

6-1-94
E8652

NASA Technical Memorandum 106526

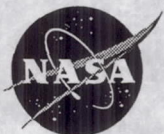
A Coupled/Uncoupled Deformation and Fatigue Damage Algorithm Utilizing the Finite Element Method

Thomas E. Wilt
University of Toledo
Toledo, Ohio

and

Steven M. Arnold
Lewis Research Center
Cleveland, Ohio

March 1994



National Aeronautics and
Space Administration

A Coupled/Uncoupled Deformation and Fatigue Damage Algorithm Utilizing The Finite Element Method

T. E. Wilt*
University of Toledo
Toledo, Ohio 43606

S. M. Arnold
NASA Lewis Research Center
Cleveland Ohio

Abstract

A fatigue damage computational algorithm utilizing a multiaxial, isothermal, continuum based fatigue damage model for unidirectional metal matrix composites has been implemented into the commercial finite element code MARC using MARC user subroutines. Damage is introduced into the finite element solution through the concept of effective stress which fully couples the fatigue damage calculations with the finite element deformation solution. An axisymmetric stress analysis was performed on a circumferentially reinforced ring, wherein both the matrix cladding and the composite core were assumed to behave elastic-perfectly plastic. The composite core behavior was represented using Hill's anisotropic continuum based plasticity model, and similarly, the matrix cladding was represented by an isotropic plasticity model. Results are presented in the form of S-N curves and damage distribution plots.

1.0 Introduction

In advanced engine designs, materials which allow higher operating speeds and longer durability in addition to decreased weight are desirable. The use of metal matrix composites (MMCs) may provide these benefits. For example, titanium metal matrix composite (TMCs) rotors are projected to have significant benefits in terms of increased rotor speeds and lower weight, as compared to the nickel and titanium rotors currently in service. However, to fully realize the benefits offered by MMCs, computationally efficient design and life prediction methods must be developed.

Analysis of typical aerospace structures subjected to complex thermomechanical load histories requires the use of computational approaches such as the finite element method.

*Resident Research Associate at Lewis Research Center

In this regard, it is desirable to develop a life prediction algorithm that can be used in conjunction with the finite element method. Historically, two basic approaches have been used in predicting the life of structures; uncoupled or fully-coupled deformation-damage methods.

A typical uncoupled analysis consists of obtaining the stress state for each element from a finite element analysis, and then, using the stress state data as input to a fatigue damage model, the number of cycles to the initiation of a crack are predicted. Subsequently, a "local" fracture mechanics approach is then used to propagate the crack. This requires a new finite element mesh be constructed to model the crack tip zone. For example, the crack may be modeled using a series of double nodes. The propagation of the crack is then controlled by a strain energy release rate criteria in conjunction with a node release scheme.

As mentioned, the alternative to the uncoupled method is a fully-coupled deformation and damage method. This is the approach taken in the present study. Specifically, the computational scheme developed uses MARC, a nonlinear finite element code in which the fatigue damage algorithm is coupled to MARC through the use of provided user subroutines. By utilizing the concept of effective stress, the effects of damage are accounted for in the finite element solution. The effective stress concept allows the current damage state to be incorporated into the deformation response through the degradation of the material properties, which, in the context of the finite element method.

In Section 2, the requisite fatigue damage equations are presented. In Section 3 the computationally-coupled fatigue damage algorithm will be outlined. Finally, an example of a reinforced MMC ring, representing a typical engine component, will be presented and results will be presented in terms of the evolution of damage in the ring cross section.

2.0 Fatigue Damage Formulation

The fatigue damage calculations utilize a recently developed multiaxial, isothermal, continuum damage mechanics model for the fatigue of unidirectional metal matrix composites [1]. The model is phenomenological, stress based, and assumes a single scalar internal damage variable, the evolution of which is anisotropic. The present multiaxial, isothermal, continuum damage model for unidirectional metal matrix composites may be expressed as, [1]

$$\int_{D_{k-1}}^{D_k} dD = \int_0^{\Delta N_k} [1 - (1 - D)^{\beta+1}]^{\alpha_k} \left[\frac{\hat{F}_m^\beta}{1 - D} \right]^\beta dN \quad (\text{EQ 1})$$

where D_k and D_{k-1} is the amount of damage at the current and previous increments, respectively, and ΔN_k is the number of cycles at the current stress state (σ^k), and α_k (which is a function of the current stress state) is defined as,

$$\alpha_k = 1 - a \frac{\langle \Phi_{fl} \rangle}{\langle \Phi_u \rangle} \quad (\text{EQ 2})$$

where $\langle \rangle$ are the Macauley brackets. Note, the subscript/superscript k and $k-1$ denote the current and previous increments, respectively. The fatigue limit surface, Φ_{fl} , is defined as,

$$\Phi_{fl} = \frac{1}{2} \max_{t_0} \max_t F_{(f)} (\sigma_{ij}^k(t) - \sigma_{ij}^k(t_0)) - 1 \quad (\text{EQ 3})$$

The static fracture surface, Φ_u , is defined as,

$$\Phi_u = 1 - \max_t F_{(u)} (\sigma_{ij}^k(t)) \quad (\text{EQ 4})$$

Lastly, the normalized stress amplitude, \hat{F}_m , is defined as,

$$\hat{F}_m = \frac{1}{2} \max_{t_0} \max_t F_{(m)} (\sigma_{ij}^k(t) - \sigma_{ij}^k(t_0)) \quad (\text{EQ 5})$$

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

$$F_{()} = \sqrt{\frac{1}{()^2 L} \left\{ (4\omega_{()}^2 - 1)I_1 + \frac{4\omega_{()}^2 - 1}{\eta_{()}^2} I_2 + \frac{9}{4} I_3 \right\}} \quad (\text{EQ 6})$$

in which I_1 , I_2 , and I_3 are physically meaningfull invariants [1], i.e.,

$$\begin{aligned} I_1 &= \frac{1}{2} S_{ij} S_{ij} - d_i d_j S_{jk} S_{ki} + \frac{1}{4} (d_i d_j S_{ij})^2 \\ I_2 &= d_i d_j S_{jk} S_{ki} - (d_i d_j S_{ij})^2 \\ I_3 &= (d_i d_j S_{ij})^2 \end{aligned} \quad (\text{EQ 7})$$

which are a function of the current deviatoric stress state, $S_{ij}^k = \sigma_{ij}^k - \frac{1}{3} \sigma_{ll}^k \delta_{ij}$, and a vector d_i denoting the materials' fiber orientation.

Note, that in the expression for α_k , $\langle \Phi_u \rangle = 0$ indicates static fracture and, as will be discussed in the following section, the finite element is considered to have failed completely. On the other hand, whenever $\langle \Phi_{fl} \rangle = 0$ indicates that the current stress state is below the fatigue limit and thus α_k is set equal to 1. This presents a special case when integrating the fatigue damage expression, EQ. 1, and will be considered later in this section.

First, consider a current state of stress, σ^k , which is above the fatigue limit, i.e. $\alpha_k \neq 1$. Thus, integrating EQ 1.,

$$\frac{[1 - (1 - D)^{\beta+1}]^{1-\alpha_k}}{(1 - \alpha_k)(\beta + 1)} \Bigg|_{D_{k-1}}^{D_k} = \hat{F}_m^\beta \Delta N_k \quad (\text{EQ 8})$$

which results in an expression for the number of cycles, ΔN_k , at the current stress, σ^k , i.e.,

$$\Delta N_k = \frac{([1 - (1 - D_k)^{\beta+1}]^{1-\alpha_k} - [1 - (1 - D_{k-1})^{\beta+1}]^{1-\alpha_k})}{\hat{F}_m^\beta (1 - \alpha_k)(\beta + 1)} \quad (\text{EQ 9})$$

Note that D_{k-1} is the total amount of damage at the beginning of the load block and D_k is the total amount of damage at the end of this load block.

To calculate the cycles to failure, let $D_k = 1$, which results in the following,

$$\Delta N_{F_k} = \frac{(1 - [1 - (1 - D_{k-1})^{\beta+1}]^{1-\alpha_k})}{\hat{F}_m^\beta (1 - \alpha_k)(\beta + 1)} \quad (\text{EQ 10})$$

In the present computational scheme, since the damage increment is controlled, both D_k and D_{k-1} are known. That is,

$$D_k = D_{k-1} + \Delta D \quad (\text{EQ 11})$$

where ΔD is the user specified increment in damage. Thus EQ. 9 is used to predict the increment in the number of cycles for each element, ΔN_k^e , due to the increment in damage.

As will be shown in the following section, it is also necessary to re-write EQ. 9 in terms of the damage D_k , i.e.,

$$D_k = 1 - \left(1 - \{ [1 - (1 - D_{k-1})^{\beta+1}]^{1-\alpha_k} + (1 - \alpha_k)(\beta + 1) \hat{F}_m^\beta \Delta N_k \}^{\frac{1}{\beta+1}} \right) \quad (\text{EQ 12})$$

Now consider the case in which the current stress state is below the fatigue limit, i.e. $\alpha_k = 1$. Thus, EQ. 1 takes the form,

$$\int_{D_{k-1}}^{D_k} \frac{(1 - D)^\beta}{1 - (1 - D)^{\beta+1}} dD = \int_0^{\Delta N_k} \hat{F}_m^\beta dN \quad (\text{EQ 13})$$

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

$$\Delta N_k = \left(\frac{\log [1 - (1 - D_k)^{\beta+1}] - \log [1 - (1 - D_{k-1})^{\beta+1}]}{\hat{F}_m^\beta (\beta + 1)} \right) \quad (\text{EQ 14})$$

For the number of cycles to failure, ΔN_{F_k} , let $D_k = 1$, i.e.,

$$N_{F_k} = \frac{-\log [1 - (1 - D_{k-1})^{\beta+1}]}{\hat{F}_m^\beta (\beta + 1)} \quad (\text{EQ 15})$$

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

$$D_k = 1 - \{1 - [1 - (1 - D_{k-1})^{\beta+1}] \exp((\beta + 1) \hat{F}_m^\beta \Delta N_k)\}^{\frac{1}{\beta+1}} \quad (\text{EQ 16})$$

The effect of damage is included in the finite element stress analysis by utilizing the concept of effective stress [2]. Based on the hypothesis of strain-equivalence [3,4], the effect of damage may be accounted for by simply degrading the elastic and plastic material properties. The degraded elastic constitutive matrix is calculated by,

$$[\bar{C}] = (1 - D_k) [C] \quad (\text{EQ 17})$$

and similarly, the plastic material properties, e.g. yield stress σ_y , are degraded,

$$\bar{\sigma}_y = (1 - D_k) \sigma_y \quad (\text{EQ 18})$$

3.0 Computational Scheme

The present implementation of the fatigue damage calculations were developed in the context of the finite element code MARC. MARC provides various user subroutines [5] that allow implementation of constitutive models, failure criteria, new elements, etc. By using a few select MARC user subroutines, the continuum-based fatigue damage model has been coupled with the nonlinear finite element solution scheme.

The MARC user subroutines required were, ELEVAR, HOOKLW, and ANPLAS. The subroutines HOOKLW and ANPLAS are used to degrade the elastic and plastic material properties, respectively. The subroutine ELEVAR is called at the end of each load increment once global convergence has been attained, and is intended to be used to output element quantities at the end of a given increment. In this algorithm ELEVAR is used to store the current converged stress state for each element during the "applied" load cycle. The meaning of "applied" load cycle will be discussed later in this section.

The present version of the fatigue damage algorithm utilizes average quantities in the damage calculations. For example, the stresses for each integration point are determined and then all integration points are averaged to give one stress state for each element. All subsequent damage calculations use these average quantities. However, the program was written in a sufficiently general form so that all of the damage calculations may be per-

formed at each integration point with minimal modifications. Specifically, all that is required is increased dimensions for various storage arrays.

A flowchart of the developed life prediction scheme is shown in Figure 1. The deformation analysis is the actual finite element run. The fatigue and failure calculations are contained in the MARC subroutine ELEVAR. First, note there are two levels of failure criteria checks; element level and structural level. The element level includes a static fracture surface check which is part of the fatigue damage model, (EQ. 4). Additional criteria may be included such as a check on total mechanical strains, etc. If an element violates one of these failure criteria, that element is considered to have "failed" and its damage, D , is set equal to the maximum amount of damage allowed. As shown in Fig. 1, for a coupled analysis the damage calculations are terminated and a deformation analysis is re-run in order to account for the stress redistribution due to that element's failure. For an uncoupled analysis, when element failure occurs the analysis does not loop back and perform another deformation analysis, instead, it continues to the next element and performs the damage calculations. The structural level criteria monitors the global response of the structure. This could take the form of a check upon selected nodal displacements which if they violate a specified displacement criteria the structure is considered to have failed. For example, the tip displacement of a turbine blade may be required to stay within a given tolerance. Again, note that for an uncoupled analysis no structural failure criteria check is performed since the present fatigue damage algorithm assumes the structure when subjected to the initial "applied" load cycle is in a completely undamaged state. In the future, the algorithm may include the option to include initial damage states for each element at the beginning of the analysis.

The above mentioned "applied" load cycles are user defined through the MARC input data file, and are used to achieve the stress redistribution in the structure due to the occurrence of damage, i.e. material degradation. In addition, the number of load increments per "applied" cycle must be specified by the user. This is necessary so that the program can internally monitor when a given "applied" load cycle has been completed and begin the damage calculation phase. The number of load increments used in the load cycle usually depends on the nonlinearity of the structural response and requires experience on the part of the user. Note, since the algorithm, in its present form, requires the user to specify the number of increments in a cycle, automatic load incrementing/stepping, i.e. MARC's AUTOLOAD option, cannot be used.

The subroutine ELEVAR is called at each increment during the "applied" load cycle and stores the average stress (strain is optional) state for each element. At the last load increment of the current "applied" load cycle, the subroutine DAMAGE is now called from within ELEVAR, see Figure 2

When DAMAGE is entered for each element, various element quantities, such as, α_k , Φ_{fl} , Φ_u , \hat{F}_m are calculated and stored. When DAMAGE has been called for the last element in the mesh, the fatigue damage calculations are performed. Figure 2 shows the general algorithm for the fatigue damage calculations. In addition, Appendix I contains the FORTRAN source code for the fatigue damage algorithm. The code is presented in a form

that allows the fatigue damage calculations to be used as a subroutine in a finite element, or any other deformation analysis program.

Presently, the damage calculations are controlled by the increment in damage, ΔD . The user specifies the allowable increment in damage, for example $\Delta D = 0.15$, (15%). In CALCN, using EQ. 9 or EQ. 12, based upon the new value of damage and the given element's stress state, the number of cycles to failure, ΔN_F^e , is calculated and stored. Next a "sorting" subroutine, SORTN, is called (see Fig. 2) to determine which element has the minimum number of cycles to failure and is chosen as the "controlling element", i.e.,

$$N_{F_{min}} = \min \Delta N_{F_k}^e \quad e = 1 \rightarrow \text{numel} \quad (\text{EQ 19})$$

Once the controlling number of cycles has been determined, the corresponding, actual amount of damage, D_k^e , in all of the remaining elements must be re-calculated. This is performed in subroutine CALCD using EQ. 10 or EQ. 13. Note that since the damage was incremented by a specified amount, the controlling element's damage is already known.

Figure 3 shows the cycle scheme used in the code. Recall that the "applied" load cycle is the actual load history that is applied and used in the finite element analysis. The subsequent cycles, shown in dashed lines, are the predicted cycles corresponding to $N_{F_{min}}$ which is determined in SORTN. Here it is assumed that the stress state in each element remains constant during the predicted $N_{F_{min}}$ cycles and at the end of $N_{F_{min}}$, each element has incurred an amount of damage as calculated in CALCD. Note for a coupled analysis, the next "applied" load cycle is run in the finite element analysis to account for the stress redistribution due to the new damage state in each element (i.e. D_k^e) and again a new $N_{F_{min}}$ is predicted. This sequence of "applied" load cycle and predicted cycles is repeated until the structure has failed. For an uncoupled analysis, only one sequence, i.e. one "applied" load cycle, is performed as shown in Fig. 3. The resulting $N_{F_{min}}$ would be used to merely indicate the location of damage initiation.

In preparation for the subsequent "applied" load cycle, the element material properties are degraded according to the newly calculated element damage D_k^e . This is done through the MARC subroutines HOOKLW, for the elastic constants, and ANPLAS, for the anisotropic yield stress ratios using EQS. 17 and 18.

Finally, a subroutine, PATSTR, was also written which generates PATRAN [6] element results files. These files contain damage distributions at specified increments during the fatigue damage analysis. In addition, output files containing a summary table of the current number of fatigue cycles and remaining cycles to failure for each element, and a summary table showing the damage evolution in each element are also generated.

Some general comments on the fatigue damage algorithm need to be made. First, once an element attains the user specified maximum allowable damage, the element is assumed to fail and is no longer considered in any subsequent damage calculations. In the example presented in the next section, a cutoff value of 95% was specified based upon preliminary experience with difficulty in achieving global convergence when the element stiffness was reduced below 5%. Further investigation of convergence difficulties needs to be

addressed. Second, in the present fully coupled damage-deformation analysis, a perfect plasticity model was used. This idealization eliminates the need to account for cyclic hardening, and the corresponding update of the internal variables, which may occur during a specific block of fatigue cycles. In order to accurately account for the hardening, a projection/update of the internal variables through the load block would be required. In addition, it is usually assumed that the fatigue damage calculations are applied to a "stabilized" stress redistribution. Thus, when hardening is present, more than one "applied" load cycle may be necessary in order to achieve the stabilized redistribution.

4.0 Example Application: A Cladded MMC Ring Insert

As stated previously, one of the primary motivations of this research is to establish a computationally efficient method for predicting the fatigue life of typical aerospace components. This includes the ability to predict the location(s) of damage initiation and to be able track the propagation of damage throughout the structure. With this in mind, the fatigue damage algorithm was applied to a cladded MMC ring. The reasons for choosing this specific structure are two-fold. First, it represents a MMC rotor insert currently under consideration in advanced engine designs. Secondly, because of its axisymmetric geometry and load conditions, qualitative stress distributions are known *a priori*. For example, maximum circumferential stress in the core occurs at its inner diameter and likewise for the cladding, thus providing some intuitive feel for where damage initiation will occur as well as how it may propagate.

The composite core was described by Hill's anisotropic elastic-plastic constitutive model available in MARC [5], while the matrix cladding was assumed to be isotropic and elastic perfectly-plastic. The elastic and inelastic material parameters required for the deformation analysis are given in Table 1, while the associated material parameters for the fatigue damage model are given in Table 2. Note that the matrix cladding utilizes the isotropic form of the fatigue damage model, i.e. $\omega_u = \omega_{fl} = \omega_m = \eta_u = \eta_{fl} = \eta_m = 1$, whereas the composite core is represented by the transversely isotropic form of the model. The finite element model, representing the cross-section of the ring, Figure 4, consisted of 225 nodes and 64 8-node axisymmetric elements (MARC element number 28). A uniform pressure load was applied along the inner diameter of the ring.

With regards to the deformation analysis, burst pressure predictions have been previously made and compared to limited experimental data [7,8]. Very good correlation was observed thus providing a level of confidence in the finite element modeling of the ring.

Two types of fatigue life analyses were performed, namely, an uncoupled and a coupled analysis. The uncoupled life prediction results were obtained by taking the stress state in each of the four elements in the radial direction of the composite core of the ring, Figure 4b. Here it was assumed that the stress state was relatively constant in the z-direction, thus one element would represent all of the elements in a column of the composite core. In the uncoupled analysis, no fatigue calculations were performed on the elements associated with the matrix cladding. This is due to the initially low stress levels in the matrix cladding causing infinite fatigue lives to be calculated. As will be shown, it is only in the coupled

analysis that finite lives and damage are predicted for the cladding, due to stress redistribution effects.

Figure 5 shows the results of the uncoupled fatigue damage analysis. As expected, element 1 has the shortest fatigue life, thus, damage is predicted to initiate along the composite core inner diameter. Figure 6 shows the fully coupled deformation and fatigue damage analysis results. By comparing Figs. 5 and 6, one observes that at pressures close to the burst pressure, the fatigue life predicted by the coupled analysis is close to that of the uncoupled analysis, since at high stress levels, once the damage initiates in the core, "structural" failure of the ring occurred rapidly. On the other hand, at low stress levels, the fatigue life as predicted by the coupled analysis is longer than that predicted from the uncoupled analysis. This difference may be viewed as the effect of propagation of the damage in the ring cross-section. This propagation is caused by the stress redistribution effects which are automatically captured by performing a fully coupled deformation and fatigue damage analysis.

Finally, Fig. 7 shows two selected damage distribution plots in the ring cross-section produced by the coupled fatigue damage analysis. Note that in Fig. 7a, the damage initiates along the inner diameter of the composite core. Conversely, in Fig. 7b structural failure of the ring is depicted (i.e. the composite core has completely failed) and due to stress redistribution, the matrix cladding has accumulated significant amounts of damage.

5.0 Summary

A coupled/uncoupled deformation and fatigue damage algorithm has been presented. The algorithm utilizes a multiaxial, isothermal, stress-based, transversely isotropic continuum fatigue damage model in which the fatigue damage calculations are coupled with the nonlinear finite element solution using the concept of effective stress. Incorporated in the life prediction scheme are failure criteria checks at both the element and structural level. The algorithm has been applied to a cladded MMC ring insert representing a typical aerospace component and results have been presented in terms of S-N curves along with damage distribution plots over the ring cross-section. All of the fatigue damage results presented are qualitative in nature since no experimental results are currently available. However, full scale burst pressure and fatigue tests are currently being performed on similar cladded MMC rings under contract with Textron Lycoming, NAS3-27027. Once these test results become available, a similar finite element analysis will be conducted to verify the present fatigue damage algorithm and continuum fatigue damage model.

Acknowledgment

The first author would like to acknowledge support for this work under a cooperative grant, NCC3-248, at NASA Lewis Research Center.

6.0 References

1. Arnold, S. M., and Kruch, S.: Differential Continuum Damage Mechanics Models for Creep and Fatigue of Unidirectional Metal Matrix Composites, NASA TM-105213, 1991.
2. Lemaitre, J. and Chaboche, J. L., Mechanics of Solid Materials, Cambridge University Press, 1990.
3. Chaboche, J. L.: Continuum Damage Mechanics: Part I - General Concepts, *J. Appl. Mech.*, Vol. 55, 1988, pp. 59-64.
4. Chaboche, J. L.: Continuum Damage Mechanics: Part II - Damage Growth, Crack Initiation and Crack Growth, *J. Appl. Mech.*, Vol. 55, 1988, pp. 65-72.
5. MARC, Revision K.5, Volume D: User Subroutines, MARC Analysis Research Corporation, Palo Alto, CA.
6. PATRAN User Manual
7. Arnold, S. M., and Wilt, T. E.: A Deformation and Life Prediction Of A Circumferentially Reinforced SiC/Ti 15-3 Ring, DE-Vol. 55, Reliability, Stress Analysis and Failure Prevention, Ed. R. J. Schaller, 1993, pp. 231-238.
8. Wilt, T. E., and Arnold, S. M.: A Computationally-Coupled Deformation and Damage Finite Element Methodology, HITEMP Review 1993, Vol. II, NASA CP19117, pp. 35:1-15

**TABLE 1. Material Properties For Deformation Model, Ref. [7]
(MPa)**

SiC/Ti 15-3 Composite Material

Elastic: (1 denotes fiber direction)
 $E_1 = 183959.$ $E_2 = E_3 = 114457.$ $\nu_{12} = 0.28$ $\nu_{13} = \nu_{23} = 0.32$

Inelastic:
 $\sigma_y = 276.$ $\frac{\sigma_{y1}}{\sigma_y} = 5.$ $\frac{\sigma_{y2}}{\sigma_y} = \frac{\sigma_{y3}}{\sigma_y} = 1.$

Matrix Material (Ti 15-3)

Elastic:
 $E = 74466.$ $\nu = 0.32$

Inelastic:
 $\sigma_y = 514.$

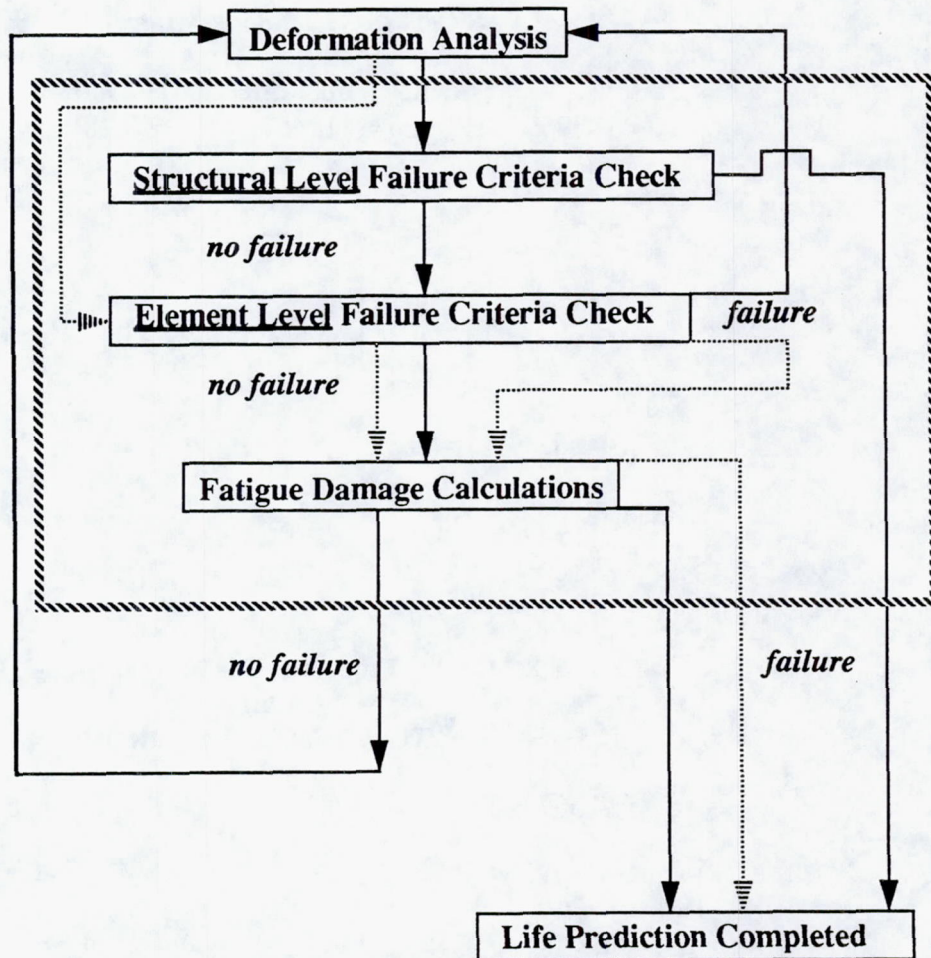
**TABLE 2. Material Properties For Fatigue Model, Ref. [7]
(MPa)**

SiC/Ti 15-3 Composite Material


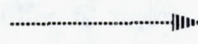
$\sigma_u = 10694.$ $\omega_u = 5.5$
 $\sigma_{fl} = 1972.$ $\omega_{fl} = 12.482$
 $\beta = 1.842$ $\omega_m = 11.8$
 $a = 0.012$ $\eta_u = \eta_{fl} = \eta_m = 1.0$
 $M = 22371.$

Matrix Material (Ti 15-3) - Isotropic Simplification

$\sigma_u = 6081.$ $\omega_u = 1.0$
 $\sigma_{fl} = 965.$ $\omega_{fl} = 1.0$
 $\beta = 2.27$ $\omega_m = 1.0$
 $a = 0.0365$ $\eta_u = \eta_{fl} = \eta_m = 1.0$
 $M = 6205.$



where type of arrow indicates

 Coupled Analysis Path
 Uncoupled Analysis Path


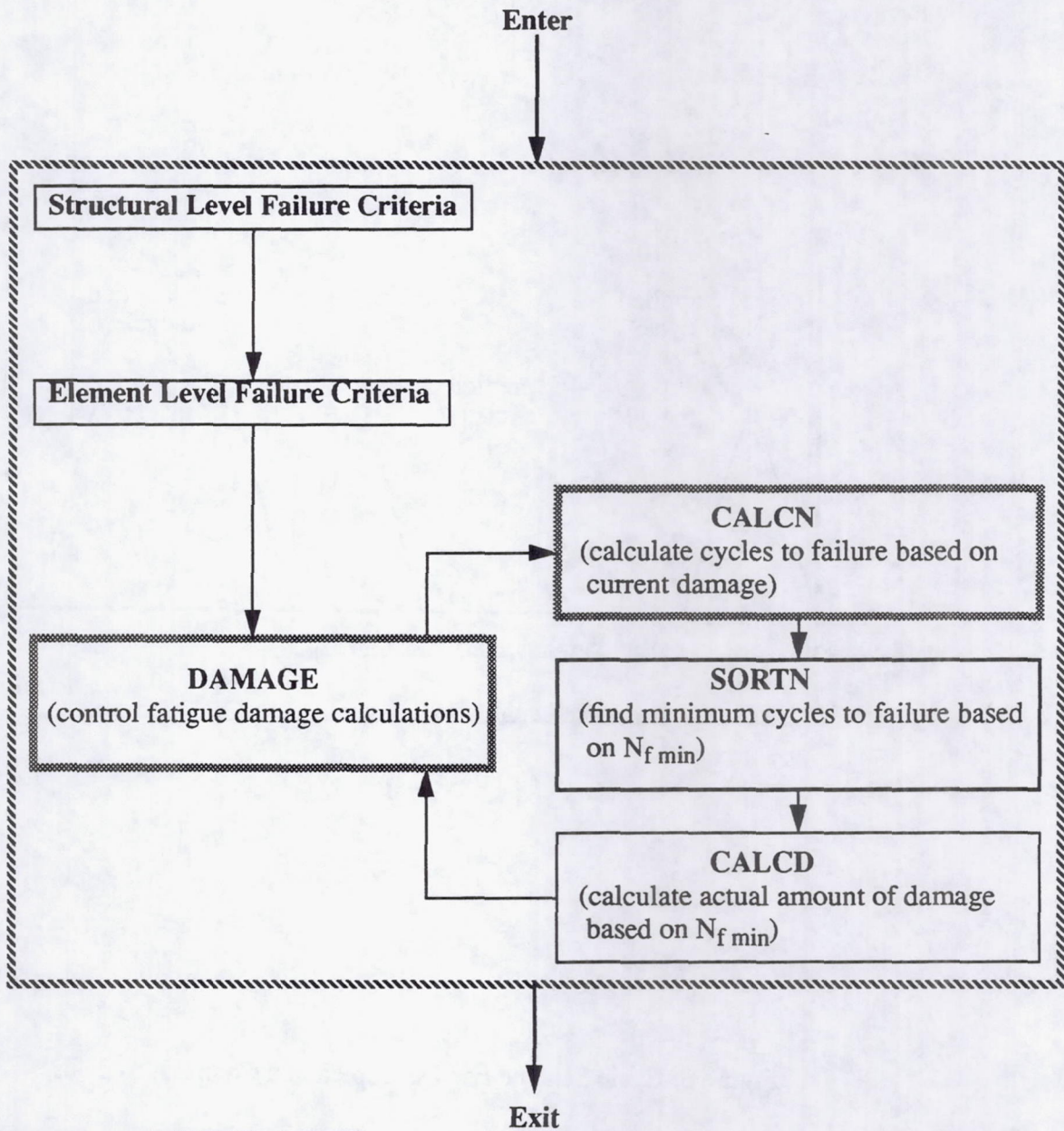
 Box indicates basic modules of life prediction algorithm

Figure 1: Coupled/Uncoupled Life Prediction Scheme

SUBROUTINE ELEVAR



Boxes contain code found in Appendix I

Figure 2: ELEVAR Subroutine Calculations

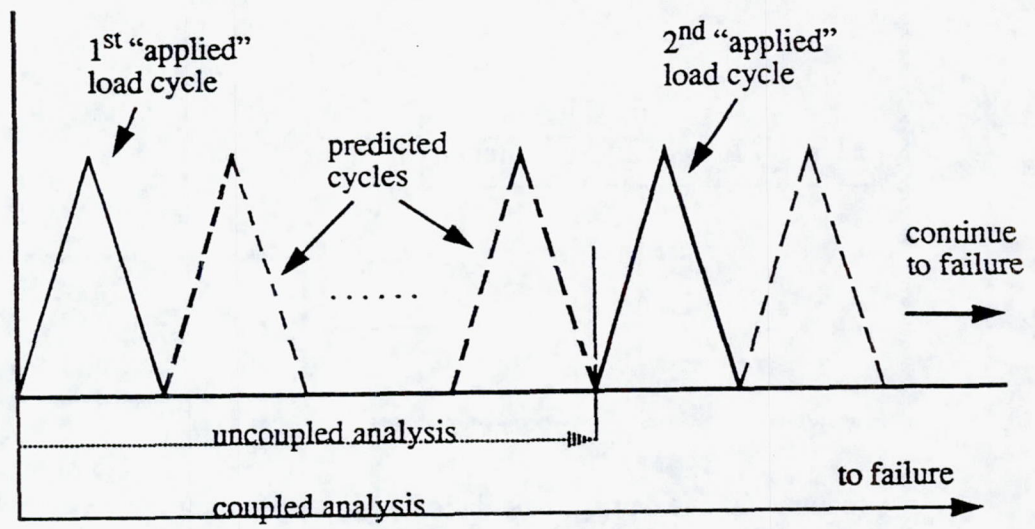
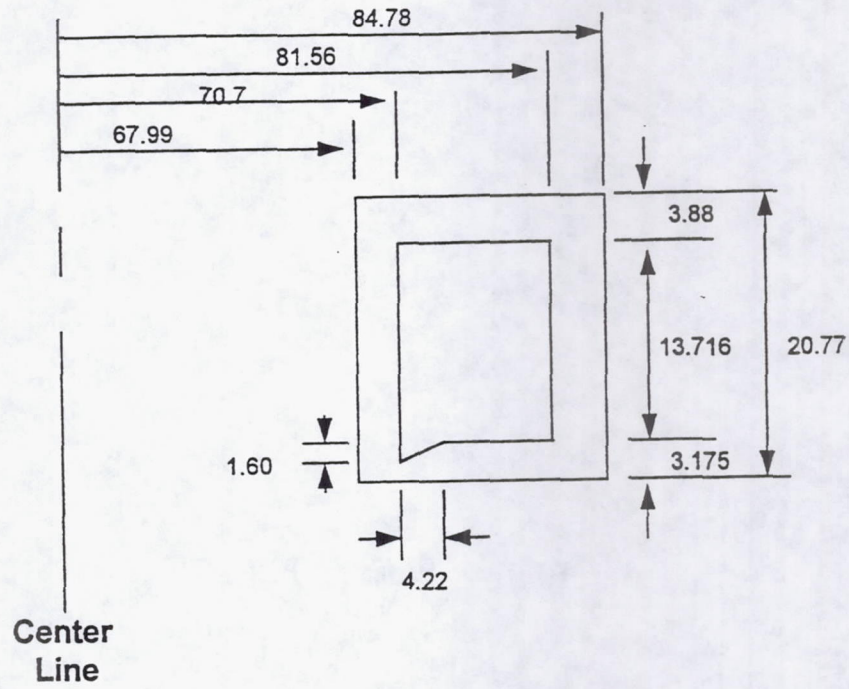
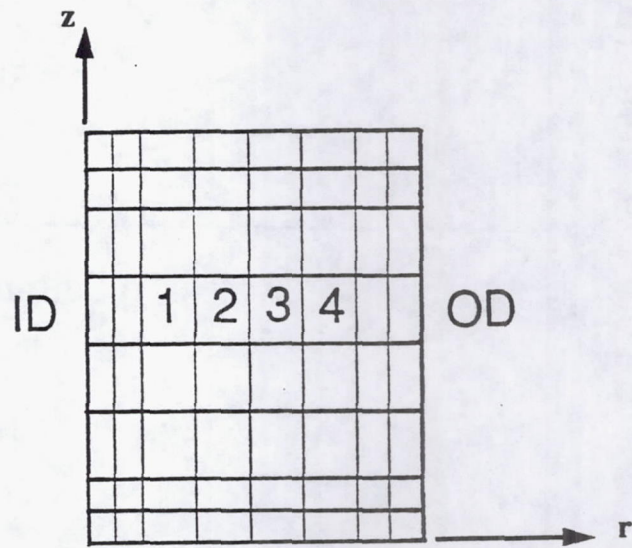


Figure 3: Cycle Scheme For Uncoupled and Coupled Analysis



a) Actual Ring Geometry
(dimensions are in mm.)



b) Idealized Ring Geometry

Figure 4: Cladded MMC Ring Geometry and Finite Element Model

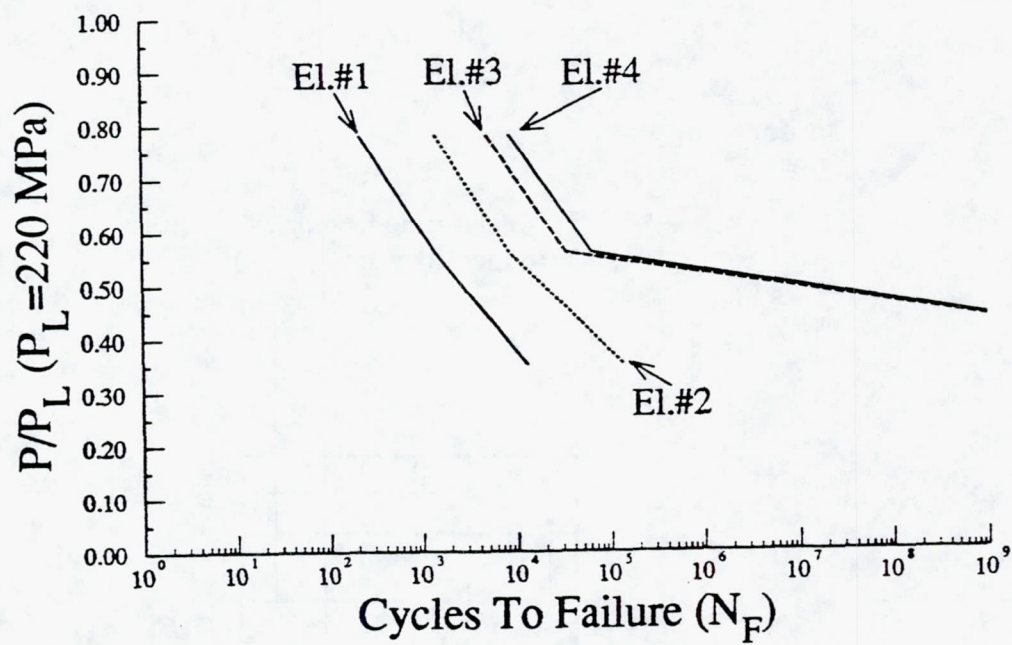


Figure 5: Life Prediction For Uncoupled Analysis

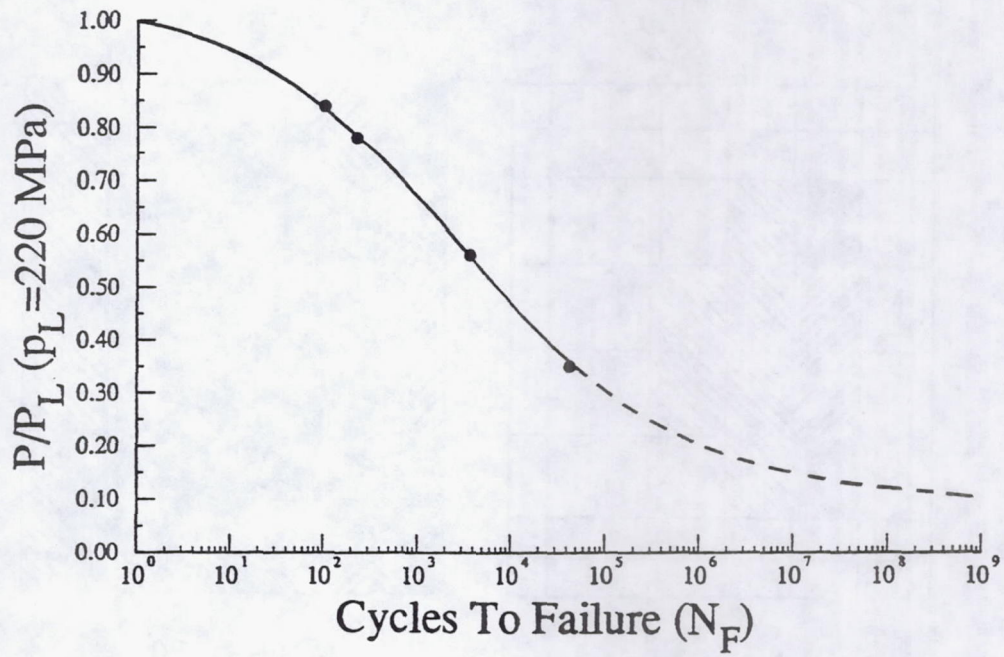
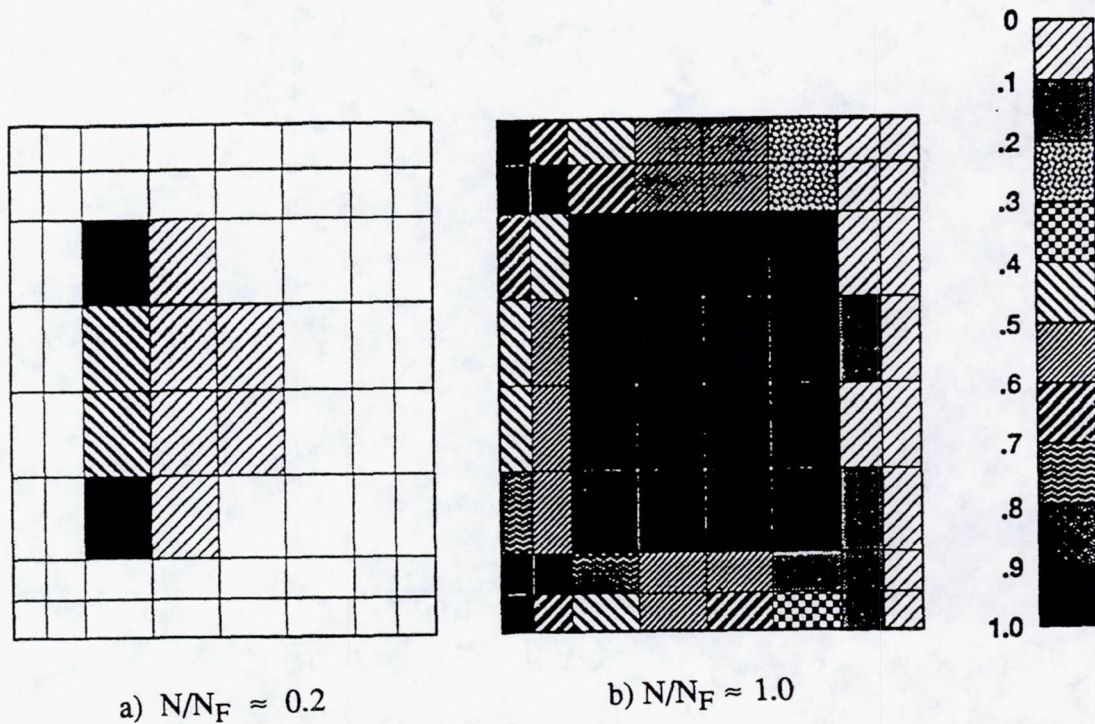


Figure 6: Life Prediction For Coupled Analysis



Cycles to failure, $N_F = 155$

Figure 7: Fatigue Damage Distribution in Ring Cross Section

APPENDIX I:
Fatigue Damage Subroutine

```

C#####
C      SUBROUTINE DAMAGE(FUMAX,FFLMAX,FMMAX,XNF,WAA2,
C      &                D0,DELD,DDN,ITOT)
C
C      PURPOSE: PERFORM FATIGUE DAMAGE CALCULATIONS
C
C      INPUT: STRESS HISTORY FOR EACH ELEMENT (PASSED
C             THROUGH ARRAY WAA2)
C             DAMAGE INCREMENT, DELD
C             NUMBER OF INCREMENTS IN "APPLIED" LOAD CYCLE, ITOT
C             PREVIOUS AMOUNT OF DAMAGE, D0
C
C      OUTPUT: CURRENT NUMBER OF CYCLES TO FAILURE, DDN,
C              BASED UPON DAMAGE INCREMENT, DELD
C
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      LOGICAL IDAM,FALT
C
C      DIMENSION AMP(6),WAA2(126)
C
C-----
C MATERIAL CONSTANTS
C-----
C
C      ANGDEG = 0.0
C      ANGRAD = (3.141592654/180.)*ANGDEG
C      OMU = 1.
C      OMFL = 1.
C      OMM = 1.
C      ETAU = 1.
C      ETAFL = 1.
C      ETAM = 1.
C      BETA = 2.27
C      DENM = 1.
C      A = 0.2302
C      SIGFL = 20.3
C      XML = 900.
C      SIGU = 128.
C
C
C LOOP OVER
C TOTAL NUMBER OF INCREMENTS, ITOT
C
C      CALL ZEROR(AMP,6)

```

```

DO 50 J=1,ITOT
DO 55 K=1,6
  INDX = ((J-1)*6)+K
  AMP(K) = WAA2(INDX)
55 CONTINUE
C
C-----
C CALCULATE F_U
C-----
C
  CALL FFUNC(AMP,ANGRAD,OMU,ETAU,SIGU,FU)
C
  IF(FU.GT.FUMAX) THEN
    FUMAX = FU
    FFU = 1.0 - FUMAX
  ENDIF
50 CONTINUE
C
C LOOP OVER
C TOTAL NUMBER OF INCREMENTS, ITOT
C
  DO 100 I=1,ITOT-1
  JS=I+1
  DO 110 J=JS,ITOT
  CALL ZEROR(AMP,6)
  DO 120 K=1,6
    INDX1 = ((I-1)*6)+K
    INDX2 = ((J-1)*6)+K
    AMP(K) = WAA2(INDX1) - WAA2(INDX2)
120 CONTINUE
C
C-----
C CALCULATE F_FL
C-----
C
  CALL FFUNC(AMP,ANGRAD,OMFL,ETAFL,SIGFL,FFL)
C
  IF(FFL.GT.FFLMAX) THEN
    FFLMAX = FFL
    FFFL = 0.5*FFLMAX - 1.0
  ENDIF
C
C-----
C CALCULATE F_M
C-----
C
  CALL FFUNC(AMP,ANGRAD,OMM,ETAM,XML,FM)
C
  IF(FM.GT.FMMAX) THEN
    FMMAX = FM
    FFM = 0.5*FMMAX
  ENDIF
C
110 CONTINUE

```

```

100      CONTINUE
C
C-----
C <_> FUNCTION
C-----
C
      IF(FFFL.LT.0.0) THEN
          FFFLH = 0.0
      ELSEIF(FFFL.GE.0.0) THEN
          FFFLH = FFFL
      ENDIF
      IF(FFU.LT.0.0) THEN
          FFUH = 0.0
      ELSEIF(FFU.GE.0.0) THEN
          FFUH = FFU
      ENDIF

C
C-----
C CALCULATE ALPHA
C-----
C
      NFAIL = 0
      IF(FFUH.NE.0.00) THEN
          ALPHA = A*(FFFLH/FFUH)

C-----
C IF FFUH=0 FAILURE CASE
C-----
      ELSE
          NFAIL = 1
          XNF = -999.
      ENDIF

C
C-----
C IDAM=.FALSE. IF NO PRIOR DAMAGE EXISTS
C IDAM=.TRUE. IF PRIOR DAMAGE EXISTS
C-----
C
      IDAM = .FALSE.
      IF(DO.NE.0.0) THEN
          IDAM = .TRUE.
      ENDIF
      FALT = .FALSE.
      IF(IDAM .AND. ALPHA.EQ.0.) THEN
          FALT = .TRUE.
      ELSEIF(.NOT.IDAM .AND. ALPHA.EQ.0.) THEN
          XNF = 1.E9
      ENDIF

C
      DSTAR = D0 + DELD
      IF(DSTAR.GT.0.95) DSTAR = 0.95

C
      IF(XNF .NE. -999.) THEN
          CALL CALCN(BETA,DSTAR,D0,ALPHA,FFM,FALT,XNF,DDN)
      ENDIF

```

```

C
C
      STOP
      END
C#####
      SUBROUTINE FFUNC(STRS,ANG,OMEGA,ETA,DENM,F)
C
C      PURPOSE: CALCULATE; F_U F_FL F_M
C
C      CALLED FROM: DAMAGE
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      DIMENSION STRS(6)
C
C-----
C CALCULATE INVARIANTS
C-----
C
      CALL INVAR(STRS,ANG,XI1,XI2,XI3)
C
      A = 1.0/(DENM*DENM)
      B = 4.0*OMEGA*OMEGA-1.0
      C = B/(ETA*ETA)
C
      F = DSQRT(A*(B*XI1+C*XI2+(9./4.)*XI3))
C
      RETURN
      END
C#####
      SUBROUTINE INVAR(SIG,ANG,XI1,XI2,XI3)
C
C      PURPOSE: CALCULATE INVARIANTS; I_1 I_2 I_3
C
C      CALLED FROM: FFUNC
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      DIMENSION SIG(6)
C
      D1 = DCOS(ANG)
      D2 = DSIN(ANG)
      D3 = 0.0
      D11 = D1*D1
      D22 = D2*D2
      D33 = D3*D3
      D12 = D1*D2
      D23 = D2*D3
      D13 = D1*D3
C
      PRESS = (SIG(1)+SIG(2)+SIG(3))/3.
      S11 = SIG(1) - PRESS
      S22 = SIG(2) - PRESS
      S33 = SIG(3) - PRESS

```

```

S12 = SIG(4)
S23 = SIG(5)
S13 = SIG(6)
C
C-----
C [S_ij]*[S_ij]
C-----
C
      XJ2 = 0.5*(S11*S11 + S22*S22 + S33*S33 + 2.*S12*S12
& + 2.*S23*S23 + 2.*S13*S13)
C
C-----
C [D_ij]*[S_ij]
C-----
C
      XI = D11*S11 + D22*S22 + D33*S33 + 2.*(D12*S12 +
& D23*S23 + D13*D13)
C
C-----
C [D_ij]*[S_jk]*[S_ki]
C-----
C
      XIH = D11*(S11*S11+S12*S12+S13*S13) +
& D22*(S12*S12+S22*S22+S23*S23) +
& D33*(S13*S13+S23*S23+S33*S33) +
& 2.*D12*(S11*S12+S22*S12+S13*S23) +
& 2.*D13*(S11*S13+S12*S23+S33*S13) +
& 2.*D23*(S12*S13+S22*S23+S23*S33)
C
      XI3 = XI**2
      XI1 = XJ2 - XIH + 0.25*XI3
      XI2 = XIH - XI3
C
      RETURN
      END
C#####
      SUBROUTINE ZEROR(A,IDIM)
C
C      PURPOSE: ZERO AN ARRAY
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      DIMENSION A(IDIM)
C      DO 100 I=1,IDIM
C        A(I) = 0.0
100    CONTINUE
      RETURN
      END
C#####
      SUBROUTINE CALCN(BETA,D,D0,ALPHA,FFM,FALT,XNF,DDN)
C
C      IMPLICIT REAL*8 (A-H,O-Z)
C
C      PURPOSE: CALCULATE CYCLES TO FAILURE

```


C
C
C
C
C

 CALLED FROM: DAMAGE

 LOGICAL FALT

```
          XFAC1 = 1. - (1. - D)**(BETA+1.)
          XFAC2 = 1. - (1. - D0)**(BETA+1.)
          IF(XNF.LT. 1.E9) THEN
            IF(.NOT. FALT) THEN
              DENOM = (ALPHA*(BETA+1.)*(FFM**BETA))
              DDN = ((XFAC1**ALPHA) - (XFAC2**ALPHA))/DENOM
              IF(DDN.LT.1) THEN
                XNF = -999.
              ENDIF
            ELSEIF(FALT) THEN
              DDN = (LOG(XFAC1) - LOG(XFAC2))/((BETA+1)*(FFM**BETA))
              IF(DDN.LT.1) THEN
                XNF = -999.
              ENDIF
            ENDIF
          ELSEIF(XNF.EQ. 1.E9) THEN
            DDN = 1.E9
          ENDIF
```

C

 RETURN

 END

C#####

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (<i>Leave blank</i>)	2. REPORT DATE March 1994	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE A Coupled/Uncoupled Deformation and Fatigue Damage Algorithm Utilizing the Finite Element Method		5. FUNDING NUMBERS WU-505-63-12	
6. AUTHOR(S) Thomas E. Wilt and Steven M. Arnold		7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		8. PERFORMING ORGANIZATION REPORT NUMBER E-8652	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-106526	
11. SUPPLEMENTARY NOTES Steven M. Arnold, NASA Lewis Research Center and Thomas E. Wilt, University of Toledo, Toledo, Ohio 43606 (work funded by NASA Cooperative Agreement NCC3-248) and Resident Research Associate at the Lewis Research Center. Responsible person, Steven M. Arnold, organization code 5220, (216) 433-3334.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 39		12b. DISTRIBUTION CODE	
13. ABSTRACT (<i>Maximum 200 words</i>) A fatigue damage computational algorithm utilizing a multiaxial, isothermal, continuum based fatigue damage model for unidirectional metal matrix composites has been implemented into the commercial finite element code MARC using MARC user subroutines. Damage is introduced into the finite element solution through the concept of effective stress which fully couples the fatigue damage calculations with the finite element deformation solution. An axisymmetric stress analysis was performed on a circumferentially reinforced ring, wherein both the matrix cladding and the composite core were assumed to behave elastic-perfectly plastic. The composite core behavior was represented using Hill's anisotropic continuum based plasticity model, and similarly, the matrix cladding was represented by an isotropic plasticity model. Results are presented in the form of S-N curves and damage distribution plots.			
14. SUBJECT TERMS Fatigue; Continuum damage; Isothermal; Anisotropic; Inelastic; Finite element method		15. NUMBER OF PAGES 26	
		16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT