INL/EXT-08-14111

New Mechanical Model for the Transmutation Fuel Performance Code

Gregory K. Miller

April 2008

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-08-14111

New Mechanical Model for the Transmutation Fuel Performance Code

Gregory K. Miller

April 2008

Idaho National Laboratory
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

INTENTIONALLY BLANK

New Mechanical Model for the Transmutation Fuel Performance Code

INL/EXT-08-14111
Revision 0

April 2008

Approved by:

Pavel Medvedev for G. Miller per e-mail — 4/16/08
Gregory K. Miller — Date
Author

Pavel Medvedev — 4/17/08
Pavel Medvedev — Date
Principal Investigator

S C Hayes — 4/18/08
Steven Hayes — Date
GNEP Fuels Tech Lead for Irradiation Testing

INTENTIONALLY BLANK

SUMMARY

A new mechanical model has been developed for implementation into the TRU fuel performance code. The new model differs from the existing FRAPCON-3 model, which it is intended to replace, in that it will include structural deformations (elasticity, plasticity, and creep) of the fuel. Also, the plasticity algorithm is based on the "plastic strain–total strain" approach, which should allow for more rapid and assured convergence. The model treats three situations relative to interaction between the fuel and cladding: (1) an open gap between the fuel and cladding, such that there is no contact, (2) contact between the fuel and cladding where the contact pressure is below a threshold value, such that axial slippage occurs at the interface, and (3) contact between the fuel and cladding where the contact pressure is above a threshold value, such that axial slippage is prevented at the interface.

The first stage of development of the model included only the fuel. In this stage, results obtained from the model were compared with those obtained from finite element analysis using ABAQUS on a problem involving elastic, plastic, and thermal strains. Results from the two analyses showed essentially exact agreement through both loading and unloading of the fuel. After the cladding and fuel/clad contact were added, the model demonstrated expected behavior through all potential phases of fuel/clad interaction, and convergence was achieved without difficulty in all plastic analysis performed.

The code is currently in stand-alone form. Prior to implementation into the TRU fuel performance code, creep strains will have to be added to the model. The model will also have to be verified against an ABAQUS analysis that involves contact between the fuel and cladding.

INTENTIONALLY BLANK

CONTENTS

FIGURES

TABLES

ACRONYMS

ARR	Advanced Recycle Reactor
GNEP	Global Nuclear Energy Partnership
TRU	transuranic

INTENTIONALLY BLANK

New Mechanical Model for the Transmutation Fuel Performance Code

1. INTRODUCTION

One of the objectives of the Global Nuclear Energy Partnership (GNEP) is to facilitate the licensing and operation of Advanced Recycle Reactors (ARRs) for transmutation of the transuranic (TRU) elements present in spent fuel. A TRU fuel performance code will be an essential element in the licensing process ensuring that behavior of the transmutation fuel elements in the reactor is understood and predictable. Even more important in the near term, a fuel performance code will assist substantially in the fuels research and development, design, irradiation testing, and interpretation of the post-irradiation examination results.

The ultimate goal of the nuclear fuel behavior modeling is to predict the length of time that the cladding performs its primary function of separating the coolant from the fuel given the geometry of the fuel element, the initial materials properties of the fuel and cladding, and the fuel pin power history. Historically, the nuclear industry has relied on fuel performance codes to model fuel behavior in nuclear reactors.

Within the TRU fuel performance code, a mechanical model is used to determine stresses and deformations of the fuel and cladding. Deformations of the fuel include the effects of fuel thermal expansion, swelling, densification, and restructuring. A new mechanical model is under development that also allows for elastic, plastic, and creep behavior of the fuel caused by internal pressure, temperature variations, and potential contact with the cladding.

The new model is intended to replace the rigid-fuel model that is currently contained in the TRU fuel performance code. The rigid-fuel model (which comes from the FRAPCON-3 code) calculates axial and radial displacements, crack volume, and porosity of the fuel. It assumes that thermal expansion (with swelling and densification) and fuel restructuring are the only sources of fuel deformation, and that the cladding provides no resistance to fuel expansion. The FRAPCON-3 model does allow for elastic, plastic, and creep deformation of the cladding, and allows for contact between the fuel and cladding.

The new deformable-fuel model will include elastic, plastic, and creep deformations of the fuel in addition to thermal expansion, swelling, and densification. It will calculate stresses, strains, and displacements in both the fuel and cladding, and will include all deformations of the cladding that are currently contained in the FRAPCON-3 model.

The model as described herein contains all of the capabilities mentioned, except that creep deformations are not yet included. Also, strains in the fuel caused by swelling and densification of the fuel are not yet explicitly included in the equations. Because these strains behave like thermal strains, though, they can very readily be added.

This report describes all of the theory behind the new model, including the differing conditions associated with fuel/clad contact. It describes the interaction of the mechanical model with the remainder of the TRU fuel performance code, and identifies the input data needed to execute the model. It also provides a comparison with finite element analysis when only the fuel portion of the model is considered. Appendix A identifies specific components of the fundamental matrices in the model, whereas Appendix B presents source code for the model. The model currently exists as a stand-alone program, that, when properly verified, will be implemented into the TRU fuel performance code.

2. NOMENCLATURE

σ	stress component
ε	strain component
E	elastic modulus
ν	Poisson's ratio
a	area of a region (between nodes) in the model
r	radius at a node
gap_{ini}	initial gap between the fuel and cladding
α	coefficient of thermal expansion
ΔT	change in temperature during an increment
ε^P	component of plastic strain
ε^{cr}	component of creep strain
ε^{sw}	component of swelling strain
$d\varepsilon^P$	component of plastic strain increment
P_I	pressure internal to the inner radius of the fuel (if any)
P_O	gas pressure
P_E	pressure external to the outer radius of the cladding
F_z	total axial force across the fuel
ε_{et}	equivalent modified total strain
$d\varepsilon_p$	effective plastic strain increment
ε'	component of modified total strain
σ_e	equivalent stress or yield strength
$d\sigma_e/d\varepsilon_p$	plastic modulus of stress-strain curve
u	displacement in the radial direction
w	displacement in the axial direction

2.1 Subscripts

r	radial direction
θ	tangential direction
z	axial direction
N	number of nodes in fuel portion of model
i	node number
j	region number

3. BASIC EQUATIONS FOR ONE-DIMENSIONAL (AXISYMMETRIC) CYLINDRICAL COORDINATES

As presented in Reference 1, the equations governing the stress-strain behavior of either the fuel or cladding are:

Equilibrium

$$\frac{d\sigma_r}{dr} + \frac{\sigma_r - \sigma_\theta}{r} = 0 \tag{1}$$

Strain compatibility

$$\frac{d\varepsilon_\theta}{dr} + \frac{\varepsilon_\theta - \varepsilon_r}{r} = 0 \tag{2}$$

Stress-strain relations (including elasticity, thermal expansion, swelling, plasticity, and creep)

$$\varepsilon_r = \frac{1}{E}\left[\sigma_r - \nu\left(\sigma_\theta + \sigma_z\right)\right] + \int \alpha_r dT + \varepsilon_r^{sw} + \varepsilon_r^P + d\varepsilon_r^P + \varepsilon_r^{cr} \tag{3}$$

$$\varepsilon_\theta = \frac{1}{E}\left[\sigma_\theta - \nu\left(\sigma_r + \sigma_z\right)\right] + \int \alpha_\theta dT + \varepsilon_\theta^{sw} + \varepsilon_\theta^P + d\varepsilon_\theta^P + \varepsilon_\theta^{cr} \tag{4}$$

$$\varepsilon_z = \frac{1}{E}\left[\sigma_z - \nu\left(\sigma_r + \sigma_\theta\right)\right] + \int \alpha_z dT + \varepsilon_z^{sw} + \varepsilon_z^P + d\varepsilon_z^P + \varepsilon_z^{cr} \tag{5}$$

With loading applied incrementally, the $d\varepsilon_j^P$ are plastic strain increments. These equations contain creep strains, which are not currently included in the model. They also include swelling strains, which are not yet explicitly in the model, but behave identically to the thermal strains. All of these strain components will be included in the final model.

Because this is a one-dimensional model, an assumption must be made relative to behavior in the axial (z) direction. The assumption used in Reference 1, and that is commonly used for a structure having an extensive length relative to its radial dimension, is that of generalized plane strain, i.e.:

ε_z = constant (6)

As described in Reference 1, the fuel can be divided into annular regions and a finite difference approach can be applied. With this approach, Equations (1)–(6) are combined into a matrix equation

$$\{\sigma\}_{i+1} = [L]_i \{\sigma\}_i + \{M\}_i \tag{7}$$

where i = node number. This equation transfers the stresses at node i to stresses at node $i + 1$. The development of the matrices L and M is as described in Reference 1.

Additionally, successive application of Equation (7) from node 1 forward results in the following:

$$\{\sigma\}_{i+1} = [A]_i \{\sigma\}_1 + \{B\}_i \tag{8}$$

This equation transfers the stresses at node 1 to the stresses at any node $i + 1$, which is particularly useful for applying boundary conditions as discussed below.

4. BOUNDARY CONDITIONS

Solving Equation (7) requires that appropriate boundary conditions be established. Prior to contact with the cladding, the radial stress at the outer surface of the fuel is set equal to the gas pressure. Using the convention that tensile stresses are positive, this is expressed as follows:

$$\sigma_{r,N} = -P_O \tag{9}$$

A second condition is that the radial and tangential stresses are equal at the center of the fuel if the fuel is a solid cylinder (due to equilibrium requirements)

$$\sigma_{r,1} = \sigma_{\theta,1} \qquad \text{if } r_1 = 0 \tag{10}$$

or the radial stress at the inner surface of the fuel is equal to the internal pressure if there is an opening in the fuel

$$\sigma_{r,1} = -P_I \qquad \text{if } r_1 \neq 0 \tag{11}$$

The final condition is that the sum of the axial forces acting on each region j in the model is equal to the net axial force acting across the fuel:

$$\sum_{j=1}^{N-1} \frac{\left(\sigma_{z,j} + \sigma_{z,j+1}\right)}{2} a_j = F_z \tag{12}$$

Generally this axial force F_z is expected to be small or zero. The boundary conditions will be used to solve for the three stress components at node 1. The first boundary condition is readily applied using Equation (8):

$$-P_O = A_{11,N-1}\sigma_{r,1} + A_{12,N-1}\sigma_{\theta,1} + A_{11,N-1}\sigma_{z,1} + B_{1,N-1} \tag{13}$$

The second boundary condition involves only stresses at node 1, so is already in the form needed.

To apply the third condition, Equation (8) is used to write the axial stress in each of the regions in terms of the stresses at node 1.

The three boundary conditions then result in a matrix equation

$$[C]\{\sigma\}_1 = \{D\} \tag{14}$$

which can be solved for the stresses σ_r, σ_θ, and σ_z at node 1. The transfer matrices A and B are then used to calculate stresses throughout the fuel.

5. PLASTIC STRAIN INCREMENTS

Plastic strain increments are determined using the Prandtl-Reuss relations. These relations generally express the strain increments in terms of stresses. However, Reference 2 recommends use of the "plastic strain–total strain" method for reaching more rapid and assured convergence in a plastic analysis. With this method, the plastic strain increments are written in terms of modified total strains as follows:

$$d\varepsilon_r^P = \frac{d\varepsilon_P}{3\varepsilon_{et}}\left(2\varepsilon_r' - \varepsilon_\theta' - \varepsilon_z'\right) \tag{15}$$

$$d\varepsilon_\theta^P = \frac{d\varepsilon_P}{3\varepsilon_{et}}\left(2\varepsilon_\theta' - \varepsilon_r' - \varepsilon_z'\right) \tag{16}$$

$$d\varepsilon_z^P = -d\varepsilon_r^P - d\varepsilon_\theta^P \tag{17}$$

where

$$\varepsilon_r' = \varepsilon_r - \varepsilon_r^P \tag{18}$$

$$\varepsilon_\theta' = \varepsilon_\theta - \varepsilon_\theta^P \tag{19}$$

$$\varepsilon_z' = \varepsilon_z - \varepsilon_z^P \tag{20}$$

$$\varepsilon_{et} = \frac{\sqrt{2}}{3}\sqrt{\left(\varepsilon_r' - \varepsilon_\theta'\right)^2 + \left(\varepsilon_r' - \varepsilon_z'\right)^2 + \left(\varepsilon_\theta' - \varepsilon_z'\right)^2} \tag{21}$$

$$d\varepsilon_P = \frac{\varepsilon_{et} - \frac{2}{3}\left[(1+\nu)/E\right]\sigma_{e,k-1}}{1 + \frac{2}{3}\left[(1+\nu)/E\right]\left(d\sigma_e / d\varepsilon_P\right)_{k-1}} \tag{22}$$

where σ_e is the equivalent stress as described below and *k-1* refers to the end of the previous increment. Equation (17) is based on the assumption that there is no volumetric change associated with plastic deformation.

6. METHOD OF SUCCESSIVE APPROXIMATIONS

The model applies the method of successive approximations to solve for plastic deformation, as described in Reference 2. This method involves iterating to a solution for each load increment in the analysis, using the following steps:

1. Assume initial plastic strain increments $d\varepsilon_i^P$ are zero for all nodes. This assumption will be correct at nodes that undergo only elastic deformation.
2. Solve for stresses using Equations (14) and (8).
3. Calculate strains using Equations (3)–(5).
4. Calculate equivalent stress σ_e and compare to current yield strength $\sigma_{e,k-1}$, (Von Mises yield criterion):

$$\sigma_e = \frac{1}{\sqrt{2}}\sqrt{\left(\sigma_r - \sigma_\theta\right)^2 + \left(\sigma_r - \sigma_z\right)^2 + \left(\sigma_\theta - \sigma_z\right)^2}$$

5. Calculate revised plastic strain increments using Equations (15)–(22) for nodes that have yielded ($\sigma_e > \sigma_{e,k-1}$). Use the stress-strain curve to determine the plastic modulus $(d\sigma_e/d\varepsilon_p)_{k-1}$ in Equation (22).
6. Iterate until convergence is reached.
7. Retain σ_e as the new yield strength for the next increment. It was found that, once the converged plastic strain increment for a load increment is determined, the yield strength, σ_e, to be used for the next increment is best determined from

$$\sigma_e = \sigma_{e,k-1} + \left(\frac{d\sigma_e}{d\varepsilon_P}\right)_{k-1} d\varepsilon_P$$

   rather than the Von Mises equation above.
8. Add the plastic strain increments to the accumulated plastic strain, and calculate total strains.
9. Calculate displacements, and update the radii.

7. DISPLACEMENTS

Once the stresses and strain increments are determined, then the total strain components at each node are calculated from Equations (3)–(5). The following strain-displacement equations are then used to calculate displacements in the radial (u) and axial (w) directions for each node,

$$u_i = r_i \varepsilon_{\theta,i} \tag{23}$$

$$dw = \varepsilon_z dz \tag{24}$$

and the radii are updated at the end of each increment as follows

$$r_i = r_{i,0} + u_i \tag{25}$$

where the $r_{i,0}$ are the radii at the beginning of irradiation.

8. NONLINEAR GEOMETRY

In the plastic analyses described above, the radii at the start of a load increment are maintained throughout the increment. After a convergent plastic solution is achieved, the radii are then updated and used as the radii for the following load increment. It may seem that, if nonlinear geometric effects take place, each plastic analysis iteration may have to account for modified radii. It was discovered, though, that accounting for a changing displacement within the plasticity loop had a negligible effect on the resulting strains and displacements even when deformations became large. This indicates that displacements need only be modified after convergence on plasticity, whether deformations are small or large. This is the way the model is constructed, which saves significantly on computational time.

A second consideration relative to nonlinear geometry is that Equations (23) and (24) are based on first-order approximations for the strains $\varepsilon_{\theta,i}$ and ε_z, and are, therefore, small displacement equations. Large deformation equations have an additional term as follows:

$$\varepsilon_r = \frac{du}{dr} - \frac{1}{2}\left(\frac{du}{dr}\right)^2 \tag{26}$$

$$\varepsilon_\theta = \frac{u}{r} - \frac{1}{2}\left(\frac{u}{r}\right)^2 \tag{27}$$

$$\varepsilon_z = \frac{dw}{dz} - \frac{1}{2}\left(\frac{dw}{dz}\right)^2 \tag{28}$$

If large deformations were to occur in fuel rod behavior, these equations may have to be used to accurately capture their effect. Since these equations are inconsistent with the strain compatibility equation (Equation [2]), though, the small deformation equations are currently used in the model. Radii are updated after each time increment per Equation (25). If future developments indicate that large deformations are an issue in fuel rod behavior, the formulation could potentially be modified to incorporate the large deformation equations. This would likely, though, significantly complicate some of the equations.

9. CLADDING AND FUEL/CLADDING CONTACT

The cladding is modeled as one layer with a node located at each of its surfaces. The model also, however, accounts for the gap between the fuel and cladding. The treatment of the gap depends on whether contact occurs between the fuel and cladding, and on the contact stress at the interface if contact has been made. The modeling for three potential scenarios is discussed below.

9.1 Prior to Contact between the Fuel and Cladding

In this phase, the fuel and cladding are treated separately until contact occurs. An L and M matrix are developed for the cladding (in the same way as described in Section 3 for the fuel) which transfer stresses from the inner to the outer surface of the cladding. The gas pressure is applied to the outer surface of the fuel and to the inner surface of the cladding. Any pressure external to the cladding is applied to the outer surface of the cladding. The boundary conditions applied in this phase to the fuel are the same as described in Section 4. Boundary conditions for the cladding are:

$$\sigma_{r,N+1} = -P_O \tag{29}$$

$$\sigma_{r,N+2} = -P_E \tag{30}$$

$$\sigma_{z,N+1} = -\sigma_{z,N+2} \tag{31}$$

The third condition, (Eq.[31]), is based on the assumption that the net axial force on the cladding is essentially zero.

9.2 Contact between the Fuel and Cladding

The fuel must overcome the gas pressure as it makes contact with the cladding. After contact occurs, the contact pressure exceeds the gas pressure, and complete structural bonding between the fuel and cladding is assumed as follows

$$u_N = u_{N+1} + gap_{ini} \tag{32}$$

$$\sigma_{r,N} = \sigma_{r,N+1} \tag{33}$$

$$d\varepsilon_{z,N} = d\varepsilon_{z,N+1} \tag{34}$$

where the strain increments $d\varepsilon_z$ are those occurring after contact is made. The third condition is based on the assumption that no axial slippage between the fuel and cladding occurs during this phase.

These three interface conditions are used to develop an L and M matrix for the gap, which then transfer stresses across the gap to inner surface of the cladding. The boundary conditions for the fuel/cladding system then are:

$$\sigma_{r,N+2} = -P_E \tag{35}$$

$$\sigma_{r,1} = \sigma_{\theta,1} \qquad \text{if } r_1 = 0 \tag{36}$$

$$\sigma_{r,1} = -P_I \qquad \text{if } r_1 \neq 0 \tag{37}$$

$$\sum_{j=1}^{N-1} \frac{\left(\sigma_{z,j} + \sigma_{z,j+1}\right)}{2} a_j + \frac{\sigma_{z,N+1} + \sigma_{z,N+2}}{2} a_{N+1} = F_z \tag{38}$$

These boundary conditions result in a matrix equation

$$[C]\{\sigma\}_1 = \{D\} \tag{39}$$

that is used to calculate the stresses $\sigma_{r,}$, σ_θ, and σ_z at node 1.

The contact stress is compared to the gas pressure with each load increment. If, upon unloading, the contact stress becomes less than the internal pressure, then slippage is allowed. In this case,

$$u_N = u_{N+1} + gap_{ini} \tag{40}$$

$$\sigma_{r,N} = \sigma_{r,N+1} \tag{41}$$

but the strain increments $d\varepsilon_z$ are no longer equated at the fuel/clad interface. For this phase, the following conditions

$$\{\sigma\}_{N+2} = [L]_{N+1}\{\sigma\}_{N+1} + \{M\}_{N+1} \tag{42}$$

$$\sum_{j=1}^{N-1} \frac{\left(\sigma_{z,j} + \sigma_{z,j+1}\right)}{2} a_j = F_z \tag{43}$$

$$u_N = u_{N+1} + gap_{ini} \tag{44}$$

were combined to produce a set of five simultaneous equations in the quantities σ_{r1} ($\sigma_{\theta 1}$, if $r_1 \neq 0$), σ_{z1}, $\sigma_{\theta,N+1}$, $\sigma_{\theta,N+2}$, and $\sigma_{z,N+1}$. In these equations, it was recognized that

$$\sigma_{r,1} = \sigma_{\theta,1} \qquad \text{if } r_1 = 0 \tag{45}$$

$$\sigma_{r,1} = -P_I \qquad \text{if } r_1 \neq 0 \tag{46}$$

Other conditions applied during this phase were:

$$\sigma_{r,N+2} = -P_E \tag{47}$$

$$\sigma_{z,N+1} = -\sigma_{z,N+2} \tag{48}$$

9.3 Re-Opening of Gap between the Fuel and Cladding

If unloading continues to the extent that the contact pressure reaches zero, then the gap between the layers is re-opened. When this occurs, the contact conditions at the interface are released, and the radial stresses at the outer surface of the fuel and inner surface of the cladding are set equal to the gas pressure. The modeling of the fuel and cladding behavior is then returned to what is described in Section 9.1 above.

If reloading occurs, then subsequent contact between the fuel and cladding is modeled as described in Section 9.2.

10. PLACEMENT IN THE TRU CODE

The mechanical model will be placed in the TRU Code as shown in Figure 1.

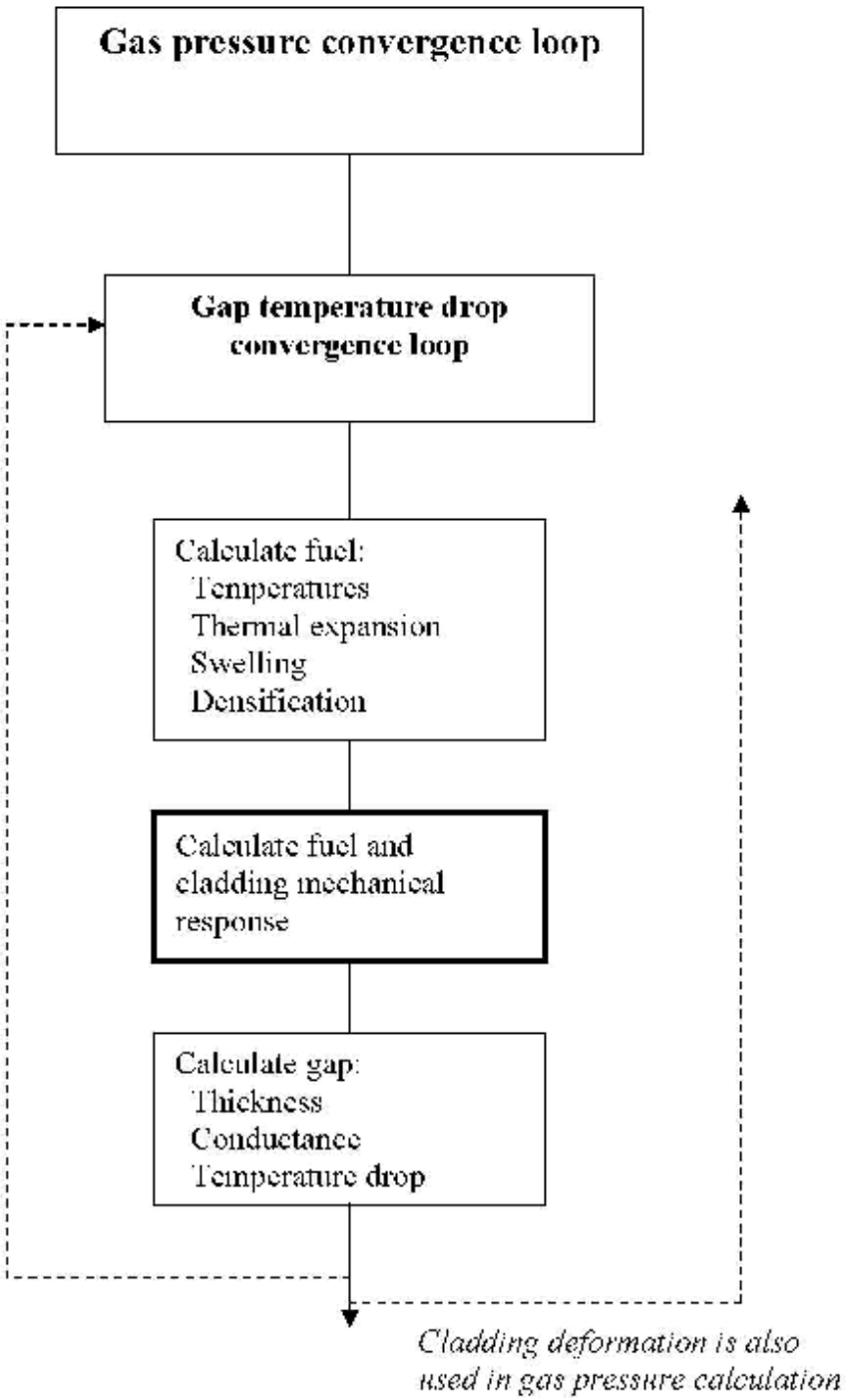

Figure 1. Flow diagram showing placement of the mechanical model in the TRU code.

11. DATA NEEDED FOR EXECUTION OF THE MODEL

Data that must be supplied to the mechanical model at each load increment include:

- α, E, ν, r, σ_e, $d\sigma_e/d\varepsilon_p$, temperature, creep parameters (for each annular region or node)
- gas pressure
- pressure external to the cladding
- total axial force
- internal pressure in fuel (if any).

12. FINITE DIFFERENCE EQUATIONS

Quantities in the finite difference equations followed the pattern shown in the equations below:

$$\frac{d}{dr}\left(\frac{\sigma_\theta}{E}\right)_j = \left(\frac{\sigma_{\theta,j+1}}{E_{j+1}} - \frac{\sigma_{\theta,j}}{E_j}\right)\frac{1}{r_{j+1} - r_j}$$

$$\left(\frac{1+\nu}{E}\frac{\sigma_r}{r}\right)_j = \left(\frac{1+\nu_{j+1}}{E_{j+1}}\sigma_{r,j+1} + \frac{1+\nu_j}{E_j}\sigma_{r,j}\right)\frac{1}{r_{j+1} + r_j}$$

Application of finite difference equations throughout the model results in matrix equations, such as Equations (7) and (8). The components of the fundamental matrices of the model are presented in Appendix A.

13. COMPARISON WITH FINITE ELEMENT ANALYSIS USING ABAQUS

The mechanical model was developed in two stages. The first stage addressed the fuel only, while the second stage brought in the cladding and fuel/cladding contact. After development of the first stage, results obtained from the model were compared with results from finite element analysis using the ABAQUS code. The parameters used in the sample problem for which comparisons were made are summarized in Table 1.

Table 1. Parameters used in sample problem for which comparisons were made.

Parameter	Value						
Outer radius (in.)	0.1						
Nodes (elements)	11 (10)						
Elastic modulus (psi)	1.2e6						
Poisson's ratio	0.3						
Initial yield stress (psi)	1.0e4						
Plastic modulus (psi)	1.0e4 (held constant)						
Coefficient thermal expansion $(°C)^{-1}$	1.6e−5						
	Load			Unload		Reload	
Temperature (°C), center	100	→	700	→	400	→	1,000
Temperature, outer radius	100	→	200	→	150	→	250
External pressure (psi), acting outward	0	→	12,000	→	6,000	→	18,000
Total axial force (lb)	0	→	0	→	0	→	0

NOTE: *Temperature is linear across fuel*

A comparison of the calculated displacement histories for the outer surface of the fuel is shown in Figure 2. A comparison of calculated stress histories at the fuel center is shown in Figure 3, and a comparison of stress profiles across the fuel at the end of the solution is presented in Figure 4. These results show that there is essentially perfect agreement between the two models.

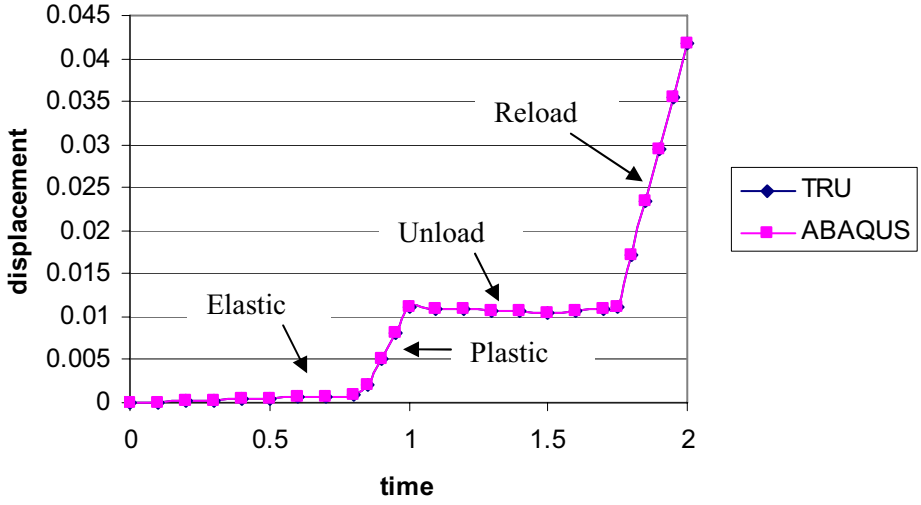

Figure 2. Radial displacement at outer radius.

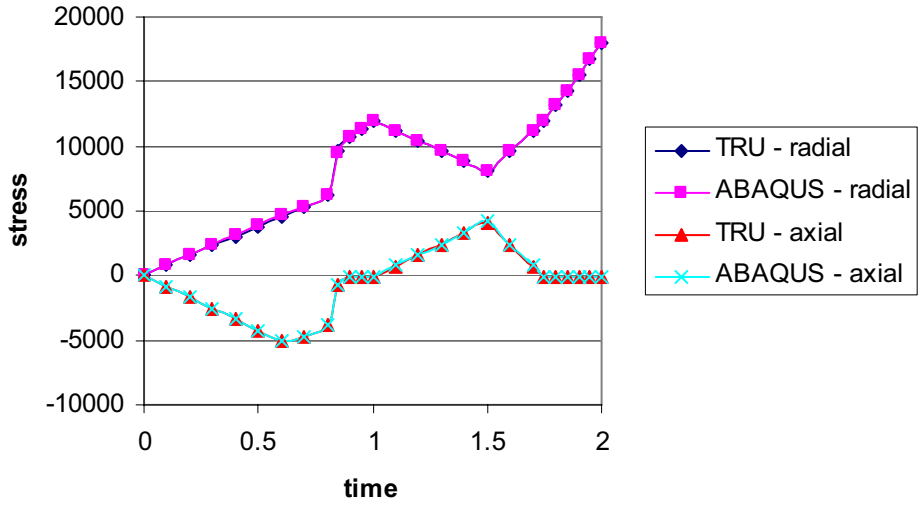

Figure 3. Stresses at fuel center.

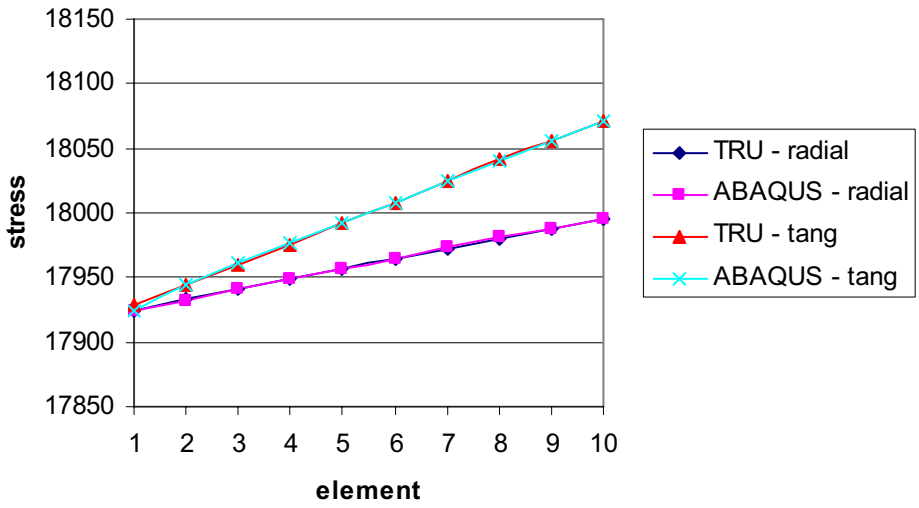

Figure 4. Stress profile at end of solution.

After development of the second stage, where the cladding and fuel/cladding contact were brought into the model, the model was exercised through its full range of capabilities. It was demonstrated to behave as expected (1) prior to contact between the fuel and cladding, (2) during contact between the fuel and cladding as loading continued, (3) during unloading when the contact stress still exceeded the gas pressure, (4) during unloading when the gas pressure exceeded the contact stress and slippage occurred, (5) during unloading after the fuel detached from the cladding, and (6) during reloading after the fuel again contacted the cladding. There were no difficulties encountered in reaching convergence in the plastic analyses performed. The model has not yet been evaluated against finite element analysis using ABAQUS on a problem involving contact between the fuel and cladding.

14. CONCLUSIONS

A new mechanical model has been developed for implementation into the TRU fuel performance code. The new model differs from the existing FRAPCON-3 model, which it is intended to replace, in that it will include structural deformations (elasticity, plasticity, and creep) of the fuel. Also, the plasticity algorithm is based on the "plastic strain–total strain" approach, which should allow for more rapid and assured convergence. The model treats three situations relative to interaction between the fuel and cladding: (1) an open gap between the fuel and cladding, such that there is no contact, (2) contact between the fuel and cladding where the contact pressure is below a threshold value, such that axial slippage occurs at the interface, and (3) contact between the fuel and cladding where the contact pressure is above a threshold value, such that axial slippage is prevented at the interface.

The first stage of development of the model included only the fuel. In this stage, results obtained from the model were compared with those obtained from finite element analysis using ABAQUS on a problem involving elastic, plastic, and thermal strains. Results from the two analyses showed essentially exact agreement through both loading and unloading of the fuel. After the cladding and fuel/clad contact were added, the model demonstrated expected behavior through all potential phases of fuel/clad interaction, and convergence was achieved without difficulty in all plastic analysis performed.

The code is currently in stand-alone form. Prior to implementation into the TRU fuel performance code, creep strains will have to be added to the model. The model will also have to be verified against an ABAQUS analysis that involves contact between the fuel and cladding.

15. REFERENCES

1. Siefken, L. J., C. M. Allison, M. P. Bohn, S. O. Peck, 1981, "FRAP-T6: A Computer Code for the Transient Analysis of Oxide Fuel Rods," EG&G Idaho, Inc., NUREG/CR-2148, May 1981.
2. Mendelson, A., 1983, Plasticity: Theory and Application, Krieger Publishing Co., Reprint Edition.

Appendix A

Components of Fundamental Matrices

Appendix A
Components of Fundamental Matrices

A.1 TRANSFER MATRICES

To perform a finite difference solution to the governing equations, the fuel is represented with *N* nodes, with the first node located at the inner surface or center of the fuel, and the *N*th node located at the outer surface of the fuel. The nodes *N + 1* and *N + 2* are located at the inner and outer surfaces of the cladding, respectively. Equations (1) through (6) can be reduced to a set of three equations for each of the regions between nodes. In this reduction, Equations (3) and (4) are substituted into Equation (2). Equation (6) for any region *j* is written as

$$\varepsilon_{z,j+1} = \varepsilon_{z,j}$$

and Equation (5) is substituted into this equation.

When written in finite difference form (per Section 12 of this report), the three equations result in the following matrix equation, which transfers stresses from node *i* to node *i* + 1

$$[E]_i \{\sigma\}_{i+1} = [F]_i \{\sigma\}_i + \{G\}_i \tag{A.1}$$

where the components of matrices *E*, *F*, and *G* are:

$$E_{11i} = 2r_{i+1}$$

$$E_{12i} = -(r_{i+1} - r_i)$$

$$E_{13i} = 0$$

$$E_{21i} = -\frac{\nu_{i+1}}{E_{i+1}(r_{i+1} - r_i)} - \frac{1+\nu_{i+1}}{E_{i+1}(r_{i+1} + r_i)}$$

$$E_{22i} = \frac{1}{E_{i+1}(r_{i+1} - r_i)} - \frac{1+\nu_{i+1}}{E_{i+1}(r_{i+1} + r_i)}$$

$$E_{23i} = -\frac{\nu_{i+1}}{E_{i+1}(r_{i+1} - r_i)}$$

$$E_{31i} = -\frac{\nu_{i+1}}{E_{i+1}}$$

$$E_{32i} = -\frac{\nu_{i+1}}{E_{i+1}}$$

$$E_{33i} = \frac{1}{E_{i+1}}$$

$$F_{11i} = 2r_i$$

$$F_{12i} = r_{i+1} - r_i$$

$$F_{13i} = 0$$

$$F_{21i} = -\frac{\nu_i}{E_i\left(r_{i+1} - r_i\right)} + \frac{1+\nu_i}{E_i\left(r_{i+1} + r_i\right)}$$

$$F_{22i} = \frac{1}{E_i\left(r_{i+1} - r_i\right)} - \frac{1+\nu_i}{E_i\left(r_{i+1} + r_i\right)}$$

$$F_{23i} = -\frac{\nu_i}{E_i\left(r_{i+1} - r_i\right)}$$

$$F_{31i} = -\frac{\nu_i}{E_i}$$

$$F_{32i} = -\frac{\nu_i}{E_i}$$

$$F_{33i} = \frac{1}{E_i}$$

$$G_{1i} = 0$$

$$\begin{aligned} G_{2i} = &-\left[\left(\alpha_\theta \Delta T\right)_{i+1} - \left(\alpha_\theta \Delta T\right)_i\right]\frac{1}{r_{i+1} - r_i} - \left[\left(\varepsilon_\theta^P\right)_{i+1} - \left(\varepsilon_\theta^P\right)_i\right]\frac{1}{r_{i+1} - r_i} \\ &- \left[\left(d\varepsilon_\theta^P\right)_{i+1} - \left(d\varepsilon_\theta^P\right)_i\right]\frac{1}{r_{i+1} - r_i} - \left[\left(\alpha_\theta \Delta T\right)_{i+1} - \left(\alpha_r \Delta T\right)_{i+1}\right]\frac{1}{r_{i+1} + r_i} - \left[\left(\alpha_\theta \Delta T\right)_i - \left(\alpha_r \Delta T\right)_i\right]\frac{1}{r_{i+1} + r_i} \\ &- \left[\left(\varepsilon_\theta^P\right)_{i+1} - \left(\varepsilon_r^P\right)_{i+1}\right]\frac{1}{r_{i+1} + r_i} - \left[\left(\varepsilon_\theta^P\right)_i - \left(\varepsilon_r^P\right)_i\right]\frac{1}{r_{i+1} + r_i} \\ &- \left[\left(d\varepsilon_\theta^P\right)_{i+1} - \left(d\varepsilon_r^P\right)_{i+1}\right]\frac{1}{r_{i+1} + r_i} - \left[\left(d\varepsilon_\theta^P\right)_i - \left(d\varepsilon_r^P\right)_i\right]\frac{1}{r_{i+1} + r_i} \end{aligned}$$

$$G_{3i} = -\left[\left(\alpha_z \Delta T\right)_{i+1} - \left(\alpha_z \Delta T\right)_i\right] - \left[\left(\varepsilon_z^P\right)_{i+1} - \left(\varepsilon_z^P\right)_i\right] - \left[\left(d\varepsilon_z^P\right)_{i+1} - \left(d\varepsilon_z^P\right)_i\right]$$

The *L* and *M* matrices (see Equation [7]) for a region are obtained by multiplying Equation (A.1) by E^{-1}. The *A* and *B* matrices (see Equation [8]) for a region are obtained by applying Equation (7) successively from node 1 forward.

A.2 BOUNDARY CONDITION EQUATIONS

A.2.1 No Fuel/Clad Contact

When there is no fuel/clad contact, application of the boundary conditions to the fuel results in a matrix equation (see Section 4 of this report)

$$[C]\{\sigma\}_1 = \{D\}$$

where the components of C and D are:

$$C_{11} = A_{11,N-1}$$

$$C_{12} = A_{12,N-1}$$

$$C_{13} = A_{13,N-1}$$

$$C_{21} = 1$$

$$C_{22} = -1 \qquad \text{if } r_1 = 0$$

$$C_{22} = 0 \qquad \text{if } r_1 \neq 0$$

$$C_{23} = 0$$

$$C_{31} = A_{311}\frac{a_1}{2} + \sum_{j=2}^{N-1}\left(A_{31,j-1} + A_{31,j}\right)\frac{a_j}{2}$$

$$C_{32} = A_{321}\frac{a_1}{2} + \sum_{j=2}^{N-1}\left(A_{32,j-1} + A_{32,j}\right)\frac{a_j}{2}$$

$$C_{33} = A_{331}\frac{a_1}{2} + \sum_{j=2}^{N-1}\left(A_{33,j-1} + A_{33,j}\right)\frac{a_j}{2}$$

$$D_1 = -P_O - B_{1,N-1}$$

$$D_2 = 0 \qquad \text{if } r_1 = 0$$

$$D_2 = -P_I \qquad \text{if } r_1 \neq 0$$

$$D_3 = F_z - B_{31}\frac{a_1}{2} - \sum_{j=2}^{N-1}\left(B_{3,j-1} + B_{3,j}\right)\frac{a_j}{2}$$

Application of boundary conditions to the clad results in the following matrix equation

$$[CC]\{\sigma\}_{N+1} = \{DC\}$$

where the second "C" refers to the clad.

The components of the matrices *CC* and *DC* are:

$$CC_{11} = L_{11,N+1}$$

$$CC_{12} = L_{12,N+1}$$

$$CC_{13} = L_{13,N+1}$$

$$CC_{21} = 1$$

$$CC_{22} = 0$$

$$CC_{23} = 0$$

$$CC_{31} = L_{31,N+1}$$

$$CC_{32} = L_{32,N+1}$$

$$CC_{33} = L_{33,N+1} + 1$$

$$DC_1 = -P_E - M_{1,N+1}$$

$$DC_2 = -P_O$$

$$DC_3 = -M_{3,N+1}$$

A.2.2 Fuel/Clad Contact with Axial Slippage

When there is fuel/clad contact, and axial slippage between the fuel and clad is allowed to occur, application of boundary conditions and compatibility conditions at the interface results in the following matrix equation

$$[CCON]\{\sigma\} = \{D\}$$

where the matrix $\boldsymbol{\sigma}$ consists of the five unknown stresses σ_{r1} ($\sigma_{\theta 1}$, if $r_1 \neq 0$), σ_{z1}, $\sigma_{\theta,N+1}$, $\sigma_{\theta,N+2}$, and $\sigma_{z,N+1}$. To solve this equation, the components of *D* are treated as a sixth column of matrix *CCON*, and the system of equations is solved by Gaussian elimination. The components of this 5 × 6 matrix *CCON* are:

$$CCON_{11} = L_{11,N+1}A_{11,N-1} + A_{12,N-1}$$

$$CCON_{12} = L_{11,N+1}A_{13,N-1}$$

$$CCON_{13} = 0$$

$$CCON_{14} = L_{12,N+1}$$

$$CCON_{15} = L_{13,N+1}$$

$$CCON_{21} = L_{21,N+1}A_{11,N-1} + A_{12,N-1}$$

$$CCON_{22} = L_{21,N+1} A_{13,N-1}$$

$$CCON_{23} = -1$$

$$CCON_{24} = L_{22,N+1}$$

$$CCON_{25} = L_{23,N+1}$$

$$CCON_{31} = L_{31,N+1} A_{11,N-1} + A_{12,N-1}$$

$$CCON_{32} = L_{31,N+1} A_{13,N-1}$$

$$CCON_{33} = 0$$

$$CCON_{34} = L_{32,N+1}$$

$$CCON_{35} = L_{33,N+1} + 1$$

$$CCON_{41} = \left(A_{311} + A_{321}\right)\frac{a_1}{2} + \sum_{j=2}^{N-1}\left(A_{31,j-1} + A_{31j} + A_{32,j-1} + A_{32j}\right)\frac{a_j}{2}$$

$$CCON_{42} = \left(1 + A_{331}\right)\frac{a_1}{2} + \sum_{j=2}^{N-1}\left(A_{33,j-1} + A_{33j}\right)\frac{a_j}{2}$$

$$CCON_{43} = 0$$

$$CCON_{44} = 0$$

$$CCON_{45} = 0$$

$$CCON_{51} = \left[A_{21,N-1} + A_{22,N-1} - \nu_N\left(A_{11,N-1} + A_{12,N-1} + A_{31,N-1} + A_{32,N-1}\right)\right]\frac{r_N}{E_N} + \nu_{N+1}\left(A_{11,N-1} + A_{12,N-1}\right)\frac{r_{N+1}}{E_{N+1}}$$

$$CCON_{52} = \left[A_{23,N-1} - \nu_N\left(A_{13,N-1} + A_{33,N-1}\right)\right]\frac{r_N}{E_N} + \nu_{N+1} A_{13,N-1}\frac{r_{N+1}}{E_{N+1}}$$

$$CCON_{53} = 0$$

$$CCON_{54} = -\frac{r_{N+1}}{E_{N+1}}$$

$$CCON_{55} = \nu_{N+1}\frac{r_{N+1}}{E_{N+1}}$$

$$CCON_{16} = -P_E - L_{11,N+1} B_{1,N-1} - M_{1,N+1}$$

$$CCON_{26} = -L_{21,N+1} B_{1,N-1} - M_{2,N+1}$$

$$CCON_{36} = -L_{31,N+1}B_{1,N-1} - M_{3,N+1}$$

$$CCON_{46} = F_z - B_{31}\frac{a_1}{2} - \sum_{j=2}^{N-1}\left(B_{3,j-1} + B_{3j}\right)\frac{a_j}{2}$$

$$CCON_{56} = -\nu_{N+1}B_{1,N-1}\frac{r_{N+1}}{E_{N+1}} + \left[\left(\alpha_\theta \Delta T\right)_{N+1} + \left(\varepsilon_\theta^P\right)_{N+1} + \left(d\varepsilon_\theta^P\right)_{N+1}\right]r_{N+1} + gap_{ini}$$
$$- \left[B_{2,N-1} - \nu_N\left(B_{1,N-1} + B_{3,N-1}\right)\right]\frac{r_N}{E_N} - \left[\left(\alpha_\theta \Delta T\right)_N + \left(\varepsilon_\theta^P\right)_N + \left(d\varepsilon_\theta^P\right)_N\right]r_N$$

If $r_1 \neq 0$, then $\sigma_{r1} = -P_I$, and $\sigma_{\theta 1}$ is treated as an unknown instead of σ_{r1}. Several components of the *CCON* matrix change as follows:

$$CCON_{11} = L_{11,N+1}A_{12,N-1}$$

$$CCON_{21} = L_{21,N+1}A_{12,N-1}$$

$$CCON_{31} = L_{31,N+1}A_{12,N-1}$$

$$CCON_{41} = A_{321}\frac{a_1}{2} + \sum_{j+2}^{N-1}\left(A_{32,j-1} + A_{32j}\right)\frac{a_j}{2}$$

$$CCON_{51} = \left[A_{22,N-1} - \nu_N\left(A_{12,N-1} + A_{32,N-1}\right)\right]\frac{r_N}{E_N} + \nu_{N+1}A_{12,N-1}\frac{r_{N+1}}{E_{N+1}}$$

$$CCON_{16} = -P_E - L_{11,N+1}B_{1,N-1} - M_{1,N+1} + L_{11,N+1}A_{11,N-1}P_I$$

$$CCON_{26} = -L_{21,N+1}B_{1,N-1} - M_{2,N+1} + L_{21,N+1}A_{11,N-1}P_I$$

$$CCON_{36} = -L_{31,N+1}B_{1,N-1} - M_{3,N+1} + L_{11,N+1}A_{11,N-1}P_I$$

$$CCON_{46} = F_z - B_{31}\frac{a_1}{2} + A_{311}P_I\frac{a_1}{2} + \sum_{j=2}^{N-1}\left[-B_{3,j-1} - B_{3j} + \left(A_{31,j-1} + A_{31j}\right)P_I\right]\frac{a_j}{2}$$

$$CCON_{56} = -\nu_{N+1}B_{1,N-1}\frac{r_{N+1}}{E_{N+1}} + \left[\left(\alpha_\theta \Delta T\right)_{N+1} + \left(\varepsilon_\theta^P\right)_{N+1} + \left(d\varepsilon_\theta^P\right)_{N+1}\right]r_{N+1} + gap_{ini}$$
$$- \left[B_{2,N-1} - \nu_N\left(B_{1,N-1} + B_{3,N-1}\right)\right]\frac{r_N}{E_N} - \left[\left(\alpha_\theta \Delta T\right)_N + \left(\varepsilon_\theta^P\right)_N + \left(d\varepsilon_\theta^P\right)_N\right]r_N$$
$$+ \left\{\left[A_{21,N-1} - \nu_N\left(A_{11,N-1} + A_{31,N-1}\right)\right]\frac{r_N}{E_N} + \nu_{N+1}A_{11,N-1}\frac{r_{N+1}}{E_{N+1}}\right\}P_I$$

A.2.3 Fuel/Clad Contact with No Axial Slippage

When there is fuel/clad contact, and axial slippage between the fuel and clad is prevented, the compatibility conditions at the fuel/clad interface are used to create *E*, *F*, and *G* matrices that transfer stresses from the outer surface of the fuel to the inner surface of the cladding. The components of these matrices are

$$E_{11N} = 1$$

$$E_{12N} = 0$$

$$E_{13N} = 0$$

$$E_{21N} = -\frac{\nu_{N+1}}{E_{N+1}}$$

$$E_{22N} = -\frac{\nu_{N+1}}{E_{N+1}}$$

$$E_{23N} = \frac{1}{E_{N+1}}$$

$$E_{31N} = -\nu_{N+1}\frac{r_{N+1}}{E_{N+1}}$$

$$E_{32N} = \frac{r_{N+1}}{E_{N+1}}$$

$$E_{33N} = -\nu_{N+1}\frac{r_{N+1}}{E_{N+1}}$$

$$F_{11N} = 1$$

$$F_{12N} = 0$$

$$F_{13N} = 0$$

$$F_{21N} = -\frac{\nu_N}{E_N}$$

$$F_{22N} = -\frac{\nu_N}{E_N}$$

$$F_{23N} = \frac{1}{E_N}$$

$$F_{31N} = -\nu_N\frac{r_N}{E_N}$$

$$F_{32N} = \frac{r_N}{E_N}$$

$$F_{33N} = -\nu_N \frac{r_N}{E_N}$$

$$G_{1N} = 0$$

$$G_{2N} = \left(\alpha_z \Delta T\right)_N - \left(\alpha_z \Delta T\right)_{N+1} + \left(\varepsilon_z^P\right)_N - \left(\varepsilon_z^P\right)_{N+1} + \left(d\varepsilon_z^P\right)_N - \left(d\varepsilon_z^P\right)_{N+1} - \left(\varepsilon_{z,N}\right)_0 + \left(\varepsilon_{z,N+1}\right)_0$$

$$G_{3N} = \left[\left(\alpha_\theta \Delta T\right)_N + \left(\varepsilon_\theta^P\right)_N + \left(d\varepsilon_\theta^P\right)_N\right] r_N - \left[\left(\alpha_\theta \Delta T\right)_{N+1} + \left(\varepsilon_\theta^P\right)_{N+1} + \left(d\varepsilon_\theta^P\right)_{N+1}\right] r_{N+1} - gap_{ini}$$

where the subscript "0" in the expression for G_2 indicates that the value for the strain component is that occurring at the beginning of the load increment.

The boundary conditions applied to the fuel/clad composite result in the following matrix equation

$$[C]\{\sigma\}_1 = \{D\}$$

where the components of the C and D matrices are:

$$C_{11} = A_{11,N+1}$$

$$C_{12} = A_{12,N+1}$$

$$C_{13} = A_{13,N+1}$$

$$C_{21} = 1$$

$$C_{22} = -1 \qquad \text{if } r_1 = 0$$

$$C_{22} = 0 \qquad \text{if } r_1 \neq 0$$

$$C_{23} = 0$$

$$C_{31} = A_{311}\frac{a_1}{2} + \sum_{j=2}^{N+1}\left(A_{31,j-1} + A_{31,j}\right)\frac{a_j}{2}$$

$$C_{32} = A_{321}\frac{a_1}{2} + \sum_{j=2}^{N+1}\left(A_{32,j-1} + A_{32,j}\right)\frac{a_j}{2}$$

$$C_{33} = A_{331}\frac{a_1}{2} + \sum_{j=2}^{N+1}\left(A_{33,j-1} + A_{33,j}\right)\frac{a_j}{2}$$

$$D_1 = -P_E - B_{1,N+1}$$

$$D_2 = 0 \qquad \text{if } r_1 = 0$$

$$D_2 = -P_I \qquad \text{if } r_1 \neq 0$$

$$D_3 = F_z - B_{31}\frac{a_1}{2} - \sum_{j=2}^{N+1}\left(B_{3,j-1} + B_{3,j}\right)\frac{a_j}{2}$$

Appendix B

Source Code

Appendix B
Source Code

The mechanical model is currently a stand-alone code, which is intended to be incorporated as a subroutine into the TRU fuel performance code. This Appendix presents the source coding for the model. In addition to the primary coding, it contains two subroutines called INVERT and GAUSS. The former inverts a 3 × 3 matrix, which is needed in several instances in the model. The latter performs Gaussian elimination to solve a set of linear simultaneous equations such as the set of five simultaneous equations described in Section 9.2 of this report.

```
      PROGRAM MECHMODEL
c   New mechanical model to calculate stress, strains, and
c    displacements throughout the fuel and cladding.
c   Currently includes elastic, thermal expansion, plastic response of
c    the fuel and cladding.  Creep, fuel swelling to be added.
      IMPLICIT NONE
      DOUBLE PRECISION  ER,E,AL1,AL2,AL3,ALT1,ALT2,
     &  ALT3,EP,EPT,EPET,EP1P,EP2P,EP3P,
     &  DEP1P,DEP2P,DEP3P,DEP1PR,DEP2PR,DEP3PR,
     &  DEPEQP,EQSTR,SIGEQ,MOD,T,R,R0,UR,UZ,DELZ,DELZ0,
     &  EM,FM,GM,EMINV,A,
     &  B,SIG,DR,DA,
     &  C,D,CINV,LM,MM,NUR,NU,PI,PO,FZ,PIE,BB,CC,
     &  DC,PE,CCINV,CCON,X,EPT0,EG,EGINV,Y,YINV,C1,GAPINI
      INTEGER*4 I,J,JJ,K,N,IPLAST,ICON
c     DIMENSION ER(N+1),E(N+2),AL1(N+1),AL2(N+1),AL3(N+1),ALT1(N+2),
c    &  ALT2(N+2),ALT3(N+2),EP(3,N+2),EPT(3,N+2),EPET(N+2),EP1P(N+2),
c    &  EP2P(N+2),EP3P(N+2),DEP1P(N+2),DEP2P(N+2),DEP3P(N+2),
c    &  DEP1PR(N+2),DEP2PR(N+2),DEP3PR(N+2),DEPEQP(N+2),EQSTR(N+2),
c    &  SIGEQ(N+2),MOD(N+2),T(N+2),R(N+2),R0(N+2),UR(N+2),UZ(N+2),
c    &  DELZ(N+2),DELZ0(N+2),EM(3,3,N+1),FM(3,3,N+1),GM(3,N+1),
c    &  EMINV(3,3,N+1),A(3,3,N+1),B(3,N+1),SIG(3,N+2),DR(N+1),DA(N+1),
c    &  C(3,3),D(3),CINV(3,3),LM(3,3,N+1),MM(3,N+1),NUR(N+1),
c    &  NU(N+2),IPLAST(N+2),
c    &  CC(3,3),DC(3),CCINV(3,3),CCON(5,6),X(5),EPT0(3,N+2),EG(3,3),
c    &  EGINV(3,3),Y(3,3),YINV(3,3)
c   Parameters are dimensioned to specific values for purposes of this
c    stand-alone program
      DIMENSION ER(12),E(13),AL1(12),AL2(12),AL3(12),ALT1(13),ALT2(13),
     &  ALT3(13),EP(3,13),EPT(3,13),EPET(13),EP1P(13),EP2P(13),EP3P(13),
     &  DEP1P(13),DEP2P(13),DEP3P(13),DEP1PR(13),DEP2PR(13),DEP3PR(13),
     &  DEPEQP(13),EQSTR(13),SIGEQ(13),MOD(13),T(13),R(13),R0(13),
     &  UR(13),UZ(13),DELZ(13),DELZ0(13),EM(3,3,12),FM(3,3,12),GM(3,12),
     &  EMINV(3,3,12),A(3,3,12),B(3,12),SIG(3,13),DR(12),DA(12),
     &  C(3,3),D(3),CINV(3,3),LM(3,3,12),MM(3,12),NUR(12),
     &  NU(13),IPLAST(13),
     &  CC(3,3),DC(3),CCINV(3,3),CCON(5,6),X(5),EPT0(3,13),EG(3,3),
     &  EGINV(3,3),Y(3,3),YINV(3,3)

c   Set values for basic input parameters
      PIE = 3.141592654D0
c      Number of nodes across fuel portion of model
      N = 11
c      Elastic moduli
      ER = 1.2D6
```

```
c      Poisson's ratio
       NUR = 0.3D0
c      Initial yield strength
       EQSTR = 1.D4
c      Initial plastic modulus
       MOD = 1.D4
c      Coefficients of thermal expansion
       AL1 = 5.0D-5
       AL2 = 5.0D-5
       AL3 = 5.0D-5
c      Internal pressure (PI), gas pressure (PO), axial force (FZ)
       PI = 0.D0
       PO = 0.D0
       FZ = 0.D0
c      Initial radii
       R(1) = 0.D0
       DO I = 2,N
        R(I) = R(I-1) + 1.D-2
       END DO
       R(N+1) = 0.101D0
       R(N+2) = 0.111D0
       R0 = R
c      Initialize plastic strain components to zero
       EP1P = 0.D0
       EP2P = 0.D0
       EP3P = 0.D0
c      Initialize axial displacement (UZ) and axial thickness (DELZ)
c       to zero
       UZ = 0.D0
       DELZ = 0.01D0
       DELZ0 = 0.01D0
c      Calculate elastic moduli and Poisson's ratios at nodes
       E(1) = ER(1)
       NU(1) = NUR(1)
       E(N) = ER(N-1)
       NU(N) = NUR(N-1)
       DO I = 2,N-1
        E(I) = (ER(I-1)+ER(I))/2.D0
        NU(I) = (NUR(I-1)+NUR(I))/2.D0
       END DO
c  Initialize fuel/clad contact condition to zero (no contact)
       ICON = 0
c      Set properties for cladding
       ER(N+1) = 29.D6
       NUR(N+1) = 0.3D0
       EQSTR(N+1) = 3.D4
       EQSTR(N+2) = 3.D4
       MOD(N+1) = 3.D4
       MOD(N+2) = 3.D4
       AL1(N+1) = 4.D-6
       AL2(N+1) = 4.D-6
       AL3(N+1) = 4.D-6
       PE = 0.D0
c  Initial width of gap between fuel and clad (GAPINI)
       GAPINI = R(N+1)-R(N)
       E(N+1) = ER(N+1)
       E(N+2) = ER(N+1)
```

```
      NU(N+1) = NUR(N+1)
      NU(N+2) = NUR(N+1)
c  Initialize total strains at the beginning of a load increment to
c   zero  (will be calculated later)
      EPT0 = 0.D0
c  Initialize accumulated thermal expansion strains to zero
      ALT1 = 0.D0
      ALT2 = 0.D0
      ALT3 = 0.D0
c   Loading is applied in increments
      DO 300 JJ = 1,2001
      IPLAST = 0
c   Initialize plastic strain increments to zero at start of load
c    increment
      DEP1P = 0.D0
      DEP2P = 0.D0
      DEP3P = 0.D0
c   Initialize plastic strain increments for previous iteration to zero
      DEP1PR = 0.D0
      DEP2PR = 0.D0
      DEP3PR = 0.D0
c   Calculate accumulated strains due to thermal expansion at each node
c   Incremental strain is that due to temperature change (T) occurring
c    during the increment
      ALT1(1) = ALT1(1)+AL1(1)*T(1)
      ALT2(1) = ALT2(1)+AL2(1)*T(1)
      ALT3(1) = ALT3(1)+AL3(1)*T(1)
      ALT1(N) = ALT1(N)+AL1(N-1)*T(N)
      ALT2(N) = ALT2(N)+AL2(N-1)*T(N)
      ALT3(N) = ALT3(N)+AL3(N-1)*T(N)
      DO I = 2,N-1
       ALT1(I) = ALT1(I)+(AL1(I-1)+AL1(I))*T(I)/2.D0
       ALT2(I) = ALT2(I)+(AL2(I-1)+AL2(I))*T(I)/2.D0
       ALT3(I) = ALT3(I)+(AL3(I-1)+AL3(I))*T(I)/2.D0
      END DO
      ALT1(N+1) = ALT1(N+1)+AL1(N+1)*T(N+1)
      ALT2(N+1) = ALT2(N+1)+AL2(N+1)*T(N+1)
      ALT3(N+1) = ALT3(N+1)+AL3(N+1)*T(N+1)
      ALT1(N+2) = ALT1(N+2)+AL1(N+1)*T(N+2)
      ALT2(N+2) = ALT2(N+2)+AL2(N+1)*T(N+2)
      ALT3(N+2) = ALT3(N+2)+AL3(N+1)*T(N+2)
c   Create EM and FM matrices for each region
      DO 100 I = 1,N+1
       IF(I.EQ.N) GO TO 100
c   Calculate thickness (DR) and area (DA) for region "i"
       DR(I) = R(I+1) - R(I)
       DA(I) = PIE*(R(I+1)**2-R(I)**2)
       EM(1,1,I) = 2.D0*R(I+1)
       EM(1,2,I) = - (R(I+1) - R(I))
       EM(1,3,I) = 0.D0
       EM(2,1,I) = - NU(I+1)/(E(I+1)*DR(I)) - (1.D0+NU(I+1))/
     A     (E(I+1)*(R(I+1)+R(I)))
       EM(2,2,I) = 1.D0/(E(I+1)*DR(I)) + (1.D0+NU(I+1))/
     A     (E(I+1)*(R(I+1)+R(I)))
       EM(2,3,I) = -NU(I+1)/(E(I+1)*DR(I))
       EM(3,1,I) = -NU(I+1)/E(I+1)
       EM(3,2,I) = EM(3,1,I)
```

```
        EM(3,3,I) = 1.D0/E(I+1)
        FM(1,1,I) = 2.D0*R(I)
        FM(1,2,I) = R(I+1) - R(I)
        FM(1,3,I) = 0.D0
        FM(2,1,I) = -NU(I)/(E(I)*DR(I))
     A      + (1.D0+NU(I))/(E(I)*(R(I+1)+R(I)))
        FM(2,2,I) = 1.D0/(E(I)*DR(I))
     A      - (1.D0+NU(I))/(E(I)*(R(I+1)+R(I)))
        FM(2,3,I) = -NU(I)/(E(I)*DR(I))
        FM(3,1,I) = -NU(I)/E(I)
        FM(3,2,I) = FM(3,1,I)
        FM(3,3,I) = 1.D0/E(I)
c   Invert the EM matrix for region "i"
        DO 11 J = 1,3
         DO 12 K = 1,3
   12    Y(J,K) = EM(J,K,I)
   11   CONTINUE
        CALL INVERT(Y,YINV)
        DO 13 J = 1,3
         DO 14 K = 1,3
   14    EMINV(J,K,I) = YINV(J,K)
   13   CONTINUE
c   Calculate LM matrix for region "i"
        DO 30 J = 1,3
         DO 40 K = 1,3
   40     LM(J,K,I) = EMINV(J,1,I)*FM(1,K,I) + EMINV(J,2,I)*FM(2,K,I)
     A      + EMINV(J,3,I)*FM(3,K,I)
   30   CONTINUE
  100 CONTINUE
c   Determine GM and MM matrices for each region
  110 DO 120 I = 1,N+1
        IF (I.EQ.N) GO TO 120
        GM(1,I) = 0.D0
        GM(2,I) = -(ALT2(I+1)-ALT2(I))/DR(I) - (EP2P(I+1)-EP2P(I))/DR(I)
     A    - (DEP2P(I+1)-DEP2P(I))/DR(I) - (ALT2(I+1)-ALT1(I+1))/
     A    (R(I+1)+R(I)) - (ALT2(I)-ALT1(I))/(R(I+1)+R(I))
     A    - (EP2P(I+1)-EP1P(I+1))/(R(I+1)+R(I)) - (EP2P(I)-EP1P(I))/
     A    (R(I+1)+R(I)) - (DEP2P(I+1)-DEP1P(I+1))/(R(I+1)+R(I))
     A    - (DEP2P(I)-DEP1P(I))/(R(I+1)+R(I))
        GM(3,I) = - (ALT3(I+1)-ALT3(I)) - (EP3P(I+1)-EP3P(I))
     A    - (DEP3P(I+1) - DEP3P(I))
        DO J = 1,3
         MM(J,I) = EMINV(J,1,I)*GM(1,I) + EMINV(J,2,I)*GM(2,I)
     A     + EMINV(J,3,I)*GM(3,I)
        END DO
        IF (I.EQ.N+1) GO TO 120
c   Determine A and B matrices for region "i"
        DO 60 J = 1,3
         IF (I.EQ.1) THEN
          B(J,I) = MM(J,I)
         ELSE
          B(J,I) = LM(J,1,I)*B(1,I-1) + LM(J,2,I)*B(2,I-1)
     A      + LM(J,3,I)*B(3,I-1) + MM(J,I)
         END IF
        DO 70 K = 1,3
         IF (I.EQ.1) THEN
          A(J,K,I) = LM(J,K,I)
```

```
        ELSE
         A(J,K,I) = LM(J,1,I)*A(1,K,I-1) + LM(J,2,I)*A(2,K,I-1)
     A     + LM(J,3,I)*A(3,K,I-1)
        END IF
   70   CONTINUE
   60  CONTINUE
  120 CONTINUE
c   Calculate C and D matrices for fuel, and CC and DC matrices for
c    cladding when there is no fuel/clad contact
      IF (ICON.EQ.0) THEN
       C(1,1) = A(1,1,N-1)
       C(1,2) = A(1,2,N-1)
       C(1,3) = A(1,3,N-1)
       C(2,1) = 1.D0
       D(1) = - PO - B(1,N-1)
       IF (R(1).EQ.0.D0) THEN
        C(2,2) = -1.D0
        D(2) = 0.D0
       ELSE
        C(2,2) = 0.D0
        D(2) = - PI
       END IF
       C(2,3) = 0.D0
       C(3,1) = (0.D0+A(3,1,1))*DA(1)/2.D0
       C(3,2) = (0.D0+A(3,2,1))*DA(1)/2.D0
       C(3,3) = (1.D0+A(3,3,1))*DA(1)/2.D0
       D(3) = FZ - B(3,1)*DA(1)/2.D0
       DO I = 2,N-1
        C(3,1) = C(3,1) + (A(3,1,I-1)+A(3,1,I))*DA(I)/2.D0
        C(3,2) = C(3,2) + (A(3,2,I-1)+A(3,2,I))*DA(I)/2.D0
        C(3,3) = C(3,3) + (A(3,3,I-1)+A(3,3,I))*DA(I)/2.D0
        D(3) = D(3) - (B(3,I-1)+B(3,I))*DA(I)/2.D0
       END DO
c   Invert C matrix
       CALL INVERT(C,CINV)
       CC(1,1) = LM(1,1,N+1)
       CC(1,2) = LM(1,2,N+1)
       CC(1,3) = LM(1,3,N+1)
       CC(2,1) = 1.D0
       CC(2,2) = 0.D0
       CC(2,3) = 0.D0
       CC(3,1) = LM(3,1,N+1)
       CC(3,2) = LM(3,2,N+1)
       CC(3,3) = LM(3,3,N+1)+1.D0
       DC(1) = -PE-MM(1,N+1)
       DC(2) = -PO
       DC(3) = -MM(3,N+1)
c   Invert CC matrix
       CALL INVERT(CC,CCINV)
      END IF
c   Calculate CCON matrix when fuel and cladding are in contact but are
c    allowed to slip
      IF (ICON.EQ.1) THEN
       CCON(1,1) = LM(1,1,N+1)*(A(1,1,N-1)+A(1,2,N-1))
       CCON(1,2) = LM(1,1,N+1)*A(1,3,N-1)
       CCON(1,3) = 0.D0
       CCON(1,4) = LM(1,2,N+1)
```

```
      CCON(1,5) = LM(1,3,N+1)
      CCON(2,1) = LM(2,1,N+1)*(A(1,1,N-1)+A(1,2,N-1))
      CCON(2,2) = LM(2,1,N+1)*A(1,3,N-1)
      CCON(2,3) = -1.D0
      CCON(2,4) = LM(2,2,N+1)
      CCON(2,5) = LM(2,3,N+1)
      CCON(3,1) = LM(3,1,N+1)*(A(1,1,N-1)+A(1,2,N-1))
      CCON(3,2) = LM(3,1,N+1)*A(1,3,N-1)
      CCON(3,3) = 0.D0
      CCON(3,4) = LM(3,2,N+1)
      CCON(3,5) = LM(3,3,N+1)+1.D0
      CCON(4,1) = (A(3,1,1)+A(3,2,1))*DA(1)/2.D0
      CCON(4,2) = (1.D0+A(3,3,1))*DA(1)/2.D0
      CCON(4,3) = 0.D0
      CCON(4,4) = 0.D0
      CCON(4,5) = 0.D0
      CCON(4,6) = FZ-B(3,1)*DA(1)/2.D0
      DO I = 2,N-1
       CCON(4,1) = CCON(4,1)+(A(3,1,I-1)+A(3,1,I)
     &      +A(3,2,I-1)+A(3,2,I))*DA(I)/2.D0
       CCON(4,2) = CCON(4,2)+(A(3,3,I-1)+A(3,3,I))*DA(I)/2.D0
       CCON(4,6) = CCON(4,6)-(B(3,I-1)+B(3,I))*DA(I)/2.D0
      END DO
      CCON(5,1) = (A(2,1,N-1)+A(2,2,N-1)-NU(N)*(A(1,1,N-1)+A(1,2,N-1)
     &      +A(3,1,N-1)+A(3,2,N-1)))*R(N)/E(N)+NU(N+1)*(A(1,1,N-1)
     &      +A(1,2,N-1))*R(N+1)/E(N+1)
      CCON(5,2) = (A(2,3,N-1)-NU(N)*(A(1,3,N-1)+A(3,3,N-1)))*R(N)/E(N)
     &      +NU(N+1)*A(1,3,N-1)*R(N+1)/E(N+1)
      CCON(5,3) = 0.D0
      CCON(5,4) = -R(N+1)/E(N+1)
      CCON(5,5) = NU(N+1)*R(N+1)/E(N+1)
      CCON(1,6) = -PE-LM(1,1,N+1)*B(1,N-1)-MM(1,N+1)
      CCON(2,6) = -LM(2,1,N+1)*B(1,N-1)-MM(2,N+1)
      CCON(3,6) = -LM(3,1,N+1)*B(1,N-1)-MM(3,N+1)
      CCON(5,6) = -NU(N+1)*B(1,N-1)*R(N+1)/E(N+1)+(ALT2(N+1)+EP2P(N+1)
     &      +DEP2P(N+1))*R(N+1)+GAPINI-(B(2,N-1)-NU(N)*(B(1,N-1)
     &      +B(3,N-1)))*R(N)/E(N)-(ALT2(N)+EP2P(N)+DEP2P(N))*R(N)
      IF(R(1).GT.0.D0) THEN
       CCON(1,1) = LM(1,1,N+1)*A(1,2,N-1)
       CCON(2,1) = LM(2,1,N+1)*A(1,2,N-1)
       CCON(3,1) = LM(3,1,N+1)*A(1,2,N-1)
       CCON(4,1) = A(3,2,1)*DA(1)/2.D0
       CCON(5,1) = (A(2,2,N-1)-NU(N)*(A(1,2,N-1)+A(3,2,N-1)))*
     &      R(N)/E(N)+NU(N+1)*A(1,2,N-1)*R(N+1)/E(N+1)
       CCON(4,6) = FZ-B(3,1)*DA(1)/2.D0+A(3,1,1)*DA(1)*PI/2.D0
       DO I = 2,N-1
        CCON(4,1) = CCON(4,1)+(A(3,2,I-1)+A(3,2,I))*DA(I)/2.D0
        CCON(4,6) = CCON(4,6)-(B(3,I-1)+B(3,I))*DA(I)/2.D0
     &      +(A(3,1,I-1)+A(3,1,I))*DA(I)*PI/2.D0
       END DO
       CCON(1,6) = CCON(1,6)+LM(1,1,N+1)*A(1,1,N-1)*PI
       CCON(2,6) = CCON(2,6)+LM(2,1,N+1)*A(1,1,N-1)*PI
       CCON(3,6) = CCON(3,6)+LM(3,1,N+1)*A(1,1,N-1)*PI
       CCON(5,6) = CCON(5,6)+((A(2,1,N-1)-NU(N)*(A(1,1,N-1)
     &     +A(3,1,N-1)))*R(N)/E(N)+NU(N+1)*A(1,1,N-1)*R(N+1)/E(N+1))*PI
      END IF
c   Perform Gaussian elimination to calculate five stress components
```

```
       CALL GAUSS(5,CCON,X)
      END IF
c   Calculate EM, FM, and GM  matrices for gap region when fuel and
c     clad are in contact, and no slippage is allowed
      IF (ICON.EQ.2) THEN
       EM(1,1,N) = 1.D0
       EM(1,2,N) = 0.D0
       EM(1,3,N) = 0.D0
       EM(2,1,N) = -NU(N+1)/E(N+1)
       EM(2,2,N) = -NU(N+1)/E(N+1)
       EM(2,3,N) = 1.D0/E(N+1)
       EM(3,1,N) = -NU(N+1)*R(N+1)/E(N+1)
       EM(3,2,N) = R(N+1)/E(N+1)
       EM(3,3,N) = -NU(N+1)*R(N+1)/E(N+1)
       FM(1,1,N) = 1.D0
       FM(1,2,N) = 0.D0
       FM(1,3,N) = 0.D0
       FM(2,1,N) = -NU(N)/E(N)
       FM(2,2,N) = -NU(N)/E(N)
       FM(2,3,N) = 1.D0/E(N)
       FM(3,1,N) = -NU(N)*R(N)/E(N)
       FM(3,2,N) = R(N)/E(N)
       FM(3,3,N) = -NU(N)*R(N)/E(N)
       GM(1,N) = 0.D0
       GM(2,N) = ALT3(N)-ALT3(N+1)+EP3P(N)-EP3P(N+1)
     &      +DEP3P(N)-DEP3P(N+1)-EPT0(3,N)+EPT0(3,N+1)
       GM(3,N) = (ALT2(N)+EP2P(N)+DEP2P(N))*R(N)
     &      -(ALT2(N+1)+EP2P(N+1)+DEP2P(N+1))*R(N+1)-GAPINI
c  Invert EM matrix for gap region
       DO 33 J = 1,3
        DO 43 K = 1,3
   43    EG(J,K) = EM(J,K,N)
   33  CONTINUE
       CALL INVERT(EG,EGINV)
c  Determine LM and MM matrices for gap region
       DO 35 J = 1,3
        MM(J,N) = EGINV(J,1)*GM(1,N)+EGINV(J,2)*GM(2,N)
     &      +EGINV(J,3)*GM(3,N)
        DO 45 K = 1,3
   45    LM(J,K,N) = EGINV(J,1)*FM(1,K,N)+EGINV(J,2)*FM(2,K,N)
     &      +EGINV(J,3)*FM(3,K,N)
   35  CONTINUE
c  Determine A and B matrices for gap and clad regions
       DO 150 I = N,N+1
        DO 80 J = 1,3
         B(J,I) = LM(J,1,I)*B(1,I-1)+LM(J,2,I)*B(2,I-1)
     &      +LM(J,3,I)*B(3,I-1)+MM(J,I)
        DO 75 K = 1,3
   75     A(J,K,I) = LM(J,1,I)*A(1,K,I-1)+LM(J,2,I)*A(2,K,I-1)
     &       +LM(J,3,I)*A(3,K,I-1)
   80   CONTINUE
  150  CONTINUE
c   Determine C and D matrices for condition of a closed gap, and
c    no slippage
       C(1,1) = A(1,1,N+1)
       C(1,2) = A(1,2,N+1)
       C(1,3) = A(1,3,N+1)
```

```
        C(2,1) = 1.D0
        D(1) = -PE-B(1,N+1)
        IF(R(1).EQ.0.D0) THEN
         C(2,2) = -1.D0
         D(2) = 0.D0
        ELSE
         C(2,2) = 0.D0
         D(2) = -PI
        END IF
        C(2,3) = 0.D0
        C(3,1) = (0.D0+A(3,1,1))*DA(1)/2.D0
        C(3,2) = (0.D0+A(3,2,1))*DA(1)/2.D0
        C(3,3) = (1.D0+A(3,3,1))*DA(1)/2.D0
        D(3) = FZ - B(3,1)*DA(1)/2.D0
        DO I = 2,N+1
         C(3,1) = C(3,1) + (A(3,1,I-1)+A(3,1,I))*DA(I)/2.D0
         C(3,2) = C(3,2) + (A(3,2,I-1)+A(3,2,I))*DA(I)/2.D0
         C(3,3) = C(3,3) + (A(3,3,I-1)+A(3,3,I))*DA(I)/2.D0
         D(3) = D(3) - (B(3,I-1)+B(3,I))*DA(I)/2.D0
        END DO
        CALL INVERT(C,CINV)
       END IF
c   Calculate stresses and strains at all nodes for condition of no
c    fuel/clad contact
       IF (ICON.EQ.0) THEN
c   For node 1, calculate:  radial stress     SIG(1,1),
c                           tangential stress SIG(2,1),
c                           axial stress      SIG(3,1)
        SIG(1,1) = CINV(1,1)*D(1) + CINV(1,2)*D(2) + CINV(1,3)*D(3)
        SIG(2,1) = CINV(2,1)*D(1) + CINV(2,2)*D(2) + CINV(2,3)*D(3)
        SIG(3,1) = CINV(3,1)*D(1) + CINV(3,2)*D(2) + CINV(3,3)*D(3)
c   Calculate equivalent stress SIGEQ(1) at node 1
        SIGEQ(1) = (1.D0/2.D0**.5D0)*((SIG(1,1)-SIG(2,1))**2
     A    + (SIG(1,1)-SIG(3,1))**2+(SIG(2,1)-SIG(3,1))**2)**.5D0
c   Calculate components of modified total strain EP at node 1
        EP(1,1) = (SIG(1,1)-NU(1)*(SIG(2,1)+SIG(3,1)))/E(1)
     A    + ALT1(1) + DEP1P(1)
        EP(2,1) = (SIG(2,1)-NU(1)*(SIG(1,1)+SIG(3,1)))/E(1)
     A    + ALT2(1) + DEP2P(1)
        EP(3,1) = (SIG(3,1)-NU(1)*(SIG(1,1)+SIG(2,1)))/E(1)
     A    + ALT3(1) + DEP3P(1)
c   Calculate stress components, equivalent stress, and modified
c     total strains at all remaining nodes
        DO I = 2,N
         SIG(1,I) = A(1,1,I-1)*SIG(1,1) + A(1,2,I-1)*SIG(2,1)
     A    + A(1,3,I-1)*SIG(3,1) + B(1,I-1)
         SIG(2,I) = A(2,1,I-1)*SIG(1,1) + A(2,2,I-1)*SIG(2,1)
     A    + A(2,3,I-1)*SIG(3,1) + B(2,I-1)
         SIG(3,I) = A(3,1,I-1)*SIG(1,1) + A(3,2,I-1)*SIG(2,1)
     A    + A(3,3,I-1)*SIG(3,1) + B(3,I-1)
         SIGEQ(I) = (1.D0/2.D0**.5D0)*((SIG(1,I)-SIG(2,I))**2
     A    + (SIG(1,I)-SIG(3,I))**2+(SIG(2,I)-SIG(3,I))**2)**.5D0
         EP(1,I) = (SIG(1,I)-NU(I)*(SIG(2,I)+SIG(3,I)))/E(I)
     A    + ALT1(I) + DEP1P(I)
         EP(2,I) = (SIG(2,I)-NU(I)*(SIG(1,I)+SIG(3,I)))/E(I)
     A    + ALT2(I) + DEP2P(I)
         EP(3,I) = (SIG(3,I)-NU(I)*(SIG(1,I)+SIG(2,I)))/E(I)
```

```
     A    + ALT3(I) + DEP3P(I)
          END DO
          SIG(1,N+1) = CCINV(1,1)*DC(1)+CCINV(1,2)*DC(2)
     A   +CCINV(1,3)*DC(3)
          SIG(2,N+1) = CCINV(2,1)*DC(1)+CCINV(2,2)*DC(2)
     A   +CCINV(2,3)*DC(3)
          SIG(3,N+1) = CCINV(3,1)*DC(1)+CCINV(3,2)*DC(2)
     A   +CCINV(3,3)*DC(3)
          SIG(1,N+2) = LM(1,1,N+1)*SIG(1,N+1)+LM(1,2,N+1)*SIG(2,N+1)
     A   +LM(1,3,N+1)*SIG(3,N+1)+MM(1,N+1)
          SIG(2,N+2) = LM(2,1,N+1)*SIG(1,N+1)+LM(2,2,N+1)*SIG(2,N+1)
     A   +LM(2,3,N+1)*SIG(3,N+1)+MM(2,N+1)
          SIG(3,N+2) = LM(3,1,N+1)*SIG(1,N+1)+LM(3,2,N+1)*SIG(2,N+1)
     A   +LM(3,3,N+1)*SIG(3,N+1)+MM(3,N+1)
          DO I = N+1,N+2
           SIGEQ(I) = (1.D0/2.D0**.5D0)*((SIG(1,I)-SIG(2,I))**2
     A     + (SIG(1,I)-SIG(3,I))**2+(SIG(2,I)-SIG(3,I))**2)**.5D0
           EP(1,I) = (SIG(1,I)-NU(I)*(SIG(2,I)+SIG(3,I)))/E(I)
     A     + ALT1(I) + DEP1P(I)
           EP(2,I) = (SIG(2,I)-NU(I)*(SIG(1,I)+SIG(3,I)))/E(I)
     A     + ALT2(I) + DEP2P(I)
           EP(3,I) = (SIG(3,I)-NU(I)*(SIG(1,I)+SIG(2,I)))/E(I)
     A     + ALT3(I) + DEP3P(I)
          END DO
         END IF
c    Calculate stresses and strains at all nodes for condition of
c     fuel/clad contact with slippage allowed
         IF (ICON.EQ.1) THEN
          SIG(1,1) = X(1)
          SIG(3,1) = X(2)
          SIG(2,N+2) = X(3)
          SIG(2,N+1) = X(4)
          SIG(3,N+1) = X(5)
          SIG(2,1) = X(1)
          IF (R(1).GT.0.D0) SIG(1,1) = -PI
          SIGEQ(1) = (1.D0/2.D0**.5D0)*((SIG(1,1)-SIG(2,1))**2
     A     + (SIG(1,1)-SIG(3,1))**2+(SIG(2,1)-SIG(3,1))**2)**.5D0
          EP(1,1) = (SIG(1,1)-NU(1)*(SIG(2,1)+SIG(3,1)))/E(1)
     A     + ALT1(1) + DEP1P(1)
          EP(2,1) = (SIG(2,1)-NU(1)*(SIG(1,1)+SIG(3,1)))/E(1)
     A     + ALT2(1) + DEP2P(1)
          EP(3,1) = (SIG(3,1)-NU(1)*(SIG(1,1)+SIG(2,1)))/E(1)
     A     + ALT3(1) + DEP3P(1)
          DO I = 2,N
           SIG(1,I) = A(1,1,I-1)*SIG(1,1) + A(1,2,I-1)*SIG(2,1)
     A     + A(1,3,I-1)*SIG(3,1) + B(1,I-1)
           SIG(2,I) = A(2,1,I-1)*SIG(1,1) + A(2,2,I-1)*SIG(2,1)
     A     + A(2,3,I-1)*SIG(3,1) + B(2,I-1)
           SIG(3,I) = A(3,1,I-1)*SIG(1,1) + A(3,2,I-1)*SIG(2,1)
     A     + A(3,3,I-1)*SIG(3,1) + B(3,I-1)
           SIGEQ(I) = (1.D0/2.D0**.5D0)*((SIG(1,I)-SIG(2,I))**2
     A     + (SIG(1,I)-SIG(3,I))**2+(SIG(2,I)-SIG(3,I))**2)**.5D0
           EP(1,I) = (SIG(1,I)-NU(I)*(SIG(2,I)+SIG(3,I)))/E(I)
     A     + ALT1(I) + DEP1P(I)
           EP(2,I) = (SIG(2,I)-NU(I)*(SIG(1,I)+SIG(3,I)))/E(I)
     A     + ALT2(I) + DEP2P(I)
           EP(3,I) = (SIG(3,I)-NU(I)*(SIG(1,I)+SIG(2,I)))/E(I)
```

```
     A    + ALT3(I) + DEP3P(I)
          END DO
          SIG(1,N+1) = SIG(1,N)
          SIG(1,N+2) = -PE
          SIG(3,N+2) = -SIG(3,N+1)
          DO I = N+1,N+2
           SIGEQ(I) = (1.D0/2.D0**.5D0)*((SIG(1,I)-SIG(2,I))**2
     A    + (SIG(1,I)-SIG(3,I))**2+(SIG(2,I)-SIG(3,I))**2)**.5D0
           EP(1,I) = (SIG(1,I)-NU(I)*(SIG(2,I)+SIG(3,I)))/E(I)
     A    + ALT1(I) + DEP1P(I)
           EP(2,I) = (SIG(2,I)-NU(I)*(SIG(1,I)+SIG(3,I)))/E(I)
     A    + ALT2(I) + DEP2P(I)
           EP(3,I) = (SIG(3,I)-NU(I)*(SIG(1,I)+SIG(2,I)))/E(I)
     A    + ALT3(I) + DEP3P(I)
          END DO
         END IF
c    Calculate stresses and strains at all nodes for condition of
c     fuel/clad contact with no slippage allowed
         IF (ICON.EQ.2) THEN
          SIG(1,1) = CINV(1,1)*D(1) + CINV(1,2)*D(2) + CINV(1,3)*D(3)
          SIG(2,1) = CINV(2,1)*D(1) + CINV(2,2)*D(2) + CINV(2,3)*D(3)
          SIG(3,1) = CINV(3,1)*D(1) + CINV(3,2)*D(2) + CINV(3,3)*D(3)
          SIGEQ(1) = (1.D0/2.D0**.5D0)*((SIG(1,1)-SIG(2,1))**2
     A    + (SIG(1,1)-SIG(3,1))**2+(SIG(2,1)-SIG(3,1))**2)**.5D0
          EP(1,1) = (SIG(1,1)-NU(1)*(SIG(2,1)+SIG(3,1)))/E(1)
     A    + ALT1(1) + DEP1P(1)
          EP(2,1) = (SIG(2,1)-NU(1)*(SIG(1,1)+SIG(3,1)))/E(1)
     A    + ALT2(1) + DEP2P(1)
          EP(3,1) = (SIG(3,1)-NU(1)*(SIG(1,1)+SIG(2,1)))/E(1)
     A    + ALT3(1) + DEP3P(1)
          DO I = 2,N+2
           SIG(1,I) = A(1,1,I-1)*SIG(1,1) + A(1,2,I-1)*SIG(2,1)
     A    + A(1,3,I-1)*SIG(3,1) + B(1,I-1)
           SIG(2,I) = A(2,1,I-1)*SIG(1,1) + A(2,2,I-1)*SIG(2,1)
     A    + A(2,3,I-1)*SIG(3,1) + B(2,I-1)
           SIG(3,I) = A(3,1,I-1)*SIG(1,1) + A(3,2,I-1)*SIG(2,1)
     A    + A(3,3,I-1)*SIG(3,1) + B(3,I-1)
           SIGEQ(I) = (1.D0/2.D0**.5D0)*((SIG(1,I)-SIG(2,I))**2
     A    + (SIG(1,I)-SIG(3,I))**2+(SIG(2,I)-SIG(3,I))**2)**.5D0
           EP(1,I) = (SIG(1,I)-NU(I)*(SIG(2,I)+SIG(3,I)))/E(I)
     A    + ALT1(I) + DEP1P(I)
           EP(2,I) = (SIG(2,I)-NU(I)*(SIG(1,I)+SIG(3,I)))/E(I)
     A    + ALT2(I) + DEP2P(I)
           EP(3,I) = (SIG(3,I)-NU(I)*(SIG(1,I)+SIG(2,I)))/E(I)
     A    + ALT3(I) + DEP3P(I)
          END DO
         END IF
c    Calculate plastic strain increments at each node where
c     yielding occurs
         DO 160 I = 1,N+2
c    If no yielding occurs at node "i", go to next node
          IF (SIGEQ(I).LE.EQSTR(I).AND.IPLAST(I).EQ.0) GO TO 160
           IPLAST(I) = 1
c    Calculate equivalent modified total strain at node "i"
          EPET(I) = ((2.D0**.5D0)/3.D0)*((EP(1,I)-EP(2,I))**2
     A    +(EP(1,I)-EP(3,I))**2+(EP(2,I)-EP(3,I))**2)**.5D0
c    Calculate effective plastic strain increment at node "i"
```

```
      DEPEQP(I) = (EPET(I)-(2.D0/3.D0)*((1.D0+NU(I))/E(I))*EQSTR(I))/
     A    (1.D0+(2.D0/3.D0)*((1.D0+NU(I))/E(I))*MOD(I))
c   Calculate components of plastic strain increment at node "i"
      DEP1P(I) = (DEPEQP(I)/(3.D0*EPET(I)))*(2.D0*EP(1,I)
     A    -EP(2,I)-EP(3,I))
      DEP2P(I) = (DEPEQP(I)/(3.D0*EPET(I)))*(2.D0*EP(2,I)
     A    -EP(1,I)-EP(3,I))
      DEP3P(I) = - DEP1P(I) - DEP2P(I)
  160 CONTINUE
c  Check for convergence of plastic strain increments at all nodes
      DO 200 I = 1,N+2
       IF (DABS(DEP1P(I)-DEP1PR(I)).LT.1.D-6.AND.
     A     DABS(DEP2P(I)-DEP2PR(I)).LT.1.D-6.AND.
     A     DABS(DEP3P(I)-DEP3PR(I)).LT.1.D-6) THEN
        GO TO 200
       ELSE
c  Retain plastic strain increments to allow for a convergence check
c   in next iteration
        DO J = 1,N+2
          DEP1PR(J) = DEP1P(J)
          DEP2PR(J) = DEP2P(J)
          DEP3PR(J) = DEP3P(J)
        END DO
c  Perform another iteration in plastic analysis
        GO TO 110
       END IF
  200 CONTINUE
c  Once convergence is reached calculate total plastic strain,
c   total strain, new yield strength, and displacements at each node
      DO I = 1,N+2
c  Components of accumulated plastic strain at node "i"
       EP1P(I) = EP1P(I) + DEP1P(I)
       EP2P(I) = EP2P(I) + DEP2P(I)
       EP3P(I) = EP3P(I) + DEP3P(I)
c  Components of total strain at node "i"
       EPT(1,I) = (SIG(1,I)-NU(I)*(SIG(2,I)+SIG(3,I)))/E(I)
     A    + ALT1(I) + EP1P(I)
       EPT(2,I) = (SIG(2,I)-NU(I)*(SIG(1,I)+SIG(3,I)))/E(I)
     A    + ALT2(I) + EP2P(I)
       EPT(3,I) = (SIG(3,I)-NU(I)*(SIG(1,I)+SIG(2,I)))/E(I)
     A    + ALT3(I) + EP3P(I)
c  Calculate new yield strength at node "i" for next increment
       IF (SIGEQ(I).GT.EQSTR(I)) EQSTR(I) = EQSTR(I)+DEPEQP(I)*MOD(I)
c  Radial displacement at node "i"
       UR(I) = EPT(2,I)*R(I)
c  Updated radius at node "i"
       R(I) = R0(I) + UR(I)
c  Axial displacement at node "i"
       UZ(I) = EPT(3,I)*DELZ(I)
c  Updated thickness at node "i"
       DELZ(I) = DELZ0(I) + UZ(I)
      END DO
c   Check contact condition at end of load increment to be used in
c    next increment
      IF (ICON.EQ.2) THEN
       IF (-SIG(1,N).LT.PO) ICON=1
       IF (SIG(1,N).GT.0.D0) ICON=0
```

```
       GO TO 333
      END IF
      IF (ICON.EQ.1) THEN
       IF (-SIG(1,N).GT.PO) ICON=2
       IF (SIG(1,N).GT.0.D0) ICON=0
       GO TO 333
      END IF
      IF (R(N).GE.R(N+1)) THEN
        ICON = 1
        R(N) = (R(N)+R(N+1))/2.D0
        R(N+1) = R(N)
        IF (-SIG(1,N).GE.PO) ICON=2
      END IF
c  Save total strain in axial direction for gap region (to be used for
c   non-slip condition)
  333 DO I = N,N+1
       EPT0(3,I) = EPT(3,I)
      END DO
c  Increment loading for next load increment
       IF (JJ.GT.1000) GO TO 499
       T(1) = 6.0D-1
       T(2) = 5.5D-1
       T(3) = 5.0D-1
       T(4) = 4.5D-1
       T(5) = 4.0D-1
       T(6) = 3.5D-1
       T(7) = 3.0D-1
       T(8) = 2.5D-1
       T(9) = 2.0D-1
       T(10) = 1.5D-1
       T(11) = 1.0D-1
       T(12) = 0.6D-1
       T(13) = 0.3D-1
        GO TO 699
  499  IF (JJ.GT.1500) GO TO 599
       T(1) = -6.0D-1
       T(2) = -5.5D-1
       T(3) = -5.0D-1
       T(4) = -4.5D-1
       T(5) = -4.0D-1
       T(6) = -3.5D-1
       T(7) = -3.0D-1
       T(8) = -2.5D-1
       T(9) = -2.0D-1
       T(10) = -1.5D-1
       T(11) = -1.0D-1
       T(12) = -0.6D-1
       T(13) = -0.3D-1
        GO TO 699
  599  T(1) = 12.0D-1
       T(2) = 11.0D-1
       T(3) = 10.0D-1
       T(4) = 9.0D-1
       T(5) = 8.0D-1
       T(6) = 7.0D-1
       T(7) = 6.0D-1
       T(8) = 5.0D-1
```

```
      T(9) = 4.0D-1
      T(10) = 3.0D-1
      T(11) = 2.0D-1
      T(12) = 1.2D-1
      T(13) = 0.6D-1
  699 IF (JJ.GT.1000) GO TO 799
      PO = PO + 1.0D0
      GO TO 300
  799 IF (JJ.GT.1500) GO TO 899
      PO = PO - 1.0D0
      GO TO 300
  899 PO = PO + 2.0D0
c 899 PO = PO +0.D0
c     FZ = FZ + 1.D-2
  300 CONTINUE
c  All load steps are complete, notify user
      write (*,*) 'program end'
     END PROGRAM MECHMODEL

     SUBROUTINE INVERT(A,AINV)
c   Subroutine inverts a 3x3 matrix
      IMPLICIT NONE
      DOUBLE PRECISION A,AINV,DETA
      INTEGER J,K
      DIMENSION A(3,3),AINV(3,3)
      AINV(1,1) = A(2,2)*A(3,3)-A(3,2)*A(2,3)
      AINV(1,2) = -(A(1,2)*A(3,3)-A(3,2)*A(1,3))
      AINV(1,3) = A(1,2)*A(2,3)-A(2,2)*A(1,3)
      AINV(2,1) = -(A(2,1)*A(3,3)-A(3,1)*A(2,3))
      AINV(2,2) = A(1,1)*A(3,3)-A(3,1)*A(1,3)
      AINV(2,3) = -(A(1,1)*A(2,3)-A(2,1)*A(1,3))
      AINV(3,1) = A(2,1)*A(3,2)-A(3,1)*A(2,2)
      AINV(3,2) = -(A(1,1)*A(3,2)-A(3,1)*A(1,2))
      AINV(3,3) = A(1,1)*A(2,2)-A(2,1)*A(1,2)
      DETA = A(1,1)*(A(2,2)*A(3,3)-A(3,2)*A(2,3))
     &       - A(1,2)*(A(2,1)*A(3,3)-A(3,1)*A(2,3))
     &       + A(1,3)*(A(2,1)*A(3,2)-A(3,1)*A(2,2))
      DO 10 J = 1,3
       DO 20 K = 1,3
  20    AINV(J,K) = AINV(J,K)/DETA
  10  CONTINUE
     END SUBROUTINE INVERT

     SUBROUTINE GAUSS(N,A,X)
c  Subroutine performs Gaussian elimination to solve a set of N
c   simultaneous linear equations
      IMPLICIT NONE
      DOUBLE PRECISION A,SAVE,R,X
      INTEGER N,IM1,K,L,I,J,M,MM
      DIMENSION A(N,N+1),X(N)
      DO 20 I = 2,N
       DO 20 J = I,N
        IF (DABS(A(I-1,I-1)).LT.1.D-12) A(I-1,I-1)=0.D0
        IF (A(I-1,I-1)) 1,2,1
   2    IM1 = I-1
        DO 21 M = I,N
         IF (A(M,IM1)) 3,21,3
```

```
 3      DO 22 MM = IM1,N+1
         SAVE = A(M,MM)
         A(M,MM) = A(IM1,MM)
22      A(IM1,MM) = SAVE
21     CONTINUE
 1    R = A(J,I-1)/A(I-1,I-1)
      DO 20 K = I,N+1
20   A(J,K) = A(J,K) - R*A(I-1,K)
     DO 30 I = 2,N
      K = N-I+2
      R = A(K,N+1)/A(K,K)
      DO 30 J = I,N
       L = N-J+1
30   A(L,N+1) = A(L,N+1)-R*A(L,K)
     DO 40 I = 1,N
40   X(I) = A(I,N+1)/A(I,I)
    END SUBROUTINE GAUSS
```