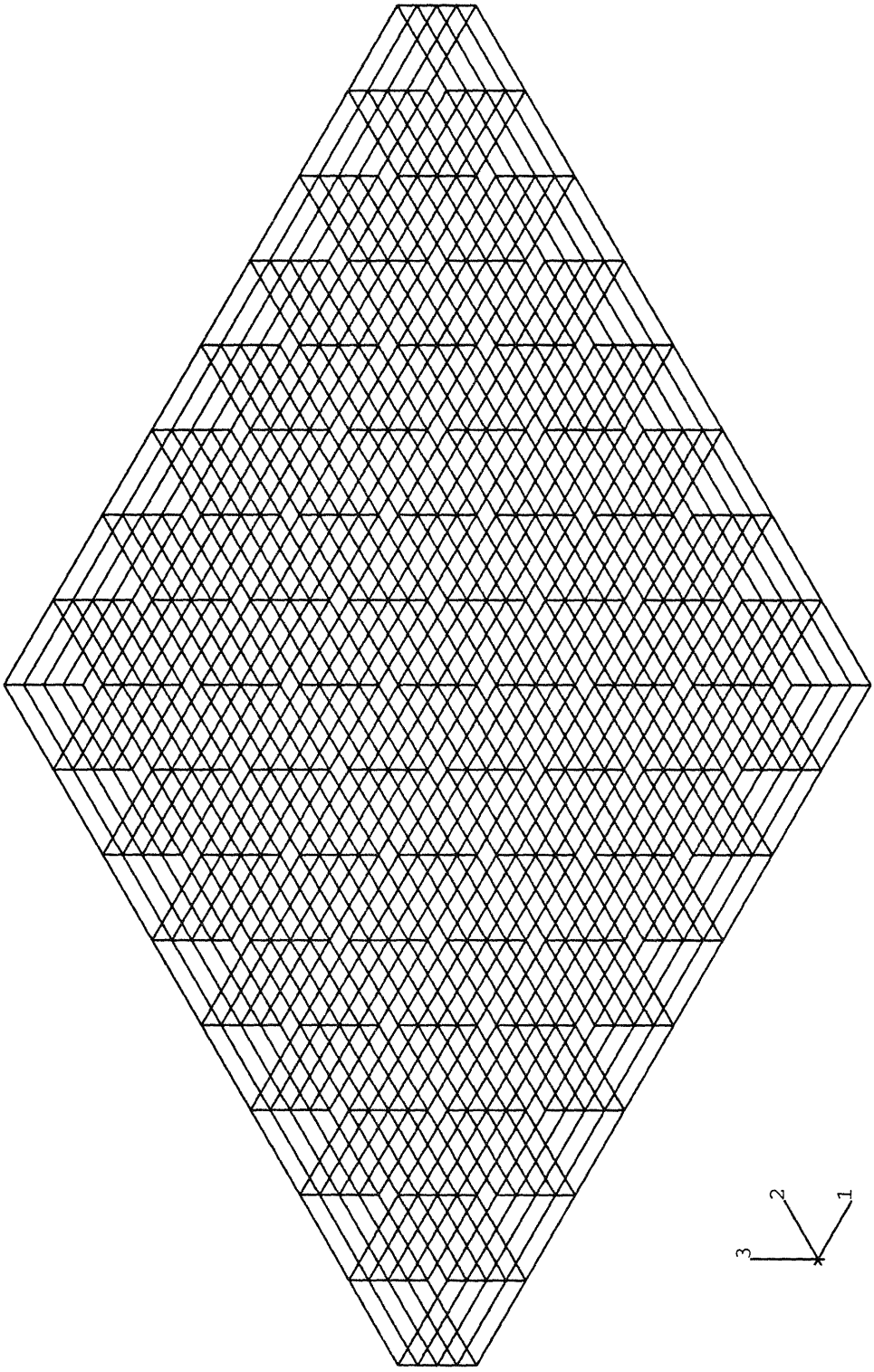


Figure 4-2: Mesh of Plate Problem

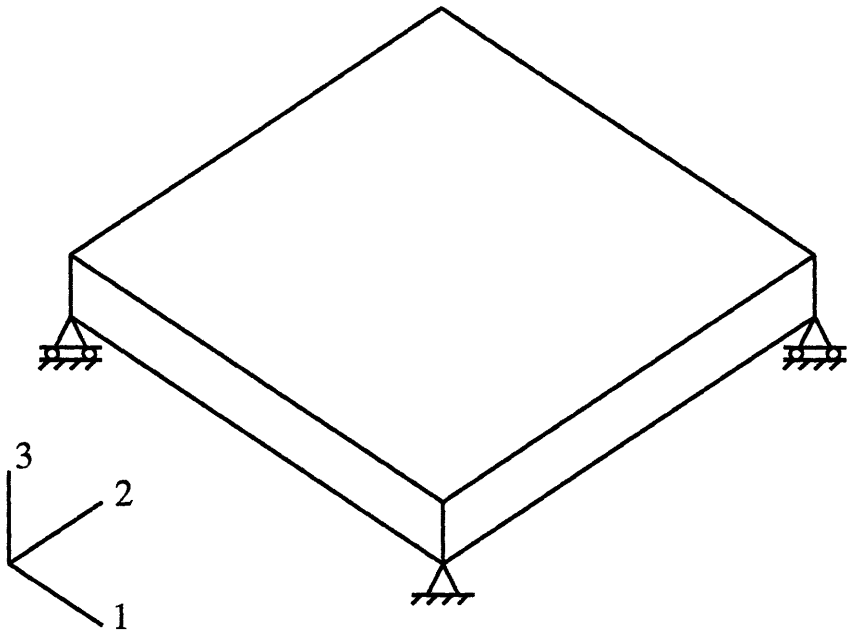


Figure 4-3: Boundary Conditions for Thermal Loading of Plate Problems

in the two and three directions while contracting in the one direction (since the fiber has a negative coefficient of thermal expansion in the axial direction.) From looking at the failure criteria, it can be seen that the matrix is up to 30% of its failure value even though the global stresses are zero. This indicates very strongly the potential for failure of the matrix when thermal strains are present in addition to other forms of loading, yet this effect would not be captured in a macroscopic analysis. This also illustrates the importance of accounting for the localized stresses introduced by the difference in the coefficients of thermal expansion for the matrix and fiber.

The temperature increase was then applied to the crossply composite. The deformed mesh is shown in Figure 4-4. The curvature in the mesh results from the fact that the plys expand in different directions due to the lack of symmetry about the mid-surface. The top ply expands the most in the two direction while the bottom ply expands most in the one direction. This behavior results from the fact that expansion in the axial direction is limited by the negative axial coefficient of thermal expansion in the fiber. A plot of the global normal stresses in the one direction shows that the global stresses, like the constituent stresses, are also no longer zero (See Figure 4-5).

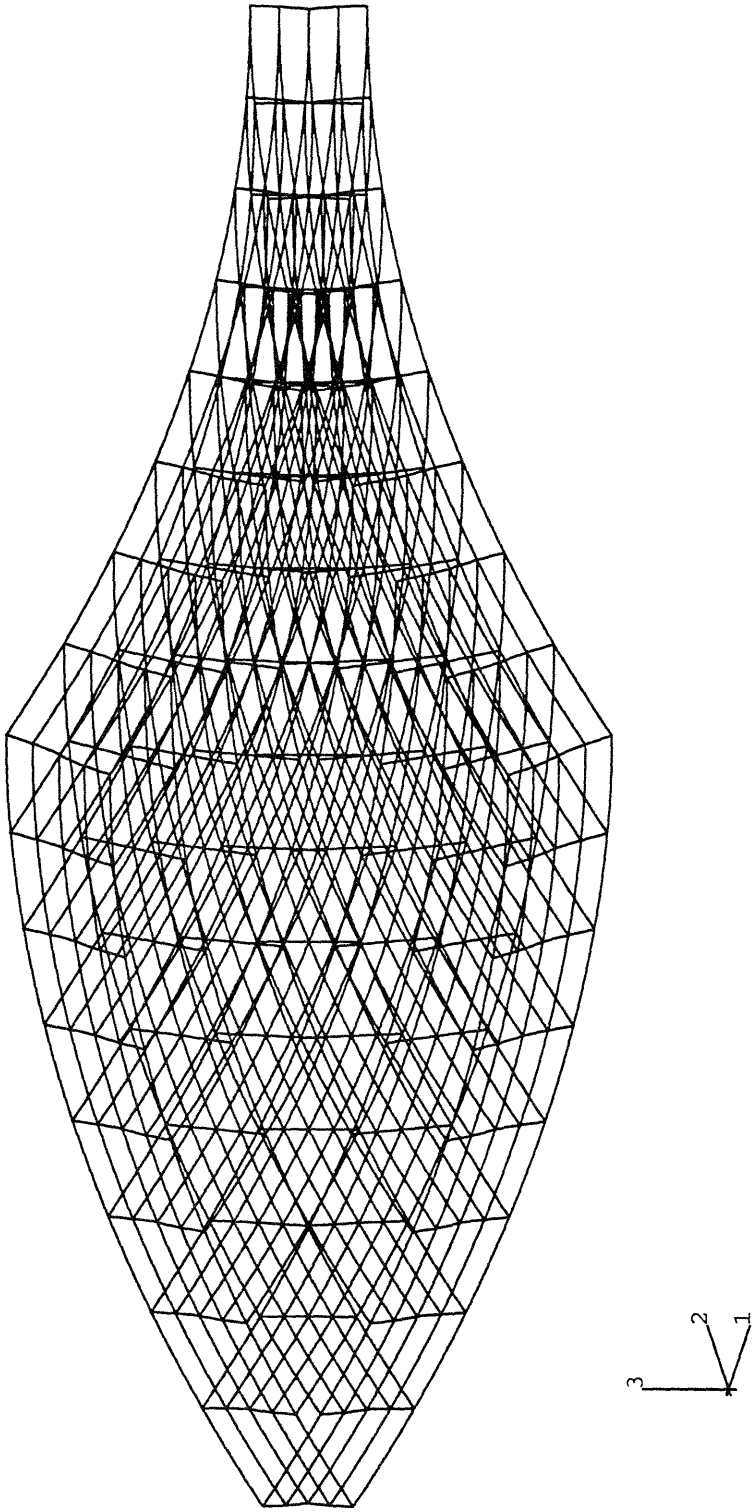


Figure 4-4: Deformed Mesh for Crossply Laminate Under Uniform Temperature Increase

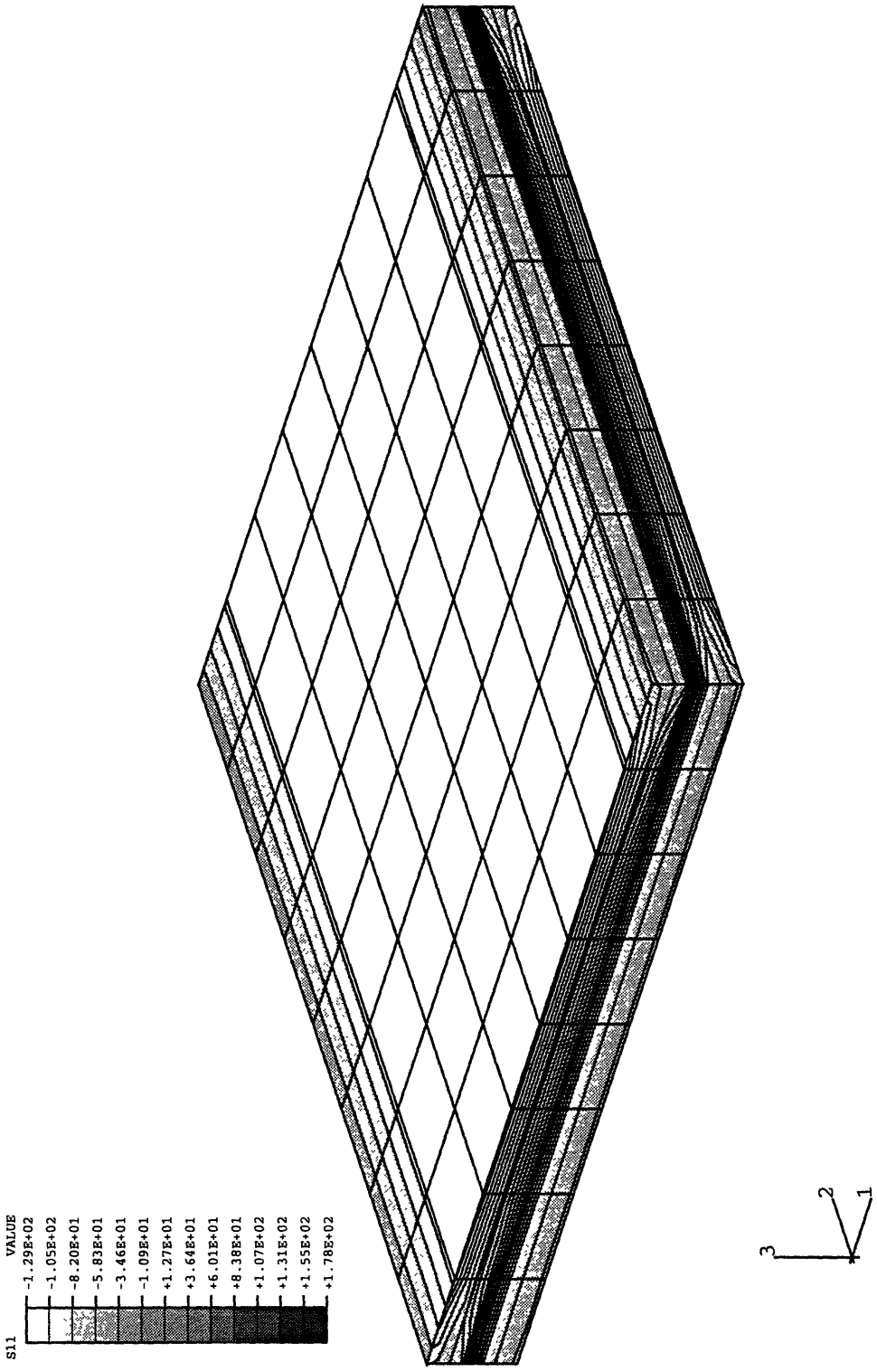


Figure 4-5: Global Normal Stresses in One Direction for Crossply Laminate Under Uniform Temperature Increase

Table 4.7: Results for Unidirectional Composite Under Constant Temperature Change

	σ_{11}	σ_{22}	σ_{33}	σ_{12}	σ_{13}	σ_{23}
Maximum Overall Stresses (MPa)	0	0	0	0	0	0
Fiber Stresses (MPa)	20.81	6.59	6.59	0	0	0
Maximum Matrix Stresses (MPa)	-42.19	-22.66	-22.66	0	0	0
Failure Criteria:[a]	Fiber			Matrix		
	0.01			0.32		

[a]Calculated for the fiber using the uniaxial criterion of equation 3.48. For the matrix the calculation was performed by dividing the maximum von Mises stress by the compressive strength (103.4 MPa). A value ≥ 1 indicates failure.

Figure 4-6 shows the deformed mesh for the unidirectional laminate when subjected to the linear thermal gradient through the thickness. The curvature results, as would be expected, since the material expands more at the top of the composite where it is hotter than at the bottom. While the global stresses for this case are zero, Figure 4-7 shows that the constituent stresses are not. As can be seen, the axial fiber stress increases linearly through the thickness of the composite. This same problem was run once again using a negative thermal gradient. The only change this introduced was found in the failure criteria. The values, while still small, increased for the negative gradient since the composite is weaker in compression.

The thermal gradient was then applied in the manner described above to the crossply laminate. The deformed mesh, shown in Figure 4-8, is similar to that for the uniform temperature problem of Figure 4-4. By comparing the global normal stresses in the one direction, Figures 4-5 and 4-9, it is found that the stress is much higher in the uniform temperature case. This is correct since the temperature change for the gradient problem is really half of that for the uniform increase problem.

4.2.2 Plate Model Under Bearing Pressure

The unidirectional and crossply laminates problems were also run subjecting the plate to a uniform pressure of magnitude 100 MPa applied to the top of the plate. This pressure was chosen to be on the order of the compressive strength of the matrix

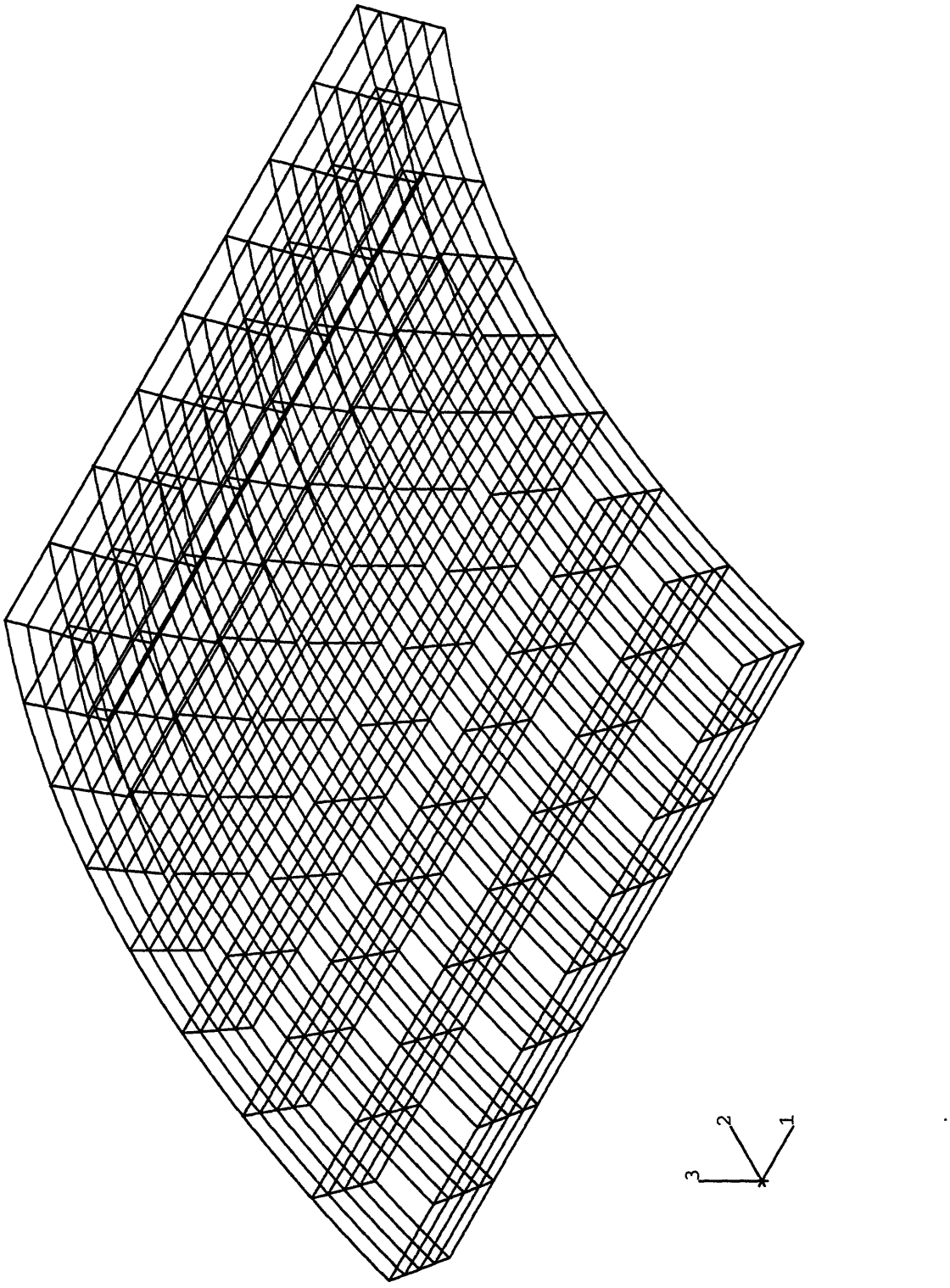


Figure 4-6: Deformed Mesh for Unidirectional Composite
Under Linear Thermal Gradient

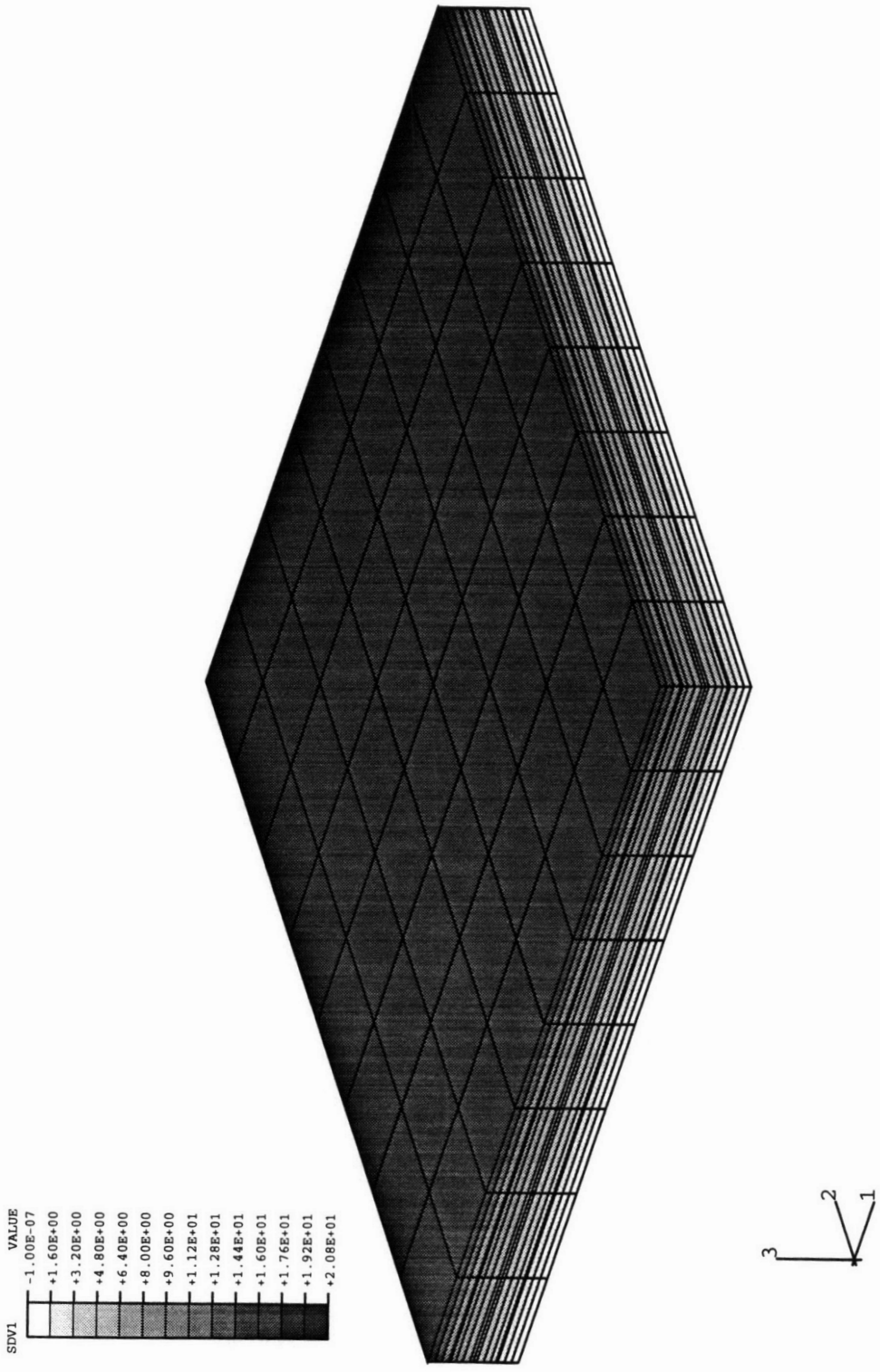


Figure 4-7: Fiber Stresses in Unidirectional Composite
Under Linear Thermal Gradient

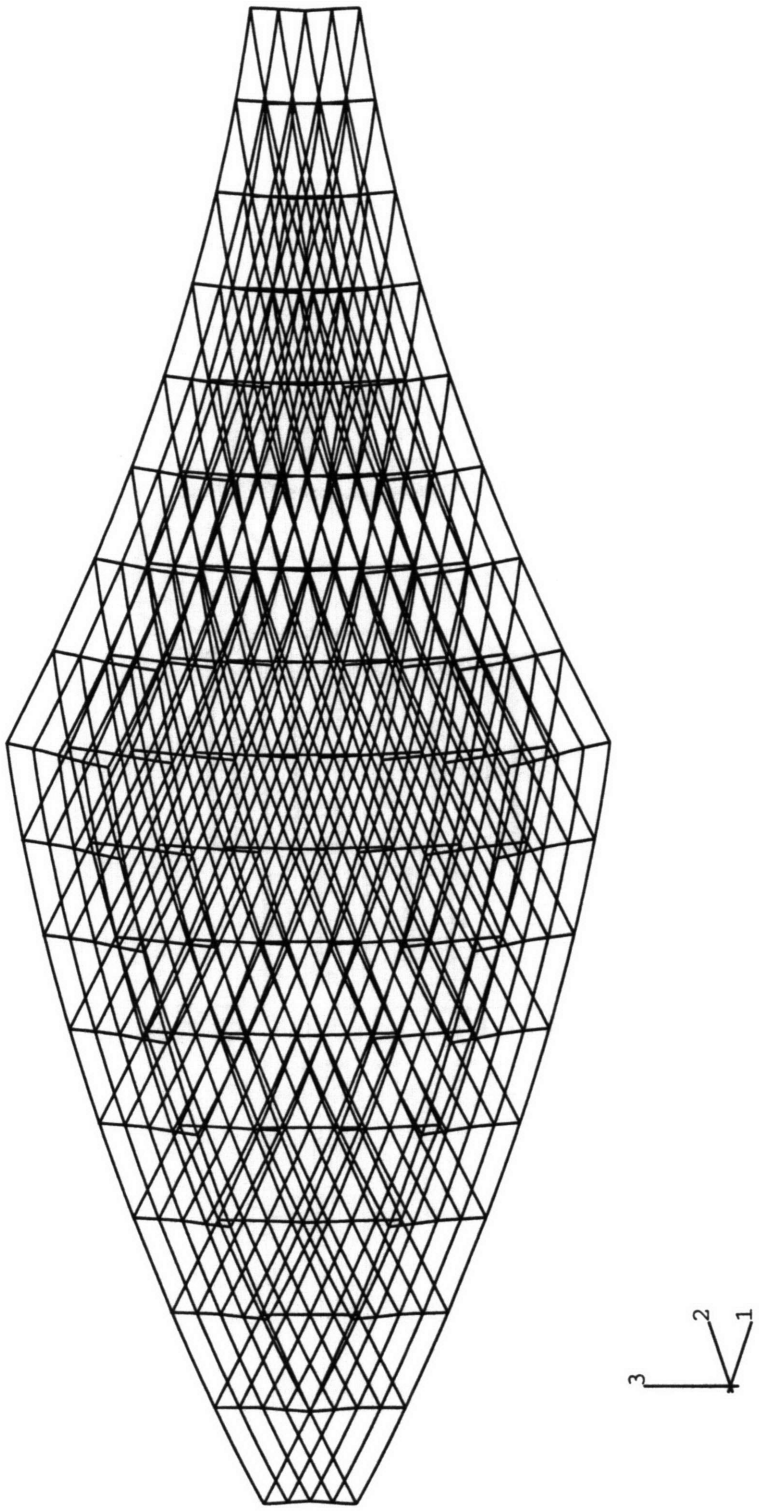


Figure 4-8: Deformed Mesh for Crossply Laminate Under Thermal Gradient

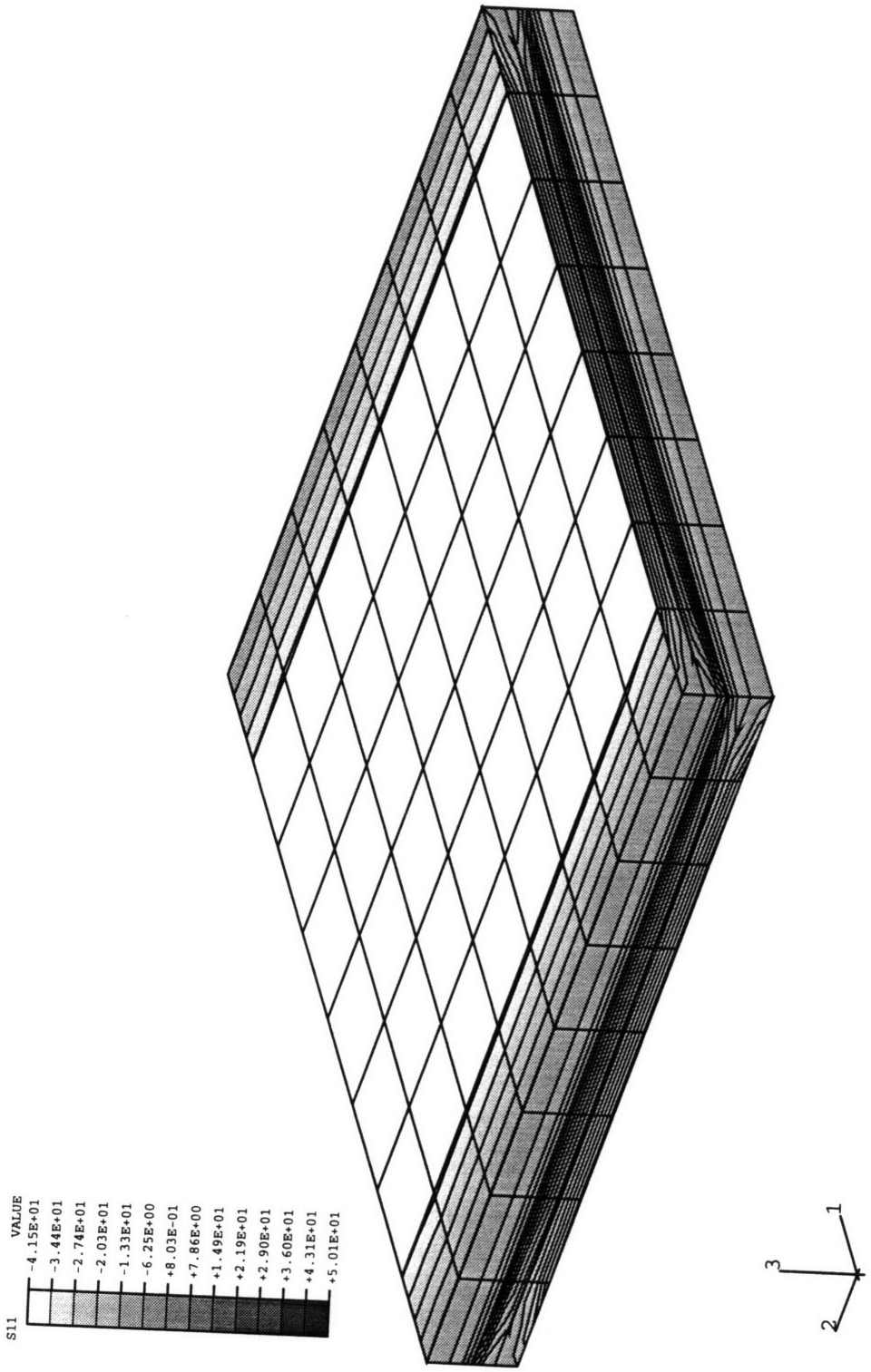


Figure 4-9: Normal Stresses in One Direction for Crossply Laminate Under Thermal Gradient

Table 4.8: Results for Unidirectional Fiber Laminate under Bearing Pressure

	σ_{11}	σ_{22}	σ_{33}	σ_{12}	σ_{13}	σ_{23}
Max. Overall Stresses (MPa)	0	0	-100.0	0	0	0
Fiber Stresses (MPa)	27.56	16.97	-118.8	0	0	0
Max. Matrix Stresses (MPa)	-75.66	-58.33	-118.8	0	0	0
Failure Criteria:[a]	Fiber			Matrix		
	0.013			1.32		
von Mises Stress in Matrix Element ($\beta\gamma$)[b] (MPa)	Element (12)		Element (21)		Element (22)	
	45.43		53.92		21.04	

[a] Calculated using equations 3.48 and 3.49. A value ≥ 1 indicates failure.
[b] This is compared to the compressive strength of the matrix, 103.4 MPa from Table 4.6.

in order to show some of the failure prediction capabilities of the micromechanics material model. The boundary conditions in this case were changed so that all nodes on the bottom face of the plate were constrained in the three direction. In addition, the pin condition and one of the roller conditions on the corner nodes shown in Figure 4-3 were kept to prevent rigid body motion. The results for the unidirectional laminate are presented in Table 4.8. The table shows that the macroscopic stress state is one of uniaxial stress in the three direction. However, from looking at the individual matrix and fiber subcell stresses, it is found that the local stress state is not simply uniaxial loading as a result of the interactions between the fiber and matrix.

It was found in testing performed at Los Alamos National Laboratory that a bearing pressure loading situation such as that applied in this problem does not produce failure within the matrix, and in fact, that the matrix was still well below failure [44]. Inconsistent with this finding, the values for the failure criteria given in Table 4.8 show the matrix to be failing. It should be remembered though that these failure criteria compare the maximum compressive stress to the uniaxial compressive strength. In actuality, the subcell matrix and fiber stresses are shown in Table 4.8 to be multiaxial. If instead the von Mises equivalent stress in the matrix is compared to the compressive failure strength, it appears that the matrix is in fact not undergoing failure. It must be recalled that it is unknown what methods were used to obtain the value for the compressive failure strength cited here. It is unclear whether this

is a yield strength value or if failure occurred before yielding; hence, the von Mises assumption that failure occurs at the onset of yielding may or may not be valid. The discrepancy between the two methods of failure prediction displays some of the shortcomings of the simple micromechanics failure criteria used in the model and the importance of choosing the appropriate criteria. The Aboudi failure criteria are intended for use in uniaxial loading situations and hence do not perform well in situations where the stress state is not simply uniaxial. The von Mises criterion, on the other hand, does take into account the presence of a multiaxial stress state, but it is a yielding criterion and is not applicable to all situations. The analysis here also points to the need for experimental data to compare with the micromechanical failure criteria. Finally, there is a need for more thorough reporting of material properties and the methods used to obtain them.

The results for the crossply laminate mesh when subjected to the same bearing pressure are shown in Table 4.9. Now, it is not only just the local stress state, but also the macroscopic stress state for each ply which is no longer uniaxial. The matrix failure criteria once again indicates failure, but as before the von Mises indicates that there is no failure.

4.2.3 Quasi-Isotropic Pressure Vessel

The final problem used to test the user material routine was a quasi-isotropic pressure vessel under an internal pressure of magnitude 100 MPa. This problem was chosen more as a demonstration illustrating how the model would be used to create a situation of quasi-isotropy than to actually show that the model was working correctly.

The mesh of the spherical pressure vessel was created by taking advantage of axisymmetry. Due to this symmetry, it was possible to model only a small wedge of the pressure vessel. In doing so, the size of the problem, and consequently the expense, was greatly reduced. The resulting mesh is shown in Figure 4-10. Two types of elements were used in the analysis. The elements at the ends of the wedge

Table 4.9: Results for Crossply Laminate under Bearing Pressure

	$\sigma_{11}{}^a$	$\sigma_{22}{}^a$	$\sigma_{33}{}^a$	$\sigma_{12}{}^a$	$\sigma_{13}{}^a$	$\sigma_{23}{}^a$
Max. Macro. Stresses, 0° Ply	105.2	1.10	-92.07	0.56	15.71	20.36
Min. Macro. Stresses, 0° Ply	-10.68	-42.80	-114.1	-0.56	-15.71	-20.36
Max. Macro. Stresses, 90° Ply	85.75	2.02	-72.48	2.11	13.05	23.83
Min. Macro. Stresses, 90° Ply	-8.01	-52.42	-126.7	-2.11	-13.05	-23.83
Max. Fiber Stresses, 0° Ply	215.1	18.14	-103.9	0.67	18.81	20.36
Min. Fiber Stresses, 0° Ply	10.20	-33.70	-128.8	-0.67	-18.81	-20.36
Max. Fiber Stresses, 90° Ply	178.1	14.77	-85.97	2.53	15.62	23.83
Min. Fiber Stresses, 90° Ply	17.93	-41.62	-145.6	-2.53	-15.62	-23.83
Max. Matrix Stresses, 0° Ply	-70.60	-57.44	-103.9	0.18	18.81	20.36
Min. Matrix Stresses, 0° Ply[b]	-87.86	-80.29	-128.8	-0.18	-18.81	-20.36
Max. Matrix Stresses, 90° Ply[b]	-54.83	-41.82	-85.97	0.68	15.62	23.83
Min. Matrix Stresses, 90° Ply[b]	-98.68	-89.63	-145.6	-0.68	-15.62	-23.83

Failure Criteria:[b]	Fiber	Matrix
0° Ply	0.10	1.74
90° Ply	0.086	2.00

von Mises Stress in Matrix Element $(\beta\gamma)^c$ (MPa)	Element (12)	Element (21)	Element (22)
0° Ply	47.28	64.28	40.57
90° Ply	51.59	64.98	45.52

[a] All stress values are in MPa.
[b] Calculated using equations 3.48 and 3.49. A value ≥ 1 indicates failure.
[c] This is compared to the compressive strength of the matrix, 103.4 MPa from Table 4.6.

have only 15 nodes, and consequently, C3D15[5] elements were used. The elements in the remainder of the wedge were constructed using C3D20 elements.

A quasi-isotropic composite is one in which the mechanical behavior is nearly isotropic despite the anisotropic nature of the composite constituents. This situation may be created in a composite by winding the fibers using three orientations, 0° and ±60° from the 0° layer. The fibers are wound around a core made to the desired shape of the end product, oftentimes a cylinder or sphere. Only a single layer of fibers is wound at any orientation before the fiber orientation is changed. For example, a single layer of 0° fibers would be wound, then the orientation would be changed to +60° and another single layer of fibers would be added. The orientation is then changed to −60° and the final layer is wound. The orientation is then returned to 0° and the whole process is repeated again. The matrix material, which is basically an epoxy, is brushed on in the form of a liquid between each fiber layer and hardens to hold the fibers in place. The oriented fiber layers themselves are extremely thin. Thus, the three differently-oriented layers are occupying nearly the same space within the composite, and they act in conjuction to provide behavior that is nearly independent of the way the composite is oriented.

This condition of quasi-isotropy can be created in a finite element program by layering three sets of elements on top of each other. For the problem at hand, the elements shown in Figure 4-10 were used along with two more sets of identical elements defined on the same nodes. Each layer was then given a different fiber orientation. In the first layer, the fibers ran circumferentially around the sphere. The remaining two layers were then set up with fibers running at ±60° from the circumferential layer. In order to orient the fibers in the correct direction, it was necessary to set up a local coordinate system at each integration point. The *ORIENTATION and the user subroutine ORIENT were used in ABAQUS to set up these local coordinate systems. Since the three layers were superimposed on top of each other, the fiber and matrix properties had to be appropriately reduced. Consequently, the moduli for each layer were equal to the original moduli divided by three.

[5]These elements are 15-noded, full integration quadratic elements

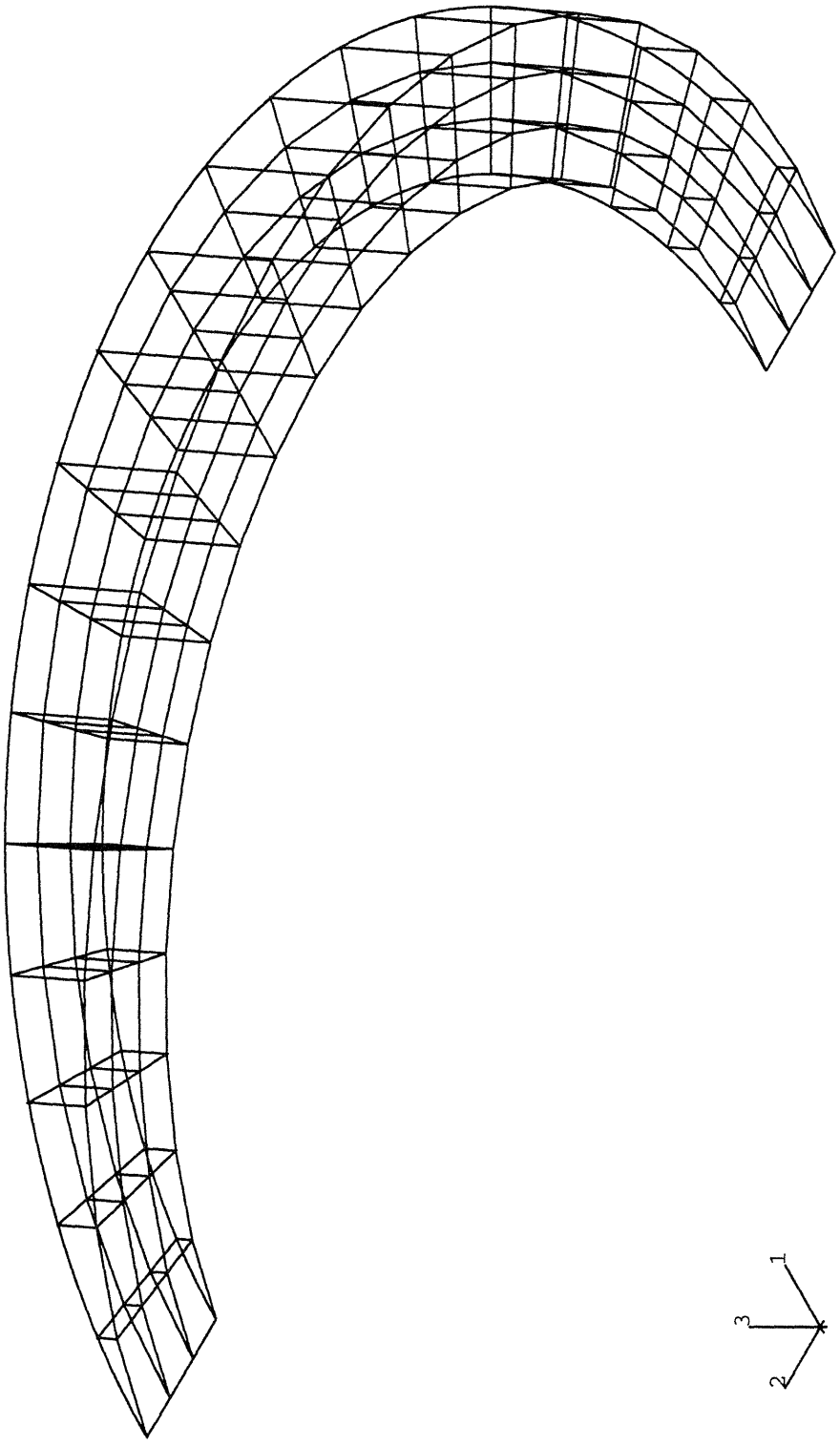


Figure 4-10: Mesh of the Spherical Pressure Vessel Wedge

Table 4.10: Finite Element Solution for Stresses in Quasi-Isotropic Pressure Vessel

	$\sigma_{11}{}^a$	$\sigma_{22}{}^a$	$\sigma_{33}{}^a$	$\sigma_{12}{}^a$	$\sigma_{13}{}^a$	$\sigma_{23}{}^a$
Stresses at Inner Surface						
0° Fibers	162.3	-31.53	-2.08	0.05	0.0001	-0.03
+60° Fibers	161.2	-31.53	-2.03	0.02	0.02	-0.04
−60° Fibers	161.2	-31.53	-2.03	-0.07	-0.02	0.01
Total[b]	239.8	-94.59	238.7	0.0	0.0001	-0.06
Stresses at Outer Surface						
0° Fibers	90.74	-0.309	6.64	-0.074	-0.003	0.084
+60° Fibers	90.76	-0.309	6.63	-0.069	-0.004	0.086
−60° Fibers	90.90	-0.310	6.63	0.143	0.0004	-0.002
Total[b]	146.1	-0.928	146.2	-0.0001	-0.063	0.168

[a] All stress values are in MPa.
[b] This is the sum of the three after transforming the ±60° stresses to align with the 0° stresses. After the transformation, σ_{22} represents the radial stress while σ_{11} and σ_{33} are the stresses in the circumferential and meridinal directions, respectively.

The results of the analysis are presented in Table 4.10. The macroscopic stresses are given for each orientation as well as the total macroscopic stress in the sphere. The values for the stresses in each orientation are given in the local coordinate systems; therefore, in order to calculate the total stresses in the sphere, the stresses in the ±60° orientations had to be transformed to align with the 0° fiber direction [26]. After this transformation was performed, the total stresses were then calculated by summing over the three layers.

The stresses of the quasi-isotropic solution may be compared with the complete elastic solution for an isotropic thick-walled sphere [51], plotted as a function of the radius in Figure 4-11. The boundary stresses from the elasticity solution at the inner and outer surfaces of the sphere are given in Table 4.11. The elasticity solution was performed using spherical coordinates; the total stresses in the finite element solution are also reported in spherical coordinates, but in a slightly different order due to the use of the orientation options in ABAQUS. The radial stress, σ_{rr} in Table 4.11, corresponds to σ_{22} in the finite element solution of Table 4.10. Similarly, the circumferential and meridinal stresses, $\sigma_{\theta\theta}$ and $\sigma_{\phi\phi}$, correspond to σ_{11} and σ_{33}, respectively, in the finite element solution.

Table 4.11: Elasticity Solution for Stresses in a Thick-Walled Sphere

	$\sigma_{rr}{}^a$	$\sigma_{\theta\theta}{}^a$	$\sigma_{\phi\phi}{}^a$	$\sigma_{r\theta}{}^a$	$\sigma_{r\phi}{}^a$	$\sigma_{\theta\phi}{}^a$
Inner Surface	-100.0	207.4	207.4	0	0	0
Outer Surface	0	157.4	157.4	0	0	0

[a] All stress values are in MPa.

Figure 4-11a shows that the magnitude of the radial stress is maximum at the inner surface of the sphere and decays through the thickness to zero at the outer surface. The maximum, -100 MPa from Table 4.11, is simply the negative of the internal pressure. In comparison, the finite element solution produces a value of -94.59 MPa at the inner surface, as found in Table 4.10, an error of only 5.4%. At the outer surface, the finite element solution provides a value of -0.928 as compared to zero for the elasticity solution. A small part of the error found here may be accounted for by remembering that the finite element solution is computed at points inside the elements and not actually at the boundaries of the sphere. In addition, another portion of the error is probably due to the fact that only three elements were used through the thickness of the sphere. As the number of elements through the thickness is increased, the solution should come closer to the true values obtained in the elasticity solution. The largest part of the error however is probably due to the fact that a quasi-isotropic state is created in the finite element problem in the local 1-3 plane. The behavior in the quasi-isotropic sphere is therefore closer to that of a transversely isotropic composite, with the anisotropy in the radial direction. Hence the comparison to the isotropic elasticity solution for the behavior through the thickness of the sphere is not completely correct. Yet despite this fact, the finite element solution still provides a decent approximation of isotropic behavior for the radial stresses in the sphere.

From symmetry arguments, it is obvious that the stresses in the circumferential and meridinal directions are equal. They will subsequently be referred to simply as the hoop stress. The hoop stress, like the radial stress, is also maximum at the inner surface and minimum at the outer surface as seen in Figure 4-11b. From Table 4.11,

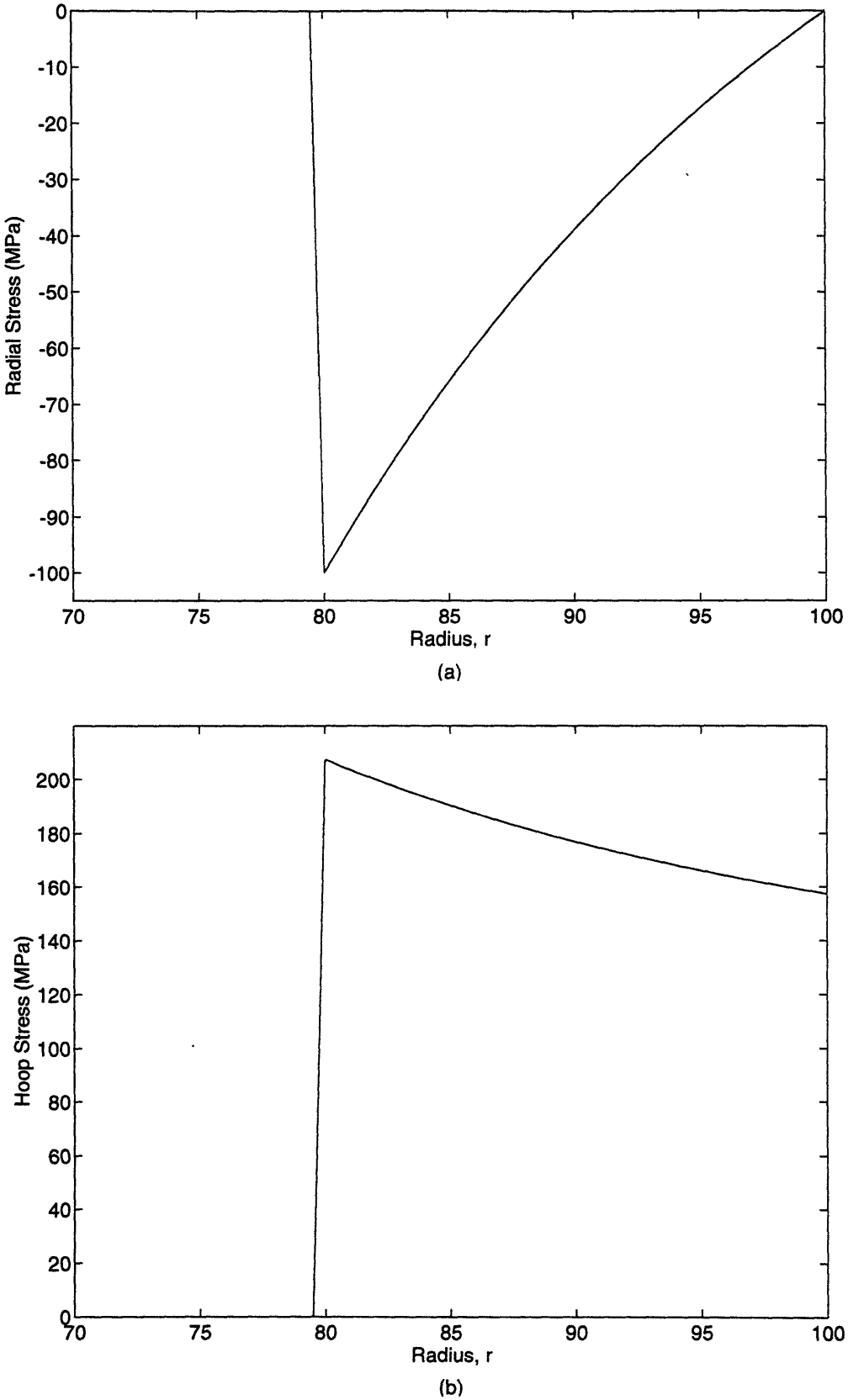


Figure 4-11: Elasticity Solution for Thick Walled Sphere
(a) Radial Stress (b) Stress in the Circumferential and Meridinal Directions

the boundary values for the elasticity solution are found to be 207.4 MPa and 157.4 MPa. When looking at the finite element solution for the hoop stress, it is seen that the values for σ_{11} and σ_{33} are not exactly equal, but very close to it. The difference at the inner surface is 1.1 MPa, off by only 0.5%. The agreement is even better at the outer surface where the stresses differ by less than 0.1%. The hoop stress values for the finite element solution do not however compare as well to the elasticity solution as did the radial stresses. The error in the finite element solution is -7.2% at the outer surface and increases to 15.6% at the inner surface. It is thus seen that while the method of cells material model did succeed in providing essentially isotropic behavior in the 1-3 plane, the fact that the model is essentially transversely isotropic rather than being completely isotropic prevents excellent agreement between the finite element solution and the elasticity solution. Once again, the accuracy of the finite element solution should increase as the number of elements through the thickness is increased.

4.3 Statistical Representation of Geometry

Another advantage of the method of cells over other micromechanical composite models is that statistical variation of both the constituent properties and the composite geometry can be introduced very easily. In this section, the unidirectional composite plate model of Section 4.2.1 is altered by allowing the fiber spacing to vary throughout the plate. This is done by using a different volume fraction from integration point to integration point in the material property calculation. In order to maintain the continuous fiber model, the variation of the volume fraction was allowed only in the two and three directions of the plate (See Figure 4-2).

Since the volume fraction was allowed to change between integration points, it was necessary to use a state variable rather than the normal property definition under the *USER MATERIAL card in ABAQUS in order to input the volume fraction into the material subroutine. The value of the state variable was set for each integration point before the analysis was begun using the user subroutine SDVINI. A random number

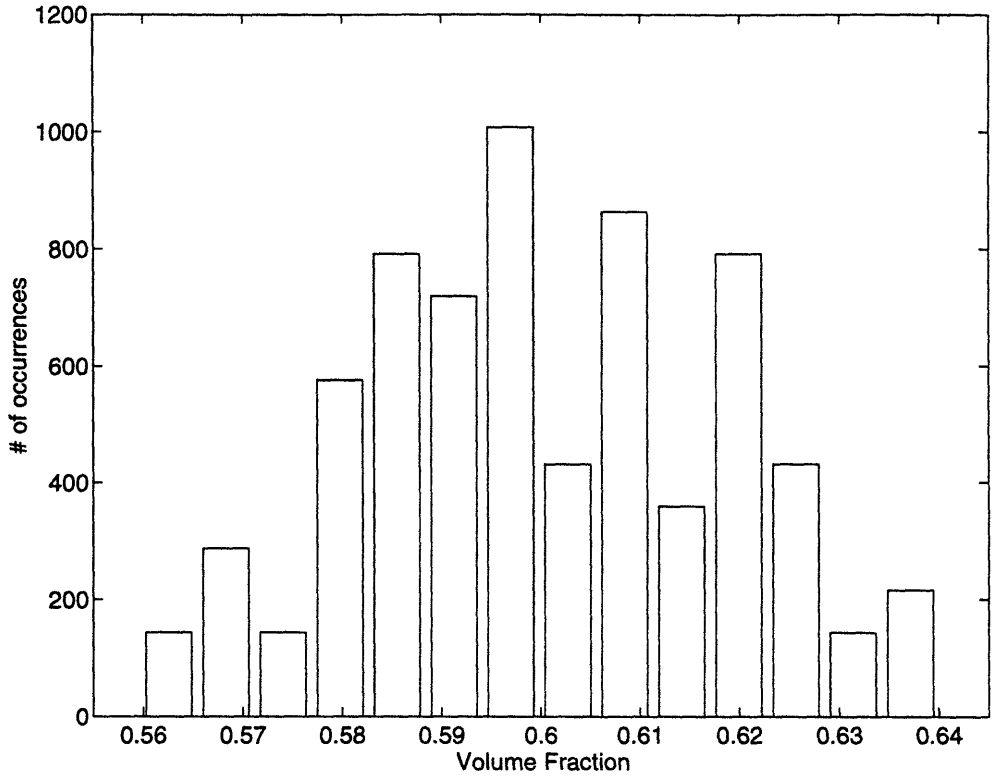


Figure 4-12: Distribution of Volume Fractions used in Material Property Calculation

generator was included in this subroutine in order to generate the volume fraction values for use in the material subroutine. The numbers provided by the random number generator ranged between zero and one. The distribution was normalized and the range was altered so that the actual volume fraction used in the material calculation was allowed to range between 0.525 and 0.675. These limits were selected based on a standard deviation of 0.025 for the volume fraction reported by Engelstad and Teply in [29]. A histogram of the resulting statistical distribution for the volume fraction is shown in Figure 4-12. The actual spatial distribution of the volume fraction throughout the plate is given in Figure 4-13. The fibers in the plate run in the one direction, and as can be seen, the volume fraction changes in the two and three directions but not in the one direction. The blacked out areas of the plot represent areas where the volume fraction is changing so much that the contour lines overlap. Figures 4-14 and 4-15 show views of the plate from the sides. In Figure 4-14 the view is along the 2-direction axis, and the variation of the volume fraction in the 1-3 plane can be seen more clearly. Similarly, Figure 4-15 views the plate along the 1-direction

Table 4.12: Stresses in Unidirectional Plate with Varying Volume Fraction

	σ_{11}	σ_{22}	σ_{33}	σ_{12}	σ_{13}	σ_{23}
Maximum Overall Stresses (MPa)	5.97	7.07	8.53	0.08	0.40	0.73
Minimum Overall Stresses (MPa)	-5.92	-7.42	-6.73	-0.08	-0.40	-0.82
Maximum Fiber Stresses (MPa)	23.83	13.26	14.92	0.10	0.49	0.73
Maximum Matrix Stresses (MPa)	-46.96	-28.11	-28.26	-0.03	-0.14	-0.82
Failure Criteria:[a]	Fiber			Matrix		
	0.012			0.34		

[a]Calculated for the fiber using the uniaxial criterion of equation 3.48. For the matrix the calculation was performed by dividing the maximum von Mises stress by the compressive strength (103.4 MPa). A value ≥ 1 indicates failure.

axis so that the variation in the 2-3 plane can be plainly seen.

As in Section 4.2.1, the analysis was conducted by subjecting the plate to a uniform temperature increase of $100\,^{\circ}C$ over the reference temperature. The resulting deformed mesh is shown in Figure 4-16 with the displacements magnified 300 times. It can be seen that the changing volume fraction introduces a small amount of waviness in the 2-3 plane of the deformed mesh. This can be seen better in the side view of the deformed mesh shown in Figure 4-17 where the displacements are further amplified to 1000 times their real value. In contrast to the results of the analysis performed with constant volume fraction (summarized in Table 4.7), the analysis here shows that the macroscopic stresses are no longer zero (See Table 4.12). Contour plots of the normal stresses in the two and three directions are shown in Figures 4-18 and 4-19. From comparing these figures to the plot of the volume fraction variation in Figure 4-13, it is seen that the pattern of stress distribution is very similar to that for the volume fraction, as would be expected. It is also worthy to note the fact that the non-uniform volume fraction has led to the development of shear stresses, the largest of which is σ_{23}. Figure 4-20 shows that the distribution of the σ_{23} shear stress is also similar to that of the input volume fraction.

It is also shown in Table 4.12 that the individual matrix and fiber stresses are higher than in the uniform volume fraction case. In general the stresses increased by 10 to 25% for the varying volume fraction case. However, the transverse normal

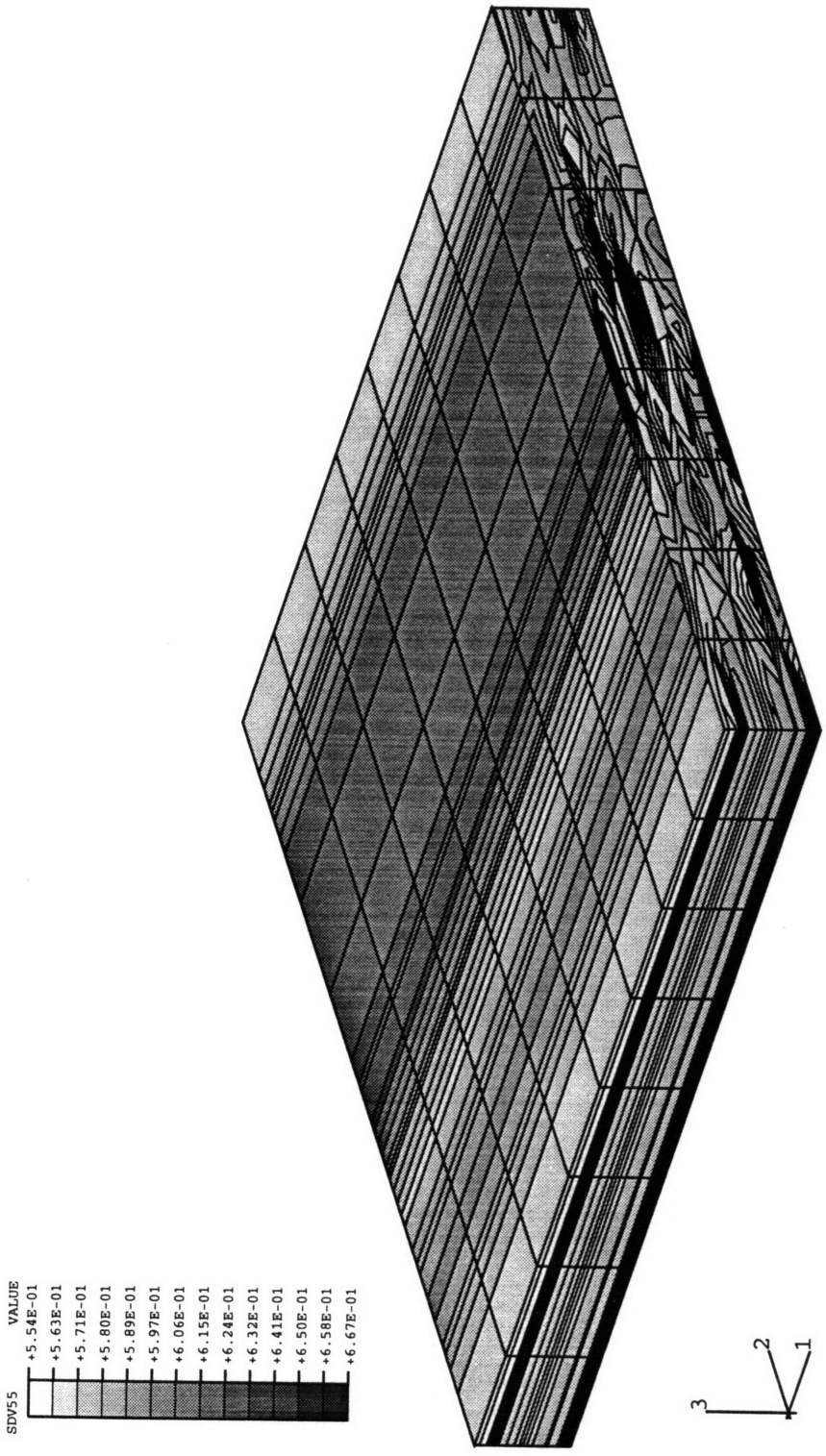


Figure 4-13: Spatial Distribution of the Volume Fraction throughout the Plate

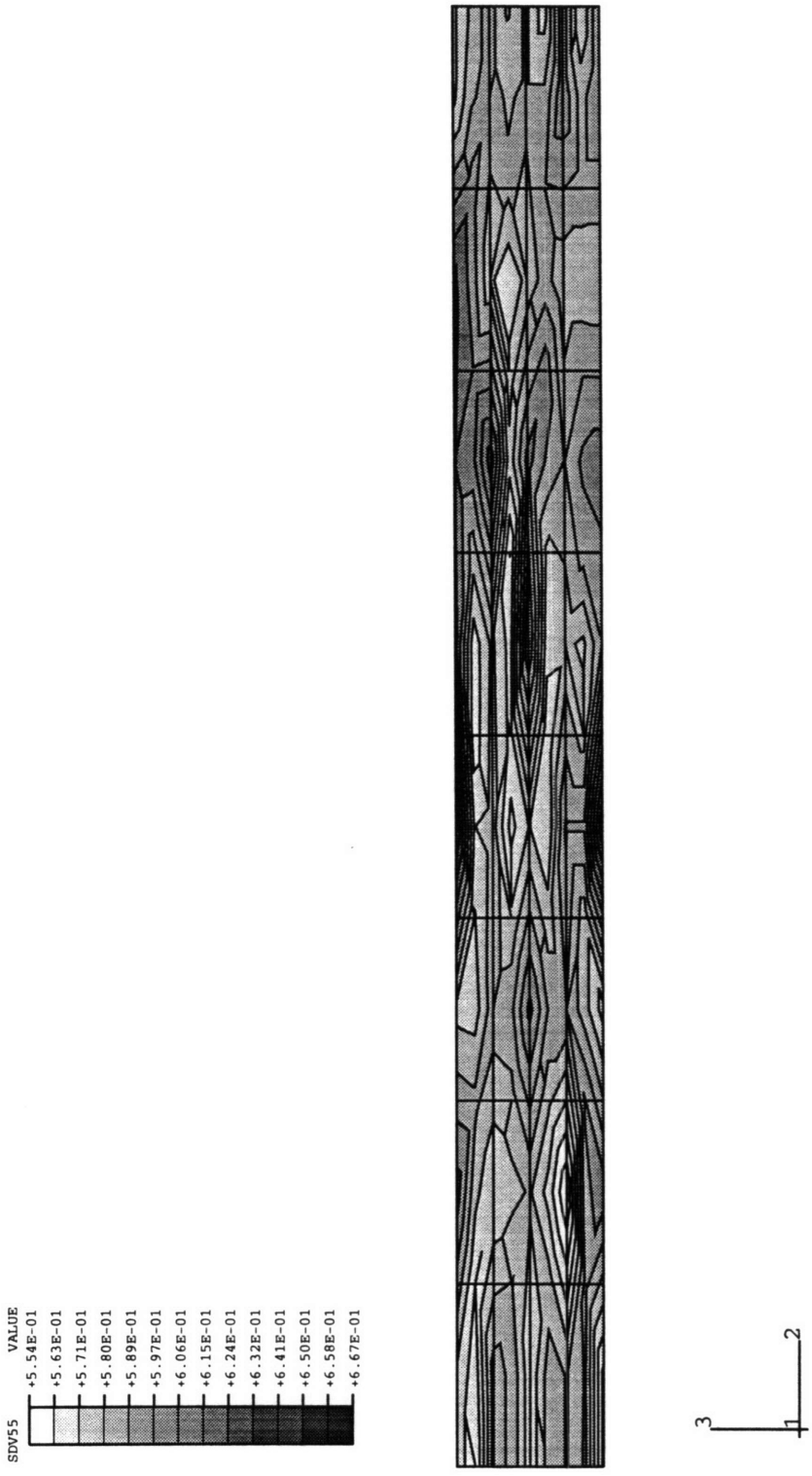


Figure 4-14: Variation of the Volume Fraction in the 2-3 Plane

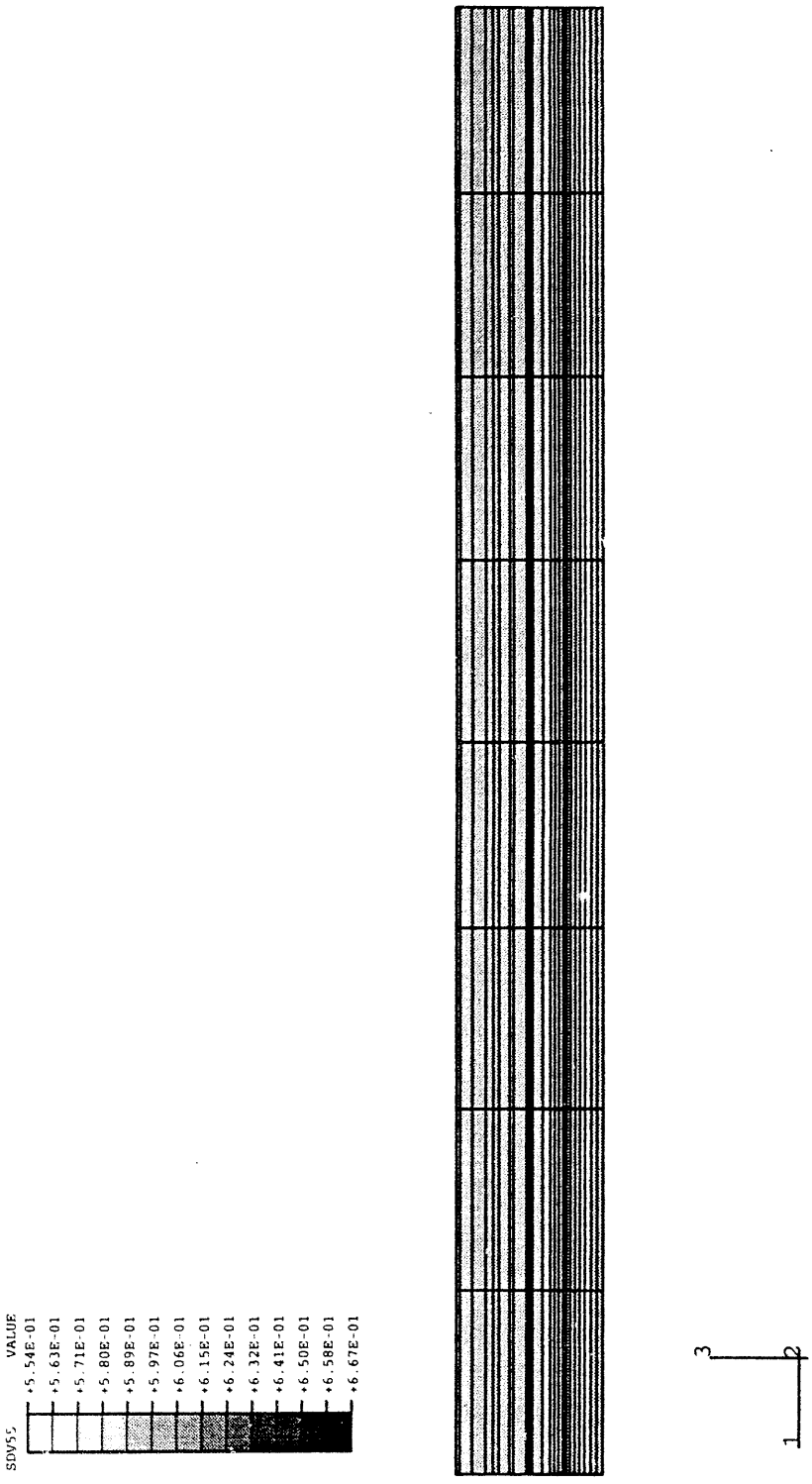


Figure 4-15: Variation of the Volume Fraction in the 1-3 Plane

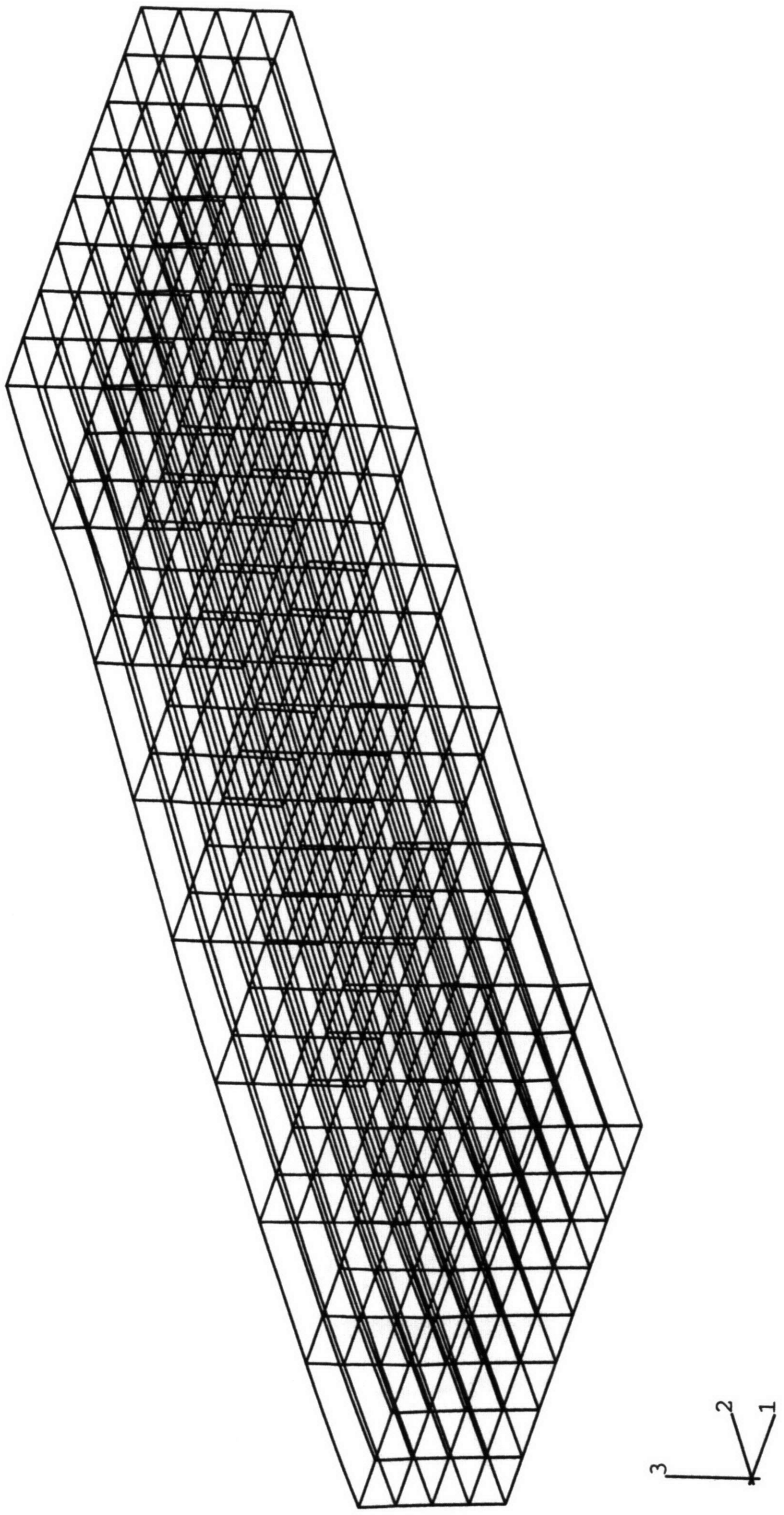


Figure 4-16: Deformed Mesh of Plate with Varying Volume Fraction
The Displacements are Magnified 300 Times

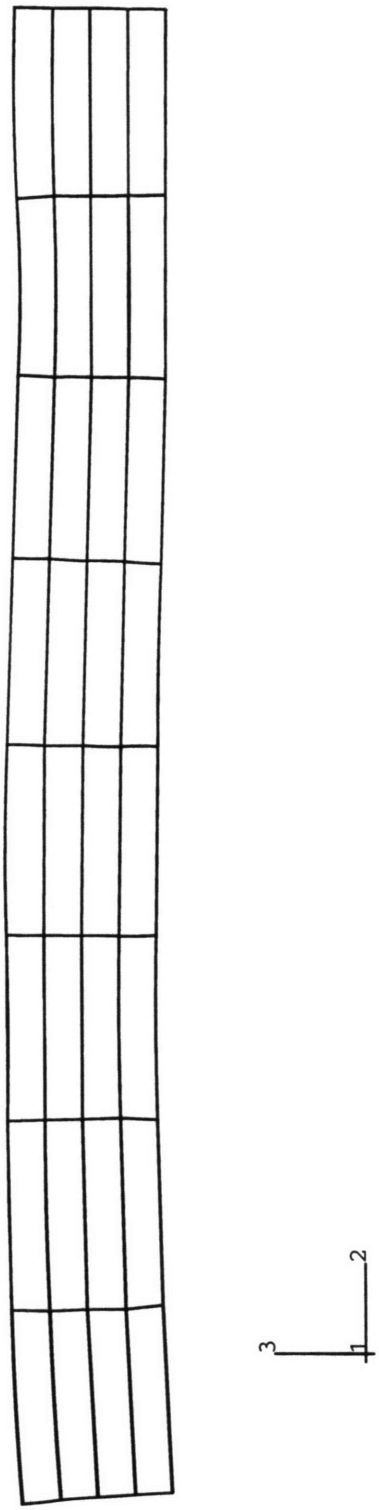


Figure 4-17: Side View of Deformed Mesh at 1000 Times Magnification

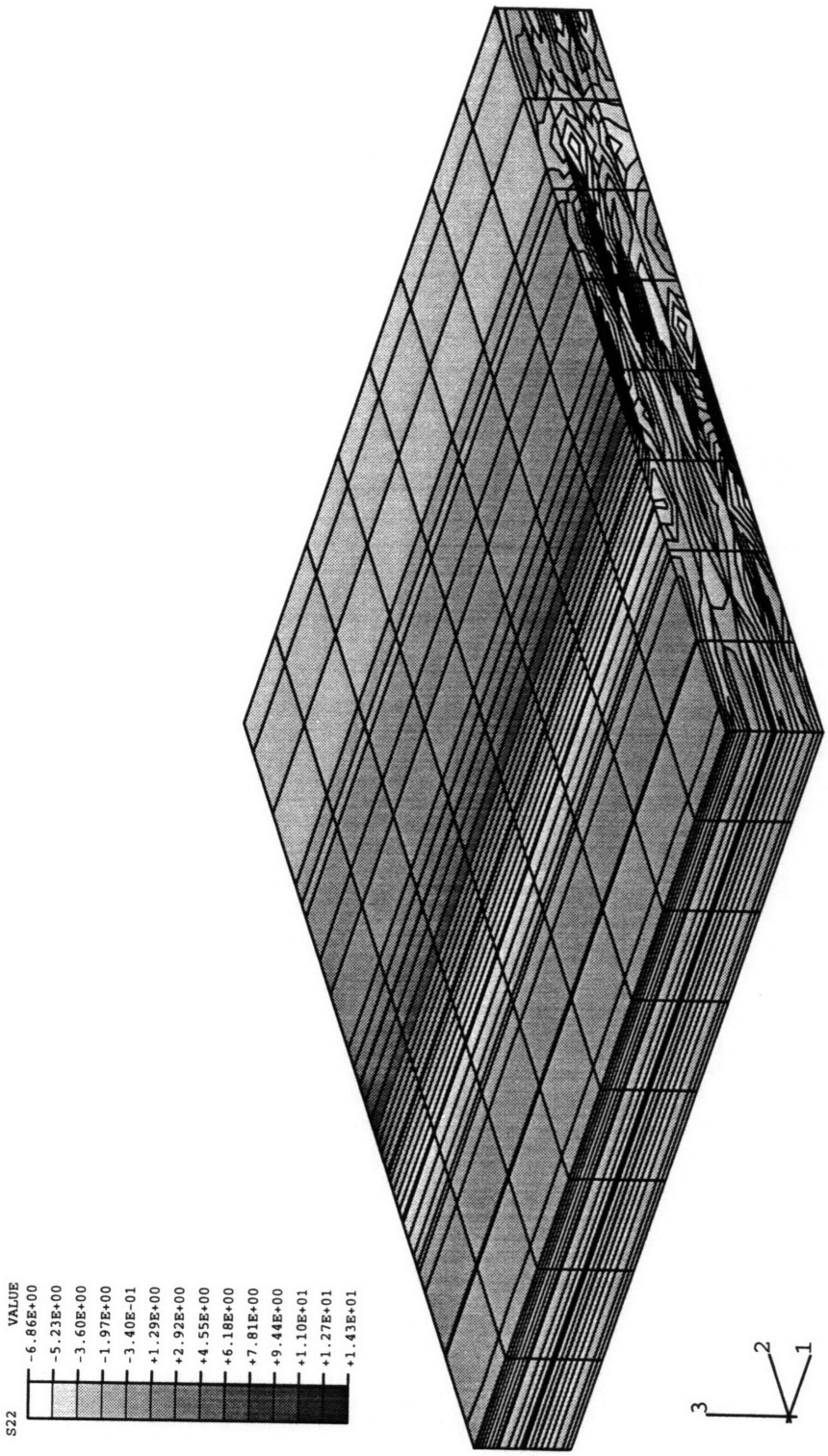


Figure 4-18: Normal Stress in the Two Direction for Plate with Varying Volume Fraction

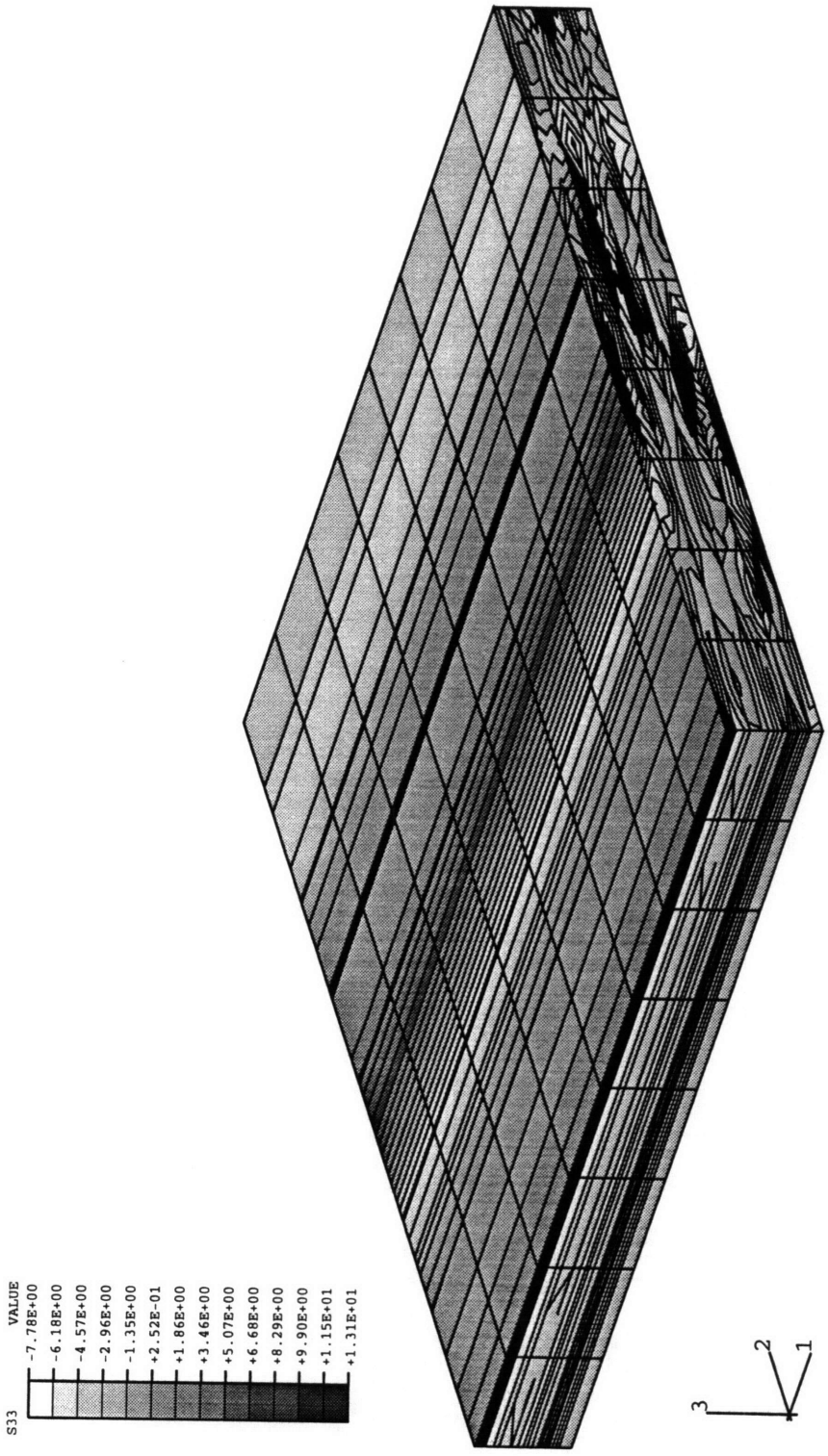


Figure 4-19: Normal Stress in the Three Direction for Plate with Varying Volume Fraction

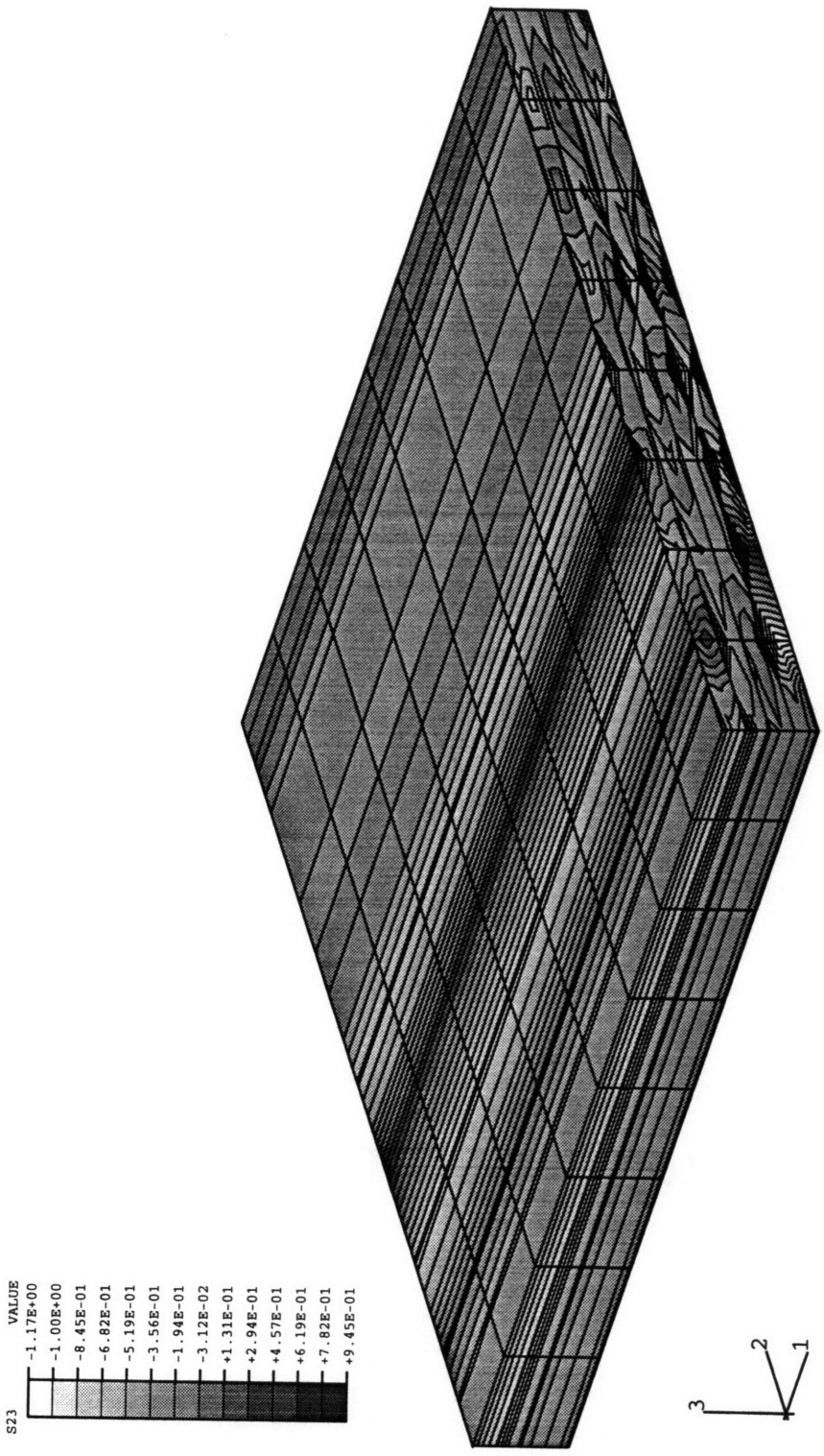


Figure 4-20: Shear Stress for Plate with Varying Volume Fraction

stresses in the fiber more than doubled when the composite was no longer assumed to be perfectly periodic. As a result of the higher stresses in the composite constituents, it was also found that the values for the failure criteria listed in the table had increased.

In conclusion, the addition of variability in the fiber spacing throughout the matrix has indicated that the periodic method of cells model may understate the effects caused by the large difference between the fiber and matrix properties. The addition of variation has no doubt produced a model truer to the actual composite, but further study is needed in which all of the input properties are allowed to vary. Even more important is the need for comparison with experimental data.

4.4 Computational Expense

The use of a micromechanical material model such as the one developed in this work is limited by the amount of time it adds to the total time required to complete the analysis. The fraction of the computation time devoted to material property calculations to the total computation time decreases as the number of degrees of freedom increases. This is due to the fact that as the number of degrees of freedom grows, the amount of time spent in elimination to solve the problem drastically increases while the increase in the amount of time spent in material calculations is approximately linear. The usefulness of a complex material model thus depends on the rate at which the proportion of time spent in the material model decreases. Since ABAQUS does not report the amount of time spent on each different part of the analysis, it was necessary to use an estimate of the way that increasing problem size reduces the relative effect of material property calculations on the total computation time. This estimate was performed by comparing computation times between problems run with the method of cells material model and problems run with a macroscopic material model.

The unidirectional plate bearing pressure problem of Section 4.2.2 was used to assess the behavior of this estimate. Before the analyses could be run, it was necessary to re-mesh the plate several times. The meshes started at using a single C3D20

Table 4.13: Comparison of Computation Time Between Plate Problem with and without Method of Cells

Problem Size	Computation Time[a]			
	Cray Y-MP 8/64[b]	DEC 3000[c]	VAX 4000-60	VAX 4000-200
One Element	0.182	0	2	4
with user routine	0.628	1	12	26
Four Elements	0.432	1	6	13
with user routine	2.167	6	45	100
32 Elements	2.688	8	46	108
with user routine	16.33	55	393	800
256 Elements	22.32	81	703	1737
with user routine	131.5	400	3439	7254
2048 Elements	338.9	1831	38153	98562
with user routine	1214.	4385	59049	142619

[a]All values are in seconds.
[b]The version of ABAQUS for the Cray reports computation time in seconds up to four decimal places while the version for VMS rounds to the nearest second.
[c]This workstation uses the new DEC alpha chip.

element to model the whole plate. The mesh was then refined by dividing the plate into four C3D20 elements. Consecutive meshes were then constructed by multiplying the number of elements by two for the width, length, and thickness, yielding meshes of 32 elements, 256 elements, and 2048 elements. These meshes were run using both the standard orthotropic model of ABAQUS and the method of cells micromechanical model. The results obtained are summarized in absolute terms in Table 4.13 while the ratios of computation times are given in Table 4.14. The problems were run on several different platforms to also estimate the effect of computer architecture on the decay of the ratio.

The trends in the ratios show that the effect of the additional computation time due to the use of the method of cells material model is large initially and then decreases rapidly as the number of degrees of freedom becomes very large (see Figure 4-21). The slight increase in the ratios at the intermediate problem sizes may be due to a couple factors. First of all, the computation times were only given to the nearest second on the DEC workstations. As a result, the ratio is highly sensitive to the rounding of the values for smaller problems. Secondly, when the number of degrees

Table 4.14: Ratio of Computation Times with and without the Method of Cells

Problem Size	Ratio			
	Cray Y-MP 8/64	DEC 3000	VAX 4000-60	VAX 4000-200
One Element	3.45	Undef.[a]	6.00	6.50
Four Elements	5.02	6.00	7.50	7.69
32 Elements	6.08	6.88	8.54	7.41
256 Elements	5.89	4.94	4.89	4.18
2048 Elements	3.58	2.39	1.55	1.45

[a]The computation time for the problem without the user routine was rounded to zero by the computer so the ratio could not be computed.

of freedom is small, the elimination is still a very minor part of the analysis. In this case the setup of the problem and the material routine no doubt dominate the total computation time.

The effect of computer architecture can be seen in Figure 4-21. The ratio for the DEC platforms falls off quickly towards the asymptotic value of one after the 32 element problem. However, the behavior on the Cray Y-MP is slightly different. This may to some extent be due to the fact that the values for the Cray were reported in seconds with significant digits up to three decimal places and henceforth are more accurate for the smaller problems. The ratios for the larger problems show however that the Cray still behaves differently as the problem size grows. The increase in the ratio for the Cray is more gradual and also decays later and slower than those for the DEC workstations. This effect is probably due to the vectorized nature of the Cray since the user routine was not written to take advantage of vectorization.

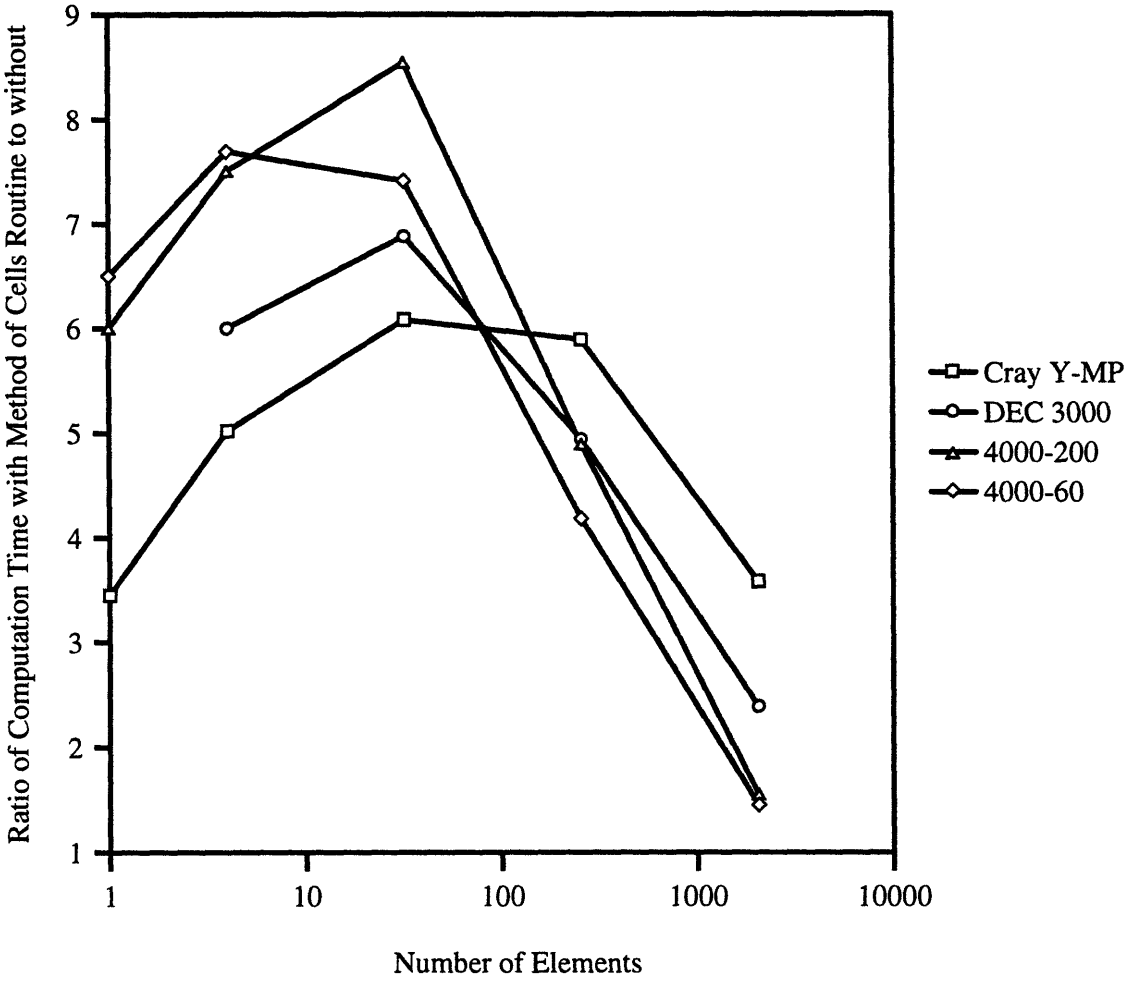


Figure 4-21: Ratio of Computation Times with and without the Method of Cells Material Model for Several Platforms

Chapter 5

Nonlinear Finite Element Adaptation

As mentioned previously, the model developed in Chapter 3 is for linear-elastic materials. In order to allow the model to handle more realistic material behaviors such as plasticity and/or damage evolution, it is necessary to include nonlinearity. In a nonlinear finite element model, the stiffness matrix is no longer independent of the loading history as in the linear-elastic case. It is instead a function of the displacements and forces internal to the representative cell, as shown in equation 5.1,

$$[K] = \frac{\partial\{R_i\}}{\partial\{u\}} \tag{5.1}$$

where $[K]$ is now the tangent or instantaneous stiffness matrix, $\{R_i\}$ is the internal force vector, and $\{u\}$ is the displacement vector. As a result, it is necessary to iterate on the displacements inside of the material model until convergence is achieved.

This chapter outlines the inclusion of a nonlinear iteration scheme into the method of cells material model. In addition, a very simple debonding model is introduced for the interface element, which is then employed to demonstrate the use of this nonlinear iteration scheme.

5.1 Newton Iteration Scheme

To perform the iterations on the displacements and internal forces, a Newton iteration scheme was included in the material model. The displacement increment at iteration step i was determined according to the difference equation:

$$\{u^{t+\Delta t}\}_i = \{u^{t+\Delta t}\}_{i-1} + [K(\{u^{t+\Delta t}\}_{i-1})]^{-1} \left\{ \{R_b^{t+\Delta t}\} - \{R_i(\{u^{t+\Delta t}\}_{i-1})\} \right\} \quad (5.2)$$

where R_b is the boundary force vector. The convergent internal displacements and the convergent internal forces from the previous time increment were used as the starting point for the first iteration at $i = 1$. That is,

$$\{u^{t+\Delta t}\}_0 = \{u^t\}_c \quad (5.3)$$

$$\left\{R_i(\{u_i^{t+\Delta t}\}_0)\right\} = \left\{R_i(\{u^t\}_c)\right\} \quad (5.4)$$

where $\{R_i(\{u^t\}_c)\}$ and $\{u^t\}_c$ are the convergent internal forces and displacements of the previous increment. In order to have the internal displacements and forces of the previous time step available at the next time increment, it was necessary to store the internal displacements at each step as state variables after convergence was reached. The convergent internal forces could then be calculated using the displacements.

Convergence for the effective constitutive matrix was determined by comparing the incremental internal force residuals at each iteration to a convergence tolerance chosen by the analyst.

5.2 Damage Model

Composite properties are heavily dependent on the strength of the bond between the two material phases. In many cases it is the breaking of these bonds that ultimately leads to the failure of the composite. In order to try and capture the effect of debonding on the composite behavior, the interface element discussed in section 3.2 was revised to allow for damage to the bonds. The degradation of these bonds over

time affects the overall material stiffness making the model nonlinear and requiring the use of an iteration scheme such as the Newton method presented above.

At this stage in the development of the nonlinear debonding model, the exact form of the debonding criteria was less important than the actual framework set up to incorporate debonding. A complicated interface model at this point would in all probability be too computationally expensive to be useful for the intended purpose of this model. The Mohr-Coulomb criterion introduced below is most likely a poor model of yield in the interface. It should be noted that modeling the interface between the constituents of a composite is currently a very active field of research, and accurate experimental data for the stiffness and failure of the interface is practically nonexistent.

The damage model introduced in this section was based on a Mohr-Coulomb form for the debonding criteria, as shown in equation 5.5

$$\tau_e = \mu\sigma_1 + (\tau_{12}^2 + \tau_{13}^2)^{1/2} \tag{5.5}$$

where τ_e is an "effective stress" and μ is a property of the bond similar to a friction coefficient. This criterion has been used in SOILS models and for polymer yielding [41]. It describes failure as pressure dependent, relating the onset of permanent deformation to the stress state. It uses the idea of internal friction to calculate an effective shear stress which depends on the normal force to determine when slippage begins. As a result, the bond will fail sooner in tension than in compression using this model.

The relation in 5.5 can be converted to a strain basis:

$$\tau_e = \mu E\epsilon_1 + G(\gamma_{12}^2 + \gamma_{13}^2)^{1/2}$$

$$\gamma_e = \frac{\mu E}{G}\epsilon_1 + (\gamma_{12}^2 + \gamma_{13}^2)^{1/2} \tag{5.6}$$

where γ_e is then the effective strain. The coefficients E and G are also properties of the bond.

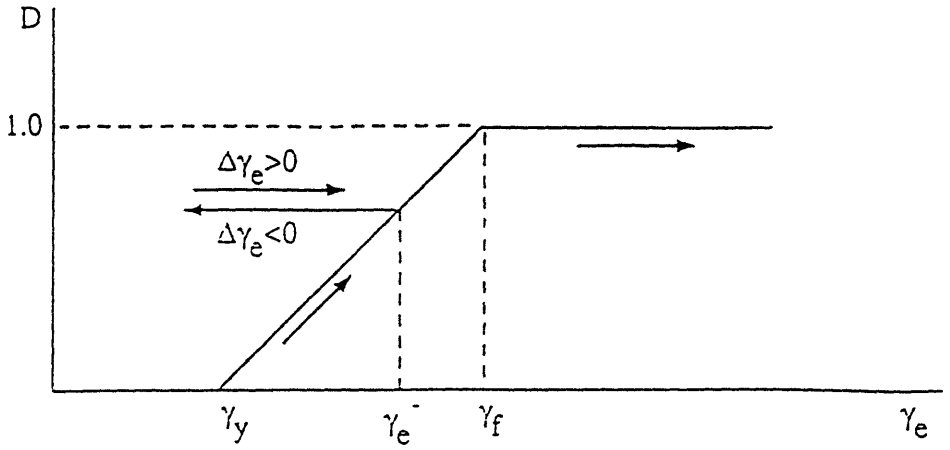


Figure 5-1: Relationship of the Damage Parameter to the Effective Strain

Using the effective strain, a functional relationship for the damage parameter was defined as shown in Figure 5-1. The parameters γ_y and γ_f can be thought of as yielding and failure strain levels, respectively, while γ_e^- is the maximum value of γ_e obtained up to the current step. The damage parameter has a physical interpretation as the ratio of the debonded area of the interface to the total area of the interface:

$$D = \frac{A_D}{(A_D + A_B)} \tag{5.7}$$

where D ranges from 0 to 1. It should be noted that D depends only on the maximum value of γ_e over time and never decreases.

The force-deflection relations for the springs of the interface element are then defined as:

$$\begin{aligned} F_1 &= K_n(1-D)\delta_1 \quad (\delta_1 > 0) \\ F_1 &= K_n\delta_1 \quad (\delta_1 < 0) \end{aligned} \tag{5.8}$$

for the normal spring (direction one) and

$$F_i = K_s(1 - D)\delta_i, \quad i = 2, 3 \tag{5.9}$$

for the shear springs in the two and three directions. The stiffnesses are defined as:

$$K_n = \frac{AE}{h}$$
$$K_s = \frac{AG}{h}$$

where h is the bond thickness and $A = A_D + A_B$. The strains are derived from the displacements as

$$\epsilon_1 = \frac{\delta_1}{h}$$
$$\gamma_{12} = \frac{\delta_2}{h}$$
$$\gamma_{13} = \frac{\delta_3}{h}$$

Using the force-deflection relationships, the tangent stiffness matrix can be found by differentiating 5.8 and 5.9.

$$\{\partial F\} = [K]\{\partial \delta\} \tag{5.10}$$

where

$$\{\partial F\} = \begin{Bmatrix} \partial F_1 \\ \partial F_2 \\ \partial F_3 \end{Bmatrix}$$

and

$$\{\partial \delta\} = \begin{Bmatrix} \partial \delta_1 \\ \partial \delta_2 \\ \partial \delta_3 \end{Bmatrix}$$

The k_{11} component of the tangent stiffness matrix is then

$$k_{11} = \frac{\partial F_1}{\partial \delta_1} = K_n(1 - D) - K_n \delta_1 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_1}\right) \tag{5.11}$$

for $\delta_1 > 0$ and

$$k_{11} = \frac{\partial F_1}{\partial \delta_1} = K_n \tag{5.12}$$

for $\delta_1 < 0$. The chain rule derivatives are:

$$\frac{\partial D}{\partial \gamma_e} = 0 \tag{5.13}$$

for $\gamma_e < \gamma_y$, $\gamma_e > \gamma_f$, or $\gamma_e < \gamma_e^-$ and

$$\frac{\partial D}{\partial \gamma_e} = \frac{1}{(\gamma_f - \gamma_y)} \tag{5.14}$$

when $\gamma_y < \gamma_e < \gamma_f$ and $\Delta\gamma_e > 0$, $\gamma_e > \gamma_e^-$ (See Figure 5-1). And,

$$\frac{\partial \gamma_e}{\partial \delta_1} = \frac{\mu E}{Gh} \tag{5.15}$$

Similarly, the off-diagonal terms are:

$$k_{12} = \frac{\partial F_1}{\partial \delta_2} = -K_n \delta_1 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_2}\right) \tag{5.16}$$

where

$$\frac{\partial \gamma_e}{\partial \delta_2} = \frac{\delta_2}{h(\delta_2^2 + \delta_3^2)^{1/2}} \tag{5.17}$$

and

$$k_{13} = \frac{\partial F_1}{\partial \delta_3} = -K_n \delta_1 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_3}\right) \tag{5.18}$$

where

$$\frac{\partial \gamma_e}{\partial \delta_3} = \frac{\delta_3}{h(\delta_2^2 + \delta_3^2)^{1/2}} \tag{5.19}$$

The remaining stiffness terms then follow as:

$$k_{21} = \frac{\partial F_2}{\partial \delta_1} = -K_s \delta_2 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_1}\right) \tag{5.20}$$

$$k_{22} = \frac{\partial F_2}{\partial \delta_2} = K_s(1-D) - K_s \delta_2 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_2}\right) \tag{5.21}$$

$$k_{23} = \frac{\partial F_2}{\partial \delta_3} = -K_s \delta_2 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_3}\right) \tag{5.22}$$

$$k_{31} = \frac{\partial F_3}{\partial \delta_1} = -K_s \delta_3 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_1}\right) \tag{5.23}$$

$$k_{32} = \frac{\partial F_3}{\partial \delta_2} = -K_s \delta_3 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_2}\right) \tag{5.24}$$

$$k_{33} = \frac{\partial F_3}{\partial \delta_3} = K_s(1-D) - K_s \delta_3 \left(\frac{\partial D}{\partial \gamma_e}\right)\left(\frac{\partial \gamma_e}{\partial \delta_3}\right) \tag{5.25}$$

A complete listing of the code for the nonlinear material model with the damage interface elements is provided in Appendix B.

5.3 Example Results

The reader is advised once again that actual values for the properties of the bond described above are unavailable. The following exercises are hence simply numerical experiments to illustrate both the use of the nonlinear routine and the prediction of failure due to both matrix-fiber debonding and fiber breakage.

Along with the introduction of the damage criterion in the interface element, additional "interface" elements were added to the original representative cell mesh so that matrix cracking and fiber breakage in the axial direction were also possible (See Figure 5-2). The word interface is used loosely here since the same element and failure criterion as used for modeling matrix-fiber debonding is also used to model the fiber breakage and matrix cracking. A summary of the failure modes allowed by the resulting model is given in Table 5.1. Also listed in the table are the state variables used to store the damage parameter for each particular element.

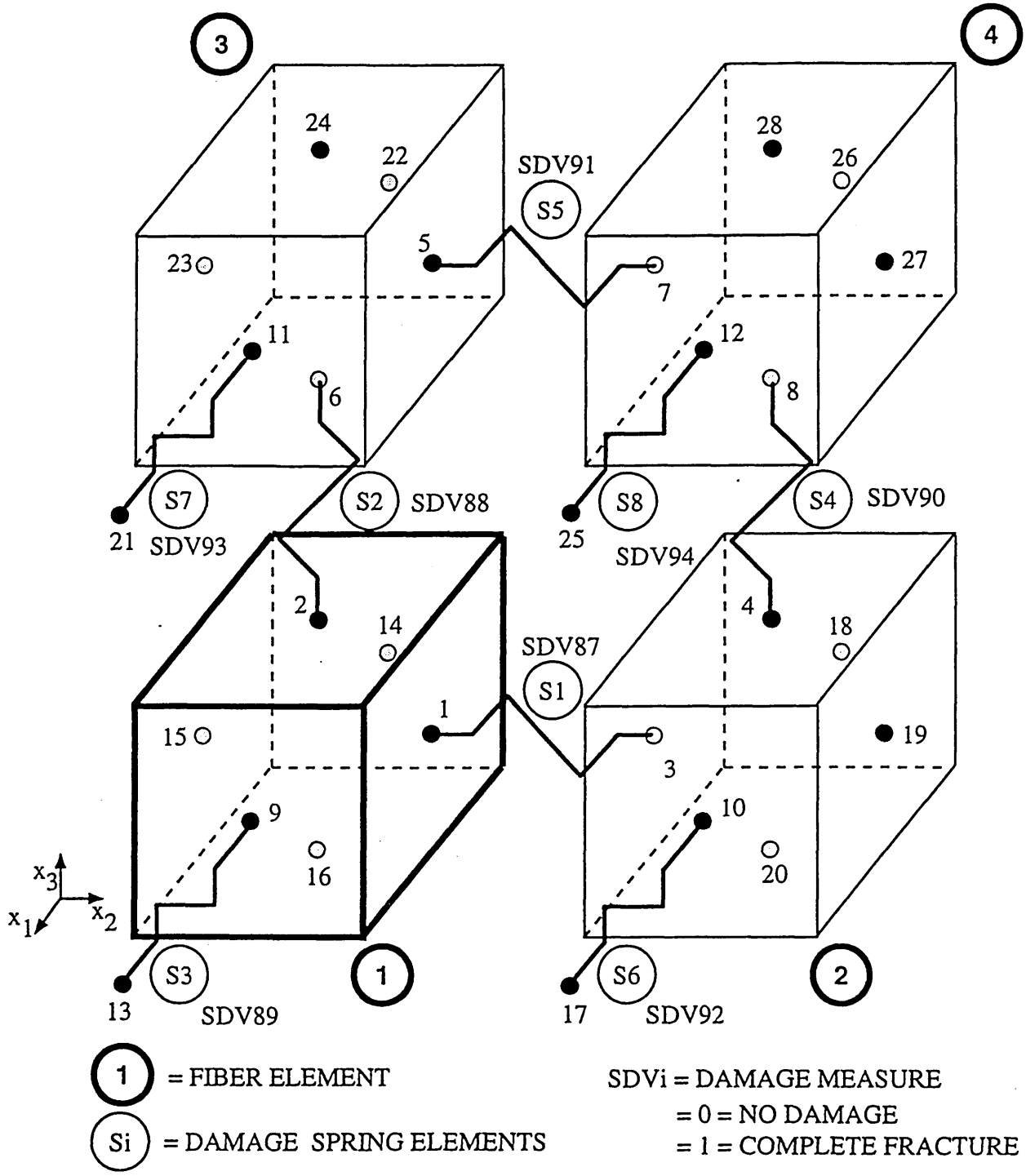


Figure 5-2: Representative Cell with Added "Interface" Elements In Axial Direction
The SDVi are the actual state variables used to store the damage parameter
for the interface elements.

Table 5.1: Failure Modes of the Revised Method of Cells Material Model

Interface Element[a]	Type of Failure Modeled	Associated State Variable
S1	Matrix-Fiber Debonding	SDV87
S2	Matrix-Fiber Debonding	SDV88
S3	Fiber Breakage	SDV89
S4	Matrix Cracking	SDV90
S5	Matrix Cracking	SDV91
S6	Matrix Cracking	SDV92
S7	Matrix Cracking	SDV93
S8	Matrix Cracking	SDV94

[a]As designated in Figure 5-2.

5.3.1 Matrix-Fiber Debonding

To illustrate the effect of matrix-fiber debonding on the behavior of a composite structure, a mesh was constructed for a cantilever beam subjected to a prescribed end deflection. During the analysis, the deflection was ramped up to the maximum value and then ramped back down to zero. The fibers in the model were unidirectional and ran along the length of the beam (the global one direction in Figure 5-3). The interface properties were specified so that failure would occur by shearing at the midplane of the beam. The form of the effective strain-damage parameter curve was such that brittle fracture occurred when γ_f was reached, similar to the behavior of many common matrix materials. The problem was then run for three temperature ranges. In the first, the beam remained at the reference temperature for zero thermal strain, $20\,^{\circ}C$, throughout the loading. In the second, the temperature was uniformly raised to $70\,^{\circ}C$ throughout the composite before the loading was begun; likewise, in the third case the temperature was lowered to $-30\,^{\circ}C$ before the beam was loaded.

A plot of the damage parameter in the $S1$ spring for the $-30^{\circ}C$ case is shown in Figure 5-3. The beam is shown just after the first fracture of the matrix-fiber bond has occurred. The displacement at the end of the beam is equal to 15.76 mm at this point. It should be noted that even though the legend shows the maximum damage to be 0.5, the data shows the true value to be 1.0 at the midplane. The value in the legend

is incorrect due to the way that the postprocessor averages values in making contour plots. As can be seen from the plot, the beam has already completely debonded along the midplane, effectively dividing it in two. The displacements in the figure are magnified five times to show how the two halves of the beam consequently slide over each other after the debonding has occurred. Figure 5-4 shows the beam at a later point where the end displacement has increased to 21.39 mm (the displacements are magnified three times). The region of the composite which is completely debonded has expanded, especially at the end where the beam is attached. Finally, in Figure 5-5 when the end deflection has reached 27 mm, it is seen that the fiber and matrix have debonded on such a large scale at the attached end of the beam that it is no longer able to carry a bending load. The cases run at $20\,^{\circ}C$ and $70\,^{\circ}C$ showed similar behavior. It was found however that as the temperature increased, the matrix-fiber bond failed sooner. This effect is shown in the force-deflection plot of Figure 5-6. Both the loading and unloading of the beam are shown in the plot for all three temperatures.

5.3.2 Fiber Breakage

The cantilevered beam problem was run once more, this time using the model to simulate fiber breakage in the S3 "interface" element. The beam was once again loaded by prescribing the end displacement, and it was assumed that the fibers underwent brittle fracture when γ_f was reached. The beam was heated to $70\,^{\circ}C$ before the loading was begun. Figure 5-7 shows the damage parameter in the $S3$ interface element when the fibers are just beginning to break under the load. The breaking begins at the top of the beam at the point where the beam is attached. Likewise, in Figure 5-8 the damage parameter is shown when the end displacement has reached 15.87 mm. The breakage has not yet completely propagated through the thickness of the beam. Finally, in Figure 5-9, the fibers have now failed completely at the point where the beam is attached. A force-deflection plot for the fibers is shown in Figure 5-10. The point where the fibers begin to break can be seen to occur at a deflection of about seven millimeters, whereafter the force drops off sharply to the unloading curve. The

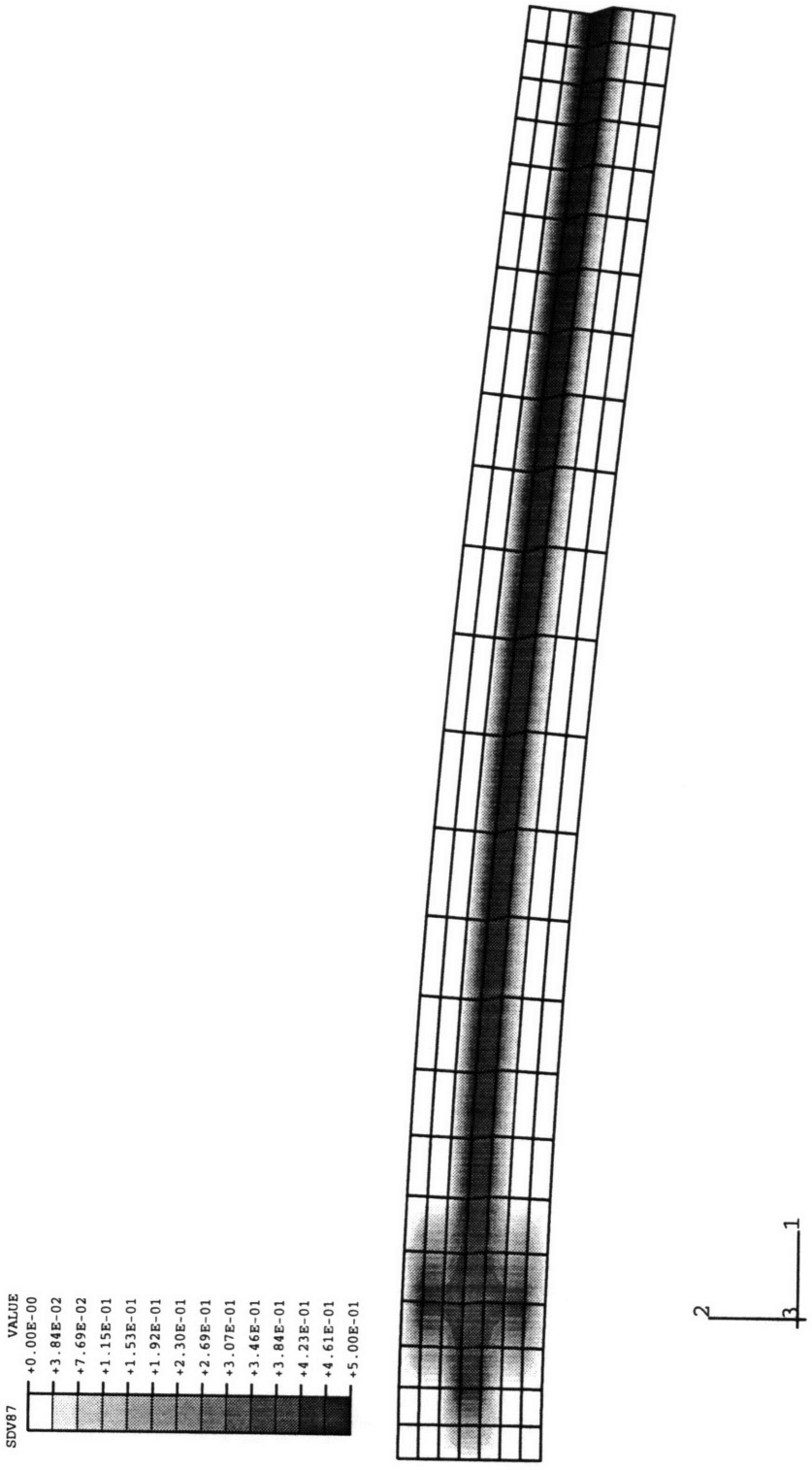


Figure 5-3: Initial Damage in S1 Interface Element for Cantilevered Beam at $-30\,°C$ The legend incorrectly shows the maximum damage to be 0.5 instead of the true value of 1. The displacements are magnified 5 times.

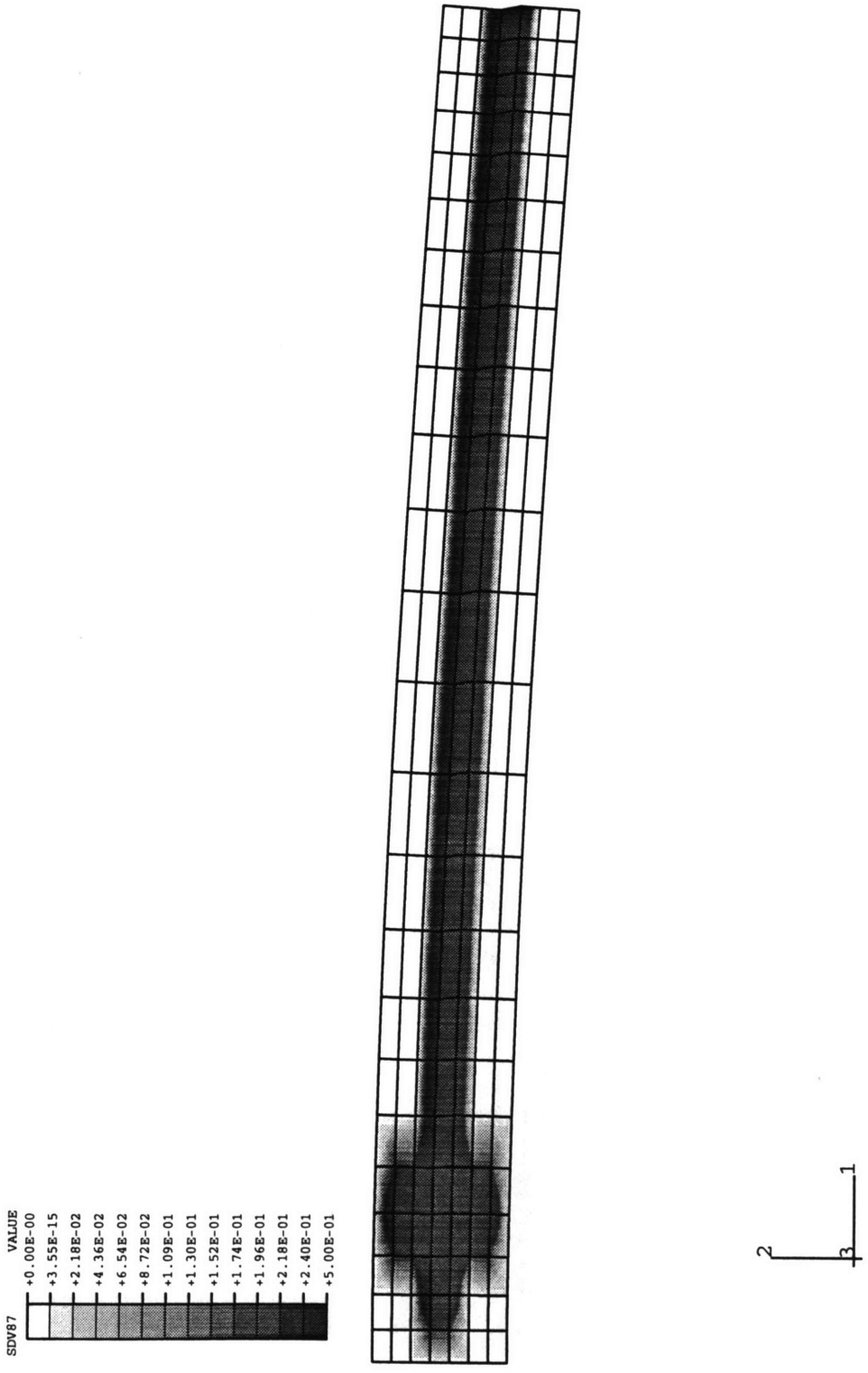


Figure 5-4: Expanded Region of Debonding in Cantilevered Beam at $-30\,^{\circ}C$
The displacements have been magnified 3 times.

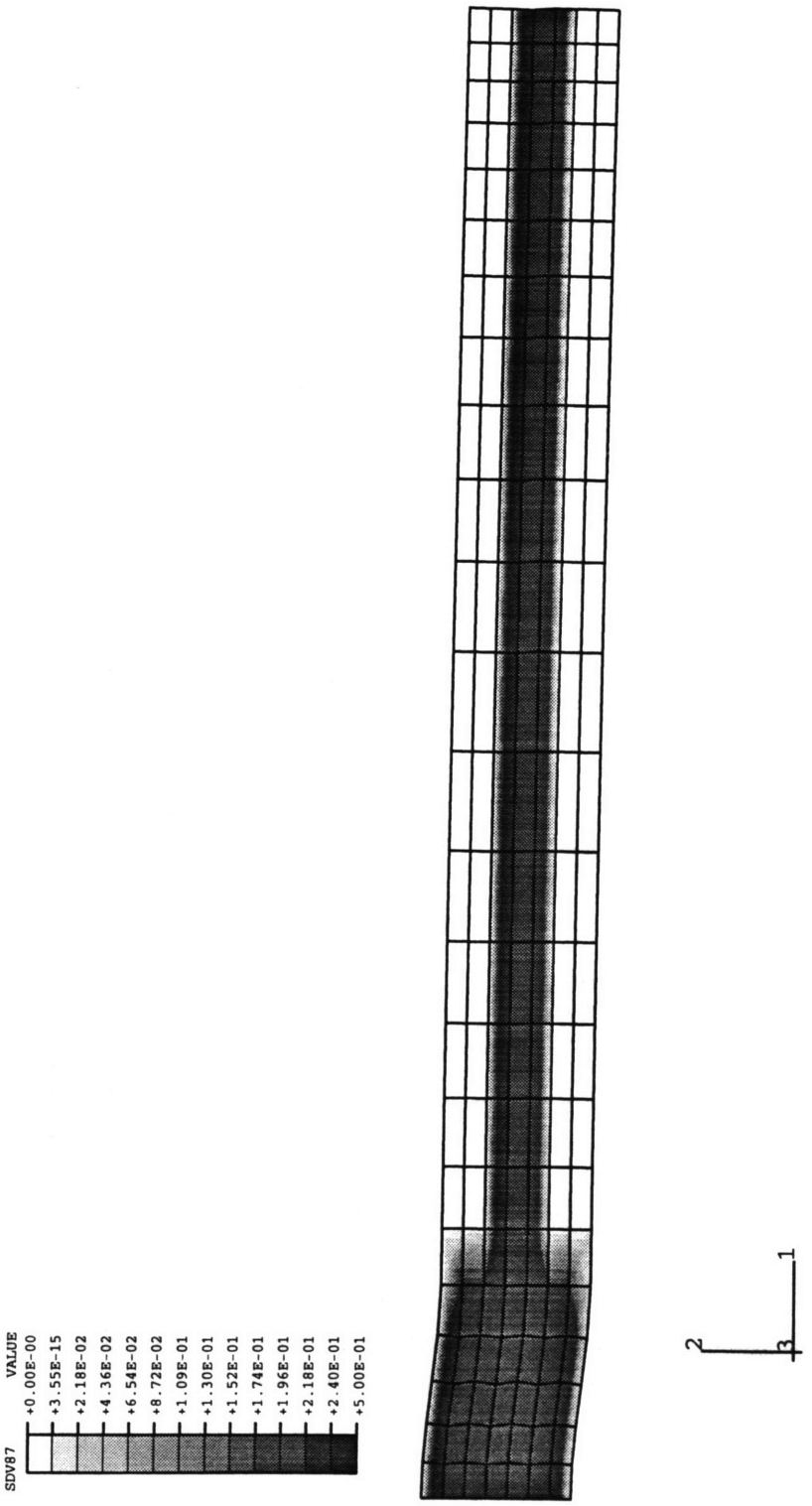


Figure 5-5: Cantilevered Beam at $-30\,°C$ After Having Lost the Ability to Carry Bending Load

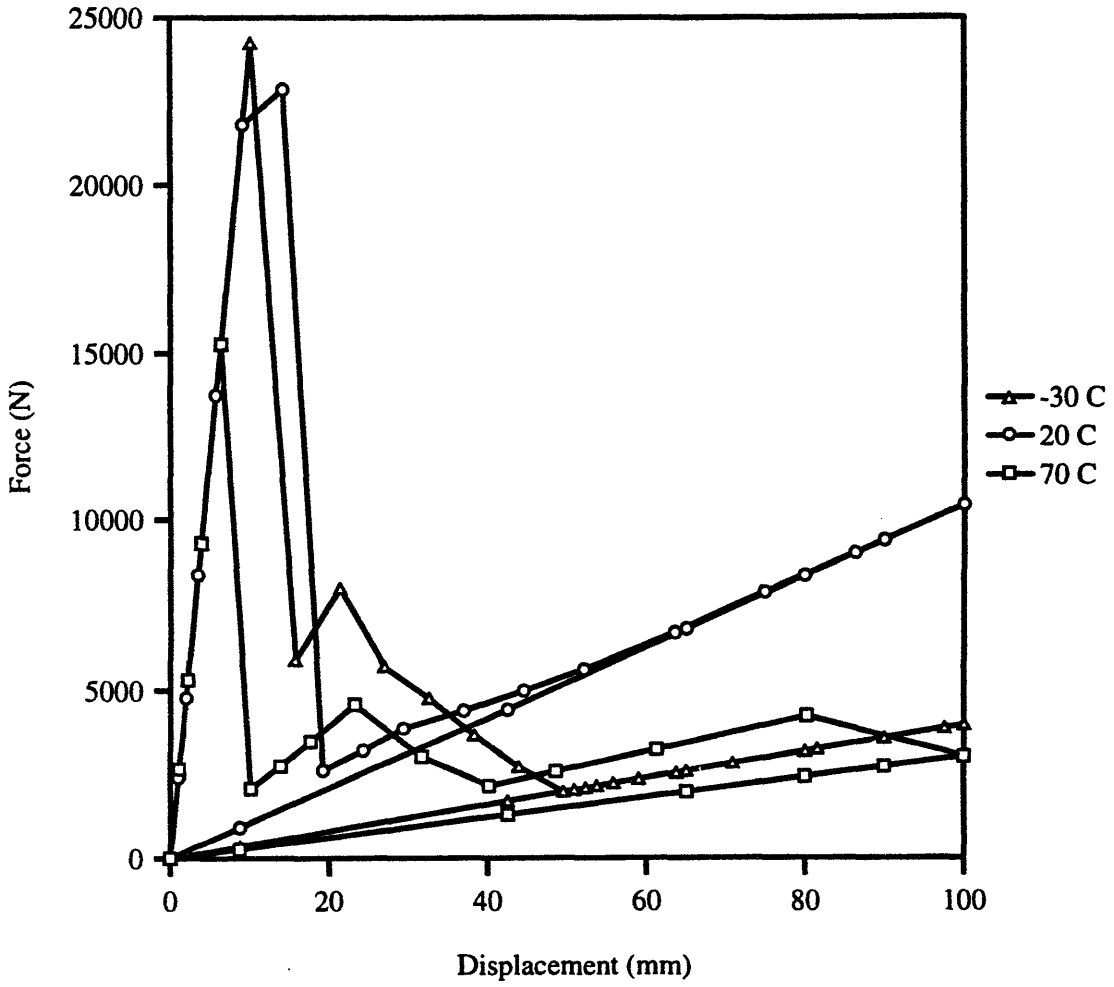


Figure 5-6: Loading and Unloading Force-Deflection Curves For Cantilevered Beam at Various Temperatures with Matrix-Fiber Debonding

sharp dropoff displays the fact that after the fibers start to break, the load is carried primarily in the matrix causing the overall composite to be much weaker.

It should be noted that even for the small problems discussed in this section, the computation time was significantly larger than for the linear-elastic cases of Chapter 4. For example, the cantilever problem was meshed with 168 C3D8R[1] elements. When compared with the 256 C3D20[2] element plate problem shown in Table 4.13 for the Cray Y-MP, it was found that the nonlinear problem took approximately three times as long to arrive at a convergent solution. This is largely due to the fact that twelve more internal degrees of freedom were used in the nonlinear debonding routine than in the linear-elastic case. The iterations themselves made only a small contribution to the increase in computation time as the number of elements undergoing failure was small.

This section was completed with the help of Richard W. Macek of Los Alamos National Laboratory [44].

[1]These are 8-noded, reduced integration elements.

[2]It may be recalled that these are 20-noded, full integration elements.

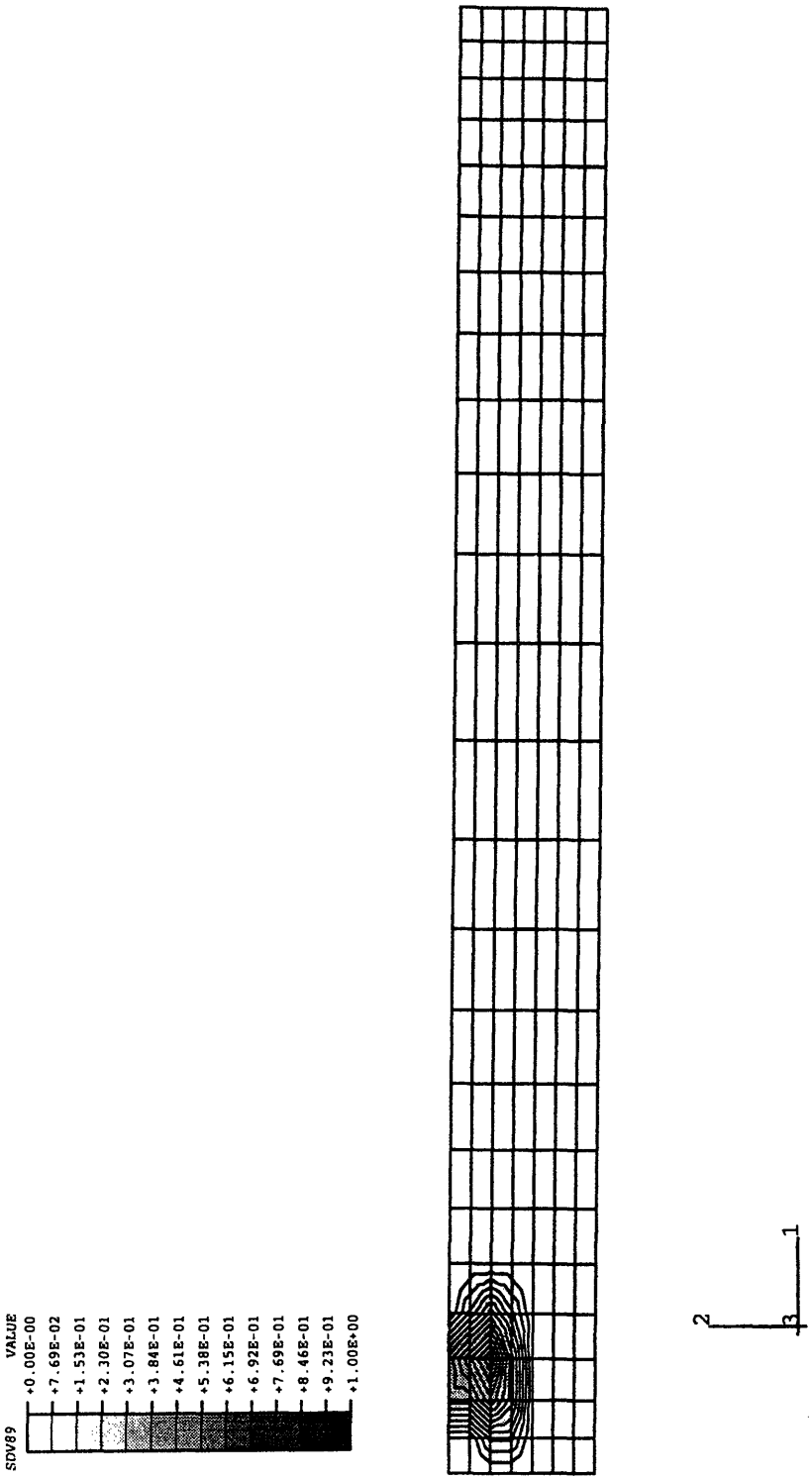


Figure 5-7: Initial Damage in the S3 Interface Element for Cantilevered Beam with Weak Fibers

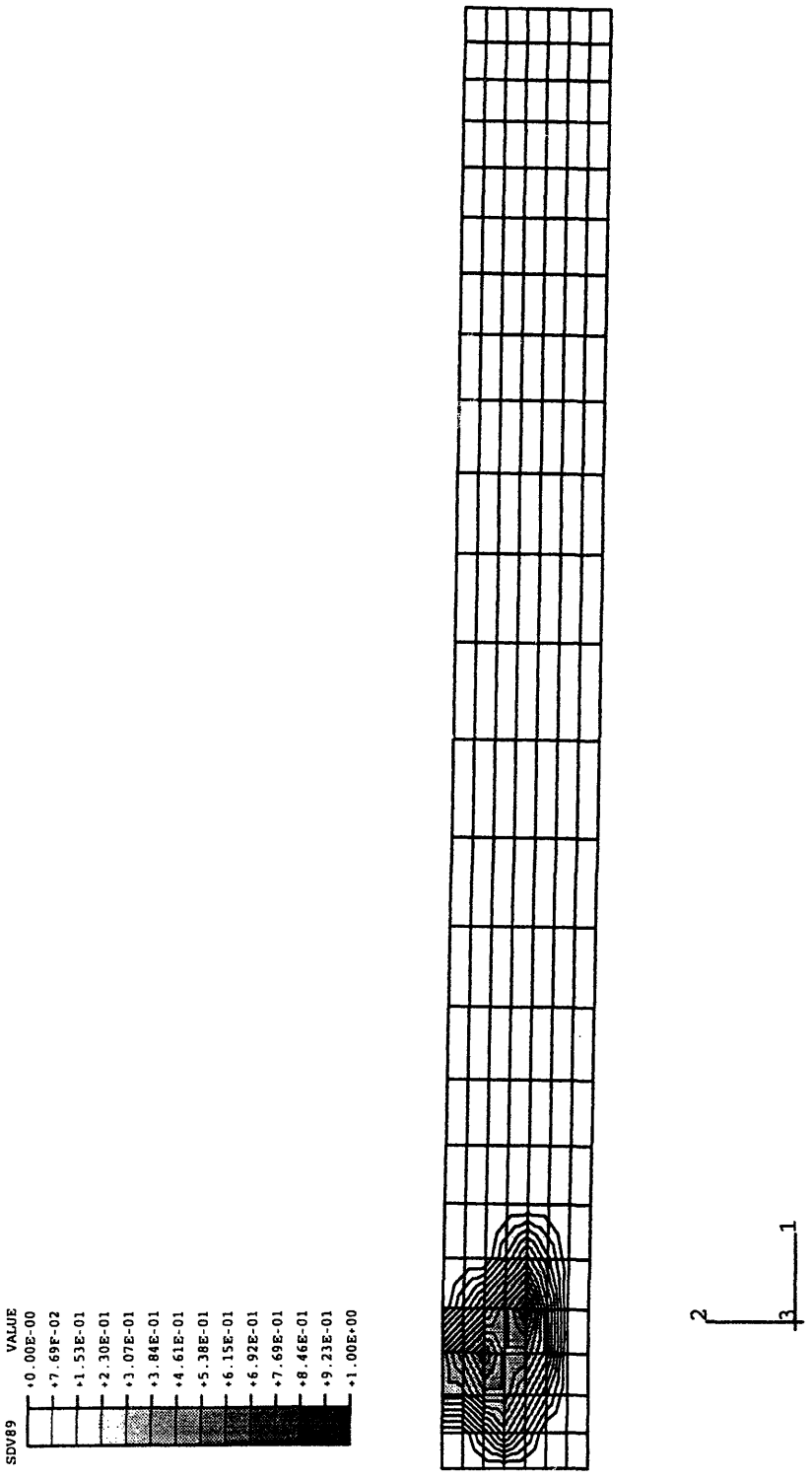


Figure 5-8: Expanded Region of Fiber Breakage for Cantilevered Beam with Weak Fibers

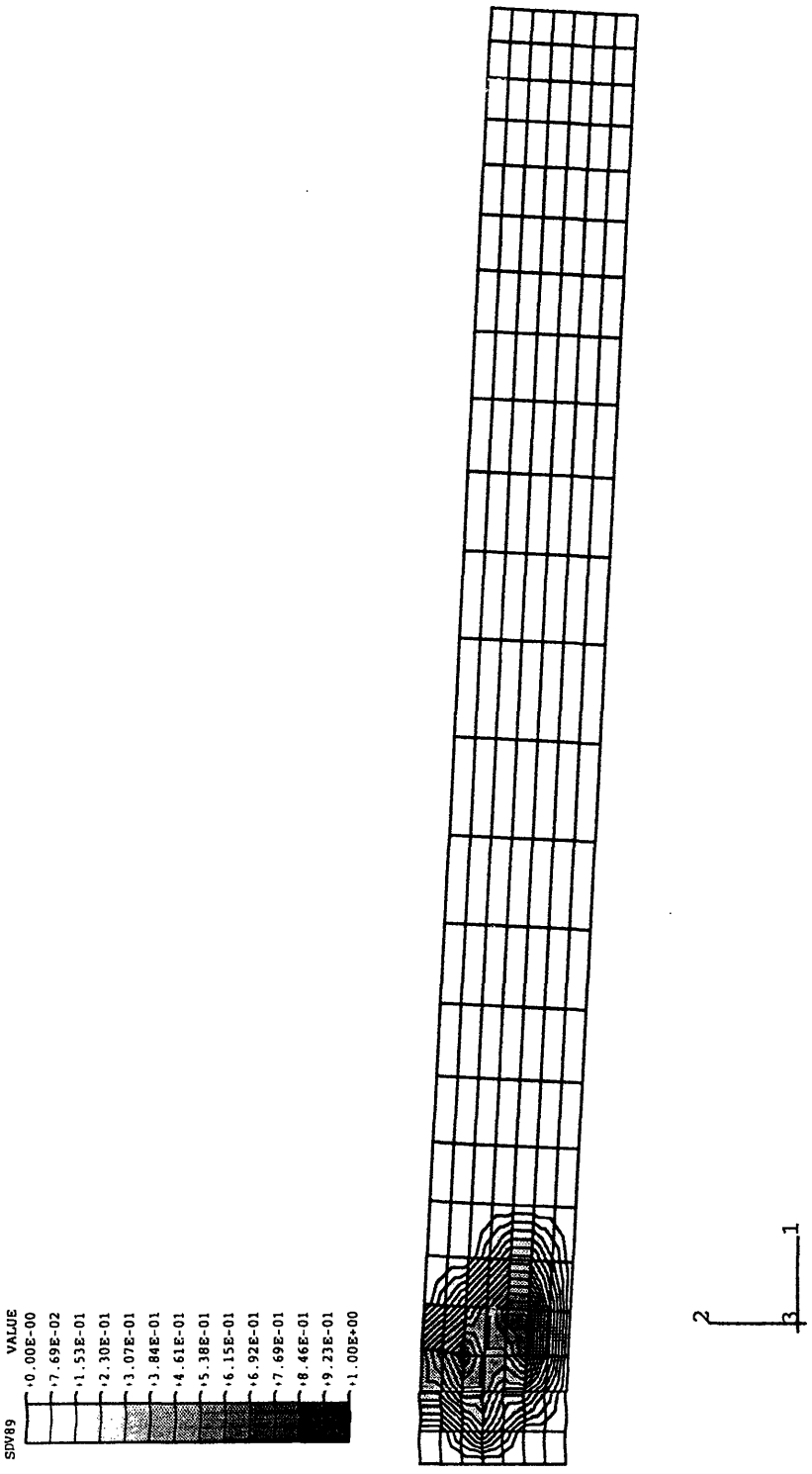


Figure 5-9: Completely Broken Fibers in Cantilevered Beam at $-70\,°C$

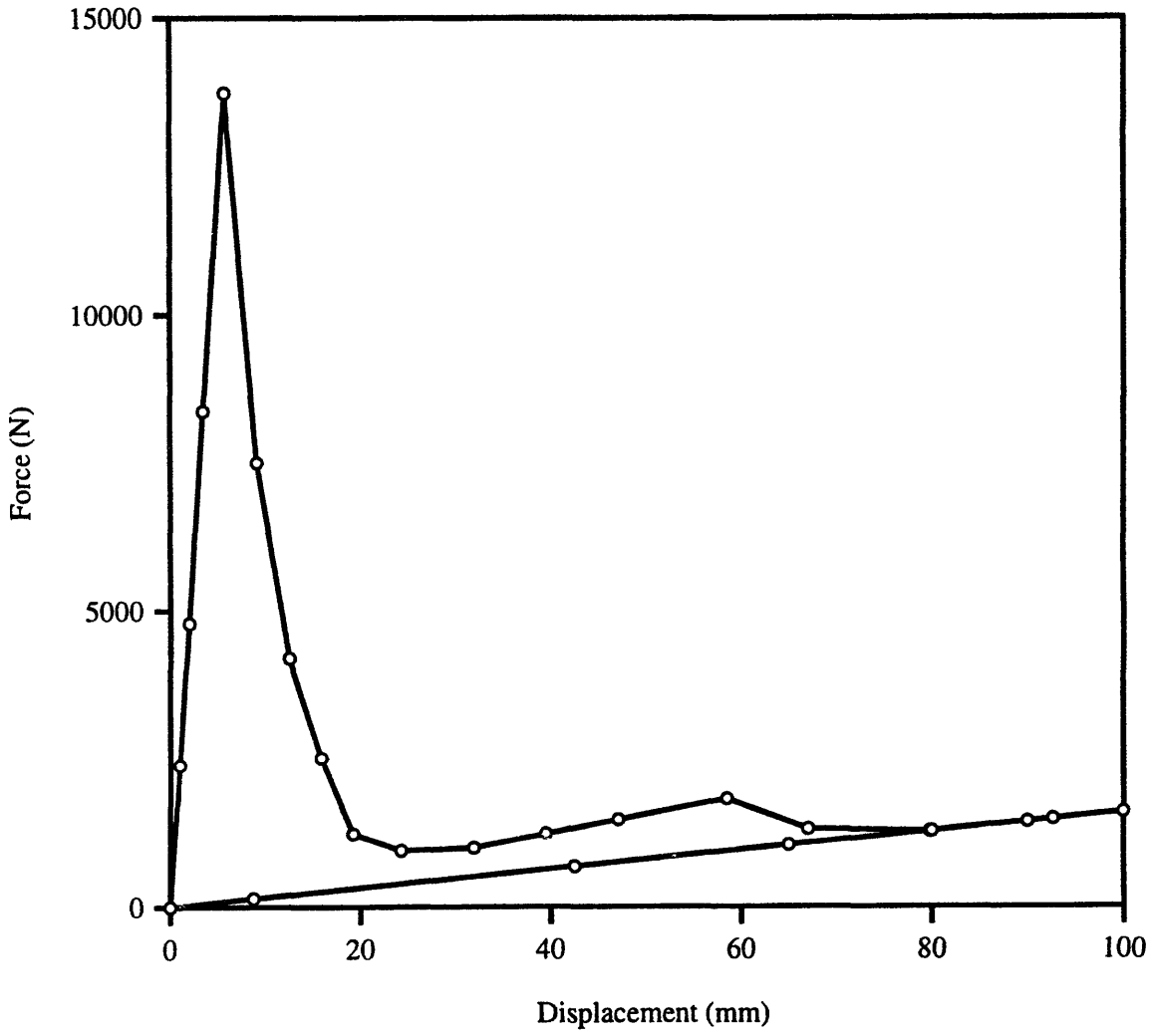


Figure 5-10: Loading and Unloading Force-Deflection Curves For Beam with Weak Fibers

Chapter 6

Conclusions

6.1 Conclusions

6.1.1 Micromechanical Framework Established

The basic framework for using micromechanical material models in finite element analysis was established in ABAQUS. The method of cells developed by Aboudi was cast into this framework for linear elastic unidirectional composites in the form of a user material subroutine. This approach is based on the assumption of a periodic rectangular array for the composite geometry and derives the average properties of the composite using a linear displacement interpolation. Interface elements were introduced into the unit cell of the method of cells during the implementation to eventually allow for the ability to evolve damage in the composite over time.

The user material routine was used to perform several common types of finite element composite structure analysis for a linear orthotropic material. The results were compared with solutions performed using ABAQUS' standard orthotropic material model, and the agreement was found to be excellent. The micromechanical model however provides much more information about the behavior of the composite than the macroscopic model. The individual matrix and fiber stresses and strains are obtained in the analysis in addition to the macroscopic state. The importance of these localized stresses was highlighted in examples of thermal loading situations where,

due to the difference in the fiber and matrix thermal properties, large stresses may be developed in the individual constituents while the macroscopic stress state remains zero. The examination of some simple failure criteria showed that the addition of thermal loading into a structural analysis may lead to constituent failure which is not predicted in a macroscopic model of the composite behavior.

The method of cells material model was then extended to allow for nonlinear behavior using a Newton iteration scheme on the displacements. To demonstrate the use of this iteration scheme, a Mohr-Coulomb form of failure criterion was introduced into the interface element to allow for damage to the composite in the form of fiber-matrix debonding, matrix cracking, and fiber breakage. A beam bending analysis was performed to show the reduction in the stiffness of the composite beam which results as damage accumulates.

The method of cells may also be used to model variation in the properties and microstructure of a composite. To illustrate the use of the method of cells model for this purpose, a statistically based representation of the fiber spacing throughout the matrix was introduced by assuming a normal distribution for the fiber volume fraction used at each material integration point. The introduction of this variation was found to increase the constituent stresses which arise due to the inequality of the fiber and matrix thermal properties.

6.1.2 Computational Expense

The use of the method of cells as a finite element material model has a significant effect upon the expense of the analysis. However, it was found that the ratio of the computation times between problems run with and without the material routine quickly falls as the size of the problem increases, approaching the asymptote of one. By running the analyses on several platforms, it was also found that computer architecture has a minor effect on the decay of this ratio. The advance of computing resources to their current status has made it possible to consider using a micromechanical model such as the method of cells in finite element analysis. It has been shown that this is a powerful method in attacking the difficult problem of composite

analysis. As computational power continues to improve, it will be possible to introduce more complicated micromechanical models using the framework established here.

6.2 Future Work

6.2.1 Optimization

During the writing of the user material model, more emphasis was placed on developing a working material model than creating an extremely streamlined routine for computational speed. Reorganization of the subroutine with optimization in mind will hopefully make its use less expensive. The single largest factor in the speed of the routine remains the solution for the internal displacements of the representative volume cell. It is possible that the gaussian elimination may be completely avoided in the material stiffness calculation by instead imposing unit strains and volume averaging the resultant subcell element stresses. The expense of the micromechanical analysis would consequently be greatly reduced.

6.2.2 Other Composite Types

The method of cells has been developed by Aboudi for short fiber and particulate composites in addition to the continuous fiber version used here. The extension of the current user material model to include these composite types simply involves the addition of more subcell elements into the representative volume cell at each material integration point. The generality of the material model will be greatly increased once this addition has been completed.

6.2.3 Constituent Models

The most important applications for the micromechanical material model in finite elements come in area of analyzing composites with nonelastic constituent materials. Such constituent material models as elastic-plastic, viscoelastic, and viscoelastic-

viscoplastic are intended for the near future. In addition, better failure criteria need to be implemented to more appropriately model fiber breakage and matrix cracking. Along with these failure criteria, experimental data must be obtained to allow for comparisons of the results.

6.2.4 Interface/Debonding Models

Perhaps the area in which the most work is needed lies in the development of accurate models for the behavior of the fiber-matrix interface. As was mentioned previously, this is currently a very active field of research, but the development necessitates testing to determine properties for the bonds. The scale of the problem has so far hindered the measurement of these properties.

6.2.5 Statistical Variation Models

A full statistical variation of the fiber and matrix properties as well as the composite geometry was performed by Engelstad and Reddy in [29]. A similar sort of variation may easily be introduced into the method of cells material model. The allowance for property and geometry variation would, in conjunction with the other proposed improvements, provide an extremely general composite material model for use in the finite element method.

Appendix A

Fortran Source Code for User Material Subroutine

```
*user subroutines
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c                this is the program that controls the use of the aboudi cell
c        elements.  it is a small finite element program which meshs the
c        area and then assembles the stiffness matrix.  the internal degrees
c        of freedom are condensed out and then we are left with the desired
c        quantities in terms of the boundary displacements.  a transformation
c        is performed on the remaining stiffness matrix to convert it to a
c        relation between the volumetric stress and the volumetric strain.      10
c        using the element routines, the element stresses are then received
c        and passed back to abaqus along with the stiffness matrix.
c
c                this is set up for abaqus version 5.3 and includes the micro-
c        failure criteria and von mises stress.
c
c                last modified 1/12/94 jpg
c
c        last change--upgrade to version 5.3
c                                                                               20
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
      subroutine umat(stress,statev,ddsdde,sse,spd,scd,
     1 rpl,ddsddt,drplde,drpldt,
     2 stran,dstran,time,dtime,temp,dtemp,predef,dpred,cmname,
     3 ndi,nshr,ntens,nstatv,props,nprops,coords,drot,pnewdt
     4 celent,dfgrd0,dfgrd1,noel,npt,layer,kspt,kstep,kinc)
c
      include 'aba_param.inc'
c
      character*8 cmname                                          30
      dimension stress(ntens),statev(nstatv),
     1 ddsdde(ntens,ntens),ddsddt(ntens),drplde(ntens),
     2 stran(ntens),dstran(ntens),time(2),predef(1),dpred(1),
     3 props(nprops),coords(3),drot(3,3),dfgrd0(3,3),dfgrd1(3,3)
c
      dimension rhs(18),amatrx(18,18),svars(12),
     1crds(3,24),u(18),lflags(4),lm(18)
c
      dimension stiff(72,72),frhs(72)
      dimension astar(48,6),tmp(48,6),tstrn(6)                    40
      dimension ub(48),ubkib(24),ui(24),lhol(4),s(6)
      real*8 volf,volm,d,l1,l2,stren
      real*8 stiffj,stfjj,ttemp,spr1,spr2,oper
      integer i,j,k,m,nel,lstop,jj,ndof,odd,high,li,ii
      integer m1,dofcon,lstopj,lstopk
c
      ndofel = 18
      mcrd = 3
      nnode = 24
      lstop = 24                                                  50
      ndof = 72
      ttemp = temp + dtemp
      lflags(3) = 1
c
c     the user input variables are arranged in the following manner:
c               props(1) = e11 fiber
c               props(2) = e22 fiber
```

```
c               props(3) = e33 fiber
c               props(4) = v12 fiber
c               props(5) = v13 fiber                                    60
c               props(6) = v23 fiber
c               props(7) = g12 fiber
c               props(8) = g13 fiber
c               props(9) = g23 fiber
c               props(10) = alpha11 fiber
c               props(11) = alpha22 fiber
c               props(12) = alpha33 fiber
c               props(13) = e matrix
c               props(14) = poisson's ratio matrix
c               props(15) = shear modulus matrix                        70
c               props(16) = alpha matrix
c               props(17) = spring stiffness in 1 dir
c               props(18) = spring stiffness in 2 dir
c               props(19) = spring stiffness in 3 dir
c               props(20) = reference temperature
c               props(21) = fiber volume fraction
c               props(22) = tensile strength-fiber
c               props(23) = compressive strength-fiber
c               props(24) = ultimate matrix tensile strength
c               props(25) = ultimate matrix compressive strength        80
c               props(26) = ultimate matrix shear strength
c               props(27) = number of aboudi cells to be used in analysis
c                     (this is usually either 4 or 8, with a default of 4)
c                      (right now its really only set up for 4)
c               note:    this program is set up so that the first element is
c      the fiber cell.
c               compute the size of the cells based on the input volume
c      fraction.    it is assumed that the integration point is at the
c      center of the cell arrangement.    (in local coordinates this is
c      (-0.5*d, 0.5*(l1+l2), 0.5*(l1+l2))).                             90
c
       nel = props(27)
       if ((nel.ne.4).and.(nel.ne.8)) then
```

```
            nel = 4
        endif
        if ((props(21).lt.0.0).or.(props(21).gt.1.0)) then
            return
        endif
        volf = props(21)
        volm = 1.0 - props(21)                                          100
        if (nel.eq.4) then
            d = 1.0
            l1 = dsqrt(volf/d)
            rad = dsqrt((4.0*l1*l1) + (4.0*volm))
            l2 = ((-2.0*l1) + rad)/(2.0)
        else
            d = 0.5
            l1 = dsqrt(2.0*volf)
            rad = dsqrt((l1*l1)+(8.0*volm)-4.0)
            l2 = (-1.0*l1+rad)/1.0                                      110
        endif
c
c           loop through the elements to assemble the stiffness matrix.
c
        do 40   j = 1,6
        do 20 i = 1,48
            astar(i,j) = 0.0
 20     continue
 40     continue
        do 80 j = 1, 72                                                 120
            frhs(j) = 0.0
        do 60 i = 1, 72
            stiff(i,j) = 0.0
 60     continue
 80     continue
c
        do 260 m = 1, nel
            m1 = m - 1
c
```

```
c               set up the matrix of coordinates                        130
c
        if (m.lt.5) then
                crds(1,1+m1*6) = 0.0
                crds(1,2+m1*6) = -d
        do 100 i = 3,6
                crds(1,i+m1*6) = -0.5*d
 100    continue
        else
                crds(1,1+m1*6) = -d
                crds(1,2+m1*6) = -2.0*d                                 140
        do 120 i = 3,6
                crds(1,i+m1*6) = -1.5*d
 120    continue
        endif
        if ((mod(m,2)).gt.0) then
                odd = 0
        else
                odd = 1
        endif
        if ((m.eq.1).or.(m.eq.2).or.(m.eq.5).or.(m.eq.6)) then          150
                high = 0
        else
                high = 1
        endif
        crds(2,1+m1*6) = 0.5*l1 + odd*(0.5*l1+0.5*l2)
        crds(2,2+m1*6) = crds(2,1+m1*6)
        crds(2,5+m1*6) = crds(2,1+m1*6)
        crds(2,6+m1*6) = crds(2,1+m1*6)
        crds(2,3+m1*6) = l1 + odd*l2
        crds(2,4+m1*6) = odd*l1                                         160
        crds(3,1+m1*6) = 0.5*l1 + high*(0.5*l1+0.5*l2)
        do 140 i = 2,4
                crds(3,i+m1*6) = crds(3,1+m1*6)
 140    continue
        crds(3,5+m1*6) = l1 + high*(l2)
```

```
      crds(3,6+m1*6) = high*l1
c
c             get the stiffness matrix for this element.
c
      do 180 i = 1,18                                                    170
             u(i) = 0.0
 180  continue
      jtype = 1
      jelem = 0
      call uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2m,lm,nprops,ttemp)
c
c             assemble the global stiffness matrix and force vector
c                                                                        180
      do 220 j = 1,18
      do 200 i = 1,18
             stiff(lm(i),lm(j))=stiff(lm(i),lm(j))+amatrx(i,j)
 200  continue
             frhs(lm(j))=frhs(lm(j))+rhs(j)
 220  continue
c
c             set up the a matrix, which converts the boundary
c      displacements into strains and the boundary forces into stresses.
c      this a matrix is multiplied times the stiffness matrix to          190
c      provide the final material stiffness matrix.    this final stiffness
c      matrix relates the volumetric strains to the volumetric stresses.
c
      do 240 i = 1,18,3
      if (lm(i).gt.lstop) then
             li = lm(i) - lstop
             ii = 1 + i/3
             astar(li,1) = crds(1,ii+m1*6)
             astar(li,4) = 0.5*crds(2,ii+m1*6)
             astar(li,5) = 0.5*crds(3,ii+m1*6)                           200
             astar(li+1,2) = crds(2,ii+m1*6)
```

```
              astar(li+1,4) = 0.5*crds(1,ii+m1*6)
              astar(li+1,6) = 0.5*crds(3,ii+m1*6)
              astar(li+2,3) = crds(3,ii+m1*6)
              astar(li+2,5) = 0.5*crds(1,ii+m1*6)
              astar(li+2,6) = 0.5*crds(2,ii+m1*6)
       endif
       if (m.eq.1) then
              dofcon = lm(9)
       endif                                                          210
 240   continue
 260   continue
c
c             put in spring elements (set up for only 4 right now)
c
       do 340 m = 1,4
       jtype = 2
       do 300 i = 1,6
              u(i) = 0.0
 300   continue                                                       220
       call uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2m,lm,nprops,ttemp)
c
c             put into the global stiffness matrix
c
       do 320 j = 1, 6
       do 310 i = 1, 6
              stiff(lm(i),lm(j))=stiff(lm(i),lm(j))+amatrx(i,j)
 310   continue                                                       230
              frhs(lm(j))=frhs(lm(j))+rhs(j)
 320   continue
 340   continue
c
c             constrain out dof 3 for node 3.    this is done to constrain
c      out the zero-energy modes.    instead of eliminating the dof outright,
c      the row and column are set to 0 with a 1 on the diagonal.
```

```
c
      do 420 i = 1,ndof
            stiff(i,dofcon) = 0.0                                          240
 420  continue
      do 440 i = 1,ndof
            stiff(dofcon,i) = 0.0
 440  continue
      stiff(dofcon,dofcon) = 1.0
      frhs(dofcon) = 0.0
c
c           now that we have the global stiffness matrix and the force
c     vector assembled, condense out the internal degrees of freedom by
c     gauss elimination.                                                   250
c
c
      do 560 j = 1, lstop
            stiffj = stiff(j,j)
      cmax = 0.0
      do 500 i = j, ndof
            stiff(j,i) = stiff(j,i)/stiffj
 500  continue
      frhs(j) = frhs(j)/stiffj
      do 540 jj = j+1, ndof                                                260
            stfjj = stiff(jj,j)
      do 520 i = j, ndof
            stiff(jj,i)=stiff(jj,i)-stiff(j,i)*stfjj
 520  continue
      frhs(jj) = frhs(jj)-frhs(j)*stfjj
 540  continue
 560  continue
c
c           develop the final material stiffness matrix, c.    this is
c     equal to the triple product:    a * stiff(reduced) * a.              270
c
      do 620 j = 1,6
      do 600 i = 1,48
```

```
                tmp(i,j) = 0.0
 600    continue
 620  continue
        do 680 k = 1,48
                lstopk = lstop + k
        do 660 j = 1,6
        do 640 i = 1,48                                                    280
                tmp(i,j)=tmp(i,j)+stiff(lstop+i,lstopk)*astar(k,j)
 640    continue
 660    continue
 680    continue
        do 720 j = 1,6
        do 700 i = 1,6
                ddsdde(i,j) = 0.0
 700    continue
 720    continue
        do 780 k = 1,48                                                    290
        do 760 j = 1,6
        do 740 i = 1,6
                ddsdde(i,j)=ddsdde(i,j)+astar(k,i)*tmp(k,j)
 740    continue
 760    continue
 780    continue
c
c               compute the boundary displacements from the volumetric
c       strains.
c                                                                          300
        do 800 i = 1, ntens
                tstrn(i) = stran(i) + dstran(i)
 800    continue
        do 820 i = 1, 48
                ub(i) = 0.0
                frhs(i+lstop) = -frhs(i+lstop)
 820    continue
        do 860 j = 1, 6
        do 840 i = 1, 48
```

```
              ub(i) = ub(i) + astar(i,j)*tstrn(j)                    310
 840    continue
 860    continue
c
c              compute the boundary forces.
c
       do 900 j = 1, 48
              lstopj = lstop + j
       do 880 i = 1+lstop, ndof
              frhs(i)=frhs(i)+stiff(i,lstopj)*ub(j)
 880    continue                                                       320
 900    continue
c
c              compute the volumetric stresses from the boundary forces
c
       do 940 i = 1, 6
              stress(i) = 0.0
       do 920 j = 1, 48
              stress(i) = stress(i) + astar(j,i)*frhs(j+lstop)
 920    continue
 940    continue                                                       330
c
c              compute the internal displacements (the vector ri).
c      this is done by backsubstituting into the top left-hand corner
c      of the stiff matrix.
c
       do 1000 i = 1, lstop
              ubkib(i) = 0.0
 1000   continue
       do 1040 j = 1, 48
              lstopj = lstop + j                                      340
       do 1020 i = 1, lstop
              ubkib(i)=ubkib(i)+stiff(i,lstopj)*ub(j)
 1020   continue
 1040   continue
       do 1060 i = 1, lstop
```

```
              ui(i) = frhs(i) - ubkib(i)
 1060   continue
        ui(lstop) = ui(lstop)/stiff(lstop,lstop)
        do 1100 i = lstop-1,1,-1
        do 1080 j = i+1, lstop                                          350
              ui(i) = ui(i)-ui(j)*stiff(i,j)
 1080   continue
 1100   continue
c
c               call the element routines again passing in the
c       displacements to retrieve the element stresses and strains.
c
        do 1300 m = 1, nel
        m1 = m - 1
        if (m.eq.1) then                                                360
              lhol(1) = 1
              lhol(2) = 3
              lhol(3) = 2
              lhol(4) = 4
        else if (m.eq.2) then
              lhol(1) = 3
              lhol(2) = 1
              lhol(3) = 2
              lhol(4) = 4
        else if (m.eq.3) then                                           370
              lhol(1) = 1
              lhol(2) = 3
              lhol(3) = 4
              lhol(4) = 2
        else if (m.eq.4) then
              lhol(1) = 3
              lhol(2) = 1
              lhol(3) = 4
              lhol(4) = 2
        endif                                                           380
        do 1200 i = 1, 6
```

```
              u(i) = ub(i+m1*12)
 1200   continue
        do 1220 i = 1, 4
        if (lhol(i).lt.3) then
              u((i-1)*3+7) = ui(1+(lhol(i)-1)*3+m1*6)
              u((i-1)*3+8) = ui(2+(lhol(i)-1)*3+m1*6)
              u((i-1)*3+9) = ui(3+(lhol(i)-1)*3+m1*6)
        else
              u((i-1)*3+7) = ub(7+(lhol(i)-3)*3+m1*12)                    390
              u((i-1)*3+8) = ub(8+(lhol(i)-3)*3+m1*12)
              u((i-1)*3+9) = ub(9+(lhol(i)-3)*3+m1*12)
        endif
 1220   continue
              jtype=1
              jelem = 1
        call uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2m,lm,nprops,ttemp)
c                                                                         400
c             the state variables are in the order of:
c                     statev(1-6) -- stresses in the fiber cell
c                     statev(7-12) -- strains in the fiber cell
c                     statev(13-18) -- stresses in matrix cell 1
c                     statev(19-24) -- strains in matrix cell 1
c                     statev(25-30) -- stresses in matrix cell 2
c                     statev(31-36) -- strains in matrix cell 2
c                     statev(37-42) -- stresses in matrix cell 3
c                     statev(43-48) -- strains in matrix cell 3
c                     statev(49) fiber failure criterion                  410
c                     statev(50) matrix failure criterion
c                     statev(51) von mises equivalent stress--fiber
c                     statev(52-54) von mises stress-matrix cells
c
        do 1240 i = 1,12
              statev(i+m1*12) = svars(i)
 1240   continue
```

```
 1300   continue
c
c                       micro-failure criteria proposed by aboudi                  420
c               the criterion for the fiber compares the stress in the
c       fiber to either the tensile or compressive strength, depending
c       upon the state of stress.    a result less than one
c               the criterion for the matrix looks at the maximum
c       principal stress in the plane perpendicular to the axis of
c       the fiber and the axial shear stresses, s12 and s13.    the
c       principal stress is squared and then divided by the square of the
c       the ultimate tensile strength of the matrix.    the axial shear
c       stresses are squared and added together.    this sum is then divided
c       by the square of the ultimate shear strength of the matrix.    these    430
c       two quotients are then added, with a sum of less than one
c       indicating that failure has not occurred in the matrix.
c
        if (statev(1).gt.0.0) then
                statev(49) = statev(1)/props(22)
        else
                statev(49) = -statev(1)/props(23)
        endif
        statev(50) = 0.0
c                                                                                  440
c               the von mises equivalent stress--fiber cell
c
        statev(51) = dsqrt((0.5*((statev(1)-statev(2))**2+
     1(statev(2)-statev(3))**2+(statev(3)-statev(1))**2))+
     2(3.0*(statev(4)*statev(4)+statev(5)*statev(5)
     3+statev(6)*statev(6))))
        do 1340 i = 1,3
        do 1320 j = 1,6
                s(j) = statev(i*12+j)
 1320   continue                                                                   450
        oper = dsqrt(((s(2)-s(3))/2.0)**2+(s(6)*s(6)))
        spr1 = (s(2)+s(3))/2.0 + oper
        spr2 = (s(2)+s(3))/2.0 - oper
```

```
        if (abs(spr2).gt.abs(spr1)) then
                spr1 = spr2
        endif
        if (spr1.le.0.0) then
                stren = props(25)
        else
                stren = props(24)                                        460
        endif
        oper = ((spr1*spr1)/(stren*stren))+
     +(((s(4)*s(4))+(s(5)*s(5)))/(props(26)*props(26)))
        if (oper.gt.statev(50)) then
                statev(50) = oper
        endif
c
c               the von mises equivalent stress--matrix cells
c
        statev(i+51) = dsqrt((0.5*((s(1)-s(2))**2+               470
     1(s(2)-s(3))**2+(s(3)-s(1))**2))+
     2(3.0*(s(4)*s(4)+s(5)*s(5)+s(6)*s(6))))
 1340   continue
c
 5000   format (3f10.5)
 5010   format (12f11.0)
 5020 format (12f8.0)
 5030 format (6g10.3)
 5040 format (3f12.5)
 5050 format (6f10.5)                                                    480
        return
        end
c
        subroutine uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2m,lm,nprops,ttemp)
c
      implicit real*8(a-h,o-z)
      dimension rhs(ndofel),amatrx(ndofel,ndofel),svars(1),
```

```
     1props(1),crds(mcrd,nnode),u(ndofel),                                    490
     2predef(1),lflags(4),lm(ndofel)
c
        if (jtype.eq.1) then
                call aboudi(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2m,lm,nprops,ttemp)
        else if (jtype.eq.2) then
                call damage(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2m,lm,nprops,ttemp)                                                      500
        endif
c
        return
        end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c                                       aboudi element
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        subroutine aboudi(rhs,amatrx,svars,ndofel,props,crds,mcrd,           510
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2m,lm,nprops,ttemp)
c
      implicit real*8(a-h,o-z)
      dimension rhs(ndofel),amatrx(ndofel,ndofel),svars(1),
     1props(1),crds(mcrd,nnode),u(ndofel),
     2predef(1),lflags(4),lm(ndofel)
c
        dimension b(6,18)
        dimension strain(6),tstr(6),thstr(6),stress(6)                       520
        dimension temp(18,6),d(6,6),abar(18,18)
        real*8 d1,d2,d3,dtemp,prod
        integer i,j,k,m1
        dimension lhol(4)
c
```

```
c              6 node aboudi cell model element for abaqus
c       the vectors are arranged in the following manner:
c                       dof 1 at node 1,
c                       dof 2 at node 1,
c                       dof 3 at node 1,                                    530
c                       dof 1 at node 2,
c                       dof 2 at node 2,
c                       dof 3 at node 2, and so on.
c
c       svars(1) to svars(6) = stresses at integration points
c       svars(7) to svars(12) = strains at integration points
c
c              the program is called in parts.    the first part calls the element
c       to obtain the element stiffness matrix and assemble the global stiffness
c       matrix.    in the second part, the element stresses and strains are      540
c       returned given the displacement field, or state of strain, that exists
c       at the boundaries.
c
c              initialize the b matrix.
c
       m1 = m - 1
       d1 = 1.0/(abs(crds(1,1+m1*6) - crds(1,2+m1*6)))
       d2 = 1.0/(abs(crds(2,3+m1*6) - crds(2,4+m1*6)))
       d3 = 1.0/(abs(crds(3,5+m1*6) - crds(3,6+m1*6)))
       prod = 1.0/(d1*d2*d3)                                                  550
c
       do 20 k = 1,18
       do 20 j = 1,6
              b(j,k) = 0.0
 20    continue
c
c      b matrix
c
       b(1,1) = d1
       b(1,4) = -d1                                                           560
       b(2,8) = d2
```

```
      b(2,11) = -d2
      b(3,15) = d3
      b(3,18) = -d3
      b(4,2) = d1
      b(4,5) = -d1
      b(4,7) = d2
      b(4,10) = -d2
      b(5,3) = d1
      b(5,6) = -d1                                                    570
      b(5,13) = d3
      b(5,16) = -d3
      b(6,9) = d2
      b(6,12) = -d2
      b(6,14) = d3
      b(6,17) = -d3
c
c             determine the d matrix (constitutive)
c
      call orthotropic(props,nprops,d,m)                              580
c
c             route the subroutine to the correct parts, depending upon
c     what is being asked for.
c
      if (jelem .eq.  1) go to 200
c
c             initialize the amatrix and check to see if the mass
c     matrix is desired.    if the mass matrix is desired
c     (lflags(3) = 3), then the amatrix is initialized to zero
c     and the subroutine returns control to abaqus.                   590
c
      do 40 j = 1,18
      do 40 i = 1,18
             amatrx(i,j) = 0.0
 40   continue
      if (lflags(3).eq.3) go to 1000
c
```

```
c               calculate the stiffness (amatrx) matrix.    this calculation is
c       the triple product:    b * d * b
c                                                                          600
        do 60 j = 1,6
        do 60 i = 1,18
                temp(i,j) = 0.0
 60     continue
        do 80 k = 1,6
        do 80 j = 1,6
        do 80 i = 1,18
                temp(i,j) = temp(i,j) + b(k,i) * d(k,j)
 80     continue
c                                                                          610
        do 100 k = 1,6
        do 100 j = 1,18
        do 100 i = 1,18
                amatrx(i,j) = amatrx(i,j) + prod * temp(i,k) * b(k,j)
 100    continue
c
c               complete the lm array
c                       lhol(1) = node 3
c                       lhol(2) = node 4
c                       lhol(3) = node 5                                   620
c                       lhol(4) = node 6
c
        if (m.eq.1) then
                lhol(1) = 1
                lhol(2) = 3
                lhol(3) = 2
                lhol(4) = 4
        else if (m.eq.2) then
                lhol(1) = 3
                lhol(2) = 1                                                630
                lhol(3) = 2
                lhol(4) = 4
        else if (m.eq.3) then
```

```
              lhol(1) = 1
              lhol(2) = 3
              lhol(3) = 4
              lhol(4) = 2
       else if (m.eq.4) then
              lhol(1) = 3
              lhol(2) = 1                                              640
              lhol(3) = 4
              lhol(4) = 2
       endif
       do 120 i = 1,3
              lm(i) = i+12+(m*12)
              lm(i+3) = i+15+(m*12)
 120   continue
       do 160 j = 1,4
       do 140 i = 1,3
              if (lhol(j) .eq.  1) then                                650
                     lm(i+3+j*3) = i+(m1*6)
              else if (lhol(j) .eq.  2) then
                     lm(i+3+j*3) = i+3+(m1*6)
              else if (lhol(j) .eq.  3) then
                     lm(i+3+j*3) = i+18+(m*12)
              else
                     lm(i+3+j*3) = i+21+(m*12)
              endif
 140   continue
 160   continue                                                         660
c
c             calculate the total strain from the displacements
c             (using strain = b * displacements)
c
 200   do 220 i = 1,6
              tstr(i) = 0.0
 220   continue
       do 240 j = 1,18
              rhs(j) = 0.0
```

```
      do 240 i = 1,6                                                    670
            tstr(i) = tstr(i) + b(i,j)*u(j)
 240  continue
c
c           calculate the thermal strains in the element
c
      dtemp = ttemp - props(20)
      do 260 i = 1,6
            thstr(i) = 0.0
 260  continue
c                                                                       680
      if (m.eq.1) then
      do 280 j = 1,3
            thstr(j) = props(j+9) * dtemp
 280  continue
      else
      do 290 j = 1,3
            thstr(j) = props(16) * dtemp
 290  continue
      endif
c                                                                       690
c           subtract the thermal strains out from the total strain
c
      do 300 i = 1,6
            strain(i) = 0.0
            strain(i) = tstr(i) - thstr(i)
 300  continue
c
c           calculate the stresses from the strain
c
      do 310 i = 1,6                                                    700
            stress(i) = 0.0
 310  continue
      do 320 j = 1,6
      do 320 i = 1,6
            stress(i) = stress(i) + d(i,j) * strain(j)
```

```
 320    continue
c
c               compute the force vector, rhs.
c
       if (lflags(3).eq.2) go to 1000                                    710
       do 360 j = 1,6
       do 340 i = 1,18
              rhs(i) = rhs(i) - prod * b(j,i) * stress(j)
 340   continue
 360   continue
c
c               set state variables
c
       do 380 i = 1,6
              svars(i) = stress(i)                                       720
              svars(i+6) = strain(i)
 380   continue
 1000  return
 5000  format (3f10.5)
 5010 format (18f6.0)
       end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c               subroutine orthotropic which determines the d (constitutive)
c      matrix for a general orthotropic material.                         730
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       subroutine orthotropic(props,nprops,d,m)
c
       implicit real*8(a-h,o-z)
       dimension props(nprops),d(6,6)
       integer nprops,m
c
       real*8 det,v12,v13,v23,v21,v31,v32
       real*8 e11,e22,e33,g12,g13,g23                                    740
       integer i,j
```

```
      if (m.eq.1) then
              e11 = props(1)
              e22 = props(2)
              e33 = props(3)
              v12 = props(4)
              v13 = props(5)
              v23 = props(6)
              v21 = v12*(props(2)/props(1))
              v31 = v13*(props(3)/props(1))                              750
              v32 = v23*(props(3)/props(2))
      det = 1 - (v12*v21) - (v23*v32) - (v13*v31) -
     + (2.0*v12*v23*v31)
              g12 = props(7)
              g13 = props(8)
              g23 = props(9)
      else
              e11 = props(13)
              e22 = e11
              e33 = e11                                                  760
              v12 = props(14)
              v13 = v12
              v23 = v12
              g12 = props(15)
              g13 = g12
              g23 = g12
              v21 = v12
              v31 = v13
              v32 = v23
      det = 1 - (v12*v21) - (v23*v32) - (v13*v31) -                      770
     + (2.0*v12*v23*v31)
      endif
c
      do 20 j = 1,6
      do 20 i = 1,6
              d(i,j) = 0.0
 20   continue
```

```
c
        d(1,1) = (1 - (v23*v32))*e11/det
        d(1,2) = (v12 + (v13*v32))*e22/det                                    780
        d(1,3) = (v13 + (v23*v12))*e33/det
        d(2,2) = (1 - (v31*v13))*e22/det
        d(2,3) = (v23 + (v21*v13))*e33/det
        d(3,3) = (1 - (v12*v21))*e33/det
        d(2,1) = d(1,2)
        d(3,1) = d(1,3)
        d(3,2) = d(2,3)
        d(4,4) = g12
        d(5,5) = g13
        d(6,6) = g23                                                          790
c
        return
        end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c                                       damage element
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        subroutine damage(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,          800
     2m,lm,nprops,ttemp)
c
      implicit real*8(a-h,o-z)
      dimension rhs(ndofel),amatrx(ndofel,ndofel),svars(1),
     1props(1),crds(mcrd,nnode),u(ndofel),
     2predef(1),lflags(4),lm(ndofel)
c
        dimension lhol(2)
        integer i,j
c                                                                             810
c       the vectors are arranged as follows:
c               dof 1 at node 1
c               dof 2 at node 1
```

```
c               dof 3 at node 1
c               dof 1 at node 2
c               dof 2 at node 2
c               dof 3 at node 2
c
c       props(17) = ea/l in the 1 direction
c       props(18) = ea/l in the 2 direction                                    820
c       props(19) = ea/l in the 3 direction
c
c       initialize and construct the amatrix
c
        do 10 i = 1, 6
                rhs(i) = 0.0
        do 10 j = 1, 6
                amatrx(i,j) = 0.0
 10     continue
        if (lflags(3).eq.3) go to 1000                                         830
c
        amatrx(1,1) = props(17)
        amatrx(1,4) = -props(17)
        amatrx(2,2) = props(18)
        amatrx(2,5) = -props(18)
        amatrx(3,3) = props(19)
        amatrx(3,6) = -props(19)
        amatrx(4,1) = amatrx(1,4)
        amatrx(4,4) = amatrx(1,1)
        amatrx(5,2) = amatrx(2,5)                                              840
        amatrx(5,5) = amatrx(2,2)
        amatrx(6,3) = amatrx(3,6)
        amatrx(6,6) = amatrx(3,3)
c
c               compute the lm array
c
        if (m .eq.  1) then
                lhol(1) = 0
                lhol(2) = 6
```

```
      else if (m.eq.2) then                                          850
             lhol(1) = 3
             lhol(2) = 15
      else if (m.eq.3) then
             lhol(1) = 9
             lhol(2) = 21
      else
             lhol(1) = 12
             lhol(2) = 18
      endif
      do 20 i = 1,3                                                  860
             lm(i) = i + lhol(1)
             lm(i+3) = i + lhol(2)
 20   continue
c
c     compute the force vector, rhs
c
      if (lflags(3).eq.2) go to 1000
      do 30 i = 1, 6
      do 30 j = 1, 6
             rhs(i) = rhs(i) - amatrx(i,j) * u(j)                    870
 30   continue
c
 1000 return
      end
```

Appendix B

Fortran Source Code for Nonlinear User Material Subroutine with Damage Interface Elements

---

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c                    this is the program that controls the use of the aboudi cell
c       elements.  it is a small finite element program which meshs the
c       area and then assembles the stiffness matrix.  the internal degrees
c       of freedom are condensed out and then we are left with the desired
c       quantities in terms of the boundary displacements.  a transformation
c       is performed on the remaining stiffness matrix to convert it to a
c       relation between the volumetric stress and the volumetric strain.
c       using the element routines, the element stresses are then received
c       and passed back to abaqus along with the stiffness matrix.
c
c     axial fiber fracture, debonding, and matrix cracking have been added.
c     the model may only be used for fully 3d analyses.
c     modified by r. w. macek (1-18-94).
c
```

```
c                  note:  this is set up for abaqus version 5.3
c
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine umat(stress,statev,ddsdde,sse,spd,scd,
     1 rpl,ddsddt,drplde,drpldt,
     2 stran,dstran,time,dtime,temp,dtemp,predef,dpred,cmname,
     3 ndi,nshr,ntens,nstatv,props,nprops,coords,drot,pnewdt,
     4 celent,dfgrd0,drgrd1,noel,npt,layer,kspt,kstep,kinc)
c
c     when actually running abaqus, take out the implicit line
c     and uncomment this include statement.  also, change the
c     input variable declarations below.
c
      include 'aba_param.inc'
c     implicit real*8(a-h,o-z)
c
      character*8 cmname
      dimension stress(ntens),statev(nstatv),
     1 ddsdde(ntens,ntens),ddsddt(ntens),drplde(ntens),
     2 stran(ntens),dstran(ntens),time(2),predef(1),dpred(1),
     4 props(nprops),coords(3),drot(3,3),dfgrd0(3,3),dfgrd1(3,3)
c
      dimension rhs(18),amatrx(18,18),svars(14),
     1crds(3,28),u(18),lflags(4),lm(72)
c
      dimension stiff(84,84),frhs(84),s(6)
      dimension astar(48,6),tmp(48,6),tstrn(6)
      dimension ub(48),ubkib(36),ui(36),lhol(4)
      dimension lms(48),area(8),h(8)
      dimension deltau(84),dofcon(4)
      real*8 volf,volm,d,l1,l2,norm,mmu,mgamy,mgamf
      real*8 stiffj,stfjj,ttemp,spr1,spr2,oper
      integer i,j,k,m,nel,lstop,jj,ndof,odd,high,li,ii
      integer dofcon,lstopj,lstopk,iter,m1st6,m1st18
c
```

```
      data lm/25,26,27,40,41,42,1,2,3,43,44,45,4,5,6,46,47,48,
     +        28,29,30,52,53,54,55,56,57,7,8,9,10,11,12,58,59,60,
     +        31,32,33,64,65,66,13,14,15,67,68,69,70,71,72,16,17,18,
     +        34,35,36,76,77,78,79,80,81,19,20,21,82,83,84,22,23,24/

c
      data lms/2,1,3,8,7,9,
     +        6,4,5,18,16,17,                                          60
     +        25,26,27,37,38,39,
     +        12,10,11,24,22,23,
     +        14,13,15,20,19,21,
     +        28,29,30,49,50,51,
     +        31,32,33,61,62,63,
     +        34,35,36,73,74,75/
      data ncon/1/
      data dofcon/3,0,0,0/
         ndofel = 18
         mcrd = 3                                                     70
         nnode = 28
         lstop = 36
         ndof = 84
         ttemp = temp + dtemp
         lflags(3) = 1
     pnewdt=1.5
c
c        the user input variables are arranged in the following manner:
c                  props(1) = e11 fiber
c                  props(2) = e22 fiber                              80
c                  props(3) = e33 fiber
c                  props(4) = v12 fiber
c                  props(5) = v13 fiber
c                  props(6) = v23 fiber
c                  props(7) = g12 fiber
c                  props(8) = g13 fiber
c                  props(9) = g23 fiber
c                  props(10) = alpha11 fiber
```

```
c               props(11) = alpha22 fiber
c               props(12) = alpha33 fiber                                    90
c               props(13) = e matrix
c               props(14) = poisson's ratio matrix
c               props(15) = shear modulus matrix
c               props(16) = alpha matrix
c               props(17) = mu, the effective spring friction coefficient
c               props(18) = e, the effective spring young's modulus
c               props(19) = g, the effective spring shear modulus
c               props(20) = reference temperature
c               props(21) = fiber volume fraction
c               props(22) = tensile strength-fiber                           100
c               props(23) = compressive strength-fiber
c               props(24) = ultimate matrix tensile strength
c               props(25) = ultimate matrix shear strength
c               props(26) = convergence tolerance
c               props(27) = number of aboudi cells to be used in analysis
c                         (this is usually either 4 or 8, with a default of 4)
c                         (right now its really only set up for 4)
c               note:  this program is set up so that the first element is
c     the fiber cell.
c               props(28) = gamma_y, the yield strain of the spring          110
c               props(29) = gamma_f, the failure strain of the spring
c               props(30) = h, the bond thickness for the springs
c               props(31) = ultimate matrix compressive strength
c
c               compute the size of the cells based on the input volume
c     fraction.  it is assumed that the integration point is at the
c     center of the cell arrangement.  (in local coordinates this is
c     (-0.5*d, 0.5*(l1+l2), 0.5*(l1+l2))).
c
c               this routine contains a nonlinear solver using newton iteration 120
c     on the displacements.  the displacements internal to the aboudi cell
c     are stored in state variables 51-86 as they are needed for the next
c     time increment.
c
```

```
      nel = 4
      if ((props(21).lt.0.0).or.(props(21).gt.1.0)) then
            return
      endif
      volf = props(21)
      volm = 1.0 - props(21)                                          130
      if (nel.eq.4) then
            d = 1.0
            l1 = dsqrt(volf/d)
            rad = dsqrt((4.0*l1*l1) + (4.0*volm))
            l2 = ((-2.0*l1) + rad)/(2.0)
      else
            d = 0.5
            l1 = dsqrt(2.0*volf)
            rad = dsqrt((l1*l1)+(8.0*volm)-4.0)
            l2 = (-1.0*l1+rad)/1.0                                    140
      endif
c
c           set svars(14), which is the initial area of the bond, and
c     initialize svars(13).
c
      area(1)=l1
      area(2)=area(1)
      area(3)=l1*l1
      area(4)=l2
      area(5)=area(4)                                                 150
      area(6)=l1*l2
      area(7)=area(6)
      area(8)=l2*l2
      h(1)=props(30)
      h(2)=h(1)
c     h(3)=.0001
      h(3)=props(30)
      h(4)=h(3)
      h(5)=h(3)
      h(6)=h(3)                                                       160
```

```
      h(7)=h(3)
      h(8)=h(3)
c
c     zero statev at start of analysis
c
      if(kstep.le.1.and.kinc.le.1) then
      do 5 i=1,nstatv
      statev(i)=0.
 5    continue
      end if                                                          170
c
c     compute fracture characteristics for fiber and matrix
c
      fibermu=1.0e6
      fibery=fibermu*props(22)/props(7)
      fiberf=1.01*fibery
c
      mmu=props(25)/props(24)
      mgamy=props(25)/props(15)
      mgamf=1.01*mgamy                                                180
c
c                 loop through the elements to assemble the stiffness matrix.
c
        do 20   j = 1,6
        do 10 i = 1,48
                astar(i,j) = 0.0
 10     continue
 20     continue
c
        do 130 m = 1, nel                                             190
                m1st6 = (m-1)*6
                m1st18 = (m-1)*18
c
c               set up the matrix of coordinates
c
        if (m.lt.5) then
```

```
            crds(1,1+m1st6) = 0.0
            crds(1,2+m1st6) = -d
      do 50 i = 3,6
            crds(1,i+m1st6) = -0.5*d                                    200
50    continue
      else
            crds(1,1+m1st6) = -d
            crds(1,2+m1st6) = -2.0*d
      do 60 i = 3,6
            crds(1,i+m1st6) = -1.5*d
60    continue
      endif
      if ((mod(m,2)).gt.0) then
            odd = 0                                                     210
      else
            odd = 1
      endif
      if ((m.eq.1).or.(m.eq.2).or.(m.eq.5).or.(m.eq.6)) then
            high = 0
      else
            high = 1
      endif
      crds(2,1+m1st6) = 0.5*l1 + odd*(0.5*l1+0.5*l2)
      crds(2,2+m1st6) = crds(2,1+m1st6)                                 220
      crds(2,5+m1st6) = crds(2,1+m1st6)
      crds(2,6+m1st6) = crds(2,1+m1st6)
      crds(2,3+m1st6) = l1 + odd*l2
      crds(2,4+m1st6) = odd*l1
      crds(3,1+m1st6) = 0.5*l1 + high*(0.5*l1+0.5*l2)
      do 70 i = 2,4
            crds(3,i+m1st6) = crds(3,1+m1st6)
70    continue
      crds(3,5+m1st6) = l1 + high*(l2)
      crds(3,6+m1st6) = high*l1                                         230
c
c           set up the a matrix, which converts the boundary
```

```
c       displacements into strains and the boundary forces into stresses.
c       this a matrix is multiplied times the stiffness matrix to
c       provide the final material stiffness matrix.  this final stiffness
c       matrix relates the volumetric strains to the volumetric stresses.
c
        do 120 i = (1+(m1st18)),(18+(m1st18)),3
      nstar=lm(i)/3+1
      nflag=0                                                          240
      if(nstar.ge.9.and.nstar.le.12) nflag=1
         if (lm(i).gt.lstop.or.nflag.eq.1) then
                  li = lm(i) - lstop
                  ii = 1 + i/3
            if(nstar.eq.9) li=37-lstop
            if(nstar.eq.10) li=49-lstop
            if(nstar.eq.11) li=61-lstop
            if(nstar.eq.12) li=73-lstop
                  astar(li,1) = crds(1,ii)
                  astar(li,4) = 0.5*crds(2,ii)                         250
                  astar(li,5) = 0.5*crds(3,ii)
                  astar(li+1,2) = crds(2,ii)
                  astar(li+1,4) = 0.5*crds(1,ii)
                  astar(li+1,6) = 0.5*crds(3,ii)
                  astar(li+2,3) = crds(3,ii)
                  astar(li+2,5) = 0.5*crds(1,ii)
                  astar(li+2,6) = 0.5*crds(2,ii)
         endif
 120     continue
 130     continue                                                      260
c
c                 compute the boundary displacements
c
         do 140 i = 1, ntens
                  tstrn(i) = stran(i) + dstran(i)
 140     continue
         do 150 i = 1, 48
                  ub(i) = 0.0
```

```
150     continue
        do 170 j = 1, 6                                                    270
        do 160 i = 1, 48
                ub(i) = ub(i) + astar(i,j)*tstrn(j)
160     continue
170     continue
c
c       set up the displacement vector using the current boundary
c       displacements and the internal displacements from the previous
c       time step.
c
        do 180 i = 1,lstop                                                 280
                ui(i) = statev(i+50)
                deltau(i) = statev(i+50)
180     continue
        do 190 i = lstop+1,84
                deltau(i) = ub(i-lstop)
190     continue
c
c     initially set displacements for axial damage springs
c     to reflect damage
c                                                                          290
      if(statev(89).lt..99) then
      dam3=h(3)/(1.-statev(89))
      deltau(25)=deltau(37)
      if(statev(1).gt.0.)
     +    deltau(25)=deltau(37)-statev(1)*dam3/props(1)
      deltau(26)=deltau(38)+statev(4)*dam3/props(7)
      deltau(27)=deltau(39)+statev(5)*dam3/props(7)
      end if
c
      if(statev(92).lt..99) then                                           300
      dam6=h(3)/(1.-statev(92))
      deltau(28)=deltau(49)
      if(statev(13).gt.0.)
     +    deltau(28)=deltau(49)-statev(13)*dam6/props(13)
```

```
      deltau(29)=deltau(50)+statev(16)*dam6/props(15)
      deltau(30)=deltau(51)+statev(17)*dam6/props(15)
      end if
c
      if(statev(93).lt..99) then
      dam7=h(3)/(1.-statev(93))                                          310
      deltau(31)=deltau(61)
      if(statev(25).gt.0.)
     +    deltau(31)=deltau(61)-statev(25)*dam7/props(13)
      deltau(32)=deltau(62)+statev(28)*dam7/props(15)
      deltau(33)=deltau(63)+statev(29)*dam7/props(15)
      end if
c
      if(statev(94).lt..99) then
      dam8=h(3)/(1.-statev(94))
      deltau(34)=deltau(73)                                              320
      if(statev(37).gt.0.)
     +    deltau(34)=deltau(73)-statev(37)*dam8/props(13)
      deltau(35)=deltau(74)+statev(40)*dam8/props(15)
      deltau(36)=deltau(75)+statev(41)*dam8/props(15)
      end if
c
c                 begin the iteration scheme
c
        iter = 0
c                                                                        330
c                 get the stiffness matrix for this element.
c
 200    do 220 j = 1, 84
                  frhs(j) = 0.0
        do 210 i = 1, 84
                  stiff(i,j) = 0.0
 210    continue
 220    continue
        do 260 m = 1, nel
                  m1st18= (m-1)*18                                       340
```

```
c
      do 230 i = 1,18
c
c     this will be used later when we make the stiffness matrix in the
c     cells nonlinear also
c
                u(i) = deltau(lm(i+(m1st18)))
c               u(i) = 0.0
230   continue
      jtype = 1                                                    350
      jelem = 0
      call uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
c
c               assemble the global stiffness matrix and force vector
c
      do 250 j = 1,18
      do 240 i = 1,18
                stiff(lm(i+(m1st18)),lm(j+(m1st18))) =             360
     +stiff(lm(i+(m1st18)),lm(j+(m1st18)))+amatrx(i,j)
240   continue
                frhs(lm(j+(m1st18)))=frhs(lm(j+(m1st18)))+rhs(j)
250   continue
260   continue
c
c               put in spring elements (set up for only 8 right now)
c
      do 340 m = 1,8
      m1st6 = (m-1)*6                                              370
      jtype = 2
     jelem = 0
      do 300 i = 1,6
                u(i) = deltau(lms(i+(m1st6)))
300   continue
     svars(13)=statev(86+m)
```

```
      svars(14)=area(m)
      if(m.ge.3) then
      sprop17=props(17)
      sprop18=props(18)                                                 380
      sprop19=props(19)
      sprop28=props(28)
      sprop29=props(29)
      sprop30=props(30)
      if(m.eq.3) then
      props(17)=fibermu
      props(18)=props(1)
      props(19)=props(7)
      props(28)=fibery
      props(29)=fiberf                                                  390
      props(30)=h(i)
      end if
      if(m.ge.4) then
      props(17)=mmu
      props(18)=props(13)
      props(19)=props(15)
      props(28)=mgamy
      props(29)=mgamf
      props(30)=h(i)
      end if                                                            400
      end if
         call uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
      if(m.ge.3) then
      props(17)=sprop17
      props(18)=sprop18
      props(19)=sprop19
      props(28)=sprop28
      props(29)=sprop29                                                 410
      props(30)=sprop30
      end if
```

```
c
c                 put into the global stiffness matrix
c
      do 320 j = 1, 6
      do 310 i = 1, 6
                stiff(lms(i+(m1st6)),lms(j+(m1st6))) =
     +stiff(lms(i+(m1st6)),lms(j+(m1st6)))+amatrx(i,j)
 310  continue                                                    420
                frhs(lms(j+(m1st6)))=frhs(lms(j+(m1st6)))+rhs(j)
 320  continue
 340  continue
c     write(*,*) 'iteration',iter
c     write(*,*) 'residual'
c     write(*,*) (frhs(ii),ii=1,lstop)
c
c       at this point we check for convergence (it is done here
c       because we need to compute the new stiffness matrix based on the
c       internal displacements from the last iteration.)  to check for   430
c       convergence, we take a norm of the internal force vector and
c       compare it to the user input convergence tolerance
c
      nconvrg=0
      if (iter.gt.0) then
      norm = 0.0
      do 600 i = 1,lstop
                norm = max(norm,abs(frhs(i)))
 600  continue
      if (norm.lt.props(26)) nconvrg=1                            440
      endif
c
c                 now that we have the global stiffness matrix and the force
c       vector assembled, condense out the internal degrees of freedom by
c       gauss elimination.
c
c
      do 560 j = 1, lstop
```

```
                stiffj = stiff(j,j)
        cmax = 0.0                                                    450
        do 500 i = j, ndof
                stiff(j,i) = stiff(j,i)/stiffj
500     continue
        frhs(j) = frhs(j)/stiffj
        do 540 jj = j+1, ndof
                stfjj = stiff(jj,j)
        do 520 i = j, ndof
                stiff(jj,i)=stiff(jj,i)-stiff(j,i)*stfjj
520     continue
        frhs(jj) = frhs(jj)-frhs(j)*stfjj                             460
540     continue
560     continue
c
c               compute the internal displacements (the vector ri).
c       this is done by backsubstituting into the top left-hand corner
c       of the stiff matrix.
c
        do 830 i = 1, lstop
                ui(i) = frhs(i)
830     continue                                                      470
        ui(lstop) = ui(lstop)/stiff(lstop,lstop)
        do 850 i = lstop-1,1,-1
        do 840 j = i+1, lstop
                ui(i) = ui(i)-ui(j)*stiff(i,j)
840     continue
850     continue
        iter = iter + 1
      do 860 i=1,lstop
      deltau(i)=deltau(i)+ui(i)
860    continue                                                       480
        if(nconvrg.eq.1) go to 900
        if (iter.le.20) then
                go to 200
        else
```

```
      if(norm.gt.10.*props(26)) pnewdt=.1
        write(*,*) 'step ',kstep,' inc ',kinc
                   write(*,*)
     + ' aboudi routine did not converge for element ',noel
        write(*,*) 'stran ',(stran(i),i=1,6)
        write(*,*) 'dstran ',(dstran(i),i=1,6)                          490
        write(*,*) 'residual ',norm
        write(6,*) 'step ',kstep,' inc ',kinc
                   write(6,*)
     + ' aboudi routine did not converge for element ',noel
        write(6,*) 'stran ',(stran(i),i=1,6)
        write(6,*) 'dstran ',(dstran(i),i=1,6)
        write(6,*) 'residual ',norm
           endif
c
c                  develop the final material stiffness matrix, c.  this is   500
c          equal to the triple product:  a * stiff(reduced) * a.
c
  900      do 920 j = 1,6
           do 910 i = 1,48
                   tmp(i,j) = 0.0
  910      continue
  920   continue
           do 950 k = 1,48
                   lstopk = lstop + k
           do 940 j = 1,6                                               510
           do 930 i = 1,48
                   tmp(i,j)=tmp(i,j)+stiff(lstop+i,lstopk)*astar(k,j)
  930      continue
  940      continue
  950      continue
           do 970 j = 1,6
           do 960 i = 1,6
                   ddsdde(i,j) = 0.0
  960      continue
  970      continue                                                      520
```

```
      do 1000 k = 1,48
      do 990 j = 1,6
      do 980 i = 1,6
            ddsdde(i,j)=ddsdde(i,j)+astar(k,i)*tmp(k,j)
980   continue
990   continue
1000  continue
c
c           compute the volumetric stresses from the boundary forces
c                                                                    530
      do 1020 i = 1, 6
            stress(i) = 0.0
      do 1010 j = 1, 48
            stress(i) = stress(i) - astar(j,i)*frhs(j+lstop)
1010  continue
1020  continue
c
c     call the aboudi element routines again passing in the
c       displacements to retrieve the element stresses and strains.
c                                                                    540
      do 1300 m = 1, nel
      m1st18 = (m-1)*18
      m1st12 = (m-1)*12
      do 1200 i = 1,18
            if(lm(i+(m1st18)).gt.lstop) then
                  u(i) = ub((lm(i+(m1st18)))-lstop)
            else
                  u(i) = deltau(lm(i+(m1st18)))
            endif
1200  continue                                                       550
            jtype=1
            jelem = 1
      call uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
c
```

```
c                the state variables are in the order of:
c                        statev(1-6) -- stresses in the fiber cell
c                        statev(7-12) -- strains in the fiber cell
c                        statev(13-18) -- stresses in matrix cell 1                    560
c                        statev(19-24) -- strains in matrix cell 1
c                        statev(25-30) -- stresses in matrix cell 2
c                        statev(31-36) -- strains in matrix cell 2
c                        statev(37-42) -- stresses in matrix cell 3
c                        statev(43-48) -- strains in matrix cell 3
c                        statev(49) fiber failure criterion
c                        statev(50) matrix failure criterion
c                        statev(51-86) -- converged internal displacements
c                        statev(87-94) -- damage parameter for springs
c                        statev(95) -- fiber von mises stress                         570
c                        statev(96-98) -- matrix von mises stresses
c
      do 1240 i = 1,12
            statev(i+m1st12) = svars(i)
 1240 continue
 1300 continue
c
c                        micro-failure criteria proposed by aboudi
c              the criterion for the fiber compares the stress in the
c      fiber to either the tensile or compressive strength, depending              580
c      upon the state of stress.  a result less than one
c              the criterion for the matrix looks at the maximum
c      principal stress in the plane perpendicular to the axis of
c      the fiber and the axial shear stresses, s12 and s13.  the
c      principal stress is squared and then divided by the square of the
c      the ultimate tensile strength of the matrix.  the axial shear
c      stresses are squared and added together.  this sum is then divided
c      by the square of the ultimate shear strength of the matrix.  these
c      two quotients are then added, with a sum of less than one
c      indicating that failure has not occurred in the matrix.                     590
c
      if (statev(1).gt.0.0) then
```

```
            statev(49) = statev(1)/props(22)
        else
            statev(49) = -statev(1)/props(23)
        endif
        statev(50) = 0.0
c
c           the von mises equivalent stress--fiber cell
c                                                                    600
        statev(95) = dsqrt((0.5*((statev(1)-statev(2))**2+
     1(statev(2)-statev(3))**2+(statev(3)-statev(1))**2))+
     2(3.0*(statev(4)*statev(4)+statev(5)*statev(5)
     3+statev(6)*statev(6))))
        do 1340 i = 1,3
        do 1320 j = 1,6
            s(j) = statev(i*12+j)
 1320   continue
        oper = dsqrt(((s(2)-s(3))/2.0)**2+(s(6)*s(6)))
        spr1 = (s(2)+s(3))/2.0 + oper                                610
        spr2 = (s(2)+s(3))/2.0 - oper
        if (abs(spr2).gt.abs(spr1)) then
            spr1 = spr2
        endif
        if (spr1.le.0.0) then
            stren = props(31)
        else
            stren = props(24)
        endif
        oper = ((spr1*spr1)/(stren*stren))+                          620
     +(((s(4)*s(4))+(s(5)*s(5)))/(props(26)*props(26)))
        if (oper.gt.statev(50)) then
            statev(50) = oper
        endif
c
c           the von mises equivalent stress--matrix cells
c
        statev(i+95) = dsqrt((0.5*((s(1)-s(2))**2+
```

```
     1(s(2)-s(3))**2+(s(3)-s(1))**2))+
     2(3.0*(s(4)*s(4)+s(5)*s(5)+s(6)*s(6))))                          630
 1340    continue
c
c       store internal dispalcements as state variables
c
          do 1400 i = 1, lstop
                    statev(50+i) = deltau(i)
 1400    continue
c
c call the spring elements to save the state variables (damage)
c                                                                     640
       do 1500 m=1,8
          m1st6 = (m-1)*6
          jtype = 2
       jelem = 1
          do 1530 i = 1,6
                    u(i) = deltau(lms(i+(m1st6)))
 1530    continue
       svars(13)=statev(86+m)
       svars(14)=area(m)
       if(m.ge.3) then                                                650
       sprop17=props(17)
       sprop18=props(18)
       sprop19=props(19)
       sprop28=props(28)
       sprop29=props(29)
       sprop30=props(30)
       if(m.eq.3) then
       props(17)=fibermu
       props(18)=props(1)
       props(19)=props(7)                                             660
       props(28)=fibery
       props(29)=fiberf
       props(30)=h(i)
       end if
```

```
      if(m.ge.4) then
      props(17)=mmu
      props(18)=props(13)
      props(19)=props(15)
      props(28)=mgamy
      props(29)=mgamf                                                    670
      props(30)=h(i)
      end if
      end if
         call uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
      if(m.ge.3) then
      props(17)=sprop17
      props(18)=sprop18
      props(19)=sprop19                                                  680
      props(28)=sprop28
      props(29)=sprop29
      props(30)=sprop30
      end if
      statev(86+m)=svars(13)
 1500  continue
c
 5000    format (3f10.5)
 5010    format (12f11.0)
 5020 format (12f8.0)                                                    690
 5030 format (6g10.3)
 5040 format (3f12.5)
 5050 format (6f10.5)
         return
         end
c
         subroutine uelab(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
c                                                                        700
```

```
      implicit real*8(a-h,o-z)
      dimension rhs(ndofel),amatrx(ndofel,ndofel),svars(1),
     1props(1),crds(mcrd,nnode),u(ndofel),
     2predef(1),lflags(4),time(2)
c
      if (jtype.eq.1) then
            call aboudi(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
      else if (jtype.eq.2) then                                          710
            call damage(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
      endif
c
      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c                                    aboudi element                    720
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine aboudi(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
c
      implicit real*8(a-h,o-z)
      dimension rhs(ndofel),amatrx(ndofel,ndofel),svars(1),
     1props(1),crds(mcrd,nnode),u(ndofel),
     2predef(1),lflags(4),time(2)                                       730
c
      dimension b(6,18)
      dimension strain(6),tstr(6),thstr(6),stress(6)
      dimension temp(18,6),d(6,6),abar(18,18)
      dimension lhol(4),nent(3,4),phi(3,4)
c
```

```
c                6 node aboudi cell model element for abaqus
c          the vectors are arranged in the following manner:
c                              dof 1 at node 1,
c                              dof 2 at node 1,                          740
c                              dof 3 at node 1,
c                              dof 1 at node 2,
c                              dof 2 at node 2,
c                              dof 3 at node 2, and so on.
c
c          svars(1) to svars(6) = stresses at integration points
c          svars(7) to svars(12) = strains at integration points
c
c                the program is called in parts.  the first part calls
c     element to obtain the element stiffness matrix and assemble the    750
c     global stiffness matrix.  in the second part, the element stresses
c     and strains are returned given the displacement field, or state of
c     strain, that exists at the boundaries.
c
c                initialize the b matrix.
c
          m1st6 = (m-1)*6
          d1 = 1.0/(abs(crds(1,1+m1st6) - crds(1,2+m1st6)))
          d2 = 1.0/(abs(crds(2,3+m1st6) - crds(2,4+m1st6)))
          d3 = 1.0/(abs(crds(3,5+m1st6) - crds(3,6+m1st6)))           760
          prod = 1.0/(d1*d2*d3)
c
          do 20 k = 1,18
          do 20 j = 1,6
                b(j,k) = 0.0
 20       continue
c
c         b matrix
c
          b(1,1) = d1                                                 770
          b(1,4) = -d1
          b(2,8) = d2
```

```
      b(2,11) = -d2
      b(3,15) = d3
      b(3,18) = -d3
      b(4,2) = d1
      b(4,5) = -d1
      b(4,7) = d2
      b(4,10) = -d2
      b(5,3) = d1                                                    780
      b(5,6) = -d1
      b(5,13) = d3
      b(5,16) = -d3
      b(6,9) = d2
      b(6,12) = -d2
      b(6,14) = d3
      b(6,17) = -d3
c
c           determine the d matrix (constitutive)
c                                                                    790
      call orthotropic(props,nprops,d,m)
c
c           route the subroutine to the correct parts, depending upon
c     what is being asked for.
c
      if (jelem .eq. 1) go to 200
c
c           initialize the amatrix and check to see if the mass
c     matrix is desired.  if the mass matrix is desired
c     (lflags(3) = 3), then the amatrix is initialized to zero       800
c     and the subroutine returns control to abaqus.
c
      do 40 j = 1,18
            rhs(j) = 0.0
      do 40 i = 1,18
            amatrx(i,j) = 0.0
 40   continue
      if (lflags(3).eq.3) go to 1000
```

```
c
c                calculate the stiffness (amatrx) matrix.  this calculation is                  810
c       the triple product:  b * d * b
c
        do 60 j = 1,6
        do 60 i = 1,18
                temp(i,j) = 0.0
 60     continue
        do 80 k = 1,6
        do 80 j = 1,6
        do 80 i = 1,18
                temp(i,j) = temp(i,j) + b(k,i) * d(k,j)                                        820
 80     continue
c
        do 100 k = 1,6
        do 100 j = 1,18
        do 100 i = 1,18
                amatrx(i,j) = amatrx(i,j) + prod * temp(i,k) * b(k,j)
 100    continue
c
c           add small stiffness for zero energy modes
c                                                                                              830
c
      nent(1,1)=2
      nent(1,2)=5
      nent(1,3)=7
      nent(1,4)=10
      phi(1,1)=d1
      phi(1,2)=-d1
      phi(1,3)=-d2
      phi(1,4)=d2
c                                                                                              840
      nent(2,1)=3
      nent(2,2)=6
      nent(2,3)=13
      nent(2,4)=16
```

```
      phi(2,1)=d1
      phi(2,2)=-d1
      phi(2,3)=-d3
      phi(2,4)=d3
c
      nent(3,1)=9                                                        850
      nent(3,2)=12
      nent(3,3)=14
      nent(3,4)=17
      phi(3,1)=d2
      phi(3,2)=-d2
      phi(3,3)=-d3
      phi(3,4)=d3
c
c      stifmin=min(amatrx(1,1),amatrx(2,2),amatrx(3,3))
      gstif=1.0e-6*d(4,4)                                                860
      do 190 l=1,3
      do 180 i=1,4
      do 180 j=1,4
      aplus=gstif*phi(l,i)*phi(l,j)
      amatrx(nent(l,i),nent(l,j))=amatrx(nent(l,i),nent(l,j))+
     +  aplus
      rhs(nent(l,i))=rhs(nent(l,i))-aplus*u(nent(l,j))
  180 continue
  190 continue
c                                                                        870
c                calculate the total strain from the displacements
c                (using strain = b * displacements)
c
  200    do 220 i = 1,6
                tstr(i) = 0.0
  220    continue
         do 240 j = 1,18
         do 240 i = 1,6
                tstr(i) = tstr(i) + b(i,j)*u(j)
  240    continue                                                        880
```

```
c
c               calculate the thermal strains in the element
c
      dtemp = ttemp - props(20)
      do 260 i = 1,6
            thstr(i) = 0.0
 260  continue
c
      if (m.eq.1) then
      do 280 j = 1,3                                                    890
            thstr(j) = props(j+9) * dtemp
 280  continue
      else
      do 290 j = 1,3
            thstr(j) = props(16) * dtemp
 290  continue
      endif
c
c               subtract the thermal strains out from the total strain
c                                                                       900
      do 300 i = 1,6
            strain(i) = 0.0
            strain(i) = tstr(i) - thstr(i)
 300  continue
c
c               calculate the stresses from the strain
c
      do 310 i = 1,6
            stress(i) = 0.0
 310  continue                                                          910
      do 320 j = 1,6
      do 320 i = 1,6
            stress(i) = stress(i) + d(i,j) * strain(j)
 320  continue
c
c               compute the force vector, rhs.
```

```
c
      if (lflags(3).eq.2) go to 1000
      do 360 j = 1,6
      do 340 i = 1,18                                                    920
            rhs(i) = rhs(i) - prod * b(j,i) * stress(j)
 340  continue
 360  continue
c
c           set state variables
c
      do 380 i = 1,6
            svars(i) = stress(i)
            svars(i+6) = strain(i)
 380  continue                                                           930
 1000 return
 5000 format (3f10.5)
 5010 format (18f6.0)
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c           subroutine orthotropic which determines the d (constitutive)
c     matrix for a general orthotropic material.
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc  940
      subroutine orthotropic(props,nprops,d,m)
c
      implicit real*8(a-h,o-z)
      dimension props(nprops),d(6,6)
      integer nprops,m
c
      real*8 det,v12,v13,v23,v21,v31,v32
      real*8 e11,e22,e33,g12,g13,g23
      integer i,j
      if (m.eq.1) then                                                   950
            e11 = props(1)
            e22 = props(2)
```

```
            e33 = props(3)
            v12 = props(4)
            v13 = props(5)
            v23 = props(6)
            v21 = v12*(props(2)/props(1))
            v31 = v13*(props(3)/props(1))
            v32 = v23*(props(3)/props(2))
      det = 1 - (v12*v21) - (v23*v32) - (v13*v31) -                    960
     +  (2.0*v12*v23*v31)
            g12 = props(7)
            g13 = props(8)
            g23 = props(9)
      else
            e11 = props(13)
            e22 = e11
            e33 = e11
            v12 = props(14)
            v13 = v12                                                 970
            v23 = v12
            g12 = props(15)
            g13 = g12
            g23 = g12
            v21 = v12
            v31 = v13
            v32 = v23
      det = 1 - (v12*v21) - (v23*v32) - (v13*v31) -
     +  (2.0*v12*v23*v31)
      endif                                                           980
c
      do 20 j = 1,6
      do 20 i = 1,6
            d(i,j) = 0.0
 20   continue
c
      d(1,1) = (1 - (v23*v32))*e11/det
      d(1,2) = (v12 + (v13*v32))*e22/det
```

```
      d(1,3) = (v13 + (v23*v12))*e33/det
      d(2,2) = (1 - (v31*v13))*e22/det                                    990
      d(2,3) = (v23 + (v21*v13))*e33/det
      d(3,3) = (1 - (v12*v21))*e33/det
      d(2,1) = d(1,2)
      d(3,1) = d(1,3)
      d(3,2) = d(2,3)
      d(4,4) = g12
      d(5,5) = g13
      d(6,6) = g23
c
      return                                                              1000
      end
      subroutine damage(rhs,amatrx,svars,ndofel,props,crds,mcrd,
     1 nnode,u,jtype,time,dtime,kstep,kinc,jelem,predef,npred,lflags,
     2 m,nprops,ttemp)
c
      implicit real*8(a-h,o-z)
      dimension rhs(ndofel),amatrx(ndofel,ndofel),svars(1),
     1props(1),crds(mcrd,nnode),u(ndofel),
     2predef(1),lflags(4),time(2)
c                                                                         1010
      dimension lhol(2)
      real*8 eps1,eps2,eps3,delta1,delta2,delta3
      real*8 dam,mu,e,g,gammay,gammaf,h
      real*8 gammae,curdam,kn,ks,divis,tol
      real*8 dddge,dgedd1,dgedd2,dgedd3
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c                                    damage element
c
c        this is now a nonlinear damage element.  the stress at any point  1020
c        is dependent on the maximum strain seen by the element at any
c        previous time.
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```

```
c       the vectors are arranged as follows:
c               dof 1 at node 1
c               dof 2 at node 1
c               dof 3 at node 1
c               dof 1 at node 2
c               dof 2 at node 2                                          1030
c               dof 3 at node 2
c
c       props(17) = mu, the effective spring friction coefficient
c       props(18) = e, the effective spring young's modulus
c       props(19) = g, the effective spring shear modulus
c       props(28) = gamma_y, the yield strain for the spring
c       props(29) = gamma_f, the failure strain for the spring
c               (this must be greater than gamma_y)
c       props(30) = h, the bond thickness for the spring
c                                                                        1040
c       svars(13) = d, the damage parameter (cumulative)
c       svars(14) = a, the initial total area of the bond
c
c       initialize and construct the amatrix
c
        do 10 i = 1, 6
               rhs(i) = 0.0
        do 10 j = 1, 6
               amatrx(i,j) = 0.0
 10     continue                                                         1050
        if (lflags(3).eq.3) go to 1000
c
        mu = props(17)
        e = props(18)
        g = props(19)
        gammay = props(28)
        gammaf = props(29)
        h = props(30)
c
c              tol is the tolerance value which i use to check the       1060
```

```
c       deltas against 0.
c
        tol = 0.01*h
c
        kn = svars(14)*e/h
        ks = svars(14)*g/h
c
c       calculate the effective strain, gammae
c
        delta1 = u(4)-u(1)                                                    1070
        delta2 = u(5)-u(2)
        delta3 = u(6)-u(3)
        eps1 = delta1/h
        eps2 = delta2/h
        eps3 = delta3/h
        gammae = (mu*e*eps1/g)+dsqrt((eps2*eps2)+(eps3*eps3))
        dddge = 0.0
        if (gammae.lt.gammay) then
                curdam = 0.0
        else if (gammae.gt.gammaf) then                                       1080
                curdam = 1.0
        else
c               curdam = gammae/(gammaf-gammay)
                curdam=1.-(gammay/gammae)*((gammaf-gammae)/(gammaf-gammay))
        endif
        if (curdam.gt.svars(13)) then
                dam = curdam
                if (curdam.lt.1.0) then
c                       dddge = 1.0/(gammaf-gammay)
                        dddge = gammaf*gammay/(gammaf-gammay)/gammae/gammae   1090
                endif
                 if(dam.gt.1.)  dam=1.
                 svars(13)=dam
        else
                dam = svars(13)
        endif
```

```
        if(jelem.eq.1) go to 1000
        dgedd1 = (mu*e)/(g*h)
        divis = dsqrt((delta2*delta2)+(delta3*delta3))
        if (abs(delta2).lt.tol) then                                    1100
               dgedd2 = 0.0
        else
               dgedd2 = (delta2)/(h*divis)
        endif
        if (abs(delta3).lt.tol) then
               dgedd3 = 0.0
        else
               dgedd3 = (delta3)/(h*divis)
        endif
c                                                                       1110
        if (delta1.gt.0.0) then
               amatrx(1,1) = kn*(1.0-dam)-(kn*delta1*dddge*dgedd1)
               amatrx(1,2) = -1.0*kn*delta1*dddge*dgedd2
               amatrx(1,3) = -1.0*kn*delta1*dddge*dgedd3
        else
               amatrx(1,1) = kn
        endif
        amatrx(2,2) = ks*(1.0-dam)-(ks*delta2*dddge*dgedd2)
        amatrx(3,3) = ks*(1.0-dam)-(ks*delta3*dddge*dgedd3)
        amatrx(2,1) = -1.0*ks*delta2*dddge*dgedd1                       1120
        amatrx(2,3) = -1.0*ks*delta2*dddge*dgedd3
        amatrx(3,1) = -1.0*ks*delta3*dddge*dgedd1
        amatrx(3,2) = -1.0*ks*delta3*dddge*dgedd2
c
c symmeterize amatrx
c
c        amatrx(1,2)=.5*(amatrx(1,2)+amatrx(2,1))
c        amatrx(1,3)=.5*(amatrx(1,3)+amatrx(3,1))
c        amatrx(2,3)=.5*(amatrx(2,3)+amatrx(3,2))
c        amatrx(2,1)=amatrx(1,2)                                        1130
c        amatrx(3,1)=amatrx(1,3)
c        amatrx(3,2)=amatrx(2,3)
```

```
cc
cc check for nonpositive definite amatrx
cc
c        det=    amatrx(1,1)*amatrx(2,2)*amatrx(3,3)
c        det=det+amatrx(1,2)*amatrx(2,3)*amatrx(3,1)
c        det=det+amatrx(1,3)*amatrx(2,1)*amatrx(3,2)
c        det=det-amatrx(1,3)*amatrx(2,2)*amatrx(3,1)
c        det=det-amatrx(1,1)*amatrx(2,3)*amatrx(3,2)                    1140
c        det=det-amatrx(1,2)*amatrx(2,1)*amatrx(3,3)
c        if(det.le.0.)  then
c        do 90 i=1,3
c        do 90 j=1,3
c 90     amatrx(i,j)=0.
c        end if
       do 120 j = 1,3
       do 100 i = 1,3
              amatrx(i,j+3) = -amatrx(i,j)
              amatrx(i+3,j) = -amatrx(i,j)                             1150
              amatrx(i+3,j+3) = amatrx(i,j)
 100   continue
 120   continue

c
c              compute the force vector, rhs
c      note:  this is the negative of the internal force vector
c
       if (lflags(3).eq.2) go to 1000
       if (delta1.gt.0.0) then                                         1160
              rhs(1) = kn*(1.0-dam)*delta1
       else
              rhs(1) = kn*delta1
       endif
       rhs(2) = ks*(1.0-dam)*delta2
       rhs(3) = ks*(1.0-dam)*delta3
       rhs(4) = -rhs(1)
       rhs(5) = -rhs(2)
```

```
      rhs(6) = -rhs(3)
c                                                                      1170
 1000 return
      end
```

Bibliography

[1] J. Aboudi. A continuum theory for fiber-reinforced elastic-viscoelastic composites. *International Journal of Engineering Science*, 20:605–621, 1982.

[2] J. Aboudi. The effective moduli of short-fiber composites. *International Journal of Solids and Structures*, 19(8):693–707, 1983.

[3] J. Aboudi. Effective behavior of inelastic fiber-reinforced composites. *International Journal of Engineering Science*, 22:439–449, 1984.

[4] J. Aboudi. Effective thermoelastic constants of short-fiber composites. *Fibre Science and Technology*, 20:211–225, 1984.

[5] J. Aboudi. Minimechanics of tri-orthogonally fibre-reinforced composites: Overall elastic and thermal properties. *Fiber Science and Technology*, 21:277–293, 1984.

[6] J. Aboudi. Elastoplasticity theory for composite materials. *Solid Mechanics Archives*, 11:141–183, 1986.

[7] J. Aboudi. Closed-form constitutive equations for metal-matrix composites. *International Journal of Engineering Science*, 25:1229–1240, 1987.

[8] J. Aboudi. Damage in composites—modeling of imperfect bonding. *Composite Science and Technology*, 28:103–128, 1987.

[9] J. Aboudi. Stiffness reduction of cracked solids. *Engineering Fracture Mechanics*, 26(5):637–650, 1987.

[10] J. Aboudi. Constitutive equations for elastoplastic composites with imperfect bonding. *International Journal of Plasticity*, 4:103–125, 1988.

[11] J. Aboudi. Micromechanical analysis of the strength of unidirectional fiber composites. *Composites Science and Technology*, 33:79–96, 1988.

[12] J. Aboudi. Micromechanical analysis of composites by the method of cells. *Applied Mechanics Review*, 42:193–221, July 1989.

[13] J. Aboudi. Micromechanics prediction of fatigue failure of composite materials. *Journal of Reinforced Plastics and Composites*, 8:150–166, 1989.

[14] J. Aboudi. Viscoelastic behavior of thermo-rheologically complex resin matrix composites. *Composites Science and Technology*, 36:351–365, 1989.

[15] J. Aboudi. The nonlinear behavior of unidirectional and laminated composites—a micromechanical approach. *Journal of Reinforced Plastics and Composites*, 9:13–31, January 1990.

[16] J. Aboudi. *Mechanics of Composite Materials: A Unified Micromechanical Approach*, volume 29 of *Studies in Applied Mechanics*. Elsevier, Amsterdam, The Netherlands, 1991.

[17] J. Aboudi. The over-all instantaneous properties of metal-matrix composites. *Composites Science and Technology*, 41:411–429, 1991.

[18] R. T. Arenburg and J. N. Reddy. Applications of the aboudi micromechanics theory to metal matrix composites. In J. N. Reddy and J. L. Teply, editors, *Mechanics of Composite Materials and Structures*, number 100 in AMD Series, pages 21–32, La Jolla, CA, July 1989. The Applied Mechanics Division, ASME.

[19] Klaus-Jurgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[20] Zdenek P. Bazant. Why continuum damage is nonlocal: Micromechanics arguments. *Journal of Engineering Mechanics*, 117(5):1070–1087, 1991.

[21] M. P. Bendsoe and N. Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71:197–224, 1988.

[22] C. A. Bigelow, W. S. Johnson, and R. A. Naik. A comparison of various micromechanics models for metal matrix composites. In J. N. Reddy and J. L. Teply, editors, *Mechanics of Composite Materials and Structures*, number 100 in Applied Mechanics Division, pages 21–32, La Jolla, CA, July 1989. The Applied Mechanics Division, ASME.

[23] D. M. Blackketter, D. E. Walrath, and A. C. Hansen. Modeling damage in a plain weave fabrix-reinforced composite material. *Journal of Composites Technology and Research*, 15(2):136–142, 1993.

[24] C. S. Chang and J. W. Ju, editors. *Homogenization and Constitutive Modeling for Heterogeneous Materials*, number 166 in AMD Series, Charlottesville, VA, June 1993. The Applied Mechanics Division, ASME and The Engineering Mechanics Division, ASCE.

[25] K. K. Chawla. *Composite Materials: Science and Engineering.* Materials Research and Engineering. Springer-Verlag, New York, New York, 1987.

[26] Stephen H. Crandall, Norman C. Dahl, and Thomas J. Lardner. *An Introduction to the Mechanics of Solids*, chapter 4.6, page 218. McGraw-Hill, New York, second edition, 1978.

[27] M. P. Divakar and A. Fafitis. Micromechanics-based constitutive model for interface shear. *Journal of Engineering Mechanics*, 118(7):1317–1337, 1992.

[28] G. J. Dvorak and Y. A. Bahei-El-Din. Elastic-plastic behavior of fibrous composites. *Journal of the Mechanics and Physics of Solids*, 27:51–72, 1979.

[29] S. P. Engelstad and J. N. Reddy. Probabilistic nonlinear finite element analysis of composite structures. *AIAA Journal*, 31(2):362–369, February 1993.

[30] J. L. Ericksen, D. Kinderlehrer, R. Kohn, and J. L. Lions, editors. *Homogenization and Effective Moduli of Materials and Media*, number 1 in The IMA Volumes in Mathematics and Its Applications. IMA, Springer-Verlag, 1986.

[31] R. L. Foye. An evaluation of various engineering estimates of the transverse properties of unidirectional composites. In *Proceedings of the Tenth National SAMPE Symposium—Advanced Fibrous Reinforced Composites*, November 1966.

[32] J. Francu. Homogenization of linear elasticity equations. *Journal of Applied Mathematics*, 27:96–117, 1982.

[33] M. Gosz, B. Moran, and J. D. Achenbach. Load-dependent constitutive response of fiber composites with compliant interphases. *Journal of the Mechanics and Physics of Solids*, 40(8):1789–1803, 1992.

[34] R. M. Hackett. Private Communication.

[35] Z. Hashin and S. Shtrikman. *Journal of the Mechanics and Physics of Solids*, 11:227+, 1963.

[36] C. T. Herakovich, J. Aboudi, S. W. Lee, and E. A. Strauss. 2-d and 3-d damage effects in cross-ply laminates. In G. J. Dvorak and N. Laws, editors, *Mechanics of Composite Materials—1988*, number 92 in AMD Series, pages 143–147, Berkeley, CA, June 1988. The Committee on Composite Materials of the Applied Mechanics Division, ASME.

[37] R. Hill. In *Proceedings of the Physical Society, A65*, page 349, 1952.

[38] R. Hill. A self-consistent mechanics of composite materials. *Journal of the Mechanics and Physics of Solids*, 13:213–222, 1965. Change pages numbers.

[39] D. A. Hopkins and C. C. Chamis. A unique set of micromechanics equations for high temperature metal matrix composites. In *NASA TM-87154, Prepared for the First Symposium on Testing Technology of Metal Matrix Composites*, Nashville, TN, November 1985. American Society for Testing and Materials.

[40] J. W. Ju. A micromechanical damage model for uniaxially reinforced composites weakened by interfacial arc microcracks. *Journal of Applied Mechanics*, 58:923–930, December 1991.

[41] Herman S. Kaufman and Joseph J. Falcetta. *Introduction to Polymer Science and Technology: An SPE Textbook*, chapter 8, pages 352–356. John Wiley, New York, NY, 1978.

[42] T. R. King, D. M. Blackketter, D. E. Walrath, and D. F. Adams. Micromechanics prediction of the shear strength of carbon fiber/epoxy matrix composites: The influence of the matrix and interface strengths. *Journal of Composite Materials*, 26(4):558–573, 1992.

[43] Francoise Lene. Damage constitutive relations for composite materials. *Engineering Fracture Mechanics*, 25(5):713–728, 1986.

[44] R. W. Macek. Private Communication.

[45] Manufacturer Specifications.

[46] P. L. N. Murphy and C. C. Chamis. Micromechanics for ceramic matrix composites. In *Proceedings of the 26th National SAMPE Symposium*, April 1991.

[47] O. O. Ochoa and J. N. Reddy. *Finite Element Analysis of Composite Laminates*, volume 7 of *Solid Mechanics and Its Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, first edition, 1992.

[48] M. J. Pindera and M. W. Lin. Micromechanical analysis of the elastoplastic response of metal matrix composites. *Journal of Pressure Vessel Technology*, 111:183–190, May 1989.

[49] D. D. Robertson and S. Mall. Micromechanical relations for fiber-reinforced composites using the free transverse shear approach. *Journal of Composites Technology and Research*, 15(3):181–192, 1993.

[50] E. Sanchez-Palencia. *Non-Homogeneous Media and Vibration Theory*, volume 127 of *Lecture Notes in Physics*. Springer-Verlag, Berlin, 1980.

[51] Irving H. Shames and Francis A. Cozzarelli. *Elastic and Inelastic Stress Analysis*, chapter 9.14, pages 381–386. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[52] J. L. Teply and G. J. Dvorak. Bounds on overall instantaneous properties of elastic-plastic composites. *Journal of the Mechanics and Physics of Solids*, 36(1):29–58, 1988.

[53] J. L. Teply and J. N. Reddy. A unified formulation of micromechanics models of fiber-reinforced composites. In J. G. Dvorak, editor, *Inelastic Deformation of Composite Materials Symposium*, pages 341–370, Troy, NY, 1990. IUTAM, Springer-Verlag.

[54] R. N. Yancey and Marek-Jerzy Pindera. Micromechanical analysis of the creep responses of unidirectional composites. *Journal of Engineering Materials and Technology*, 112:157–163, April 1990.

[55] W. Yang and J. P. Boehler. Micromechanics modeling of anisotropic damage in cross-ply laminates. *International Journal of Solids and Structures*, 29(10):1303–1328, 1992.

[56] Wen-Chao Zhang and K. E. Evans. An analytical model for the elastic properties of fibrous composites with anisotropic constituents. *Composites Science and Technology*, 38:229–246, 1990.

[57] O. C. Zienkiewicz. *The Finite Element Method*, chapter 7.7, pages 162–164. McGraw-Hill, London, UK, third edition, 1977.