PNCT 841-75-17

FUEL PERFORMANCE CODE "ACTIVE-II"

July, 1975

TOKAI WORKS

POWER REACTOR & NUCLEAR FUEL DEVELOPMENT CORPORATION

PNCT841-75-17
July, 1975

Fuel Performance Code "ACTIVE-II"

Masami KATSURAGAWA*
Hiroaki KAWAMURA**

Abstract

The "ACTIVE-II" is a fuel performance analysis code for a thermal and mechanical analysis of cylindrical reactor fuel element under irradiation, as a function of the actual reactor operating history. The fuel element mechanical behavior is analyzed by a generalized plane-strain procedure using matrix, especially emphasizing fuel-clad mechanical interactions, fuel cracking behavior and fine analysis of stress and strain distributions of fuel and cladding. The code also includes models of mechanical behavior such as restructuring, swelling and creep of fuel, fission gas release, plutonium migration, and swelling and creep of cladding.

The ACTIVE-II code is written in FORTRAN-IV on IBM model 360/195. The output of the calculation is given not only in digital form but also in graphic display by plotter. Since the code is constructed by variable dimension method, number of meshes for the calculation is almost infinite as far as the computer can accept.

The code can be provided not only for sodium cooled fast breeder reactor fuels, but also for fuels of PWR, BWR and heavy water reactor like FUGEN constructing in Japan. The models, equations, materials' properties and entire program list of the ACTIVE-II code are written in this report.

The program of this code is available from Department of Nuclear Fuels Development, Power Reactor and Nuclear Fuel Development Corporation.

---

* Fuel Design Section, Plutonium Fuel Division, Tokai Works
** IBM Japan, Ltd.

PNCT 841−75−17
1975年7月

Fuel Performance Code "ACTIVE-II"

実施責任者　青木利昌*
報告者　桂川正巳*
川村博章**

要旨　ACTIVE−IIは，燃料棒の熱的・機械的挙動を原子炉の運転履歴を追って計算するコードである。円柱状の燃料要素は軸対称として扱われ，平面歪解析による方程式を行列式で解くもので，特に燃料と被覆管の機械的相互作用，燃料のクラックおよび燃料と被覆管内の応力一歪分布を詳細に計算することを重点にしている。

コードにはまた燃料の組織変化，燃料と被覆管のスエリングおよびクリープ，分裂性ガス放出，プルトニウムの拡散等の照射下での現象を考慮している。

プログラムはIBM360/195用のFORTRAN−IVで書かれており，計算結果は数字とプロッターによる図面で出される。特にこのプログラムは可変ディメンジョン方式を採用しているため，計算するケースの大きさに制限はない。

対象とする燃料はナトリウム冷却高速炉，新型転換炉，加圧水型軽水炉および沸騰水型軽水炉の各種であり，オプションで選択できる。

本報告書ではプログラムのモデル，構成式，材料の物性値，およびプログラム・リストとその詳細な説明を記した。

プログラム入手御希望の場合には，核燃料開発本部宛御連絡下さい。

*　東海事業所プルトニウム燃料部設計課
**　日本アイ・ビー・エム株式会社

Contents

1. Introduction

ACTIVE-II is a fuel performance analysis code for digital computer.

Under irradiation, fuel pin seems to be a living creature. Fuel cracks, creeps and swells. Fission gas bubbles move up to temperature gradient and form columnar grain region and central void. Fission product ingots are formed in central void, and drops during shutdown. Cladding creeps and swells, and so on. Thus, in a reactor, fuel pin is very vivid and "active".

This computer code analyzes the movements of fuel pin under irradiation, and reproduces these "active" movements following the actual reactor history. Therefore, this fuel performance code is named "ACTIVE".

The usage of this fuel pin performance code is very wide. That is, it can supply the input condition for transient, safety analysis codes of fuel pins. The code can be used to research the phenomena which is most sensitive to the fuel pin life. It may also evaluate the safety margin for a fuel pin design. Furthermore, fuel pin performance code can contribute to the operation manual of a reactor for more safety and economical usage of fuel.

There are many computer codes which calculate fuel pin performance; i.e. LIFE-I, FMODEL, CYGRO, BEHAVE, SATURN and so on. The authers had used LIFE-I code for a while, and compared the models with experimental results. But in order to compile some new models, and to shorten the computer processing time, ACTIVE-II was developed based on LIFE-I and other codes.

ACTIVE-II has some characteristics compared with these codes.

First, it can calculate fine structure of stress and strain distribution. By using variable dimension method, the user can choose as many meshes as he wishes.

Second, the code is made up by many modules. Therefore, it's easy to change any models in the code.

Third, ACTIVE-II can deal with any type of reactor fuel which has sylindrical shape. (i.e. fast breeder reactor fuel, heavy water reactor fuel, pressurized water reactor fuel and boiling water reactor fuel.)

Fourth, it can evaluate fuel cracking and healing which cause during reactor power change. Fuel cracking is evaluated until every stresses settle down below fracture stresses.

It is remarkable that the calculation time is minimized by adopting matrix method.

Although ACTIVE-II has many usages and characteristics, it is still under developing. There are many phenomena remaining unresolved which would be included in the code. These phenomena must be clarified through the verification of the code.

2. Code Abstract

2.1 Model

Assuming a cylindrical geometry, axi-symmetrical analysis is performed. As shown in Fig. 2.1, fuel column is devided into as many meshes as the user wishes by the plane perpendicular to the axis, and a plenum region is added at the top of the fuel column.

Fuel dimensions (i.e. clad outer diameter, clad inner diameter, fuel diameter and central hole diameter), pellet densities, O/M ratio, linear rod power and fast neutron flux are input for each axial region. These values are constant within each axial region, but have distributions for axial direction. Thermal and mass movements are not allowed between these axial regions.

Fuel central void is assumed to be separated from upper plenum during operation but to be connected during shutdown.

Radial cross section is shown in Fig. 2.2. Fuel is divided into three concentric zones corresponding to the regions with columner grain growth region, equiaxed grain growth region and undisturbed region, and within these regions, fuel densities are assumed to be constant. These three regions and cladding region are divided into many submeshes for stress and strain analysis. For example, columnar, equiaxed, undisturbed and cladding regions are divided into 3,3,4, and 4 submeshes, respectively.

For these sub-meshes, temperature, materials' properties including creep and swelling, stress and strain (using plane-strain analysis) and fuel cracking are calculated.

For time intervals, quasi-steady state is assumed.

2.2 Flow diagram

Fig. 2.3 shows the flow diagram of ACTIVE-II. At the beginning of the MAIN program, dimensions for the program are decided according to the variable dimension method.

Capsule dimensions, material properties' options, relative power and fast flux distributions, and fuel densities are input as the initial value, and then, operation conditions are prepared assuming stepwise power change.

First, boundary movement and fuel densities are calculated using the results of the former time step. Temperature distribution is calculated based on the coolant temperature assuming one dimensional calculation. Gap conductance between fuel and cladding is a function of gap width, and the thermal conductivity of fuel is a function of fuel density, temperature, and O/M ratio.

Using the equivalent stress of the former time step, the creep strains of fuel and cladding are calculated for each sub-region. If one of the increments of boundary movements, creep of fuel or creep of cladding during the time step is too much larger than the limit given by input, time interval is cut to half and the calculation is retried at the beginning of the temperature calculations.

Based on the temperature distribution, fuel restructuring, plutonium redistribution, fission gas release, plenum pressure, fuel burnup, fast neutron fluence and fuel and cladding swelling are also calculated.

Those results are used for the boundary conditions of stress and strain calculations, which are performed using matrix and gap condition iterations. That is, first open gap is assumed for the fuel - cladding interface and if the calculated results show no gap, the boundary conditions are reevaluated and the calculation is continued. After the calculation of stress and strain, stresses for radial, circumferential and axial directions of each sub-region are compared with the fracture stress, which depends on temperature, to check if the fuel cracking occurs. If one of the stresses is greater than the fracture stress, the code marks the fuel cracking at the region, and changes the Young's modulus of elasticity and Poisson's ratio of the region. Then the stress and strain are re-evaluated using these moderated properties, and the resulting stresses are compared with the fracture stresses.

After the convergence of the cracking routine or in the case of no cracking, healing conditions are checked and the whole calculated conditions are printed out before going to the next time step.

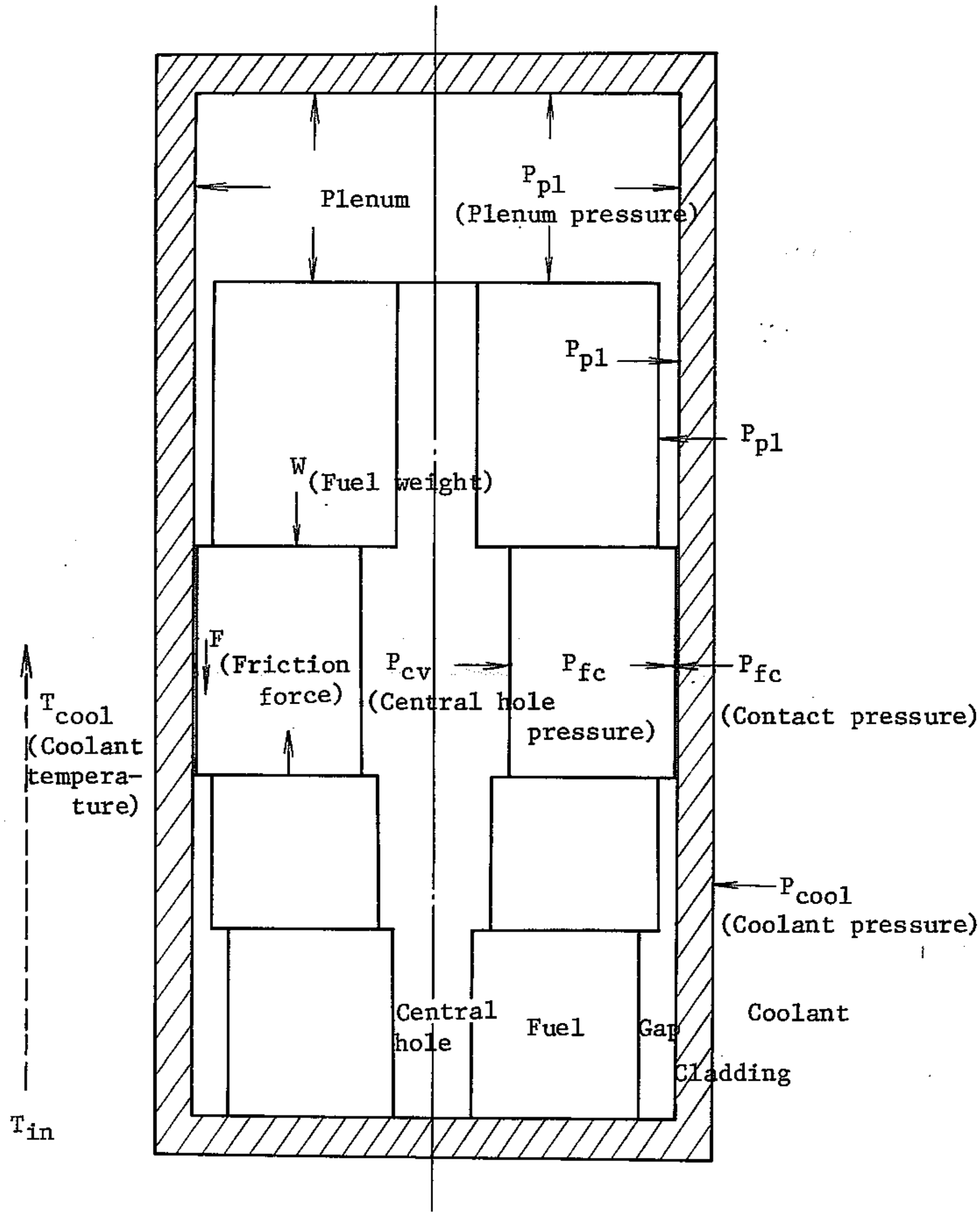


Fig. 2.1 Axial Cross Section of Fuel Pin

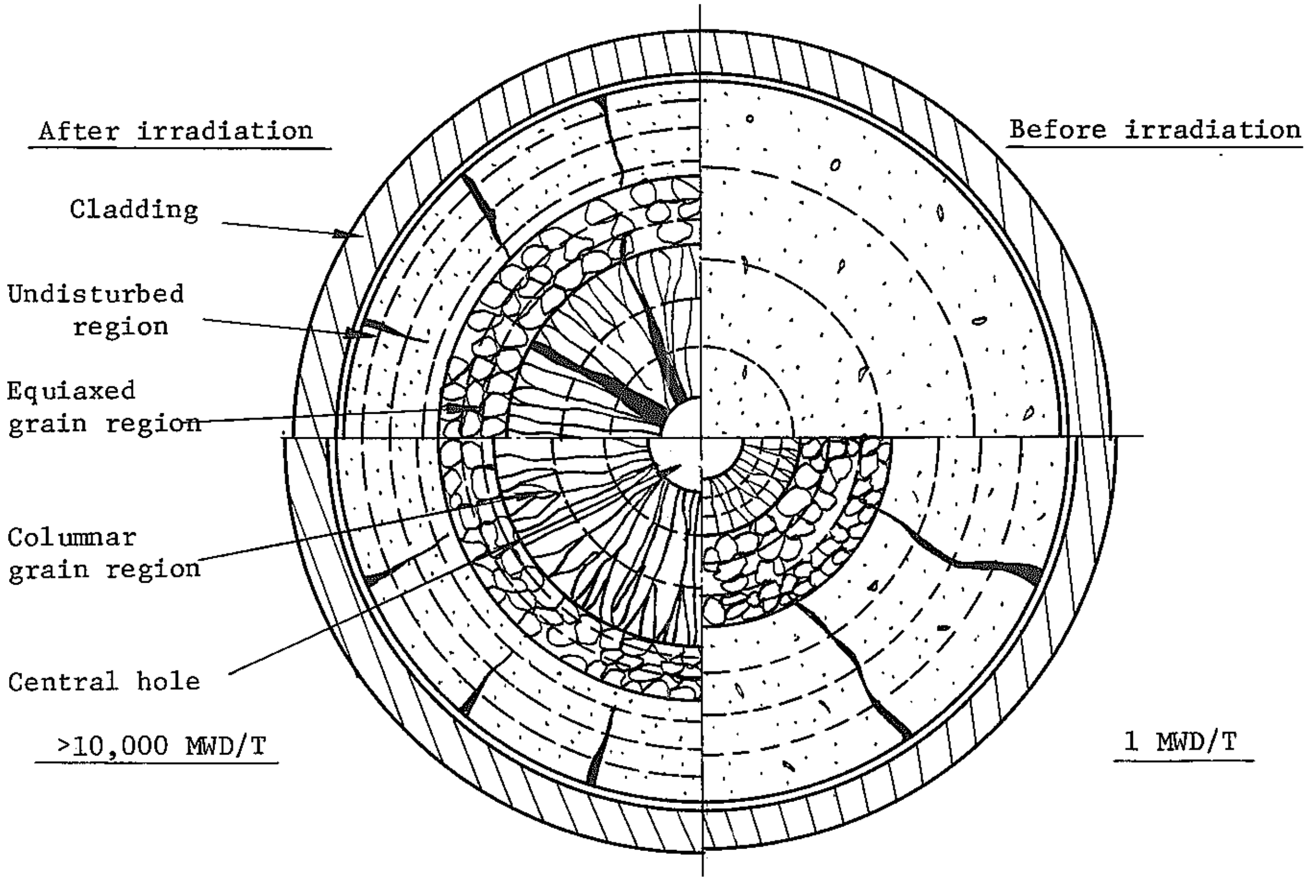


Fig. 2.2 Radial Cross Section of Fuel Pin

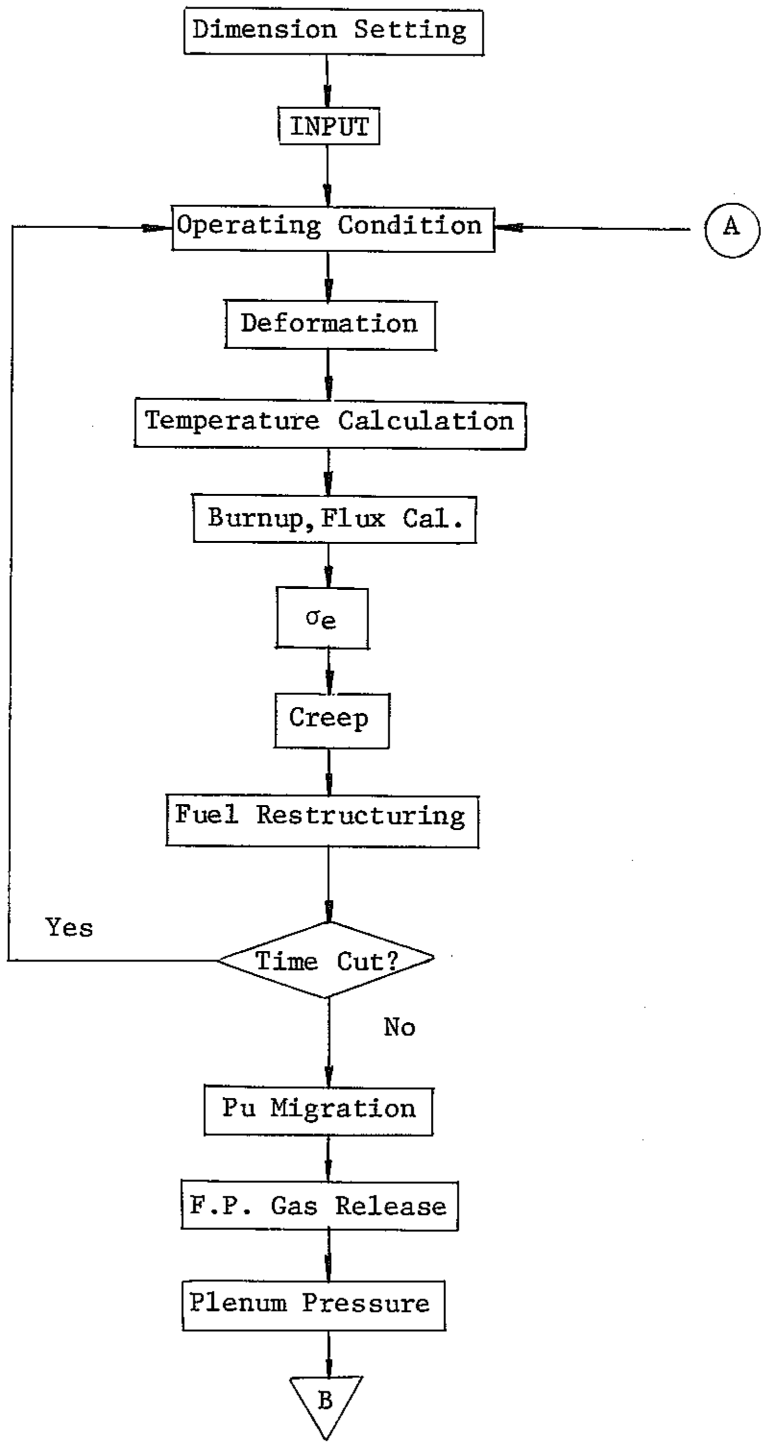


Fig. 2.3 Flow Diagram of ACTIVE-II

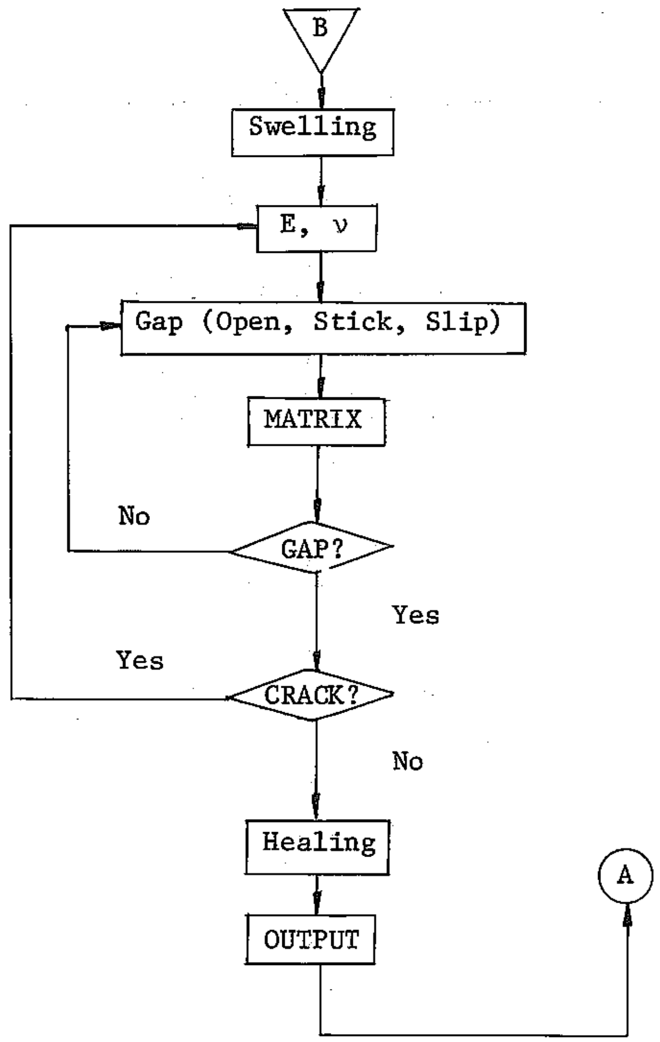


Fig. 2.3 Flow Diagram of ACTIVE-II (Continued)

3. Equations

3.1 Temperature Distribution

1) Coolant Temperature

Coolant temperature at the axial position (Z) is given by the following equation.

$$T_c(z) = T_c(0) + \frac{\int^{z} q(z)\,dZ}{\rho C_p v s} \qquad (1)$$

where, $T_c(z)$ = coolant temperature at the axial position (Z) (°C),
$T_c(0)$ = coolant inlet temperature (°C),
$q(z)$ = linear rod power (w/cm),
Z = axial distance from the bottom (cm),
C_p = coolant heat capacity (w·sec/g),
ρ = coolant density (g/cm^3),
v = coolant velocity (cm/sec),
and s = coolant flow area (cm^2).

2) Cladding Outer Surface Temperature

Cladding outer surface temperature is calculated based on coolant temperature at the same axial position.

$$T_{co} = T_c(z) + \frac{q}{2\pi r_{co} h_c} \qquad (2)$$

where, T_{co} = cladding outer surface temperature (°C),
$T_c(z)$ = coolant temperature (°C),
q = linear rod power (w/cm),
r_{co} = cladding outer radius (cm),
and h_c = film coefficient (w/cm^2·°C)

3) Temperature Distribution in Cladding

$$T_c(i) = T_{co} + \frac{(q/\pi)\,\ell n\{r_{co}/r_c(i)\}}{k_c + [k_c^2 + (a_2 q/\pi)\,\ell n\{r_{co}/r_c(i)\}]^{1/2}} \qquad (3)$$

where, $T_c(i)$ = cladding temperature at radius $r_c(i)$ (°C),
k_c = thermal conductivity of cladding ($=a_1+a_2T$) (w/cm·°C)

4) Fuel Surface Temperature

$$T_{fo} = T_{ci} + \frac{q}{2\pi r_g h_g} \tag{4}$$

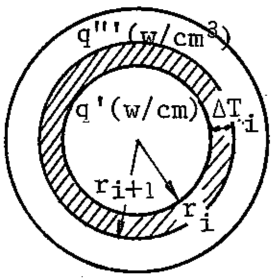
where, T_{fo} = fuel surface temperature (°C),
r_g = mean radius of fuel-clad gap (cm),
T_{ci} = clad inner surface temperature (°C),
and h_g = gap conductance (w/cm^2·°C).

5) Temperature Distribution in Fuel

Fuel radius is divided into some meshes, and in each region power generation rate is assumed to be constant. Temperature rise across a region ($r_i \le r \le r_{i+1}$) is given by the equation

$$\Delta T_i = \frac{q'''}{4k_f}(r_{i+1}^2 - r_i^2) - (\frac{q'''}{2k_f} r_i^2 - \frac{q'}{2\pi k_f}) \ \ell n \ \frac{r_{i+1}}{r_i}, \tag{5}$$

where q''' = heat generation rate in the region (w/cm^3),
k_f = fuel thermal conductivity (w/cm·°C),
r_i = inner radius of the region (cm),
r_{i+1} = outer radius of the region (cm),
and q' = linear heat rating within the radius r_i (w/cm).



In these equations, materials' properties can be selected by options.

3.2 Fuel Restructuring

It is assumed that, during irradiation, fuel is divided into three regions (i.e. columnar grain growth region, equiaxed grain growth region and undisturbed region), and central hole opens at the center. These region boundaries are assumed to grow in a certain velocity.

1) Columnar Grain Growth Region

As columnar grain is believed to be formed by the movement of voids in the fuel, the velocity of the region boundary growth depends upon the velocity of void in the fuel. That is, the velocity is calculated by the following equation.

$$v = C_1 T_k^{-3/2} e^{-\frac{Q}{RT_k}} \frac{dT}{dr} \quad \text{(cm/sec)} \tag{6}$$

where C_1 = constant given by fine tuning of the code,
T_k = temperature of columnar grain region outer boundary (°K),
Q = activation energy (cal/mole),
R = 1.987 (cal/mole·°K)
and $\frac{dT}{dr}$ = temperature gradient at columnar grain region outer boundary (°C/cm).

2) Equiaxed Grain Growth Region

Assuming that equiaxed grain growth depends on the temperature only, the velocity of the region boundary is calculated by the following equation.

$$v = C_2 e^{-\frac{Q}{RT}} \quad \text{(cm/sec)}, \tag{7}$$

where C_2 = constant calculated by fine tuning of the code,
T = region outer boundary temperature (°K),
Q = activation energy (cal/mole),
and R = 1.987 (cal/mole °K).

3) Central Void

As the results of the restructuring described above, a certain central void is formed at the center of the fuel. The radius of the central hole is calculated by the following mass balance equation.

$$\begin{aligned} M = {} & \pi\cdot\rho_{col}(r_{col}^2 - r_{cv}^2) + \pi\cdot\rho_{eq}(r_{eq}^2 - r_{col}^2) \\ & + \pi\cdot\rho_{und}(r_f^2 - r_{eq}^2) \end{aligned} \tag{8}$$

where M = fuel weight per unit length (g/cm),

ρ_i = fuel density of region i (g/cm^3),

r_i = outer radius of region i (cm),

and the suffix i means,

cv = central void,

col = columnar grain growth region,

eq = equiaxed grain growth region,

and f = fuel pellet.

4) Average Densities of the Regions

Within each regions, density is assumed to be independent on the position, and is calculated as the functions of the initial pellet density and burnup.

Density of columnar grain growth region

$$\rho_{col} = \rho^{\circ}_{col} + (0.99 \times \rho_{th} - \rho^{\circ}_{col}) \times \frac{B}{B_o}, \tag{9}$$

$$\rho^{\circ}_{col} = 0.767 \times \rho_o + 0.233 \times \rho_{th}, \tag{10}$$

where ρ_{th} = theoretical density (g/cm^3),

ρ_o = initial pellet density (g/cm^3),

B = burnup (a/o),

and B_o = constant calculated through fine tuning of the code (a/o).

Density of equiaxed grain growth region

$$\rho_{eq} = \rho^{\circ}_{eq} + (0.965 \times \rho_{th} - \rho^{\circ}_{eq}) \times \frac{B}{B_o} \tag{11}$$

$$\rho^{\circ}_{eq} = \rho_o \tag{12}$$

Density of undisturbed region

$$\rho_{und} = \rho_o \tag{13}$$

3.3 Plutonium Migration

Plutonium redistribution is calculated in a subroutine, but the plutonium generation and the feed back to the power distribution are neglected.

The fuel is devided into some concentric cylindrical shells of equal mass, and the calculation is based on the assumption that the plutonium migration is caused by thermal and chemical diffusion.

$$\rho \frac{dC^{Pu}}{dt} = -\nabla J^{Pu} , \tag{14}$$

where, ρ = fuel density,

C^{Pu} = plutonium concentration,

J^{Pu} = flux of plutonium,

$= \rho \times D' \times W^{Pu} \times W^{U} \times \nabla T - \rho \times D \times \nabla W^{Pu}$,

D' = thermal diffusion coefficient,

D = diffusion coefficient of plutonium in the mixed oxide fuel,

W^{Pu} = w/o of plutonium,

and W^{U} = w/o of uranium.

3.4 Plenum Pressure and Central Void Pressure

Enclosed gas (usually helium gas), absorbed gas in the fuel (H_2O, N_2 etc.) and fission product gas are mixed and are filled in the plenum, fuel-clad gap and central void. Central void is assumed to be a pressure vessel separated from the plenum during operation, and to be connected after shut down.

Plenum pressure is calculated by the equation

$$P_{pl} = \frac{nRT_{pl}}{V_{pl} + V_{cv} + V_g} \tag{15}$$

where P_{pl} = plenum pressure (kg/cm^2),

n = total gas moles (moles),

R = Boltzmann constant(cm^3-atm/mole °K),

T_{pl} = plenum temperature (°K),

V_{pl} = plenum volume (cm^3),

V_g = fuel-clad gap volume (cm^3),

and V_{cv} = central void volume (cm^3).

Central void pressure is calculated by the equation

$$P_{cv} = (n \times \frac{V_{cv}}{V_{pl}+V_{cv}}) \times \frac{R}{\sum_{i=1}^{NF} \frac{V_{cv}(i)}{T_{cv}(i)}} \tag{16}$$

where P = central void pressure (kg/cm^2),

NF = number of axial nodes,

$T_{cv}(i)$ = central void temperature of axial node-i (°K),

and $V_{cv}(i)$ = central void volume of axial node-i (°K),

3.5 Creep Deformation

Creep deformation is calculated based on the second creep rate of the material, and the creep rate functions include both thermal and irradiation enhanced creep.

$$\dot{\varepsilon}_p = A\sigma_e^m e^{-\frac{Q}{RT}} + B\sigma_e^n \phi \tag{17}$$

where $\dot{\varepsilon}_p$ = creep rate (1/hr),

Q = activation energy for thermal creep (cal/mole),

R = 1.987 (cal/mole °K),

T = temperature (°K),

ϕ = neutron flux ($n/cm^2 \cdot sec$),

σ_e = equivalent stress

$= \frac{1}{\sqrt{2}} [\sigma_r - \sigma_\theta)^2 + (\sigma_\theta - \sigma_z)^2 + (\sigma_z - \sigma_r)^2]^{1/2}$,

and A, B, m, n = constants.

3.6 Swelling

1) Fuel

Swelling of fuel is divided into three components, which are swelling by solid f.p., by gasious f.p. and hot pressing.

$$\dot{\varepsilon}_s = \dot{\varepsilon}_{s.s.} + \dot{\varepsilon}_{s.g.} + \dot{\varepsilon}_{s.hp} \tag{18}$$

where $\dot{\varepsilon}_s$ = total swelling rate (1/hr)

$\dot{\varepsilon}_{s.s.}$ = swelling caused by solid fission products (1/hr),

$\dot{\varepsilon}_{s.g.}$ = swelling caused by gaseous fission products(1/hr),

and $\dot{\varepsilon}_{s.hp}$ = hot pressing components (1/hr).

The equations for these components will be given in the section of materials' properties.

2) Cladding

The swelling rate of cladding is chosen from some empirical equations which are given in the section of materials' properties.

3.7 Mechanical Analysis

Equations between stress and strain are given for each concentric cylinder regions based on plane-strain analysis.
Nomenclats used in the equations are as follows.

σ_r, σ_θ, σ_z = principal stresses for r, θ and z,

ε_r, ε_θ, ε_z = total strains for r, θ and z directions,

ε_r^p, ε_θ^p, ε_z^p = creep strains for r, θ and z directions,

ε_r^s, ε_θ^s, ε_z^s = swelling strains for r, θ and z directions,

u = displacement of radial direction during time interval of Δt (excluding fuel restructure),

αT = thermal expansion,

E = Young's modulus of elasticity,

γ = Poisson's ratio,

P = pressure,

and Δ = increment during time interval of Δt.

1) Constitutive relations

Total strain is the sum of elastic strain, thermal expansion, creep strain and swelling strain.

$$\varepsilon_r^T = \frac{1}{E}\{\sigma_r - \nu(\sigma_\theta + \sigma_z)\} + \alpha T + \varepsilon_r^p + \varepsilon_r^s \tag{19}$$

$$\varepsilon_\theta^T = \frac{1}{E}\{\sigma_\theta - \nu(\sigma_z + \sigma_r)\} + \alpha T + \varepsilon_\theta^p + \varepsilon_\theta^s \tag{20}$$

$$\varepsilon_z^T = \frac{1}{E}\{\sigma_z - \nu(\sigma_r + \sigma_\theta)\} + \alpha T + \varepsilon_z^p + \varepsilon_z^s \tag{21}$$

2) Equilibrium

As axial symmetry is assumed, principal stresses are combined by the equation,

$$\frac{\partial \sigma_r}{\partial r} + \frac{\sigma_r - \sigma_\theta}{r} = 0 \tag{22}$$

3) Equation between strain and displacement

As a state of generalized plane strain is assumed, strain increments in Δt are functions of increment of radial displacement in Δt as follows.

$$\Delta\varepsilon_r^T = \frac{\partial u}{\partial r} \tag{23}$$

$$\Delta\varepsilon_\theta^T = \frac{u}{r} \tag{24}$$

$$\Delta\varepsilon_z^T = \text{constant} \tag{25}$$

4) Creep law

Prandtle-Reuss flow laws are used for creep strain calculation.

$$\Delta\varepsilon_r^p = \frac{\Delta\varepsilon_p}{2\sigma_e} \quad (2\sigma_r - \sigma_\theta - \sigma_z), \tag{26}$$

$$\Delta\varepsilon_\theta^p = \frac{\Delta\varepsilon_p}{2\sigma_e} \times (2\sigma_\theta - \sigma_z - \sigma_r), \tag{27}$$

$$\Delta\varepsilon_z^p = \frac{\Delta\varepsilon_p}{2\sigma_e} \times (2\sigma_z - \sigma_r - \sigma_\theta), \tag{28}$$

where

$$\sigma_e = \frac{1}{\sqrt{2}} \{(\sigma_r-\sigma_\theta)^2 + (\sigma_\theta-\sigma_z)^2 + (\sigma_z-\sigma_r)^2\}^{1/2} \tag{29}$$

(equivalent stress),

and

$$\Delta\varepsilon_p = f_s(\sigma_e)$$

(which is given by the equation (17)).

5) Boundary conditions

At the inner and outer boundary of each mesh, radial stress is equivalent to the contact pressure.

$$\sigma_r = -P_i \qquad \text{at } r = r_i, \tag{30}$$

$$\sigma_r = -P_o \qquad \text{at } r = r_o, \tag{31}$$

where i means inner boundary and o means outer boundary.

From the pressure balance between adjacent meshes,

$$P_{o,j-1} = P_{i,j} , \tag{32}$$

and from the continuity of the boundary,

$$U_{o,j-1} = U_{i,j} \ , \tag{33}$$

where j denotes mesh number.

At the central hole of fuel, contact pressure equals central hole gas pressure,

$$P_{i,1} = P_{cv} \ , \tag{34}$$

and at the outer surface of cladding, contact pressure equals coolant pressure,

$$P_{o,clad} = P_{cool} . \tag{35}$$

At the interface between fuel and cladding, two kinds of conditions are considered.

(i) When gap exists between fuel and clad, pressure of fuel outer surface and that of clad inner surface equal to plenum pressure.

$$P_{o,fuel} = P_{pl} \ , \tag{36}$$

$$P_{i,clad} = P_{pl} . \tag{37}$$

(See Fig. 3.1)

(ii) When fuel contacts with cladding, pressure of both fuel outer and cladding inner surface equal to contact pressure, and increments of radial displacement of both boundary are same.

$$P_{o,fuel} = P_{i,clad} = P_{fc} \tag{38}$$

$$U_{o,fuel} = U_{i,clad} \tag{39}$$

(See Fig. 3.2)

As the boundary conditions of axial direction, mean stresses of both fuel and cladding axial direction balance outside forces. Friction force between fuel and cladding is also included in the boundary conditions of axial direction. (See Fig. 3.3)

(i) Fuel

$$\bar{\sigma}_{zf} = P_{pl} + \frac{\rho_f \cdot g \cdot V_f}{\pi(r_f^2 - r_c^2)} \times \frac{(L_f - Z)}{L_f} + \frac{F \times \Delta L_f}{\pi(r_f^2 - r_c^2)} \tag{40}$$

where $\overline{\sigma}_{zf}$ = mean stress of axial direction in fuel,
ρ_f = mean density of fuel,
g = gravity constant,
V_f = fuel volume per unit axial length,
r_f = fuel outer radius,
r_c = fuel central void radius,
L_f = total fuel column length,
Z = distance from bottom of fuel,
F = $2\mu\pi r_f P_{fc}$ (friction force),
μ = friction factor,
P_{fc} = contact pressure between fuel and cladding,
and ΔL_f = radial node length.

(ii) Cladding

$$\overline{\sigma}_{zc} = \frac{1}{\pi(r_{co}^2 - r_{ci}^2)} \times \{P_T + \pi(r_{co}^2 P_c - r_{ci}^2 P_{pl}) + \rho_c \cdot g \cdot V_c \cdot (1 - \frac{Z}{L_c}) - F \times \Delta L_f\} \tag{41}$$

where $\overline{\sigma}_{zc}$ = mean stress of axial direction in cladding,
ρ_c = density of cladding,
V_c = cladding volume per unit axial length,
r_{co} = cladding outer radius,
r_{ci} = cladding inner radius,
and L_c = total cladding length.

6) Mean values in each region

Mean values in each region are given by the following equation.

$$\overline{A}(r) = \frac{\int rA(r)\,dr}{\int r\,dr} \tag{42}$$

3.8 Fuel Cracking and Healing

During the stress-strain calculation, ACTIVE-II checks whether fuel cracking would occur in any of the three principal directions. The cracking criterion is; if any of the principal stresses in a fuel region reach the fracture strength, the fuel cracks to the perpendicular direction to that principal stress in that region. The fracture stress used in the code is the data prepared by Canon et al.[1].

$$\sigma_f = 15{,}000 + 3.7 \times T_k \tag{43}$$

where σ_f is the ultimate tensile stress in psi, and T_k is the temperature in °K. If crack occurs, the modifications mentioned below are applied to that region, and the stresses and strains are recalculated until none of the principal stresses in the fuel exceed the fracture stress.

The modification used in ACTIVE-II is the method proposed by Weeks et al. for LIFE-II[2)] code. The stress component perpendicular to a crack vanishes in the region, and the elastic strain existing prior to cracking is then redistributed. For instance, when the stress of the axial direction σ_z exceeds the fracture stress σ_f, a crack occurs perpendicular to the axial direction. The axial elastic strain then decreases by σ_z/E, and the elastic strains in the other principal directions increase by $\nu\sigma_z/E$ each. As it is assumed that no volume change occures by cracking, these changes in elastic strains are compensated by equal changes in plastic strains.

In ACTIVE-II code, plastic strains are divided into creep strains and swelling strains. Therefore the changes in plastic strains by creacking are described as follows.

Swelling strains for each direction

$$\Delta\varepsilon^s = \sigma_z/9B \tag{44}$$

Creep strains

$$\Delta\varepsilon_r^p = -\sigma_z/6G \tag{45}$$

$$\Delta\varepsilon_\theta^p = -\sigma_z/6G \tag{46}$$

$$\Delta\varepsilon_z^p = \sigma_z/3G \tag{47}$$

where B is the bulk modulus and G is the shear modulus.

The mechanical properties of the cracked fuel are modified as follows:

Young's modulus

$$E^c = E \times \left(\frac{2}{3}\right)^L \tag{48}$$

Poisson's ratio

$$\nu^{c} = \nu \times \left(\frac{1}{2}\right)^{L} \tag{49}$$

Creep rate

$$\dot{\varepsilon}_{p}^{c} = \dot{\varepsilon}_{p} \times \left(\frac{3}{2}\right)^{L\times(m-1)} \tag{50}$$

where L is the number of cracks and m is the stress exponent for the thermal creep.

During steady state operation, crack healing is allowed. Cracks are assumed to heal under the following conditions.

* The averaged stress in the region becomes compressive and exceeds a certain stress ($7000 kg/cm^2$).
* The temperature in the region exceeds 1400°C, and the pressure is compressive and exceeds 70 kg/cm^2.
* These conditions have continued more than 1 hour.

When the cracks heal, the mechanical properties of the region become those of uncracked fuel, but the plastic strains caused by cracking remain as before.

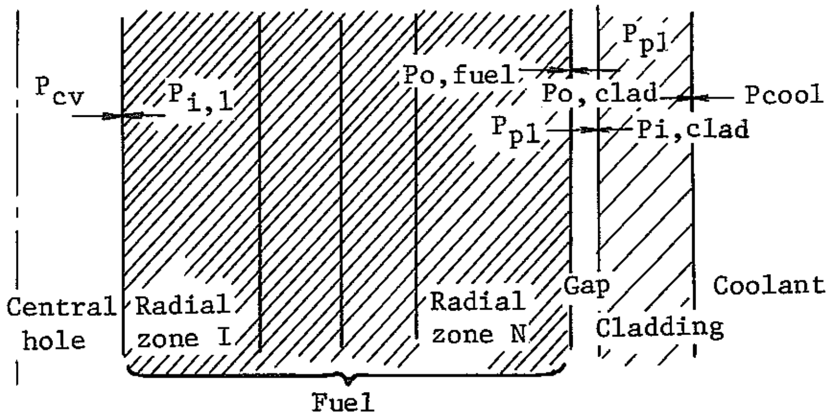


Fig. 3.1 Radial Force Balance (Gap Open)

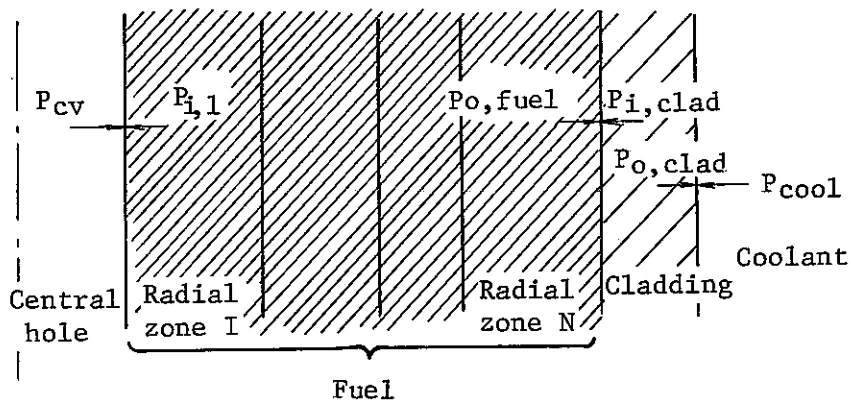


Fig. 3.2 Radial Force Balance (Gap Closed)

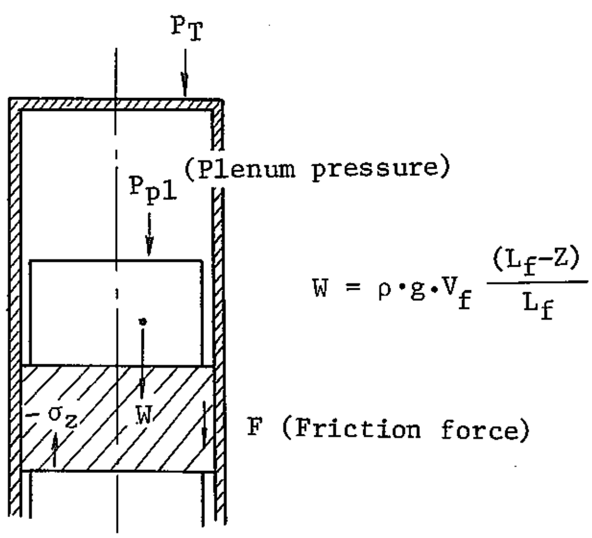


Fig. 3.3 Axial Force Balance

4. Solution Method

4.1 Program Construction

The code is written in FORTRAN-IV by using variable dimension method, in order to minimize the program size. Therefore, the MAIN routine is only used to decide the program size and the real calculation is done in subroutine ACTIVE which is called by MAIN ROUTINE.

1) MAIN program

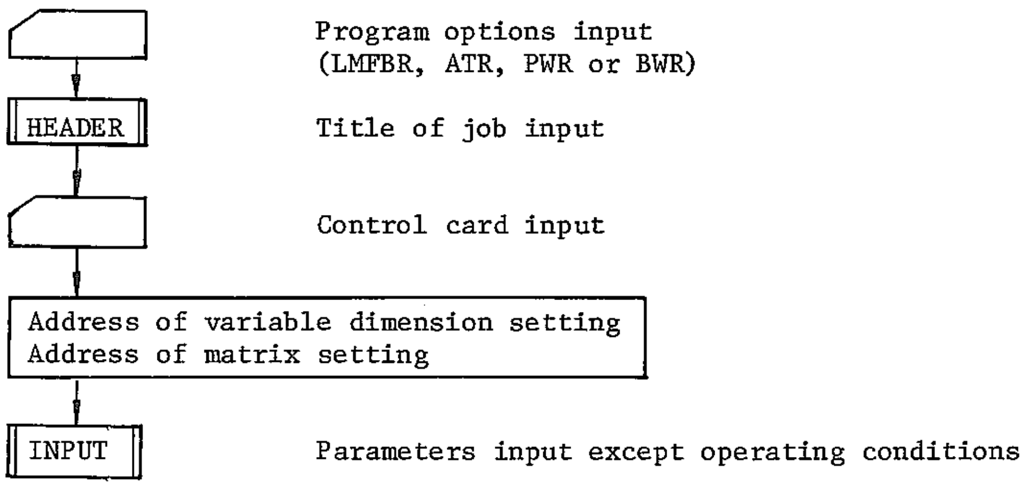
The program of MAIN is as follows.

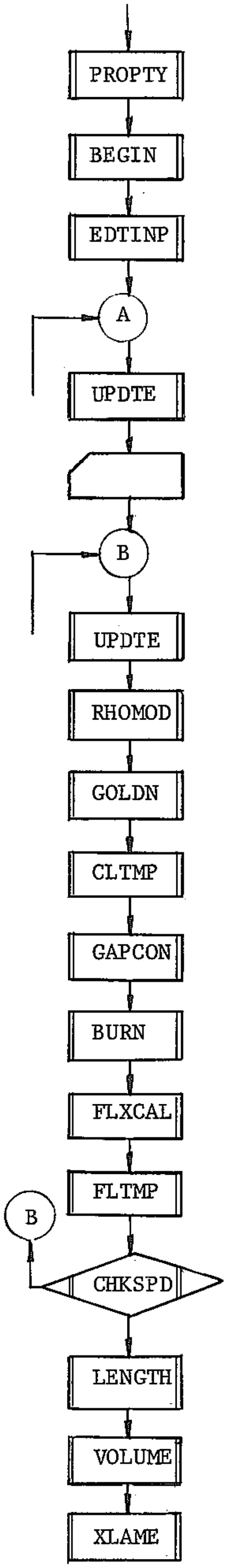
```
COMMON / DIM / IA,IAM
DIMENSION A(10000),AM(30000)
IA = 10000
IAM = 30000
CAL ACTIVE(A,AM)
STOP
END
```

In this program, symbol A denotes the all symboles which have dimensions except the symbols used in matrix solution. IA is the number of As. AM denotes the symbols used in matrix solution and IAM is the number of AMs. See section 6.1 for details.

2) Subroutine ACTIVE

As mentioned above, the calculation is done in the subroutine ACTIVE. The flow of the subroutine and the subroutines called are as follows.





Materials' properties selection

Initial values setting

Output the input parameters

Beginning of time step loop

Memory of parameters before time cut

Operating conditions input

Beginning of time step cut iteration

Renewal of parameters after time cut

Calculation of region boundaries and densities

Temperatures of coolant and cladding outer surface calculation

Temperature distribution in the cladding calculation

Gap conductance calculation

Burnup calculation

Neutron flux and fluence calculation

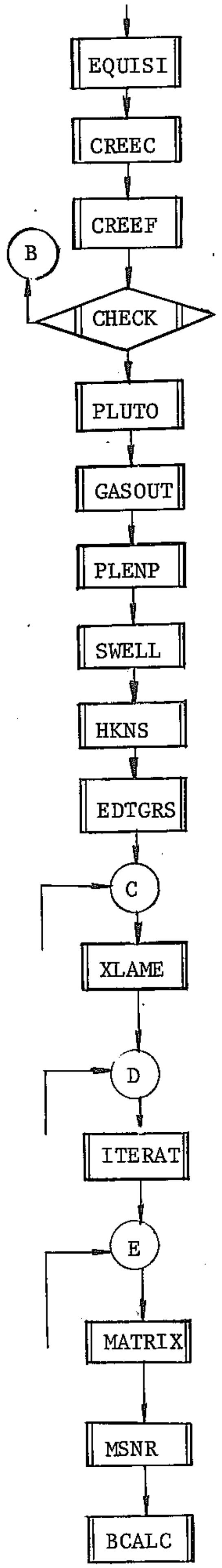
Temperature distribution in the fuel and fuel restructuring calculation

Check the fuel restructuring speed

Length of fuel and cladding calculation

Volumes of regions calculation

Young's modulus and Poisson's ratio of fuel calculation



Equivalent stress calculation

Cladding creep calculation

Fuel creep calculation

Check the creep rate

Plutonium migration calculation

F.P. gas release calculation

Plenum pressure

Fuel swelling rate by solid F.P. calculation

Cladding swelling rate calculation

Fuel pin gross output (power & temperature)

Axial position loop

Young's modulus and Poisson's ratio of fuel calculation

Fuel crack iteration loop

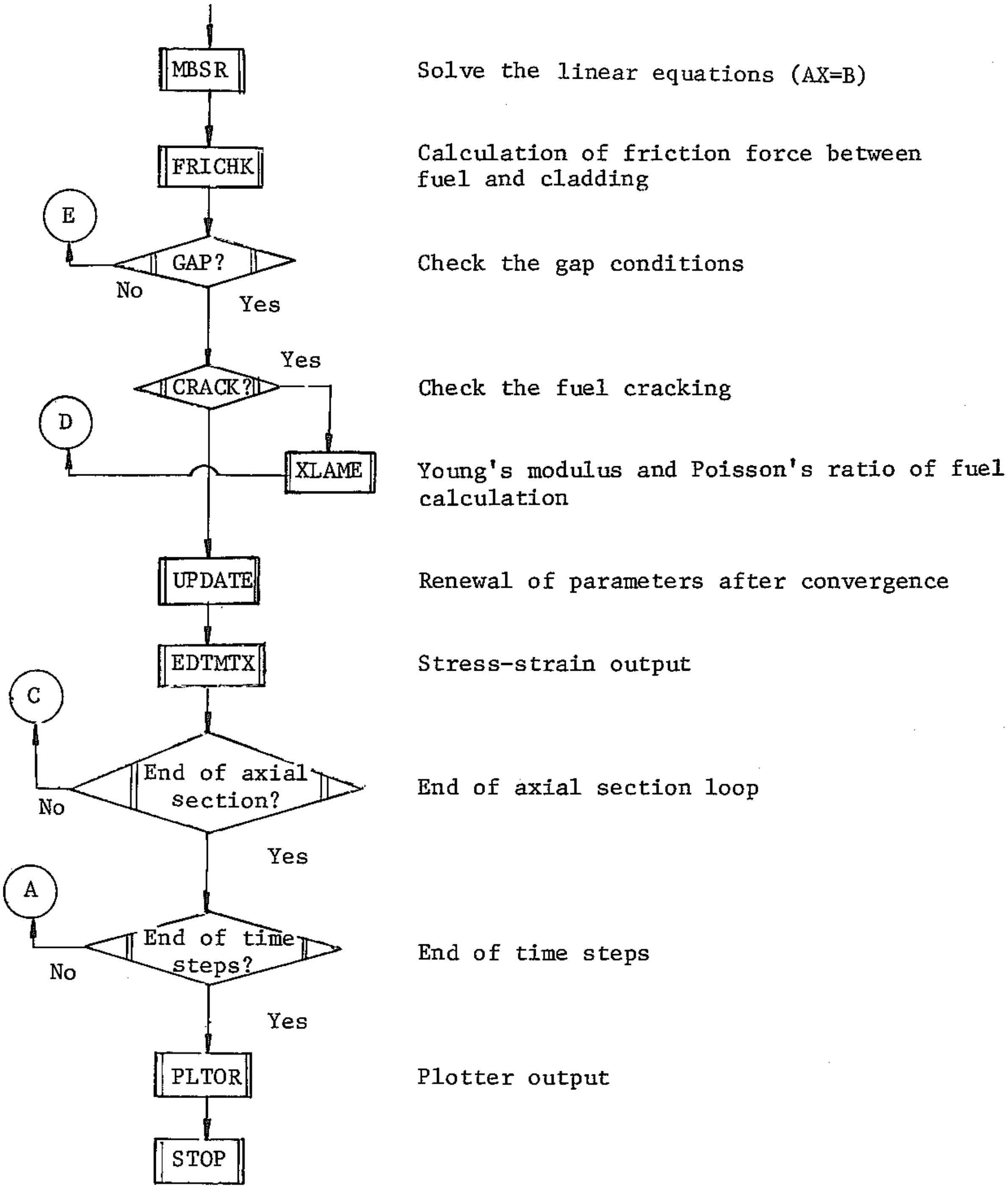
Fuel swelling rate by gaseous F.P. and hot pressing calculation

Gap condition iteration loop

Calculation of matrix A of the linear equations (AX=B)

Factorization of a non-singular sparse matrix A into a lower (L) and an upper (U) triangular (A=LU)

Calculation of matrix B of the linear equations (AX=B)



4.2 Matrix Solution

For the calculation by matrix method, the unknown factors and equations shown in section 3 are rearranged. Suffixes used hereinafter are as follows.

NCR = Number of radial meshes in the columnar grain growth region
NER = Number of radial meshes in the equiaxed grain growth region
NUR = Number of radial meshes in the undisturbed region
NCL = Number of radial meshes in the cladding region

M = NCR + NER + NUR

= Number of radial meshes in the fuel region

N = NCR + NER + NUR + NCL

= Total number of radial meshes

1) Unknown factors

Radial total strains

$\varepsilon^T_{r_1}, \varepsilon^T_{r_2}, \dots\dots\dots\dots, \varepsilon^T_{r_N}$ [1 ∿ N]

Circumferential total strains

$\varepsilon^T_{\theta_1}, \varepsilon^T_{\theta_2}, \dots\dots\dots\dots, \varepsilon^T_{\theta_N}$ [N+1 ∿ 2N]

Radial stresses

$\sigma_{r_1}, \sigma_{r_2}, \dots\dots\dots\dots, \sigma_{r_N}$ [2N+1 ∿ 3N]

Circumferential stresses

$\sigma_{\theta_1}, \sigma_{\theta_2}, \dots\dots\dots\dots, \sigma_{\theta_N}$ [3N+1 ∿ 4N]

Axial stresses

$\sigma_{z_1}, \sigma_{z_2}, \dots\dots\dots\dots, \sigma_{z_N}$ [4N+1 ∿ 5N]

Radial creep strains

$\varepsilon^p_{r_1}, \varepsilon^p_{r_2}, \dots\dots\dots\dots, \varepsilon^p_{r_N}$ [5N+1 ∿ 6N]

Circumferential creep strains

$\varepsilon_{\theta_1}, \varepsilon_{\theta_2}, \dots\dots\dots\dots, \varepsilon^p_{\theta_N}$ [6N+1 ∿ 7N]

Axial creep strains

$\varepsilon^p_{z_1}, \varepsilon^p_{z_2}, \dots\dots\dots\dots, \varepsilon^p_{z_N}$ [7N+1 ∿ 8N)

Average radial creep strains for fuel and cladding

$\overline{\varepsilon^p_{rf}}, \overline{\varepsilon^p_{rc}}$ [8N+1 ∿ 8N+2]

Average circumferential creep strains for fuel and cladding

$\overline{\varepsilon^p_{\theta f}}, \overline{\varepsilon^p_{\theta f}}$ [8N+3 ∿ 8N+4]

Average axial creep strains for fuel and cladding

$\overline{\varepsilon^p_{zf}}, \overline{\varepsilon^p_{zf}}$ [8N+5 ∿ 8N+6]

Average axial total strains for fuel and cladding

$\overline{\varepsilon^T_{zf}}, \overline{\varepsilon^T_{zc}}$ [8N+7 ∿ 8N+8]

Radial displacement

$u_1,\ u_2,\ \cdots\cdots\cdots\ u_{N+2}$ [8N+9 ∿ 9N+10]

Radial direction contact pressures

$P_1,\ P_2,\ \cdots\cdots\cdots\ P_{N+1}$ [9N+11 ∿ 10N+11]

Friction force between fuel and cladding

F [10N+12]

2) Equations

The equations mentioned in section 3 are rearranged to gather the unknown factors at the left sides.

a) Constructive relations

Equations (19) and (20) are arranged as follows.

$$\varepsilon^T_{rj} - \frac{1}{E_j}\sigma_{rj} + \frac{\nu_j}{E_j}\sigma_{\theta j} + \frac{\nu_j}{E_j}\sigma_{zj} - \varepsilon^p_{rj} = \alpha T_j + \varepsilon^s_{rj}$$

(j=1,N) [1 ∿ N]

$$\varepsilon^T_{\theta j} + \frac{\nu_j}{E_j}\sigma_{rj} - \frac{1}{E_j}\sigma_{\theta j} + \frac{\nu_j}{E_j}\sigma_{zj} - \varepsilon^p_{\theta j} = \alpha T_j + \varepsilon^s_{\theta j}$$

(j=1,N) [N+1 ∿ 2N]

b) Creep law

The equations of (19), (20) and (21) are rewrote using the relation,

$$\Delta\varepsilon^p = \varepsilon^p_k - \varepsilon^p_{ko} \qquad (k=r,\theta,z).$$

$$\frac{\Delta\varepsilon^j_p}{\sigma_{ej}}\sigma_{rj} - \frac{\Delta\varepsilon^j_p}{2\sigma_{ej}}\sigma_{\theta j} - \frac{\Delta\varepsilon^j_p}{2\sigma_{ej}}\sigma_{zj} - \varepsilon^p_{rj} = -\varepsilon^p_{roj}$$

(j=1,N) [2N+1 ∿ 3N]

$$-\frac{\Delta\varepsilon^j_p}{2\sigma_{ej}}\sigma_{rj} + \frac{\Delta\varepsilon^j_p}{\sigma_{ej}}\sigma_{\theta j} - \frac{\Delta\varepsilon^j_p}{2\sigma_{ej}}\sigma_{zj} - \varepsilon^p_{\theta j} = -\varepsilon^p_{\theta oj}$$

(j=1,N) [3N+1 ∿ 4N]

$$-\frac{\Delta\varepsilon_p^j}{2\sigma_{ej}}\sigma_{rj}-\frac{\Delta\varepsilon_p^j}{2\sigma_{ej}}\sigma_{\theta j}+\frac{\Delta\varepsilon_p^j}{\sigma_{ej}}\sigma_{zj}-\varepsilon_{zj}^p = -\varepsilon_{zoj}^p$$

$(j=1,N)$ [4N+1 ∿ 5N]

c) <u>Averaging the creep strains</u>

In order to get the mean values of creep strains in the fuel and cladding regions, equation (42) is used.

Mean radial creep strain in the fuel region is calculated by the equation

$$\overline{\varepsilon_{rf}^p} = \frac{\int_{r_i}^{r_o}\varepsilon_r^p\, r\, dr}{\int_{r_i}^{r_o} r\, dr} .$$

Therefore,

$$(r_{o1}^2 - r_{i1}^2)\,\varepsilon_{r1}^p + (r_{o2}^2 - r_{i2}^2)\,\varepsilon_{r2}^p + \cdots\cdots + (r_{oM}^2 - r_{iM}^2)\,\varepsilon_{rM}^p$$

$$-(r_{oM}^2 - r_{i1}^2)\,\overline{\varepsilon_{rf}^p} = 0 \qquad [5N+1]$$

Using the same equation,

$$(r_{o1}^2 - r_{i1}^2)\,\varepsilon_{\theta 1}^p + \cdots\cdots + (r_{oM}^2 - r_{iM}^2)\,\varepsilon_{\theta M}^p - (r_{oM}^2 - r_{i1}^2)\,\overline{\varepsilon_{\theta f}^p} = 0$$

[5N + 2]

$$(r_{o1}^2 - r_{i1}^2)\,\varepsilon_{z1}^p + \cdots\cdots + (r_{oM}^2 - r_{iM}^2)\,\varepsilon_{zM}^p - (r_{oM}^2 - r_{i1}^2)\,\overline{\varepsilon_{zf}^p} = 0$$

[5N + 3]

In the cladding region,

$$(r_{oM+1}^2 - r_{iM+1}^2)\,\varepsilon_{rM+1}^p + \cdots + (r_{oN}^2 - r_{iN}^2)\,\varepsilon_{rN}^p - (r_{oN}^2 - r_{iM+1}^2)\,\overline{\varepsilon_{rc}^p} = 0$$

[5N + 4]

$$(r_{oM+1}^2 - r_{iM+1}^2)\,\varepsilon_{\theta M+1}^p + \cdots + (r_{oN}^2 - r_{iN}^2)\,\varepsilon_{\theta N}^p - (r_{oN}^2 - r_{iM+1}^2)\,\overline{\varepsilon_{\theta c}^p} = 0$$

[5N + 5]

$$(r^2_{oM+1}-r^2_{iM+1})\ \varepsilon^P_{zM+1}+\ \ldots\ +(r^2_{oN}-r^2_{iN})\ \varepsilon^P_{zN}\ -\ (r^2_{oN}-r^2_{iM+1})\ \overline{\varepsilon^P_{zc}}\ =0$$

[5N + 6]

d) Radial displacement

From the equations of (19), (20), (21), (23) and (24), stresses (σ_r, σ_θ, σ_z) and displacement ($u(r)$) are solved using the boundary conditions described by equations (30) and (31).

Then, the average displacement of each region is calculated using the mean radius of each region ($\bar{r}$).

$$\bar{r} = \frac{\int_{r_i}^{r_o} r\cdot rdr}{\int_{r_i}^{r_o} r\ dr} = \frac{2}{3}\cdot\frac{r_o^2+r_or_i+r_i^2}{r_o+r_i}$$

The displacement of the inner radius of a region is given by the equation;

$$\begin{aligned}
u_i &+ P_i\times\frac{r_i^2}{2G(r_o^2-r_i^2)}\ \{-\frac{r_o^2}{\bar{r}}-(1-2\nu)\bar{r}\} \\
&+ Po\times\frac{r_o^2}{2G(r_o^2-r_i^2)}\ \{\frac{r_i^2}{\bar{r}}+(1-2\nu)\bar{r}\} \\
&+ \overline{\varepsilon^T_r}\times(\bar{r}-r_i) \\
&+ \overline{\varepsilon^T_z}\times\nu\bar{r} \\
&+ \overline{\varepsilon^P_r}\times\{-\frac{1-2\nu}{2}\cdot\bar{r}-Y_1+Y_2\} \\
&+ \overline{\varepsilon^P_\theta}\times\{-\frac{1-2\nu}{2}\cdot\bar{r}+Y_1-Y_2\} \\
= &\ \overline{\varepsilon^T_{r_o}}\times\{-r_i+\frac{\bar{r}}{2}-Y_1+Y_2\} \\
&+ \overline{\varepsilon^T_{\theta o}}\times\{-\frac{\bar{r}}{2}+Y_1-Y_2\} \\
&+ \overline{\alpha T}\times\{(1+\nu)\bar{r}\}
\end{aligned}$$

$$- \overline{\varepsilon_r^s} \times \{-\frac{\bar{r}}{2} - Y_1 + Y_2\}$$

$$- \overline{\varepsilon_\theta^s} \times \{-\frac{\bar{r}}{2} + Y_1 - Y_2\}$$

$$- \overline{\varepsilon_z^s} \times \{-\nu\bar{r}\} \qquad [5N+7 \sim 6N+6]$$

where,

$$Y_1 \equiv \frac{1-2\nu}{1-\nu} \cdot \frac{\bar{r}}{2} \ln \frac{\bar{r}}{r_i}$$

$$Y_2 \equiv \frac{\{r_i^2 + (1-2\nu)\bar{r}^2\}r_o^2}{2(1-\nu)\bar{r}(r_o^2-r_i^2)} \ln \frac{r_o}{r_i}$$

The displacement of the outer radius of a region is given by the equation;

$$u_o + Pi \times \frac{r_i^2}{2G(r_o^2-r_i^2)} \{-\frac{r_o^2}{\bar{r}} - (1-2\nu)\ \bar{r}\}$$

$$+ Po \times \frac{r_o^2}{2G(r_o^2-r_i^2)} \{\frac{r_i^2}{\bar{r}} + (1-2\nu)\ \bar{r}\ \}$$

$$+ \overline{\varepsilon_r^T} \times (\bar{r} - r_o)$$

$$+ \overline{\varepsilon_r^T} \times \nu\bar{r}$$

$$+ \overline{\varepsilon_r^P} \times \{-\frac{1-2\nu}{2}\ \bar{r} - Y_1 + Y_2\}$$

$$+ \overline{\varepsilon_\theta^P} \times \{-\frac{1-2\nu}{2}\ \bar{r} + Y_1 - Y_2\}$$

$$= \overline{\varepsilon_{r_0}^T} \times \{-r_o + \frac{\bar{r}}{2} - Y_1 + Y_2\}$$

$$+ \overline{\varepsilon_{\theta_0}^T} \times \{-\frac{\bar{r}}{2} + Y_1 - Y_2\}$$

$$+ \overline{\alpha T} \times \{(1+\nu)\ \bar{r}\ \}$$

$$- \overline{\varepsilon_r^s} \times \{-\frac{\bar{r}}{2} - Y_1 + Y_2\}$$

$$- \overline{\varepsilon_\theta^s} \times \{-\frac{\bar{r}}{2} + Y_1 - Y_2\}$$

$$- \overline{\varepsilon_z^s} \times \{-\nu \bar{r}\} \qquad [6N+7 \sim 7N+6]$$

e) Boundary conditions

As shown in the section 3, boundary conditions of pressure are given as follows.

At the fuel central hole,

$$P_1 = P_{cv} . \qquad [7N + 7]$$

At the cladding outer surface,

$$P_{N+1} = P_{cool} . \qquad [7N + 8]$$

f) Axial displacement

According to the assumption of plane-strain analysis, axial displacements are independent of radius. Therefore, the axial displacement in the fuel region is given by averaging the displacements of all regions:

$$(1+\nu)\overline{\varepsilon_{zf}^T} + \nu\overline{\varepsilon_{rf}^p} + \nu\overline{\varepsilon_\theta^p}$$

$$- \overline{\varepsilon_{zf}^p} - \frac{\nu}{G} \cdot \frac{r_f^2}{r_f^2 - r_c^2} P_{fc} + \frac{\Delta L_f}{2G\pi(r_f^2 - r_c^2)} \cdot F$$

$$= -\frac{1}{2G} [P_{pl} + \frac{\sigma_f g V_f}{\pi(r_f^2 - r_c^2)} (1 - \frac{Z}{L_f})]$$

$$+ \frac{\nu}{G} (1 - \frac{r_f^2}{r_f^2 - r_c^2}) P_{cv} + \frac{(1+\nu)}{r_f^2 - r_c^2} \sum_{j=1}^{M} (r_{oj}^2 - r_{ij}^2) \overline{\alpha T}_j$$

$$+ (1+\nu) \overline{\varepsilon_{zf}^s} \qquad [7N + 9]$$

In the same manner, the axial displacement in the cladding region is given by the equation,

$$(1+\nu) \overline{\varepsilon_{zc}^T} + \nu\overline{\varepsilon_{rc}^p} + \nu\overline{\varepsilon_{\theta c}^p} - \overline{\varepsilon_{zc}^p}$$

$$+ \frac{\nu}{G} \cdot \frac{r_{ci}^2}{r_{co}^2 - r_{ci}^2} \cdot P_{fc} - \frac{\Delta L}{2\pi G(r_{co}^2 - r_{ci}^2)} \cdot F$$

$$= \frac{1}{2\pi G(r_{co}^2 - r_{ci}^2)} \{- P_T - \pi (r_{co}^2 P_c - r_{ci}^2 P_p)$$

$$- \rho_c g V_c (1 - \frac{z}{L_f})\} + \frac{(1 + \nu)}{(r_{co}^2 - r_{ci}^2)} \sum_j (r_{oj}^2 - r_{ij}^2) \overline{\alpha T_j}$$

$$+ \frac{\nu}{G} \cdot \frac{r_{co}^2}{r_{co}^2 - r_{ci}^2} P_c + (1 + \nu) \overline{\varepsilon_{zc}^{s}} \qquad [7N + 10]$$

g) <u>Radial stresses</u>

Stresses calculated from equations (19), (20), (21), (23) and (24) are the functions of radius. Therefore, the mean stress of each region is given by averaging the stress function as follows.

$$\overline{\sigma_r} = \frac{\int_{r_i}^{r_o} \sigma_r r dr}{\int_{r_i}^{r_o} r dr}$$

$$\overline{\sigma_r} + \frac{r_o^2}{r_o^2 - r_i^2} \cdot (1 - \frac{2r_i \ln \frac{r_o}{r_i}}{r_o^2 - r_i^2}) \cdot P_o$$

$$- \frac{r_o^2}{r_o^2 - r_i^2} \cdot (1 - \frac{2r_o^2 \ln \frac{r_o}{r_i}}{r_o^2 - r_i^2}) \cdot P_i$$

$$+ \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2 r_o^2 (\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_r^P}$$

$$- \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2 r_o^2 (\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_\theta^P}$$

$$= \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2 r_o^2 (\ln \frac{r_i}{r_o})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_{ro}^T}$$

$$- \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2 r_o^2 (\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_{\theta o}^T}$$

$$- \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2 r_o^2 (\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_r^s}$$

$$+ \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2\ r_o^2(\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_\theta^s} \qquad [7N+11 \sim 8N+10]$$

h) <u>Circumferential stresses</u>

Circumferential stresses are averaged by the same method.

$$\overline{\sigma}_\theta + \frac{r_o^2}{r_o^2 - r_i^2} \cdot (1 + \frac{2r_i^2\ \ln \frac{r_o}{r_i}}{r_o^2 - r_i^2}) \cdot P_o$$

$$- \frac{r_i^2}{r_o^2 - r_i^2} \cdot (1 + \frac{2r_o^2\ \ln \frac{r_o}{r_i}}{r_o^2 - r_i^2}) \cdot P_i$$

$$- \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2\ r_o^2(\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_r^P}$$

$$+ \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2\ r_o^2(\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_\theta^P}$$

$$= - \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2\ r_o^2(\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_{ro}^T}$$

$$+ \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2\ r_o^2(\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_{\theta o}^T}$$

$$+ \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2\ r_o^2(\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_r^s}$$

$$- \frac{G}{2(1-\nu)} \{1 - \frac{4r_i^2\ r_o^2(\ln \frac{r_o}{r_i})^2}{(r_o^2 - r_i^2)^2}\} \cdot \overline{\varepsilon_\theta^s} \qquad [8N+11 \sim 9N+10]$$

i) <u>Axial stresses</u>

By averaging the stress distribution u(r) in each region;

$$\overline{\sigma}_z + 2\nu \cdot \frac{r_o^2}{r_o^2 - r_i^2} \cdot P_o - 2\nu \cdot \frac{r_i^2}{r_o^2 - r_i^2} \cdot P_i$$

$$- 2G\nu \cdot \overline{\varepsilon_r^P} - 2G\nu \cdot \overline{\varepsilon_\theta^P} + 2G \cdot \overline{\varepsilon_z^P} - 2G(1+\nu) \cdot \overline{\varepsilon_z^T}$$

$$= -2G(1+\nu) \cdot \overline{\alpha T} - 2G\ (1+\nu) \cdot \overline{\varepsilon_z^T} \qquad [9N+11 \sim 10N+10]$$

j) Boundary conditions between fuel and cladding

Boundary conditions between fuel and cladding are changed following the relationship between them.

(i) When gap exists

Contact pressure

$$P_{fc} = P_{pl} \qquad [10N + 11]\ (1)$$

Friction force

$$F = 0 \qquad [10N + 12]\ (1)$$

(ii) When gap does not exist (stick condition)

Strain

$$\bar{\varepsilon}_{zf} = \bar{\varepsilon}_{zc} \qquad [10N + 11]\ (2)$$

Displacement

$$u_{o,M} - u_{i,M+1} = r_{co} - r_f \qquad [10N + 12]\ (2)$$

(iii) When gap does not exist (slip condition)

At first, code calculates by the assumption of gap existance, and if the result shows no gap, code assumes the stick condition. Then if the friction force F, which is calculated by stick assumption, is greater than the value estimated by the contact pressure between fuel and cladding;

$$F > 2\pi\mu_s f_f P_{fc}$$

where μ_s is static friction constant, slip assumption is adopted.

Friction force

$$F = 2\pi\mu_d r_f P_{fc} \qquad [10N + 11]\ (3)$$

Displacement

$$u_{o,M} - u_{i,M+1} = r_{ci} - r_f \qquad [10N + 12]\ (3)$$

From these [10N + 12] equations, the [10N + 12] unknown factors are calculated using matrix method.

5. Material Properties

Material properties used in ACTIVE-II are defined in the subroutine PROPTY, except fuel thermal conductivity and clad-coolant heat conductance.

As the code ACTIVE-II can deal with not only sodium cooled fast breeder reactor fuels, but also the fuels of heavy water reactors, of pressurized water reactors and of boiling water reactors, one must select the appropriate properties for a certain reactor type. Hereinafter, the material properties are described for each reactor type, separately.

The units used in the code are standardized in CGS unit, but some of them have units in BTU or engineering units because of the data.

5.1 Material Properties for Sodium Cooled Fast Breeder Reactor Fuels

1) Coolant-cladding heat conductance[3)]

$$h_c = \frac{K_{Na}}{D_e}\left[5.0 + 0.025\left(\frac{De \cdot W \cdot Cp}{A_f \cdot K_{Na}}\right)^{0.8}\right] \tag{51}$$

where, h_c = heat conductance (Btu/hr/ft^2/°F),

K_{Na} = thermal conductivity of sodium,

$$= 54.306 - 1.878 \times 10^{-2}T + 2.0914 \times 10^{-6}T^2, \tag{52}$$

T = temperature of coolant (°F),

D_e = equivalent flow diameter (ft),

$$= \frac{4Af}{P},$$

Af = coolant flow area (ft^2),

P = wet perimeter (ft),

W = sodium mass flow rate (lb$_m$/hr),

$$= \frac{L_f \times q_{av}}{\int_{T_{in}}^{T_{out}} C_p dT},$$

L_f = fuel column length,

q_{av} = average linear heat rating (Btu/in/hr),

and, C_p = heat capacity of sodium (Btu/lb$_m$/ft),

$$= 0.3457 - 0.792 \times 10^{-4}T + 3.41178 \times 10^{-8}T^2. \tag{53}$$

2) Fuel properties

Usually, fuels for fast breeder reactors are plutonium-uranium mixed oxide, therefore properties of ($20w/oPuO_2$-$80w/oUO_2$) fuel are chosen for the representatives.

a) Thermal conductivity

Option (NOK) = 1

$$K_f = (-0.018 + \frac{54}{T} + 1.75 \times 10^{-12}T) \times \frac{1-P}{1+\beta P} \text{ (W/cm·°C)}$$

where, T = temperature (°K),

P = porosity,

and $\beta = 1$.

Sales' equation[4]

Option (NOK) = 2

$$K_f = 0.011 + \frac{1}{(0.485-0.4465 \cdot D)T} \quad \text{(W/cm·°C)}$$

where, T = temperature (°C),

and D = ratio of density to the theoretical density,

= 0.99 for columnar grain region,

= 0.965 for equiaxed grain region.

Baily-Asamoto's equation[5]

Option (NOK) = 3

$$K_f = \{\frac{1}{(3.11+355.0\times x) + 0.0272 \times T} + 5.39 \times 10^{-13} \times T^3\}$$

$$\times \frac{(\frac{1-P}{1+\beta P})}{(\frac{1-0.05}{1+\beta\times 0.05})} \quad \text{(W/cm·°C)}$$

where, T = temperature (°K),

P = porosity,

X = 2-O/M (stoichiometry)

and $\beta = 1$.

Biancheria's equation[6]

b) Thermal expansion rate[7]

Thermal expansion rate is assumed to be the same for all three regions (i.e. columnar grain region, equiaxed grain region and undisturbed region).

$$\alpha_f = 8.192\times10^{-6}+2.430\times10^{-9}\times(T-20) \quad (1/°C)$$

where, T = temperature (°C).

c) Young's modulus of elasticity[7)]

$$E_f = 3.59\times10^{7}\times(1-2.35\times P)\ (1-\frac{T}{22{,}200}) \quad (psi)$$

where, T = temperature (°F),

and P = porosity.

d) Poisson's ratio[7)]

$$\nu_f = \frac{0.317\times(1-0.46\times P)\times(22{,}200-T)+0.5\times(T-77)}{22{,}200-77}$$

where, T = temperature (°F),

and P = porosity.

e) Shear modulus

$$G_f = \frac{E_f}{2(1+\nu_f)}$$

f) Creep rate[7)]

$$\dot{\varepsilon} = A_o\times e^{-\frac{Q_o}{RT}}\times(1+\frac{\psi}{\psi_o})\times\sigma_e^{m}+A_1\times e^{-\frac{Q_1}{RT}}/d^2+C\times\phi\}\times\sigma^{n} \quad (1/hr)$$

where, $A_o = 1.83\times10^{-3}$,

$A_1 = 2.82\times10^{7}$,

Q_o = 140,000 (cal/mole),

Q_1 = 110,000 (cal/mole),

ψ = fission density (fissions/sec/cc),

$\psi_o = 3.0\times10^{12}$ (fissions/sec/cc),

σ_e = equivalent stress (psi),

$m = 4.5$,

$n = 1$,

d = 10.0 (μ) grain size,

$C = 6.0\times10^{-23}$,

ϕ = fast neutron flux ($n/cm^2/sec$) (E>0.1 MeV),

amd R = 1.1034 (cal/mole/°R).

g) Swelling rate

Fuel swelling is assumed to be caused by fission gases and solid fission products retained in each region.

Solid fission products[7)]

Swelling rate caused by solid fission products are cal-calculated based on the volume of fision products per fission.

1.147×10^{-23} (cm^3/fission)

Gaseous fission products[7)]

In the undisturbed region, gaseous fission products are assumed to be incompressible, and to have constant volume.

39.15 (cm^3/mole)

In the columnar grain region and equiaxed grain region, gaseous fission products are assumed to be controlled by ideal gas law as follows.

$$\left(\frac{\Delta V}{V}\right)_i^{gas} = \frac{n_i R \overline{T}_i}{V_i(P_i + P_i')}$$

where,

ΔV_i = volume change in the region i,

V_i = initial volume in the region i,

n_i = moles of fission gas retained in the region i,

$\overline{T}_i$ = mean temperature in the region i,

$P_i = \frac{1}{3}(\sigma_r^i + \sigma_\theta^i + \sigma_z^i)$ average stress in the region i,

P_i' = correction factor related to the surface tension of fission gas bubbles,

and R = gas constants (cal/g/mole/°K).

h) Hot pressing

$$\Delta e_i^{hp} = -\left[\frac{40\Omega b i D o e^{-Q/RT}}{KTd^2}\right] \cdot P_i \cdot \left(1 - \frac{\rho_i}{\rho_{th}}\right)\Delta t$$

where , $\Delta e_i^{hp} = (\frac{\Delta V}{V})_i^{hp}$

= volumetric strain due to hot pressing in the region i,

$\rho_{th} = \rho_o/(1-3\alpha T\ )$

= theoretical density of fuel (g/cm^3),

T_i = temperature (°K),

α = thermal expansion rate (1/°C),

Δt = time interval (sec),

$\Omega = 4.0 \times 10^{-23}$ (cm^3)

= vacancy volume,

$b_i = 4$

= stress concentration factor in the region i,

$D_o = 6.8 \times 10^{-5}$ (cm^2/sec)

= diffusion coefficient,

$K = 1.38 \times 10^{-16}$ (erg/°K)

= Boltzmann's constant,

$d = 10 \times 10^{-4}$

= average grain size,

and $Q = 98.3$ (Kcal/g/mole)

= activation energy.

In the program, these factors are combined into two constants, C and QPK[7].

$$C = \frac{40\Omega b i D_o}{K d^3}$$

$$= \frac{40\times(4\times10^{-23})\times4\times(6.8\times10^{-5})}{(1.38\times10^{-16})\times(10^{-3})^2} \times 3600\times(9.8\times10^5)$$

$$= 11.13\times10^6 \quad (cm^2\cdot°K/hr/Kg)$$

$$QPK = \frac{Q}{R} = \frac{98300}{1.987}$$

$$= 49{,}600 \quad (1/°K)$$

i) <u>Theoretical density</u>

Theoretical density of pellet is given as a function of plutonium enrichment.

$$\rho_{th} = 10.96 + 0.5 \times C_{pu} \qquad (g/cm^3)$$

where, C_{pu} = plutonium enrichment.

j) Fission gas release rate[7)]

Fission gas release rate is given as a function of temperature,

$$G_r(i) = \exp(-4.48 - \frac{8000}{T_i}) \qquad (1/hr)$$

where, T_i = temperature (°K).

But if the swelling strain caused by fission gas exceeds 25%, gas release rate is accelerated by the following equation.

$$G_r^m(i) = Gr(i) + [(\frac{\Delta V}{V})_{i\text{-}gas} - 0.25] \times 100$$
$$\times [G_r(\text{columnar}) - G_r(i)]$$

where, G_r(columnar) = gas release rate in the columnar grain region,

and $(\frac{\Delta V}{V})_{i\text{-}gas}$ = swelling strain caused by gas.

By the way, number of atoms per fission (n_a) and number of fissions per Kw-hr (n_f), which are used in the program, are as follows.

$$n_a = 0.27 \quad (n/\text{fission})$$
$$n_f = 1.116 \times 10^{17} \text{ (fissions/kw-hr)}$$

3) Fuel-cladding interface

a) Gap conductance between fuel and cladding[8)]

Gap conductance is given by the function of gap width.

$$h_g = 1873 \times e^{-53.25 \times \Delta r} \qquad (\text{Btu/ft}^2 \cdot \text{hr} \cdot {}^\circ\text{F})$$

where, Δr = gap width (mm).

b) Fruction factor between fuel and cladding [7)]

Friction factor is used to settle the gap conditions.

Static friction factor : $\mu s = 0.8$

Dynamic friction factor : $\mu d = 0.6$

4) Cladding properties

SUS 316 stainless steel (almost equivalent AISI-316 S.S.) is chosen for the representative of the cladding of fast breeder reactors.

a) Thermal conductivity

$$k_c = 0.130 + 1.36 \times 10^{-4} \times T \qquad (W/cm\cdot °C)$$

where, T = temperature (°C).

b) Thermal expansion rate[9]

$$\alpha_c = 17.0 \times 10^{-6} + 4.25 \times 10^{-9} \times T \qquad (1/°C)$$

where, T = temperature (°C).

c) Poisson's ratio[7]

$$\nu_c = 0.3015 + 0.850 \times 10^{-4} \times T$$

where, T = temperature (°C).

d) Young's modulus of elasticity[10]

$$E = 2.0793 \times 10^{6} - 8.8592 \times 10^{2} \times T \qquad (Kg/cm^2)$$

where, T = temperature (°C)

3) Creep rate

Option (NOC) = -1[7]

$$\dot{\varepsilon} = A \times e^{-\frac{Q}{RT}} \times \sigma^{m} + B \times \phi \times \sigma^{n} \qquad (1/hr)$$

where,

$A = 2.7\times10^{-11}$,

$Q = 95{,}000$ (cal/mole),

$R = 1.1034$ (cal/mole),

$T =$ (°R),

$m = 7$,

$B = C_e + \frac{A_e}{B_e} \times e^{\frac{-\phi t}{B_e}}$,

$A_e = 1.0\times10^{-8}$,

$B_e = 2.2\times10^{20}$,
$C_e = 4.3\times10^{-30}$.
ϕ = neutron flux ($n/cm^2/sec$),
t = time (sec),
and $n = 1$.

(Boltax, Biancheria & Guha's equation)

Option (NOC) = -2[7]

$$\dot{\varepsilon} = A \times e^{-\frac{Q}{RT}} \times \sigma^m + B \times \phi \times \sigma^n \qquad (1/hr)$$

where,
$A = 2.7\times10^{-11}$,
Q = 95,000 (cal/mole) ,
R = 1.1034 (cal/mole) ,
T = temperature (°R) ,
$m = 7$,

$$B = C_e + \frac{A_e}{B_e} \times e^{-\frac{\phi t}{B_e}} ,$$

$A_e = 1.0\times10^{-8}/(T/910)$,
$B_e = 2.2\times10^{20}\times(T/910)$,
$C_e = 4.3\times10^{-30}/(T/910)$,
ϕ = neutron flux ($n/cm^2/sec$) ,
t = time (sec),
and $n = 1$.

f) Swelling rate

Option (NOS) = -1[11]

$$\frac{\Delta V}{V} = [(T-40)\times10^{-10}](\phi t)^{2.05-\frac{27}{\theta}+\frac{78}{\theta^2}} \exp(32.60-\frac{5100}{T}-0.015T) \qquad (\%)$$

where,
ϕt = neutron fluence ($n/cm^2\times10^{-22}$) $E > 0.1$ MeV
T = (°K)
and $\theta = T-623$
for annealed 304 & 316 S.S
$T = 370 \sim 500$°C
$\phi t \leq 7\times10^{22}$ n/cm^2

Option (NOH) ≤ -2

$$\frac{\Delta V}{V} = 9\times10^{-35}\times(\phi t)^{1.50}(4.028-3.712\times10^{-2}\times T$$
$$+1.0145\times10^{-4}\times T^2-7.879\times10^{-8}\times T^3)$$

where,

ϕt = neutron fluence (n/cm^2)   E > 0.1 MeV

and T = temperature (°K)

for 20% cold worked 316 S.S.

$\phi t \leq 5 \times 10^{22}\ n/cm^2$

5.2 Material Properties for Heavy Water Reactor Fuels

Heavy Water Reactor means a heavy water moderated and boiling light water cooled reactor which is now being developed by PNC.

1) Coolant-cladding heat conductance

Heat conductance equation proposed by Jens and Lottes[12] is used.

$$\Delta T_{film} = 7.93 \times q^{1/4} \times \exp(-\frac{P}{63.2}) \qquad (°C)$$

where, q = surface heat flux (w/cm^2),

and P = coolant pressure.

Film coefficient is calculated using the temperature difference given by the above equation.

$$h_c = \frac{q}{2\pi r} \qquad (w/cm^2/°C)$$

where, r = clad outer radius (cm).

2) Fuel properties

Uranium dioxide or plutonium slightly enriched mixed oxide fuels are used for heavy water reactor. Therefore, the properties are the same as those used for uranium dioxide pellets.

a) Thermal conductivity

Option (NOK) = 3

Thermal conductivity of 95% T.D. fuel is,

$$k_f(0.05) = \frac{38.24}{T+129.4} + 4.79 \times 10^{-13}T^3 \qquad (W/cm\cdot°C)$$

where, T = °K

(Old Lyon's equation[13])

Modification for density is,

$$k_f(P) = \frac{1-P}{1+P(\alpha-1)} \times k_f(0)$$

where, $k_f(P)$ = thermal conductivity of porosity P fuel,

α = 1.5,

$k_f(0)$ = thermal conductivity of 100% T.D. fuel,

and P = porosity.

(Biancheria's equation[14])

Option (NOK) = 4

Thermal conductivity of 95% T.D. fuel is,

$$k_f(0.05) = \frac{38.24}{T+129.4} + 6.1256 \times 10^{-13}T^3 \qquad (W/cm\cdot°C)$$

where, T = temperature (°K).

(New Lyon's equation[15])

Modification for density is the same as option 3.

b) Thermal expansion rate

Thermal expansion rate is assumed to be the same for all three regions (i.e. columnar grain region, equiaxed grain region and undisturbed region).

$$\alpha_f = 9.61 \times 10^{-6} + 1.57 \times 10^{-9} \times T \qquad (1/°C)$$

where, T = temperature (°K).

c) Other properties of fuel

Other properties of fuel for heavy water reactor are the same as those of fast breeder reactor fuel.

3) Fuel-cladding interface

Properties related to fuel-cladding interface, such as gap conductance and friction factor, are assumed to be the same as those of fast breeder reactor fuel.

4) Cladding properties

Zircaloy-2 is chosen for the representative of the cladding of advanced thermal reactors.

a) Thermal conductivity[16)]

$$k_c = 0.121 + 1.24 \times 10^{-4} \times T \qquad (W/cm\cdot{}^\circ C)$$

where, $T = {}^\circ C$.

b) Thermal expansion rate[17)]

$$\alpha_c = 5.675 \times 10^{-6} + 1.7 \times 10^{-9} \times T \qquad (1/{}^\circ C)$$

where, $T = {}^\circ C$.

c) Poisson's ratio

$$\nu_c = 0.325 + 2.14 \times 10^{-4} \times T$$

where, $T = {}^\circ C$.

d) Young's modulus of elasticity

$$E_c = 9.898 \times 10^5 - 9.265 \times 10^2 \times T \qquad (Kg/cm^2)$$

where, $T = {}^\circ C$.

e) Creep rate

Option (NOC) = -1

$$\dot{\varepsilon} = 4 \times 10^{-27} \sigma_e \phi (T-160) \qquad (1/hr)$$

where,

$T = {}^\circ C$,

ϕ = fast neutron flux ($n/cm^2/sec$),

$E > 1$ MeV

and, σ_e = equivalent stress (psi), Option (NOC) = -2

$$\dot{\varepsilon} = 1.02\times10^{-13}\exp(\frac{14000}{RT})\phi^{0.85}\sinh(1.63\times10^{-1}\sigma_e) \quad (1/hr)$$

where,

$T = {}^\circ K$,

σ_e = equivalent stress (Kg/mm^2) ,

ϕ = fast neutron flux ($n/cm^2/sec$) ,

$E > 1$ MeV

$R = 1.985$ (cal/mole·°K) ,

and $Q = 14,000$ (cal/mole)

= activation energy .

f) Swelling rate

Swelling rate is assumed to be zero in the code.

5.3 Material Properties for Pressurized Water Reactor Fuels

Material properties for pressurized water reactors are almost the same as those for heavy water reactor, expect coolant-cladding heat conductance.

1) Coolant-cladding heat conductance

Coolant-cladding heat conductance is calculated by Dittus-Boettler's equation[18] as follows.

$$N_u = 0.023 \times R_e^{0.8} \times P_r^{0.4}$$

where,

N_u = Nusselt number,

R_e = Reynols number,

and P_r = Prandtle number,

5.4 Material Properties for Boiling Water Reactor Fuels

Material properties for boiling water reactors are assumed to be the same as those for heavy water reactors as mentioned before.

6. Input and Output Format

Using a sample calculation, core storage, input parameter preparation, input format, output format and plotter output are described in this chapter.

6.1 Core Storage

As mentioned in chapter 4, this code uses variable dimension method. Therefore, the total core storage size must be given in the MAIN routine according to the job size.

The required amount of storage space for a problem is determined with the following parameters.

NN = number of axial sections. (input)

NCR = number of meshes into which the columnar grain retion is subdivided. (input)

NER = number of meshes into which the equiaxed grain region is subdivided. (input)

NUR = number of meshes into which the undisturbed region is subdivided. (input)

NFR = NCR + NER + NUR. (total number of meshes into which the fuel is subdivided.)

NCL = number of meshes into which the cladding is subdivided. (input)

NTR = NFR + NCL = NCR + NER + NUR + NCL.

NP = number of regions (concentric cylindrical shells) of equal mass into which the fuel is subdivided for plutonium migration. (input)

NBUF = storage size (in words) required for storing plotting informations. The plotting informations are written to logical unit every NBUF steps. (input)
If NBUF = 0, then the code sets NBUF to 50.

1) Core storage required for problem data, excluding those for matrix. (in words)

$$IA = 13NFR + NUR + NP + 1 + NN\,(41NTR + NFR + NUR + 8NP + 101) + \begin{cases} 0 & \langle\text{without plotting}\rangle \\ NBUF(3NP+19) & \langle\text{with plotting}\rangle \end{cases}$$

The last term NBUF(3NP+19), is added to IA when the plotting is required.

2) Core storage required for data only for matrix. (in words)

$$IMA = 1332NTR + 656$$

3) Core storage required for program (load module).

Load module = 240K bytes

4) Region size in the REGION parameter used in the JOB card on IBM-360 or -370 series machines.

$$REGION = \left[240 + \frac{4(IA+IAM)}{1024}\right] \quad \text{K-bytes}$$

5) Example

NN=5, NCR=1, NER=1, NUR=1, NCL=1, NP=10, NBUF=50.

The storage requirement is obtained from the above equations as follows.

IA = 4241 words

IAM = 5984 words

REGION = 384 K (multiple of 128K)

The typical MAIN program is written as below.

(See the section CONTROL CARDS)

```
//xxxx JOB (xxxx, ....) ....., REGION=384K
     :
     :
//    EXEC FORTHCLG
//FORT.SYSIN DD   *
        COMMON/DIM/IA,IAM
        DIMENSION A(4250), AM(5990)
        IA=4250
        IAM=5990
        CALL ACTIVE (A, AM)
        STOP
        END
/*
        :
        :
```

6.2 Input Parameter Preparation

Since ACTIVE-II code calculates with a simplified model of fuel pin, it is very convenient to make a summary of irradiation data. As mentioned below, the irradiation data summary includes the information of fuel pin fabrication and irradiation conditions.

Table 6.1 shows the abstract of the irradiation, and the purpose of irradiation, irradiation abstracts and references are summarized.

Table 6.2 shows the fuel pin summary of before irradiation. Plenum volume of item 1-7 means the volume available for gas retaining, but excluding pellet-cladding gap volume of core region. The pellet-cladding gap volume of core region is automatically calculated and added to the plenum volume in ACTIVE-II. Adsorbed gas of item 1-10 includes moisture in the pellets, and coolant flow area per fuel pin of item 1-11 means the equivalent flow area around the fuel pin.

Table 6.3 shows the axial distributions of parameters before irradiation. Fuel pellet region is devided into some meshes along the axis, and the number and interval of meshes are optional. But the plenum region is allowed only at the top of the fuel pin, and its length must be calculated from the plenum volume equivalently (that is, equivalent length (L_{pl}) is calculated using the plenum volume (V_{pl}) of item 1-7 of Table 6.2 and the inner radius (r_{ci}) of cladding as follows).

$$L_{pl} = V_{pl}/(\pi \times r_{ci}^2)$$

In this sample, 4 axial meshes are required of 3 in fuel column region and 1 in plenum region. Rod power and fast flux distribution may be shown as relative values along the axial direction.

Table 6.4 shows the irradiation history. Integrated time means the time since the irradiation started, and the data are required for each time step where the irradiation conditions are changed. The conditions during the intervals are calculated interporatedly in the code.

ACTIVE-II code usually needs the inlet and outlet temperatures of coolant, and calculates the coolant temperature distributions and coolant-cladding heat conductance using the properties of coolant and axial power distribution. But in the case of Dounrey Fast Reactor (DFR), in which the coolant is NaK and flows from top to bottom, or in the case where the temperature distributions of cladding outer surface is calculated by other code, cladding outer surface temperatures are required instead of coolant outlet temperature.

Average linear heat rating and average fast flux mean the mean values calculated by simple averaging method along the axial meshes of fuel column.

Radii for radial profiles of rod power and fast flux mean the radii of gravity center of radial meshes which are divided using equal mass method, and the number of meshes should be the same used for plutonium migration calculation. Assuming that the density of fuel is constant, the radius of each gravity center of mesh is calculated as follows.

$$r_1 = \left(\frac{r_{fs}^2}{2N} + r_{cv}^2\right)^{1/2}$$

$$r_{i+1} = \left(\frac{r_{fs}^2}{N} + r_i^2\right)^{1/2} \quad (i=2,3,\ldots,N)$$

where, r_i = radius of gravity center of mesh i, (i=1,2,..., N)

r_{fs} = fuel outer radius,

r_{cv} = radius of fuel central void or central hole,

and N = total mesh number.

If the pellet is solid and has no central void, the equation becomes as follows.

$$r_i = \left(\frac{2i-1}{2N}\right)^{1/2} \times r_{fs}$$

For the input of ACTIVE-II, relative values are needed for the radii. The sample case shows the case that the pellet is solid and the number of meshes is 10.

Radial powers for each mesh points may be relative values, and the code normalize the power distribution as the mean value becomes unit. In this case, as EBR-II is a fast reactor, radial power distribution is flat.

Table 6.1 Irradiation Data Summary

Date :	July 7, 1975
Compiled:	T. Kajiyama
Approved:	M. Katsuragawa

Irradiation Data Summary

•Title	:	G.E. F1 Group
•Purpose	:	Investigation of mixed oxide fuel pin performance under fast neutron flux
•S/A identification	:	XG01
•Fuel pin number	:	F1A
•Irrad. duration	:	May 12, 1965 ⟶ May 22, 1965
•Reactor name	:	EBR-II
•Maximum burnup	:	1526 MWD/T_{OXIDE}
•Max.neutron fluence (E>0.1 MeV)	:	nvt
•Max.linear heat rate	:	471.6 w/cm
•Nominal values	:	

- Core pellet : Composition 20w/oPuO_2-UO_2, Diameter 5.488 mm
  Density 95.3 %T.D
- Blanket pellet : Composition ______, Diameter ______ mm
  Density ______ %T.D
- Cladding : Material 347 S.S.
  Inner dia. 5.588 mm, Outer dia. 6.350 mm
- Pin length ______ mm, Total stack length ______ mm
  Lower blanket length ______ mm, Core length 361.19 mm,
  Upper blanket length ______ mm, Plenum length ______ mm.

•Irradiation position

S/A position in the core	Pin position in the S/A

•Notes

- Material characteristics :
- Irradiation characteristics :
- Post irradiation: characteristics

•Reference

(1) S.A. Robin, et. al., Short-term Fast Flux (EBR-II) Irradiation of PuO_2-UO_2 Fuel Pins, GEAP-5570 (Oct., 1967)

(2)

(3)

(4)

(5)

Table 6.2 Fuel Pin Summary Before Irradiation

Before Irradiation - (1)

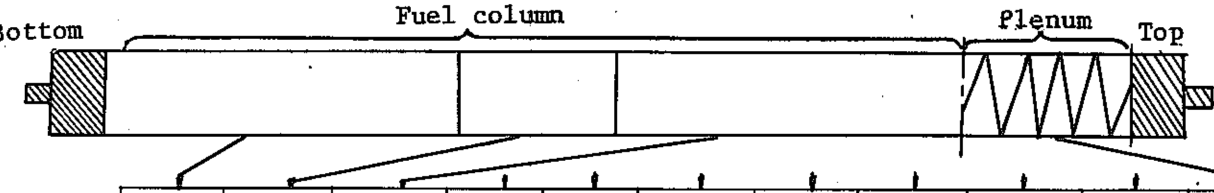
1. <u>Fuel Pin Summary</u>

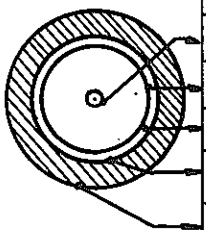
1-1	Total Fuel Pin Length		cm
1-2	Fuel Stack Length	36.119	cm
1-3	Material of Cladding/End Plug	347 S.S. /	Maker/Lot No. / Cold Worked or Solution Treated %
1-4	Total Weight of Fuel Pin		g
1-5	Total Weight of Fuel	22.51	g
1-6	Spring Volume		cm^3
1-7	Plenum Volume	7.371	cm^3 (Excluding pellet-cladding gap volume)
1-8	Spring Material/Spring Constnat		Kg/cm
1-9	Initial Internal Pressure At 20°C	1.033	Kg/cm^2
1-10	Internal Gas		
	- Total	0.407×10^{-3}	mole
	- He	0.307×10^{-3}	mole
	- Adsorbed	0.100×10^{-3}	mole or 100 μℓ/g At °C
1-11	Coolant Flow Area/1 Fuel Pin	0.822	cm^2
1-12	Av. Coolant Pressure	1.68	Kg/cm^2
1-13	Density of Cladding	7.98	g/cm^3

Table 6.3 Fuel Pin Detail

Before Irradiation -(2)

2. Fuel Pin Detail





	A	B	C	D	E	F	G	H	I	J	K
Node Length (cm)	14.45	3.61	18.06								30.06
Fuel Inner Radius (cm)											
Fuel Outer Radius (cm)	0.2744	0.2744	0.2744								
Radial Gap Size (cm)											
Clad Inner Radius (cm)	0.2794	0.2794	0.2794								0.2794
Clad Outer Radius (cm)	0.3175	0.3175	0.3175								0.3175
PuO_2 Enrichment (w/o)	0.2	0.2	0.2								
U-235 Enrichment (w/o)	0.007	0.007	0.007								
$Pu^{239}/Pu^{240}/Pu^{241}/Pu^{242}$											
Fuel Density (g/cm^3)	10.54	10.54	10.54								
Rod Power (w/cm)	0.9792	1.0723	0.9485	(Relative value)							0.0
Fast Flux ($n/cm^2 \cdot sec$)	0.9792	1.0723	0.9485	(Relative value)							0.3
O/M Ratio	1.998	1.998	1.998								
Remarks											

Capsule I.R. (cm) 0.4254 Capsule O.R. (cm) 0.4762

Table 6.4 Irradiation History

Irradiation History - (1)

1. Reactor Operating Data (1) (Reactor EBR-II Position ________)

Date Year Mo.Day	Integrated Time		Coolant			Av.Linear Heat Rating (w/cm)	Av. Fast Flux (>0.1 MeV) ($n/cm^2 \cdot sec$)	Clad Surface Temp. (°C)
	Days	Hours	Inlet Temp. °C	Outlet Temp. °C	Flow Rate g/sec			
	0.15		371.1	372.1		0.0		
	0.671		371.1	479.4		439.8	1.53×10^{15}	
	2.00		371.1	479.4		439.8		
	2.125		371.1	372.1		0.0		
	2.50		371.1	372.1		0.0		
	2.921		371.1	479.4		439.8		
	6.35		371.1	479.4		439.8		
	6.643		371.1	419.2		195.5		
	6.80		371.1	419.2		195.5		
	7.142		371.1	479.4		439.8		
	9.50		371.1	479.4		439.8		
	10.0		371.1	372.1		0.0		

2. Radial Profiles of Rod Power and F. Flux

Radius	0.224	0.387	0.500	0.592	0.671	0.742	0.806	0.866	0.922	0.975
Fast Flux	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Power	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

6.3 Input Format

Table 6.5 shows the input format, and the sample calculation is shown in Appendix. The input explanation is as follows.

Card No.	Format	Symbol	Description
1	(2I5,3E10.5)		
		JFA	Option of reactor type 1 = fast breeder reactor 2 = heavy water reactor 3 = pressurized water reactor 4 = boiling water reactor
		JOUT	Option for output style 0 = output only at the time step given in input 1 = output at the every time step computed in the code.
		EPSSPD	Limitation of restructuring speed, and each region boundary movement should be less than EPSSPD (cm) in each time step. Default = 0.01 cm
		EPSCRF	Limitation of fuel creep speed, and in each time step, fuel creep rate calculated by the equation (17) should be less than EPSCRF. Default = 0.05
		EPSCRC	Limitation of cladding creep speed, and in each time step, cladding creep rate calculated by equation (17) should be less than EPSCRC. Default = 0.01
2	(20A4)	TITLE	Title of the case, and can be continued as many cards as one wishes until the end card (*END). The title shown in the first card is written as the title of plotter output.
3	(16I5)		Problem size and options.
		NN	Number of aixial meshes including plenum

Card No.	Format	Symbol	Description
			region.
		NCR	Number of radial meshes for stress-strain calculation in the columnar grain growth region.
		NER	Number of radial meshes for stress-strain calculation in the equiaxed grain growth region.
		NUR	Number of radial meshes for stress-strain calculation in the undisturbed region.
		NCL	Number of radial meshes for stress-strain calculation in the cladding.
		NP	Number of radial meshes for plutonium migration, and for radial power distribution.
		NOK	Option of fuel thermal conductivity. (See chapter 5; Material properties)
		NOS	Option of cladding swelling equation. (See chapter 5; Materials properties)
		NOC	Option of cladding creep equation. (See chapter 5; Materials properties)
		NPLOT	Option of plotter. 0 = Plotter is not required. 1 = plotter output is required.
		NCRK	Option of fuel cracking. 0 = fuel cracking is not considered. 1 = fuel cracking routine becomes active.
		NSKIP	Option of temperature boundary. 0 = coolant outlet temperatures are input. 1 = cladding outer surface temperatures are input.
		NBUF	Number of time steps for plotter output. If JOUT of card No.1 is 0, NBUF equals the number of input time steps. Default = 50.

Using these numbers of meshes, the following symbols are difined.

NTR = NCR + NER + NUR + NCL

Total number of radial meshes

NFR = NCR + NER + NUR

Total number of radial meshes in the fuel region

Card No.	Format	Symbol	Description
4	(16I5)		Input and plotter options.
		IDLE	Options for the inputs of stress and strain conditions. 0 = input the initial conditions of stress and strain. 1 = the initial conditions of stress and strain are set to zero in the code, and the input cards from No.5 to No.18 are skipped.
		IPFUL	Number of axial position for plotter output concerned with temperature.
		IPSTN	Number of axial position for plotter output concerned with strain.
		IPSTS	Number of axial position for plotter output concerned with stress.
		IPCRK	Number of axial position for plotter output concerned with fuel cracking.
5	(5E15.8)	((EPR(I,J), I=1, NN), J=1, NTR)	Radial component of total strain at the initial condition. (At the axial position I and the radial position J.)
6	(5E15.8)	((EPC(I,J), I=1, NN), J=1, NTR)	Circumferential component of total strain at the initial condition.
7	(5E15.8)	((EPZ(I,J), I=1, NN), J=1, NTR)	Axial component of total strain at the initial condition.

Card No.	Format	Symbol	Description
8	(5E15.8)	((TPR(I,J), I=1, NN), J=1, NTR)	Radial component of accumulated creep strain at the initial condition.
9	(5E15.8)	((TPC(I,J), I=1, NN), J=1, NTR)	Circumferential component of accumulated creep strain at the initial condition.
10	(5E15.8)	((TPZ(I,J), I=1, NN), J=1, NTR)	Axial component of accumulated creep strain at the initial condition.
11	(5E15.8)	((TSR(I,J), I=1, NN), J=1, NTR)	Radial component of accumulated swelling strain at the initial condition.
12	(5E15.8)	((TSC(I,J), I=1, NN), J=1, NTR)	Circumferential component of accumurated swelling strain at the initial condition.
13	(5E15.8)	((TSZ(I,J), I=1, NN), J=1, NTR)	Axial component of accumulated swelling strain at the initial condition.
14	(5E15.8)	((SIGR(I,J), I=1, NN), J=1, NTR)	Radial component of averaged stress in the region (I,J) at the initial condition. (kg/cm^2)
15	(5E15.8)	((SIGC(I,J), I=1, NN), J=1, NTR)	Circumferential component of averaged stress in the region (I,J) at the initial condition. (kg/cm^2)
16	(5E15.8)	((SIGZ(I,J), I=1, NN), J=1, NTR)	Axial component of averaged stress in the region (I,J) at the initial condition.(kg/cm^2)

Card No.	Format	Symbol	Description
17	(5E15.8)	((ADISP(I,J), I=1, NN), J=1, NTR)	Axial displacement at the initial condition. (cm)
18	(5E15.8)	((RDISP(I,J), I=1, NN), J=1, NTR+2)	Radial displacement of the boundary at the initial condition. (cm)
19	(5E15.8)	(RVB(I), I=1, NN)	Fuel central hole radius at the initial condition. (cm)
20	(5E15.8)	(RCB(I), I=1, NN)	Radius of columnar grain growth region outer boundary at the initial condition. (cm)
21	(5E15.8)	(REB(I), I=1, NN)	Radius of equiaxed grain growth region outer boundary at the initial condition. (cm)
22	(5E15.8)	(RUB(I), I=1, NN)	Fuel outer radius at the initial condition. (cm)
23	(5E15.8)	(RCI(I), I=1, NN)	Cladding inner radius at the initial condition. (cm)
24	(5E15.8)	(RCO(I), I=1, NN)	Cladding outer radius at the initial condition. (cm)
25	(5E15.8)	(CLZO(I), I=1, NN)	Axial node length. (cm)

Card No.	Format	Symbol	Description
26	(5E15.8)	(QSFF(I), I=1, NN)	
			Axial distribution of fast neutron flux. (relative value)
27	(5E15.8)	(QSFP(I), I=1, NN)	
			Axial distribution of linear heat rating. (relative value)
28	(5E15.8)	(OTOM(I), I=1, NN)	
			Stoichiometry of each axial section.
29	(5E15.8)	(CPU(I, 1), I=1, NN-1)	
			Plutonium enrichment of each axial section.
30	(5E15.8)	(RHOP(I, 3), I=1, NN-1)	
			Pellet density of each axial section. (g/cm^3)
31	(5E15.8)	(F(1, L), L=1, NP)	
			Radial power distribution. (relative value)
32	(5E15.8)		Capsule dimension and coolant flow area
		RCPI	Capsule inner radius (cm)
		RCPO	Capusle outer radius (cm)
			(If there is no capsule, PCPI>RCPO.)
		AFNA	Coolant flow area per fuel pin. (cm^2)
33	(5E15.8)	(PG(I), I=1, 3)	
			Correction factor related to the surface tension of fission gas bubbles for the fuel swelling equation. (psi) (See chapter 5; Material properties)
34	(5E15.8)		Outer force
		PTOP	Axial force to the cladding (kg)
		PO	Average coolant pressure at the initial condition. (kg/cm^2)

Card No.	Format	Symbol	Description
35	(5E15.8)		Cladding and fuel density parameter
		RHOC	Density of cladding (g/cm^3)
		BURH1	Correction factor related to the burnup effect on the columnar grain region density of equation (10). (a/o)
		BURH2	Correction factor related to the burnup effect on the equiaxed grain region density of equation (10). (a/o)
36	(5E15.8)		Fuel restructuring parameter
		CM1(1)	Constant in the velocity equation of columnar grain region outer boundary movement of equation (6); C_1 (cm/sec)
		CM1(2)	Constant in the velocity equation of equiaxed grain region outer boundary movement of equation (7); C_2 (cm/sec)
		DHR(1)	Constant related to the activation energy of columnar grain region outer boundary movement of equation (6); (Q/R) (°K)
		DHR(2)	Constant related to the activation energy of equiaxed grain region outer boundary movement of equation (7); (Q/R) (°K)
37	(5E15.8)		
		GASI	Initial gas moles in the plenum including adsorbed gas. (moles)
		TCON	The ratio of coolant temperature rize along the fuel pin to the average linear heat rating. (°C·cm/w) (If the coolant outlet temperature is given in card No.38 or the cladding outer surface temperatures are given in card No.39, TCON= 0.0)
		PDBU	Depletion factor of linear heat rating as the burnup. (1 / a/o)

Card No.	Format	Symbol	Description
38	(E10.3, F5.1, 3E15.5, E10.5, I5)		
			Irradiation history data card. (This card and card No. 39, if necessary, should be repeated.)
		TIMEB	Integrated time of irradiation (days).
		POI	Average coolant pressure if changed. (kg/cm^2) Default; POI is set to the same value as the one of the previous time step.
		POWAV	Average linear heat rating of fuel pin. (w/cm)
		CFLX	The ratio of the average fast neutron flux to the average linear heat rating. ($(n/cm^2/sec)/(w/cm)$)
		TIN	Coolant inlet temperature. (°C)
		TOUT	Coolant outlet temperature. (°C)
		NPLOTC	Option of plotter output for fuel crack pattern. 0 or blank = no plotter output 1 = plotter output at that time step.
39	(5E15.8)	(TOUTER(I), I=1, NN)	
			Cladding outer surface temperature distirbu-tion at the time step. (°C) (If ISKIP of card No.3 equals 0, this card should be omitted.)
40	E10.3	TIMEB	Option for the next case, and this card is used at the end of irradiation history. 0.0 = next case follows. /* = end of job.

PNC – PU

Table 6.5 Input Format

INPUT DATA FORM

PAGE ____ OF ____

JOB NUMBER ____ MAIN PROGRAM LABEL ____
PROBLEM ____ MODIFICATION ____
CODED BY ____ DATE ____

KEYPUNCH RECORD DATE ____ NO. OF CARDS ____
KEYPUNCHED BY ____
VERIFIED BY ____

Card	Fields
01	JFA, JOUT, EPSSPD, EPSCRF, EPSCRC
02	TITLE
03	-- -- --
04	*END
05	NN, NCR, NER, NUR, NCL, NP, NOK, NOS, NOC, NPLOT, NCRK, NSKIP, NBUF
06	IDLE, IPFUL, IPSTN, IPSTS, IPCRK
07	EPR(I,J)
08	EPC(I,J)
09	EPZ(I,J)
10	TPR(I,J)
11	TPC(I,J)
12	TPZ(I,J)
13	TSR(I,J)
14	TSC(I,J)
15	TSZ(I,J)
16	SIGR(I,J)
17	SIGC(I,J)
18	SIGZ(I,J)
19	ADISP(I,J)
20	RDISP(I,J)

MEMO NO. ____ PAGE ____ OF ____

Table 6.5 Input Format (cOntinued)

INPUT DATA FORM

PAGE OF

JOB NUMBER MAIN PROGRAM LABEL

PROBLEM MODIFICATION

CODED BY DATE

KEYPUNCH RECORD DATE NO. OF CARDS

KEYPUNCHED BY

VERIFIED BY

Card	Columns 1–10	11–15	16–30	31–45	46–60	61–70	71–80
01	RVB(I)						
02	RCB(I)						
03	REB(I)						
04	RUB(I)						
05	RCI(I)						
06	RCO(I)						
07	CLZO(I)						
08	QSFF(I)						
09	QSFP(I)						
10	OTOM(I)						
11	CPU(I,1)						
12	RHOP(I,3)						
13	F(1,L)						
14	RCPI		RCPO	AFNA			
15	PG(1)		PG(2)	PG(3)			
16	PTOP		PO				
17	RHOC		BURH1	BURH2			
18	CM1(1)		CM1(2)	DHR(1)	DHR(2)		
19	GASI		TCON	PDBU			
20	TIMEB	POI	POWAV	CFLX	TIN	TOUT	NPLOTC

MEMO NO. PAGE OF

Table 6.5 Input Format (continued)

INPUT DATA FORM

PAGE OF

JOB NUMBER MAIN PROGRAM LABEL

PROBLEM MODIFICATION

CODED BY DATE

KEYPUNCH RECORD DATE NO. OF CARDS

KEYPUNCHED BY

VERIFIED BY

	Columns 1–10	Columns 11–20
01	TOUTER(I)	
02	/* or 0.0	

MEMO NO.

PAGE OF

6.4 Output Format

The output consists of the list of input parameters and the calculated results at each time steps. The time steps to be output can be controlled by an option.

a) Input list

Page	Title	Contents
1	———	Expression of the type of reactor.
2	———	Title of the case
3	CONTROL NUMBER AND OPTION	Options adopted in the code by the input card No.3 and 4.
	CENTRAL VOID RADIUS	Central void radii of axial section 1 to NN. (cm)
	COLUMNAR GRAIN REGION OUTER BOUNDARY	Columnar grain growth region outer radius. (cm)
	EQUIAXED GRAIN REGION OUTER BOUNDARY	Equiaxed grain growth region outer radius. (cm)
	FUEL OUTER RADIUS	Fuel outer radius. (cm)
	CLAD INNER RADIUS	Cladding inner radius. (cm)
	CLAD OUTER RADIUS	Cladding outer radius. (cm)
	CPASULE INNER RADIUS	Capsule inner radius if there is. (cm)
	CAPSULE OUTER RADIUS	Capsule outer radius if there is. (cm)
	COOLANT FLOW AREA	Coolant flow area. (cm^2)
	FUEL AXIAL LENGTH	Fuel axial node length and the total fuel column length. (cm)
	CLAD AXIAL LENGTH	Cladding axial node length and the total cladding length. (cm)

Page	Title	Contents
	DENSITY OF FUEL (3 REGIONS) AND CLAD	Fuel density of columnar, equiaxed, and undisturbed regions and cladding density are shown for each axial node. (g/cm^3)
	O/M OF FUEL	Stoichiometry of each axial node.
	PLUTONIUM CONCENTRATION (WEIGHT-FRACTION)	Radial distribution of plutonium concentration of each axial node.
	AXIAL FAST FLUX (E.GE. 0.1 MEV) RATE	Axial fast flux distribution normalized as the mean value to be unit.
	AXIAL HEAT GENERATION RATE	Axial distribution of heat generation rate normalized in the fuel region.
	RADIAL HEAT GENERATION RATE	Radial heat generation rate normalized.
	INITIAL GAS IN PLENUM	Initial gas moles in the plenum including adsorbed gas. (moles)
	*** ACTUALLY USED CORE FOR ARRAY	The actually used core for array A(N) and input value. (words)
	*** ACTUALLY USED CORE FOR MATRIX	The actually used core for array AM(N) and input value. (words)
	*** RECOMMENDED CORE SIZE	The actually used core for the program shown in multiple of 128 K. (byte)

b) Output for each time step

Title		Contents
	STEP NO.	Integrated time steps used in the code. (The output time steps are controlled by an option, JOUT of input card No.1)

Title		Contents
	TIME	Integrated irradiation time from start up (hr)
	POWAV	Average linear heat rating: Q (w/cm) $$Q = \frac{Q_o}{(1+\varepsilon_f^z)}$$ where Q_o = average heat rating of input, and ε_f^z = total axial strain of fuel column.
	TOUT	Coolant outlet temperature (°C) (If it is not given in input, the code calculate the temperature.)
	TIN	Coolant inlet temperature given in input (°C)
	DELT	Time interval of the time step (hr)
	CFLX	Ratio of fast neutron flux to linear heat rating given in input ($(n/cm^2/sec)/(w/cm)$)
	PO	Average coolant pressure (kg/cm^2)
GROSS OUTPUT NEXT		
		Integrated results of fuel pin conditions.
	POWER	Linear heat rating of each axial section (w/cm)
	FLUX	Average fast neutron flux of each axial section ($n/cm^2/sec$)
	FLUENC	Average fast neutron fluence of each axial section (n/cm^2)
	BURN UP	Average burn up of each axial section (a/o)
	BURN UP	Average burn up of each axial section (MWD/T-fuel)
	FUEL LENGTH	
		Axial node length of fuel column (cm)
	CLAD LENGTH	
		Axial node length of cladding (cm)
	THEO. DENS.	
		Theoretical density of fuel of each axial section (g/cm^3)
	COOLANT TEMP	
		Coolant temperature distribution of each axial section (°C)
	FILM H.	Coolant-cladding heat conductance of each axial section. ($w/cm^2/°C$)
	GAP H.	Fuel-cladding gap conductance of each axial section ($w/cm^2/°C$)
	DELT GAS REL.	
		Gas released during the time interval from each axial section (moles)

Title	Contents
TOTAL GAS IN PLENUM	Total gas in plenum including released fission gas and adsorbed gas. (moles)
CUMULATIVE GAS RELEASED	Cumulative fission gas released. (moles)
PERCENT OF TOTAL F.P. GAS	Fission gas release rate. (%)
PLENUM PRESSURE	Plenum pressure. (kg/cm^2)
CENTRAL HOLE PRESSURE	Pressure in the central void. (kg/cm^2)
TOTAL FUEL LENGTH AND FL/FLO	Total fuel column length and the percentage of length change. (cm) & (%)
TOTAL CLAD LENGTH AND CL/CLO	Total cladding length and the percentage of length change. (cm) & (%)
+++ PERFORMANCE	
	Average performance of each radial region (i.e. columnar grain region, equiaxed grain region, undistrubed region of fuel, cladding and capsule region) is shown for each axial section.
RADIUS	Radius of the boundary (cm)
TEMP.	Temperature of the boundary (°C)
AV. TEMP.	Average temperature in the region (°C)
ALPHA-T	Thermal expansion rate of the region
DENS	Average density of the region (g/cm^3)
MASS	Mass per unit length of the region (g/cm)
STRESS AND STRAIN OUTPUT NEXT	
	Material properties, boundary conditions, stress and strain are shown for every radial meshes and axial sections which are used in stress-strain calculation. The output are separated into capsule, cladding and fuel regions.
RADIUS	Radius of each boundary of radial mesh. (cm)

Title		Contents
	DISPLACEMENT	
		Displacement of each boundary during the time interval. (cm)
	TEMP.	Temperature of each boundary
	PRESS.	Contact pressure at the boundaries.
	ALPHA-T	Thermal expansion rate of each radial mesh.
	POISSON RATIO	
		Average poison ratio of each radial mesh.
	YOUNG M.	Young modulus of elasticity of each radial mesh. (kg/cm^2)
	NUMBER OF CRACK	
		Number of cracks in each radial mesh for radial, circumferential and axial directions.
	STRESS	Stress in each radial mesh for radial, circumferential and axial directions. (kg/cm^2)
	TOTAL STRAIN	
		Accumulative total strain in each radial mesh for radial, circumferential and axial directions.
	CREEP STRAIN	
		Accumulative creep strain in each radial mesh for radial, circumferential and axial directions.
	CREEP IN DELT	
		Creep strain increment in the time interval in each radial mesh for radial, circumferential and axial directions.
	SWELLING STRAIN	
		Accumulative swelling strain in each radial mesh.
	TOTAL	Total swelling strain.
	SOLID COMPONENT	
		Swelling caused by solid fission products. (for fuel region only)
	GAS COMPONENT	
		Swelling caused by gaseous fission products. (for fuel region only)
	HOT PRESSING	
		Strains caused by hot pressing. (for fuel region only)

Title		Contents
	PERCENT CHANGE IN ORIGINAL O.D.	
		Percent change of outer diameter vs original diameter. (%)
	AVERAGE POISSON RATIO	
		Poisson's ratio averaged in fuel or cladding region.
	AVERAGE YOUNG MODULUS	
		Young's modulus of elasticity averaged in fuel or cladding. (kg/cm^2)
	GAP SIZE	Radial gap size between fuel and cladding. (cm)
	FRICTION FORCE	
		Axial friction force between fuel and cladding. (kg/cm)
+++ PLUTONIUM MIGRATION		
		Plutonium radial distribution in each axial cross section.
	RADIUS	Boundary radius of mesh used in the plutonium migration calculation. (cm)
	TEMP.	Temperature of boundary of mesh. (°C)
	PU CONC.	Plutonium concentration in each radial mesh.
CPU TIME		Accumulative computer time from the beginning of calculation. (sec)

3) Plotter output

ACTIVE-II has the routines of plotter output provided by Calcomp. It requires Calcomp 1136 drum type plotter, and the paper of 13 inches wide and about 10 meters long. The types of pens are black ink pen of 0.5 mm, red ink pen of 0.5 mm and green ink pen of 0.5 mm for No.1, 2 and 3 pen positions, respectively.

The estimated time for plotting is about 40 minutes.

Plotter output is controlled by the options given by input card. No.3 and No.4

The first figure is the title of the output and the date of code used is given.

The second figure is a set of fuel cracking patterns of cross sections perpendicular to and parallel to the axis. Red and green dotted lines show the cracking positions. This output time position is controlled by NPLOTC given in input card No.38.

The third figure shows the gross performance of fuel pin as a function of time. Maximum fuel center temperature and maximum fuel surface temperature of fuel pin are shown by black solid lines, and red line shows the fission gas release rate and green line shows the plenum pressure as a function of time.

The 4th one shows the linear heat rating and burnup of the axial cross section specified by the option IPFUL given in input card No.4.

The 5th figure shows the maximum fuel center temperature as a function of time of the axial cross section specified by the option IPFUL given in input card No.4.

The 6th figure shows the radial temperature profiles of fuel temperature at each time step at the axial cross section specified by the option IPFUL given in input card No.4. In this figure, columnar grain region outer boundary and equiaxed grain region outer boundary are pointed by red and green marks respectively.

The 7th figure shows the fuel restructuring as a function of time. Equiaxed grain outer boundary, columnar grain outer boundary and central void radius are shown at the specified axial cross section by the option IPFUL given in the input card No.4.

The 8th figure shows the relationship between fuel-cladding gap and cladding hoop stress as a function of time. The axial cross section chosen for this figure is specified by the option IPSTS given in the input card No.4.

The 9th figure shows the total cladding circumferential strain and its creep and swelling components as a function of time. The axial cross section is specified by the option IPSTN given in the input card No.4.

The last one shows the plutonium redistribution calculated in subroutine PLUTO. Radial plutonium distribution for each time step is shown at the cross section specified by the option IPFUL given in the input card No.4

Acknowledgements

This ACTIVE-II code is not only the work of the authers but also the fruits of knowledge of many persons, to whom the authers would like to thank from the bottom of their hearts: P. G. Shewmon, B. R. T. Frost, R. W. Weeks, V. Z. Jankus, L. A. Neimark, J. D. B. Lambert, and W. F. Murphy of Materials Science Division of Argonne National Laboratory, U.S.A., A Watanabe of PNC and A. Doi of Hitachi Ltd. for their assistances for the auther to learn actual fuel behavior and LIFE-I code, which is the mother of ACTIVE-II;

T. Ohkubo of Tokyo Univ. and T. Kikuchi of JAERI for their consulting the modeling;

H. Akutsu, T. Aoki, H. Mizuta, T. Ohtake, R. Yumoto, T. Kajiyama, K. Itagawa and other members of Plutonium Design Section of PNC, and M. Yomori of Japan IBM for their continued strong support and encouragement.

References

1) R. F. Canon, J. T. A. Roberts and R. J. Beals; Deformation of UO_2 at High Temperatures, J. Am. Ceram. Soc. 54(1971) 105.
2) V. Z. Jankus and R. W. Weeks; LIFE-II, a Computer Analysis of Fast Reactor Fuel-Element Behavior as a Function of Reactor Operating History, Nucl. Eng. and Design 18 (1972) 83-96.
3) R. A. Seban and T. Shimazaki; Heat Transfer to a Fluid Flowing Turbulently in a Smooth Pipe with Walls at Constant Temperature, ASME Paper 50-A-128 (1950).
4) C. W. Sayles; Trams. Am. Nucl. Soc. 10, 458 (1967).
5) W. E. Baily; Nuclear Metallurgy Symposium, (1967).
6) A. Biancheria; WARD-4135-1, P∿6 (Sep. 1969).
7) V. Z. Jankus & R. W. Weeks; LIFE-I, a FORTRAN-IV Computer Code for the Prediction of Fast-Reactor Fuel-Element Behavior, ANL-7736 (Nov. 1970).
8) BNWL-1345-1
9) Chromium-Nickel Stainless Steel Data Section I-Bulletin B, 'Mechanical and Physical Properties of the Austenitic Chromium-Nickel Stainless Steel at Elevated Temperature', International Nickel Co. Inc. (1963).
10) Steel for Nuclear Application, Manual by U.S. Steel, (May, 1967).
11) WARD-4135-11
12) W. H. Jens & P. A. Lottes; ANL-4627, (1951).
13) M. F. Lyons; UO_2 Pellet Thermal Conductivity from Irradiations with Central Melting, GEAP-4624.
14) A. Biancheria; Trans. Am. Nucl. Soc. 9, 15 (1966).
15) M. F. Lyons; DOCKET-50331
16) Nucleonics Handbook of Nuclear Research & Technology.
17) Engineering Information & Data (AMAX)
18) F. W. Dittus and L. M. K. Boelter; Univ. Calif., Pubs. Eng. 2, P.∿443, (1930).

APPENDIX A: SUBROUTINE DICTIONARY

Appendix A: Subroutine Dictionary

ACTIVE-II requires five kinds of subroutines, that is, 1) FORTRAN library subroutines, 2) IBM Tokyo data center service library subroutines, 3) IBM support subroutine (SL-MATH), 4) plotter subroutines, and 5) original subroutines written for ACTIVE-II

The explanation of the FORTRAN library subroutines are abridged, because they are foundamental for FORTRAN language. The full lists of the original subroutines written for ACTIVE-II are shown in Appendix D, and the explanations are described in this chapter. As other kinds of subroutines are proprietary of IBM, only the explanations of these are given here.

A.1 IBM Tokyo Data Center Service Library Subroutines

1) STIMER and TTIMER

The combinative use of STIMER with TTIMER enables the users to obtain the CPU time.

Example:

```
        :
   CALL STIMER (ID)  ...........    ①
        :
   CALL TTIMER (T)   ...........    ②
        :
```

where ID=2.

The argument T in the statement ② is the CPU time from the statement ① to the statement ② . (Floating type in unit of second)

2) DATE

The subroutine DATE returns the date of today in the format of month/day/year.

Example:

```
      REAL*8    DAY
      CALL DATE (DAY)
      WRITE(6,1) DAY
    1 FORMAT(1HO, 'DATE', A8)
```

The argument DAY must be defined in double precision.

A.2 IBM Support Subroutines (SL-MATH)

SL-MATH (Subroutine Library-Mathematics)

SL-MATH provides powerful tools for the solution of many problems in industry, science and engineering. SL-MATH is a collection of subroutines dealing with matrix, algebra and numerical mathematics. The subroutines, computational building blocks written in FORTRAN-IV, operate on data that reside in main storage.

ACTIVE-II uses subroutine MSNR and MBSR in the SL-MATH to solve a stress-strain matrix, which is a non-symmetric sparse matrix.

1) Storing and processing of vectors for non-symmetric sparse matrices

A sparse N-by-N matrix A is described by three one-dimensional arrays: An array, also named A, which contains only the non-zero elements of the matrix A (stored row-wise), an array JA containing the corresponding column indices of the non-zero elements of A (stored row-wise), and an array IA containing pointers to the locations of the first element and corresponding column index of each row of A within the array A and JA, respectively. That is, A(K), for K=IA(I) is the first non-zero element of the i-th row of A, JA(K) being the corresponding column index.

The dimension of the array A (and the array JA, respectively) is (at least) the number of non-zero elements of the matrix A. The dimension of the array IA is (at least) N+1. IA(N+1) is the pointer to the (virtual) element of the array A immediately following the last location of a non-zero element, (to indicate the end of the last row).

Example:

The 6-by-6 matrix $A = \begin{pmatrix} 7 & 0 & -3 & 0 & -1 & 0 \\ 2 & 8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -3 & 0 & 0 & 5 & 0 & 0 \\ 0 & -1 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & -2 & 0 & 6 \end{pmatrix}$

with 12 non-zero elements is described by the arrays A, JA, IA as follows.

```
IA(1) = 1,      A(1)=7,    A(2)=-3,   A(3)=-1,
               JA(1)=1,   JA(2)= 3,  JA(3)= 5,
IA(2) = 4,      A(4)=2,    A(5)= 8,
               JA(4)=1,   JA(5)= 2,
IA(3) = 6,      A(6)=1,
               JA(6)=3,
IA(4) = 7,      A(7)=-3,  A(8)=5,
               JA(7)= 1, JA(8)=4,
IA(5) = 9,      A(9)=-1,  A(10)=4,
               JA(9)= 2, JA(10)=5,
IA(6) = 11,     A(11)=-2, A(12)=6,
               JA(11)=4, JA(12)=6,
IA(7) = 13.
```

2) Subroutine MSNR

MSNR factoriezes a given general non-singular sparse matrix A into a lower (L) and an upper (U) triangular factor (A=L·U). Thereby, partial pivoting based on row-density is performed, i.e., the rows of A are chosen as pivot rows in the sequence of their lengths, and the updated elements with maximum absolute value are chosen as pivot elements (column choise).

Usage:

CALL MSNR (N,IA,JA,A,IBOT,IC,JC,IU,DI,C,IPR,IPC,IER,KA,AR)

N. Given number of rows and columns to be processed for matrix A.

IA. Given vector (dimension N+1) containing the pointers to the first element in each row of A in row-wise order.

JA. Given vector (dimension NT=total number of non-zero elements of A) containing the column indices of A in row-wise order.

A. Given vector (dimension NT) containing the values of the elements of A in row-wise order.

IBOT. Given dimension of vectors JC and C (refer to the section Remarks).

IC. Resultant vector (dimension N+1) containing the pointers to the first element in each row of L.

JC. Resultant vector (dimension IBOT=total number of elements in the factors L and U, excluding the diagonals), containing the column indices of the elements of L and U in row-wise order. The column index of the last element in row I of L is followed by that of the first element in row I of U to the right of the diagonal.

IU. Resultant vector (dimension N) containing the pointers to the first element in each row of U.

DI. Resultant vector (dimension N) containing the values of the diagonal elements of L.

C. Resultant vector (dimension IBOT) containing the numerical values of L and U in row-wise order, with the exception of the diagonal elements.

IPR. Resultant vector (dimension N) determining the sequence of pivot rows.

IPC. Resultant vector (dimension N) determining the sequence of pivot columns.

IER. Resultant error indicator.

KA. Auxiliary vector (dimension N) containing the column indices of the pivot row.

AR. Auxiliary vector (dimension N) containing initially the values of the elements of row I of A and, finally, those of the rows I of L and U.

This vector must be defined in double precision.

Error Table:

Code	Explanation	Program Action	Comments
0	No error was detected.	The total number of multiplications and the number of built-ups are returned in KA(1) and KA(2), respectively.	
1000	Length of row I of A is zero.	The index I is returned in KA(1). Processing is discontinued.	
2000	Numerical value of the pivot element is zero.	The index of the last row processed correctly is returned in KA(1). KA(2) and KA(3) contain the pivot row and column last chosen. Processing is discontinued.	A is singular.
3000	Space of JC and C is exceeded.	The index of the last row processed correctly is returned in KA(i), and processing is discontinued.	IBOT must be increased.

Code	Explanation	Program Action	Comments
5000	Pivot element P at step I is approximately zero. ($-FUZZ \leq P \leq FUZZ$) (refer to the section Remarks).	Same as for code 2000.	Pivot element has a value below the limit.

Remarks:

The sequence (IPR) of pivot rows is determined by sorting the rows by their increasing lengths. At the i-th factorization step the row IRR(I) of A is transformed into the i-th row of L and U by subtracting multiples of rows of U. After this process the element with the maximum absolute value in the remaining columns is chosen as pivot. Following this modified scheme of Gaussian elimination, the numerical values for all non-zero elements U (i,j; i≠j) and L (i,j) are computed and stored row-wise as one matrix in the vector C without the diagonal elements, which are stored in vector DI. Only operations on non-zero operands are performed.

If NT1 is the number of built-ups in L and U, and NT2 is the total number of elements of A not located on the diagonal, IBOT must be equal or greater than NT1+NT2=total number of resultant element in JC or C(L,U), respectively.

At the beginning of MSNR there is a statement by which a variable FUZZ is set to zero. Only pivot elements having an absolute value greater than FUZZ will be accepted. The user may assign a small non-zero tolerance value to this variable by changing this statement.

3) Subroutine MBSR

MBSR solves the system of linear equations AX=B for a general non-singular sparse coefficient matrix A, given its right-hand sides (B) and its LU decomposition.

Usage:

CALL MBSR (N,IPR,IPC,IC,IU,JC,DI,C,B,X,Y)

N. Given number of rows and columns to be processed for matrix A.

IPR. Given permutation vector (dimension N) determining the sequence of pivot rows.

IPC. Given permutation vector (dimension N) determining the sequence of pivot columns.

IC. Given vector (dimension N+1) contaning the pointers to the first element in each row of L.

IU. Given vector (dimension N) containing the pointers to the first element in each row of U.

JC. Given vector (dimension IBOT as used in the preceding factorization by MSNR), containing the column indices of the rows in L and U stored in row-wise order as one matrix C.

DI. Given vector (dimension N) containing the diagonal elements of L.

C. Given vector (dimension IBOT) containing row-wise numerical values of L and U without the diagonal elements.

B. Given vector (dimension N) containing the right-hand sides of the system of linear equations.

X. Resultant vector (dimension N) containing the solution of AX=B.

Y. Auxiliary vector (dimension N) contaning the intermediate results of LY=B.

Remarks:

Using the permutation vectors IPR and IPC of pivot rows and columns (determined by partial pivoting based on row density by MSNR), MBSR solves first the forward substitution LY=B. Then UX=Y is solved (backward substitution), and the result is returned in X.

The method differs from the usual forward and backward substitution in that MBSR,

(1) deals with rows and columns in permuted order, and

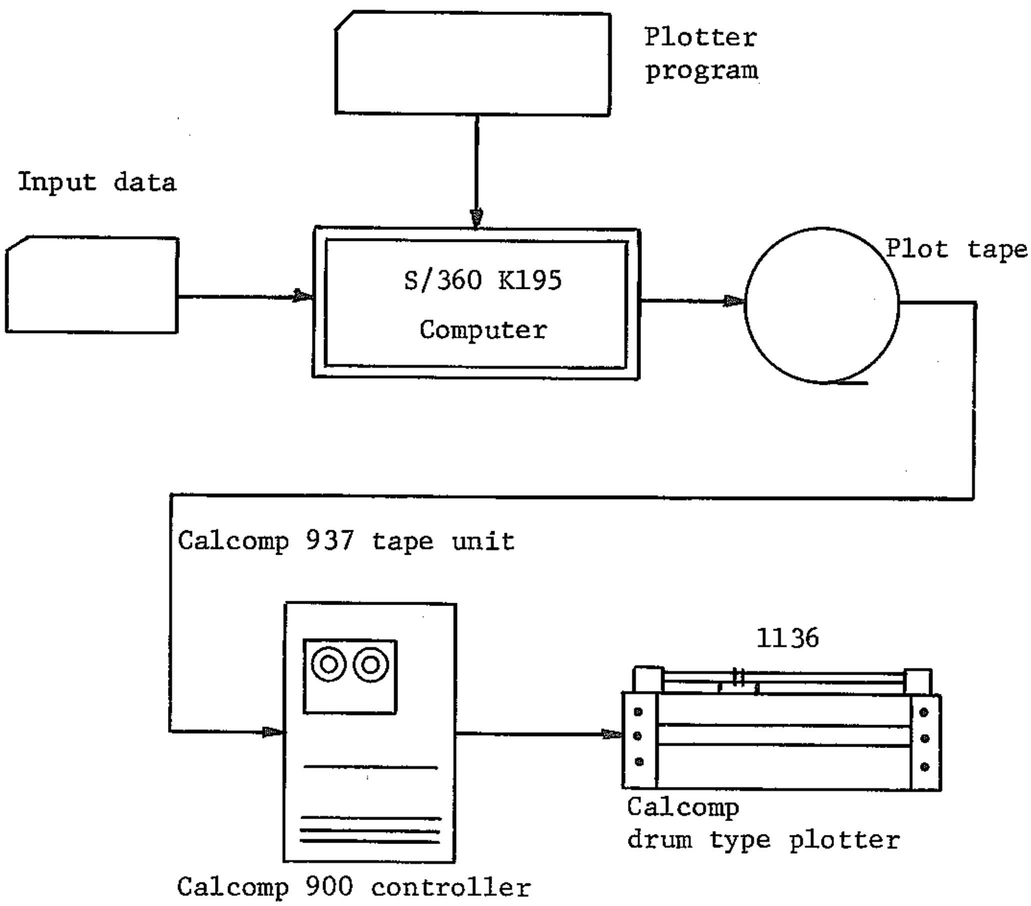
(2) processes only the non-zero elements of the factors L and U.

A.3 Plotter Routines

Calcomp provides basic and functional subroutines, which is a set of closely related subroutines that generate output for controlling a digital graphic system. The programmer is not required to communicate with the hardware in its own data structure. Instead, he communicates with the set of subroutines in a manner such as: 'move the pen to specified coordinates'; 'place some characters at a certain location on the page'; 'draw a line through a series of points'.

ACTIVE-II uses nine plotter subroutines to plot several variables and transform the values of variables which are stored in the sequence of time history into visible forms.

Plotting system configuration (offline) is shown below.



1) CALL PLOTS(X,X) or CALL PLOTS(X,X,LDEV)

The PLOTS entry is used to initialize the PLOT subroutines. It must be called only once before any other call to plotter subroutine is given.

X is not used.
LDEV is the logical output device number.
If LDEV is not specified, it is set to 77.

2) CALL PLOT (X1,Y1,IPEN)

The PLOT entry is used primarily to move the pen in a straight line to a new position, with the pen either up or down during the movement.

X1,Y1 ; are the X, Y coordinates, in millimeters from the current reference point (origin), of the terminal position to which the pen is moved. An origin (where both X, Y equal zero) may be established anywhere on the plotting surface, as explained below for negative IPEN values.

IPEN = 2 ; the pen is down during movement, thus drawing a visible line.

IPEN = 3 ; the pen is up during movement.

IPEN =-2 or -3 ; A new origin is defined at the terminal position after the movement is completed as if IPEN were positive. IPEN must be 999 in the last plotter call in a program. The effects are the same as if IPEN=-3, except that a 999 search record is written, and the output device is closed.

3) CALL NEWPEN (NPEN)

This entry enables program selection of any one of the three available pens. Pen No.1 is initially selected when the PLOTS entry is called.

NPEN ; specifies the number of the pen to be selected (1-3). The old pen is raised, and the new pen is moved to the same physical location where the old pen was positioned.

4) CALL SYMBOL (X1, Y1, HEIGHT, IBCD, ANGLE, NC)

This entry is used to draw text such as titles, captions, and legends.

X1, Y1 ; are the coordinates, in millimeters, of the lower left-hand corner (before character rotation) of the first character to be produced. The pen is up while moving to this point. If X1 and/or Y1 equals 999.0, X1 and/or Y1 is assumed to be the current position.

HEIGHT ; is the height, in millimeters, of the character to be plotted.

IBCD ; is the text to be used as annotation. The character(s) must be left-justified and contiguous in: a single variable, an array, or a Hollerith literal.

ANGLE ; is the angle, in degrees from the X-axis, at which the annotation is to be plotted. ANGLE is positive, if counter-clockwise.

NC ; is the number of characters to be plotted from IBCD. If NC>0, the data must be left-justified in the first location of IBCD, If NC=0, one alphameric character is produced, using a single character which is right-justified in the first location of IBCD.

5) CALL SYMBOL(X1, Y1, HEIGHT, ICODE, ANGLE, IPEN)

This form of SYMBOL call is used to draw special centered symbols such as a box, octagon, triangle, etc., for plotting data points. X1, Y1, HEIGHT, and ANGLE are the same as described for the call in the above section (4). If the symbol to be produced is one of the centered symbols (i.e., if ICODE is less than 14), X1, Y1 represent the geometric center of the character produced.

ICODE ; is the integer equivalent of the desired symbol. Valid integers and their symbols are listed in the symbl table of the Table A.1, If ICODE is 0 through 13, a centered symbol is produced.

IPEN =-1 ; the pen is up during movement, after which a single symbol is produced.

=-2 or less; the pen is down during movement, after which a single symbol is produced.

6) CALL NUMBER (X1, Y1, HEIGHT, FPN, ANGLE, NDEC)

This entry is used to draw a real variable (floating-point number). X1, Y1, HEIGHT, and ANGLE are the same as those arguments described for subroutine SYMBOL.

FPN ; is the location of the floating-point number that is to be plotted.

NDEC = 0 ; only the integer portion of the number and a decimal point are plotted after rounding.

=-1 ; only the integer portion of the number is plotted after rounding.

=1 through 11; specifies the number of digits to the right of the decimal point that are to be plotted.

7) CALL CIRCLE (X1, Y1, A1, A2, R1, R2, DS)

CIRCLE draws a circle, arc, or a spiral starting at a given point.

X1, Y1 ; are the coordinates of the starting point of the arc, in millimeters.

A1 ; is the angle for the start of the arc, in degrees.

A2 ; is the angle for the end of the arc, in degrees.

R1 ; is the radius at the start of the arc, in millimeters.

R2 ; is the radius at the end of the arc, in millimeters.

DS ; is a code used to specify the type of line desired.
If DS=0.0, a solid arc is drawn.
=0.5, a dashed arc is drawn.

8) CALL AROHD (X0, Y0, X1, Y1, S, W, ICODE)

AROHD draws an arrowhead at the end of a line segment.

XO, Yo ; are the coordinates of the starting point of the line segment determining the direction of the arrowhead, in millimeters.

X1, Y1 ; are the coordinates of the tip of the arrowhead, in millimeters.

S ; is the length of the arrowhead, in millimeters.

W ; is the width of the arrowhead, in millimeters.

ICODE ; is a two-digit decimal code 10*I+J where:

I is 0 if no line is desired from X0, Y0 to the arrowhead at X1, Y1.

1 if a line is desired.

2 if a line is desired and a second arrowhead pointing in the opposite direction is desired at X0, Y0.

J is 1-7 specifying the type of arrowhead desired, as shown in Table A.1.

9) CALL DASHPT (X1, Y1, W)

DASHPT draws a dashed line from the present pen position to a specified point.

X1, Y1 ; are the coordinates of the point to which the dashed line is to be drawn, in millimeters.

W ; is the length of the dash and the space between dashes, in millimeters.

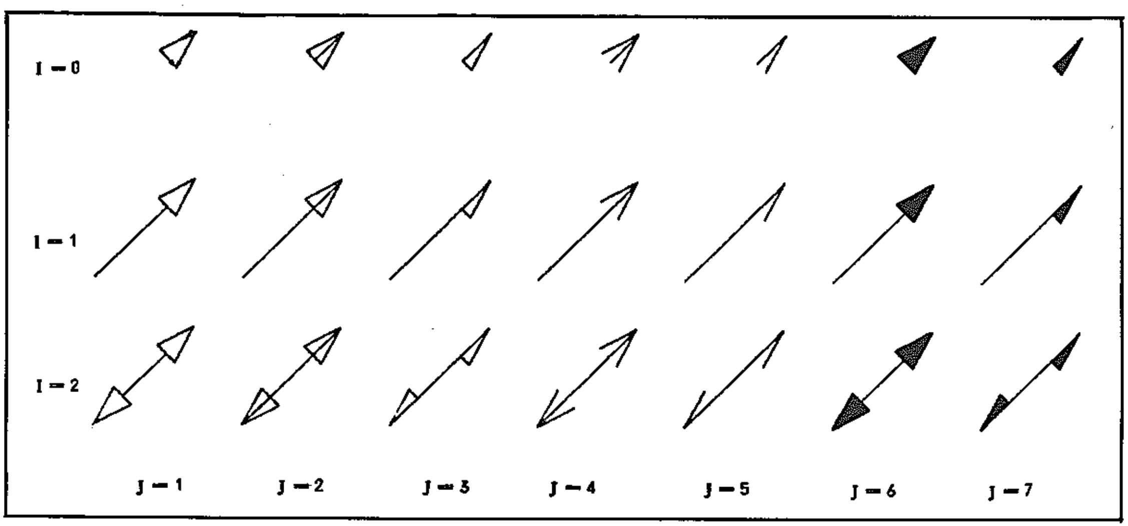
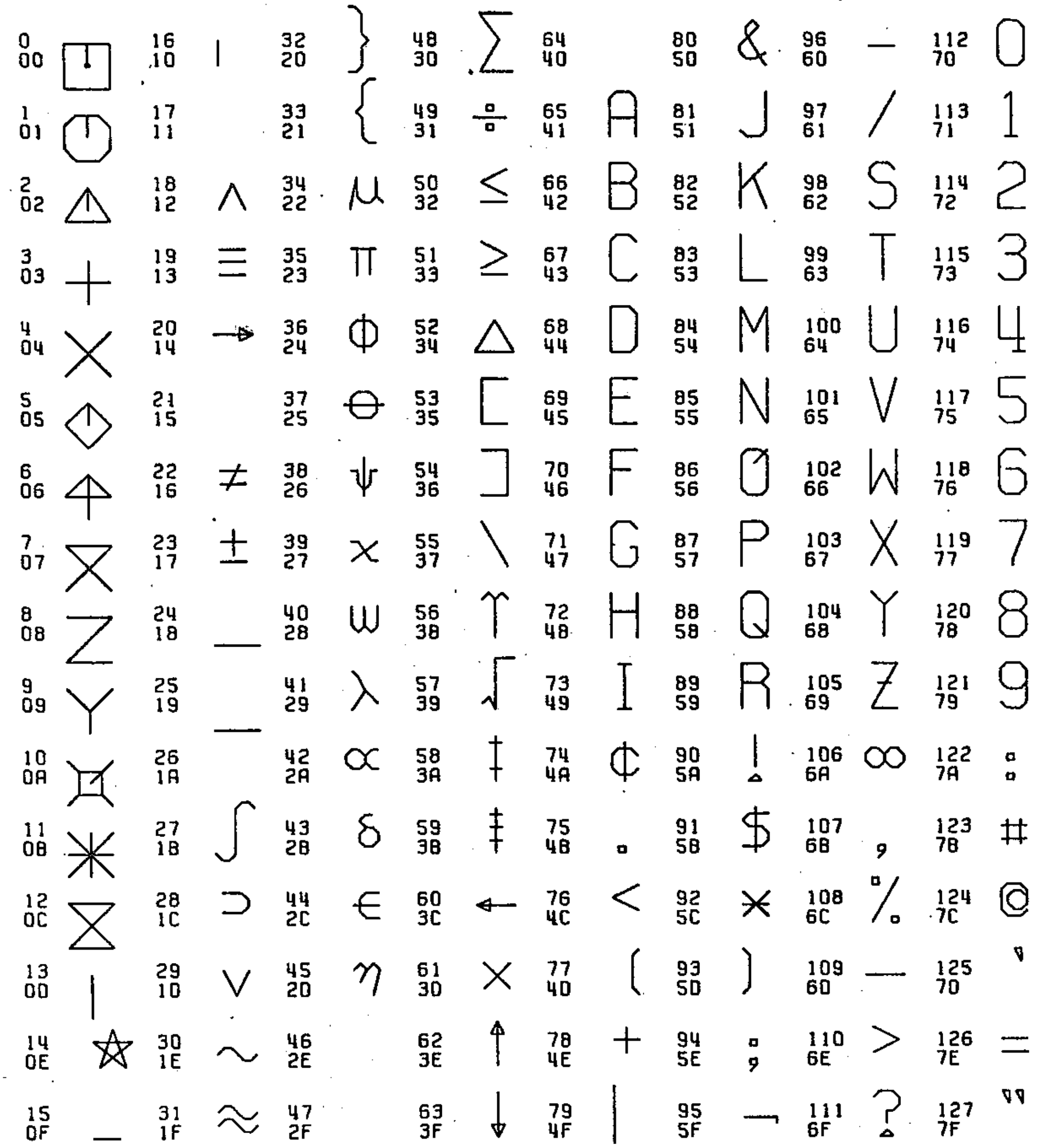


Table A.1 Symbols for plotter

CHARACTERS AVAILABLE IN THE IBM 360 SYMBOL ROUTINE

CODES NEXT TO EACH SYMBOL ARE: 1. INTEGER CODE USED IN SPECIAL SYMBOL CALL.
2. INTERNAL HEXADECIMAL CODE (EBCDIC MODULO 128)
SEE THE IBM 360 REFERENCE DATA CARD FOR CORRESPONDING PUNCHED CARD CODES.



NOTES: 1. THE FIRST 14 SYMBOLS IN THE TABLE ARE CENTERED SYMBOLS.
2. ANY INTEGER OR HEX CODE, OTHER THAN THOSE SHOWN, WILL BE CONSIDERED MODULO 128. THEREFORE EITHER OF TWO PUNCHED CARD CODES FROM THE 360 REFERENCE CARD MAY BE USED FOR EACH SYMBOL.

Table A.1 Symbols for plotter (continued)

A.4 Original Subroutines

Original subroutines written for ACTIVE-II are 59, and the whole lists are given in Appendix D.
The names of subroutines are as follows.

MAIN	ACTIVE	AREA
ATBFUN	AXIAL	AXISD
AXIST	BCALC	BCALCP
BEGIN	BURN	CHECK
CHKSPD	CLEAR	CLTMP
COEF	CORTMP	CRACK
CREEC	CREEF	EDTGRS
EDTINP	EDTMTX	EQUISI
FLCOND	FLTMP	FLXCAL
FRICHK	GAPCON	GASOUT
GOLDN	HEADER	HKNS
INPUT	IOSET	ITERAT
LENGTH	MATRX	MATRXP
OTMCAL	PLENP	PLOT1
PLOT2	PLOT3	PLOT4
PLOT5	PLTOR	PLUTO
POWCUT	PROPT	RHOMOD
SUBTIT	SWELL	TIMCUT
UPDATE	UPDATP	UPDTE
VOLUME	XLAME	

1) MAIN routine

MAIN routine must be programmed by the user when the ACTIVE-II is run. (See the section CORE STORAGE or CONTROL CARDS.)

Sample MAIN routine:

```
COMMON /DIM/ IA, IAM
DIMENSION  A(5000), AM(10000)
IA=5000
IAM=10000
CALL ACTIVE (A,AM)
STOP
END
```

2) Subroutine ACTIVE (A,AM)

This subroutine ACTIVE controls the calls to all subroutines included in the ACTIVE-II code. The calling flow is described in chapter 4.

The time step interval is determined internally in the ACTIVE. The maximum changes in time step interval, average linear power, restructuring, fuel creep, and cladding creep allowed to take place within a given time step are limited to specified values. If any of the computed changes exceed these limits, the time step interval is cut internally by this subroutine ACTIVE.

The limiting conditions are:

1. Time step interval must not exceeds the CX times DELTO.
   Presently, CX=2.0, DELTO=preceding time interval (hrs).
   (Refer to the subroutine TIMCUT.)

2. Average linear power change must not exceeds the EPSPOW.
   Presently, EPSPOW=50.0 (w/cm).
   (Refer to the subroutine TIMCUT.)

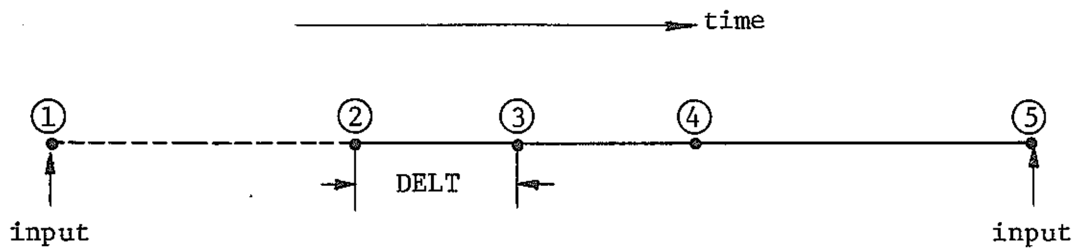
3. Restructuring velocity must not exceeds the EPSSPD.
   EPSSPD is the input value. If zero, EPSSPD is set to 0.01 (cm/sec).
   (Refer to the subroutine CHKSPD.)

4. Fuel creep increment must not exceeds the EPSCRF. EPSCRF is the input value. If zero, EPSCRF is set to 0.05.
   (Refer to the subroutine CHECK.)

5. Cladding creep increment must not exceeds the EPSCRC. EPSCRC is the input value. If zero, EPSCRC is set to 0.01.
   (Refer to the subroutine CHECK.)

If any of these limits are exceeded, the time step interval is halved, and the power and coolant temperature changes are adjusted accordingly.

Called from : MAIN (user written program)
Functions used : DATE, STIMER (library)
Commons required: ALWAYS, OTHERS, CONST, CAPSUL, IOFIL, FBRATR, DIM, PLTOPT, DAY
Variables required:
A - used for all the variables except these for matrix.
AM - used only for the variables related to the matrix.
Variables required:



Point of time

① Time specified on the preceding input card.

② Preceding step at which the calculation has just been done. (step number = IT-1)

③ Current time step at which the calculation is being done. (step number = IT)

④ Time once calculated but not accepted due to criteria for restructuring velocity and for the creep increment. (limitation 2.,4.,5.)

⑤ Time specified on the current input card.

Variables used at the above each point of time are as follows.

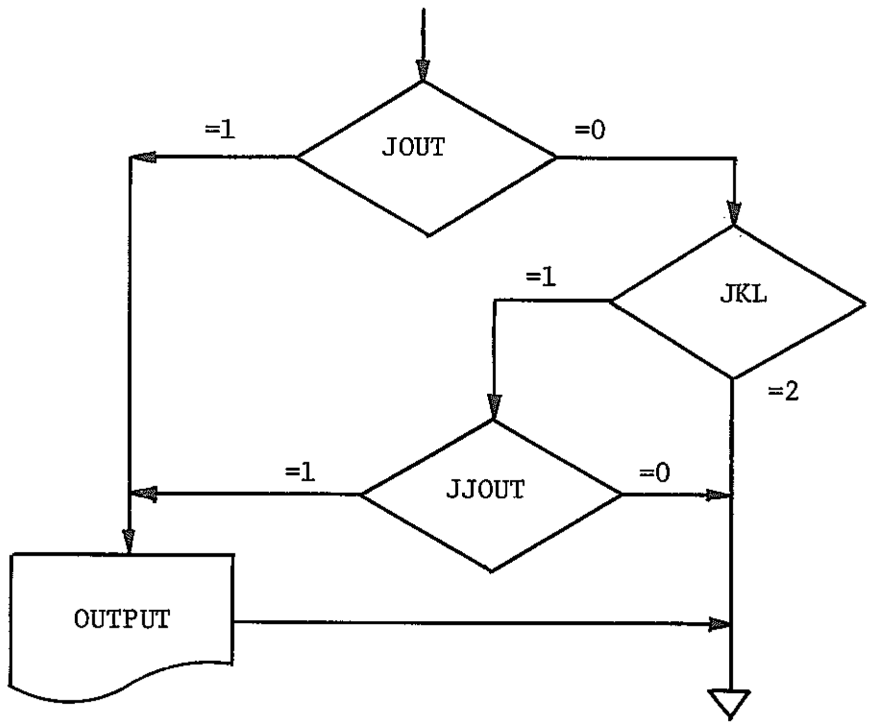
Point of time	②	③	④	⑤
time (hr)	TIP	TIM	YTIM	TIMO
average linear power (w/cm)	PPOWA	POWAV	YPOWAV	POWAO
coolant inlet temperature (°C)	PTIN	TIN	YTIN	TINO
coolant outlet temperature(°C)	PTOUT	TOUT	YTOUT	TOUTO
average external pressure on cladding (kg/cm^2)	PPO	PO	YPO	POO

The time step interval, DELT, is determined as (TIM-TIP).

IT - step counter for calculation. (step number)

ITPLOT - step counter for plotting.

Output timing is controlled by the input value, JOUT, and the internally determined values, JKL, JJOUT.



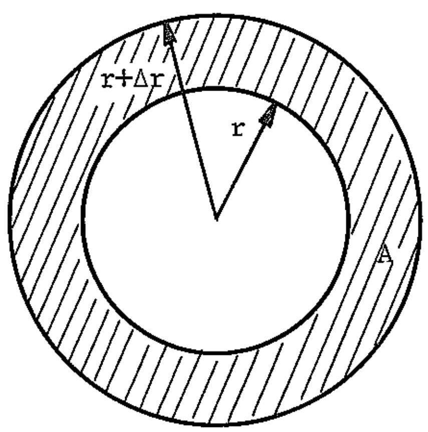
JKL - is set to 2, when the time step is cut internally due to the limiting conditions 1 or 2 as mentioned before. Otherwise, JKL is set to 1.

JJOUT - is set to 0, when the time step is cut due to the limiting conditions 3, 4 or 5 as mentioned before. Otherwise, JJOUT is set to 1.

JCR - is set to 1 or 2, when the fuel cracking or crack healing occurred, respectively. Otherwise, JCR is set to 0. (Refer to the subroutine CRACK)

3) Function AREA (R, DR)

The AREA calculates the area between the circle with radius R and the concentric circle with radius R+DR.



$$A = (r+\Delta r)^2 - r^2 = \Delta r(2r+\Delta r)$$

Called from : CLTMP

Variables required:

R - radius r. (cm)

DR - Δr. (cm)

Variables changed :

AREA - $\Delta r(2r+\Delta r)$ (cm^2)

4) Function ATBFUN (T, DT, I)

This function calculates the thermal expansion in any of the three fuel regions, i.e., columnar grain region, equiaxed grain region, and undisturbed region.

The thermal expansion is computed by integrating the thermal coefficient of expansion from the room temperature to the temperature T+DT/2.

$\alpha(T) = a_0 + a_1 T$: thermal coefficient of expansion. (°C-1)

T : temperature (°C)

Y = 20 °C : room temperature

$$\int_{Y}^{T+\frac{\Delta T}{2}} \alpha(T)\,dT = \int_{Y}^{T+\frac{\Delta T}{2}} (a_0+a_1T)\,dT = (a_0T+\frac{a_1}{2}T^2) + (a_0+a_1T)\frac{\Delta T}{2} - (a_0Y+\frac{a_1}{2}Y^2)$$

Called from : FLTMP

Commons required : OTHERS

Variables required:

T - temperature. (°C)

DT - ΔT. (°C)

Y - room temperature. (20 °C)

I - = 1, indicator of columnar region.

= 2, indicator of equiaxed region.

= 3, indicator of undisturbed region.

ALFO - a_0 in the expression a_0+a_1T. $(°C^{-1})$ (refer to the Common OTHERS)

ALF1 - a_1 in the expression a_0+a_1T. $(°C^{-1})$ (refer to the Common OTHERS)

Variables changed:

$$\text{ATBFUN} - \int_{Y}^{T+\frac{\Delta T}{2}} \alpha(T)\,dT \,. \qquad (-)$$

5) Subroutine AXIAL (XO, YO, ID)

This subroutine AXIAL is used to plot an axial section number at the right-hand side of the graph.

Called from : PLOT4, PLTOR
Subroutines called: SYMBOL, NUMBER (library)
Variables required:
XO,YO - coordinates where the characters 'AXIAL SECTION=' are plotted. (mm)
ID - axial section number.

6) Subroutine AXISD (MX, XMN, DX)

The AXISD draws the latitudinal axis with regard to distance with graduations.

Called from : PLTOR
Subroutines called: PLOT, NEWPEN, NUMBER (library)
Variables required:
MX - number of graduations.
XMN - interval between graduations.
DX - scaling factor.

7) Subroutine AXIST (MX, XMN, DX)

The AXIST draws the latitudinal axis with regard to time with graduations.

Called from : PLTOR
Subroutines called: PLOT, NEWPEN, NUMBER (library)
Variables required:
MX - number of graduations.
XMN - interval between graduations.
DX - scaling factor.

8) Subroutine BCALC (B, ATBF, ATBCL, TPR, TPC, TPZ, PRS, RF, RC, P,
BM, G, PB, GB, VOL, RH, PLPR, CVPR, PO, PTOP,
DELT, TSR, TSC, TSZ, FLUX, TBCL, KFRIC, SWG, SWS,
EHP, EPR, EPC, EPZ, MOUNT, IFRIC, NS, RB)

The BCALC provides the vector containing the right-hand sides of the stress-strain equations, i.e., stress-strain equations except for that of the plenum section.

Called from : ACTIVE
Functions used : ALOG (library)
Commons required : ALWAYS, OTHERS
Variables required:

ATBF, ATBCL, TPR, TPC, TPZ, PRS, P, BM, G, PB, GB, VOL, RH, TSR, TSC, TSZ, FLUX, TBCL, KFRIC, SWG, SWS, EHP, EPR, EPC, EPZ, RB - refer to the Symbol Dictionary (Appendix C).

RF - radii of the region boundaries in the fuel. (cm)
RC - radii of the region boundaries in the cladding. (cm)
PLPR - plenum pressure. (kg/cm^2)
CVPR - central void pressure. (kg/cm^2)
PO - average external pressure on cladding. (kg/cm^2)
PTOP - axial force on cladding. (kg)
DELT - time step interval. (hr)
MOUNT - switch used when the starting condition of the status of pellet-cladding gap is not accepted.
IFRIC - indicator of status of pellet-cladding gap.
NS - axial section number.

Variables changed:

B - vector containing the right-hand sides of the stress-strain linear equations except for those of the plenum section.

9) Subroutine BCALCP (B, ATBCL, TPR, TPC, TPZ, RC, P, G, PB, GB,
VOL, RH, TBCL, FLUX, KFRIC, CRIF, DELT, TSR,
TSC, TSZ, PLPR, PTOP, PO, EPR, EPC, EPZ, NS)

The BCALCP provides the vector containing the right-hand sides of the stress-strain equations only for the plenum section.

Called from : ACTIVE

Functions used : ALOG (library)

Commons required : ALWAYS, OTHERS

Variables required:

ATBCL, TPR, TPC, TPZ, P, G, PB, GB, VOL, RH, TBCL, FLUX, KFRIC, TSR, TSC, TSZ, EPR, EPC, EPZ, CRIF - refer to the Symbol Dictionary (Appendix C).

RC - radii of region boundaries in the cladding. (cm)

DELT - time step interval. (hr)

PLPR - plenum pressure. (kg/cm)

PTOP - axial force on cladding. (kg)

PO - external pressure on cladding. (kg/cm^2)

NS - axial section number.

Variables changed :

B - vector containing the right-hand sides of the stress-strain linear equations for the plenum section.

10) Subroutine BEGIN (XM, ZFLX, RH, RHOP, CPU, LM, EHP, SWS, GS, GR, RVB, RCB, REB, RUB, QSF, QSFF, QSFP, POW, XCPI, XCPO, BU, HG, KFRIC, FLZO, FLZ, PRS, FLNC, CLZO, CLZ, FL, CL, DXM, SWG, RCI, RCO, RVBO, RCBO, REBO, RUBO, RCIO, RCOO, XMINT, RDIST, EPSEQ, SWS3, LMR, LMC, LMZ)

This subroutine is used for initialization of some variables before any of calculations is performed.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, CONST, CAPSUL

Variables required:

RHOP, RVB, RCB, REB, RUB, QSFF, QSFP, FLZO, CLZO, RCI, RCO - refer to the Symbol Dictionary (Appendix C).

RCPI - capsule inner radius. (cm)

RCPO - capsule outer radius. (cm)

RHOC - cladding density. (g/cc)

PI - π

FA11, FA12, FA21, FA22, FA31, FA32 - refer to the Common CONST.

Variables changed (initialized):

XM, ZFLX, RH, LM, EHP, SWS, GS, GR, QSF, QSFF, POW, XCPI, XCPO, BU, HG, KFRIC, FLZ, PRS, FLNC, CLZ, DXM, SWG, RVBO, RCBO, REBO, RUBO, RCIO, RCOO, XMINT, RDIST, EPSEQ, SWS3, LMR, LMC, LMZ - refer to the Symbol Dictionary. (Appendix C)

FL - total length of the fuel. (cm)

CL - total length of the cladding. (cm)

ALFO, ALF1, SMQ, SMQ0 - refer to the Common OHTERS.

XEWTON - refer to the Common CONST.

11) Subroutine BURN (BU, POW, XM, EPZ, QSF, QSFP, DELT)

This is the subroutine which calculates the burn-up of the fuel. 2.246×10^{19} fissions per gram of the fuel are assumed to be equvalent to one atomic-percent burn-up. And 1 watt is equivalent to 3.1×10^{10} fissions per second. Using these constants and the linear power, the burn-up of the fuel is computed.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS

Variables required:

POW - average linear power during the time step. (w/cm)

XM - mass. (g/cm)

QSFP - axial power shape. (-)

DELT - the time step interval. (hr)

BUCNS - constant used for conversion from (w/cm) to (atomic-percent). $(3.1\times10^{10}\times3600/2.246\times10^{19})$

PDBU - linear power decrement due to burn-up. (1/atomic-percent)

Variables changed :

BU - burn-up. (atomic-percent)

QSF - axial power distribution factor. (g/cm)

SMQ - summation of axial power shapes multiplied by mass. (g/cm)

BUB - average burn-up of the fuel. (a/o)

12) Subroutine CHECK (DEQP, EF, EC, ISW)

The subroutine CHECK is used to check whether the equivalent creep-strain increment satisfies the given criterion or not. If the criterion is satisfied, the tentative time interval is accepted. If not, the time interval is halved in the subroutine ACTIVE.

Called from : ACTIVE
Commons required : ALWAYS
Variables required:

- DEQP - equivalent creep-strain increment in the tetative time interval.
- EF - criterion for DEQP in the fuel.
- EC - criterion for DEQP in the cladding.
- NN - total number of axial sections.
- NFR - total number of radial meshes in the fuel.
- NCL - total number of radial meshes in the cladding.

Variables changed :

- ISW - indication of acceptance of the time step interval. If ISW=1, smaller time interval is required. If ISW=0, the tentative time step interval is accepted.

13) Subroutine CHKSPD (DXM, EPS, NN, ISW, DELT)

This subroutine is used to check whether the migration velocities of each region interfaces satisfy the given criterion or not. If the criterion is satisfied, the tentative time step interval is accepted. If not, the time interval is halved in the subroutine ACTIVE.

Called from : ACTIVE
Variables required:

- DXM - migration velocity in the tentative time step interval. (cm/sec)
- EPS - criterion for DXM. (cm)
- NN - total number of axial sections.
- DELT - tentatively decided time step interval. (hr)

Variables changed :

ISW - indication of acceptance of the time step interval. If ISW=1, smaller time step interval is required. If ISW=0, the tentative time step interval is accepted.

14) Subroutine CLEAR (IFRIC, JFRIC, NS)

This subroutine is used for setting the initial assumption of fuel-cladding gap condition. Fuel-cladding gap is assumed to be open.

Called from : ACTIVE

Variables required:

JFRIC - gap condition at the preceding time step.
=0, open
=1, stick
=2, slip

NS - axial section number.

Variables changed :

IFRIC - gap condition to be assumed at the current time step.
=0, open
=1, stick
=2, slip

15) Subroutine CLTMP (POW, TO, RO, RI, R, T, TB, TBA, ATB, ATBA, RCLO, IT, RB)

The CLTMP is used to determine the temperature and the thermal expansion of the cladding and the capsule (if the fuel-element is capsulated).

First, determined are the radii of interest. The cladding or the capsule is equally divided into NCL meshes.

The temperature distribution, i.e., the temperature at each region boundary, of the cladding (or the capsule) is calculated using the given cladding (or the capsule) outer temperature. (See chapter 3.1-3)) The cladding (or the capsule) outer temperature is determined in the subroutine GOLDN.

When the temperature distribution across the cladding (or the capsule) is known, the average temperature and average thermal expansion are determined.

Called from : ACTIVE
Functions used : ALOG, SQRT (library), AREA
Commons required : ALWAYS, OTHERS, CONST
Variables required:

- POW - average linear power. (w/cm)
- TO - temperature at the outside of the cladding or the capsule. (°C)
- RO - outer radius of the cladding or the capsule. (cm)
- RI - inner radius of the cladding or the capsule. (cm)
- CK1 -
- CK2 -
- CA1 -
- CA2 -
- A - same as XEWTON

(CK1, CK2, CA1, CA2, A: refer to the Common CONST described in the Common Dictionary (Appendix B).)

- PI - π
- IT - time step number

Variables changed :

- R - radius of each region boundary. (cm)
- T - temperature at the radius R. (°C)
- TB - average temperature of the mesh. (°C)
- TBA - average temperature of the cladding or the capsule. (°C)
- ATB - average thermal expansion of the mesh. (-)
- ATBA - average thermal expansion of the cladding or the capsule. (-)
- RCLO,RB - refer to the Symbol Dictionary (Appendix C).
- RCL,TCL,ATCL - radius, temperature, and thermal expansion respectively used for the numerical integration by NEWTON-COTES-6.

16) Subroutine COEF (OTOM, N)

The numerical coefficients in the formula for fuel conductivity are calculated as a function of OTOM (fuel stoichiometry).

Called from : FLTMP
Commons required : ALWAYS, CONST, FBRATR
Variables required:

OTOM - O/M ratio (i.e., X in UOx) according to fuel stoichiometry.
N - axial section number.
JFA - option for reactor model.
=1, for FBR
=2, for ATR
=3, for PWR
=4, for BWR
NOK - option number for fuel conductivity formula.

Variables changed :

FKA1, FKB1, FKC1 -
FKA2, FKB2, FKC2 -
FKA3, FKB3, FKC3 -
refer to the Common CONST described in the section Common Dictionary.

17) Subroutine CORTMP (TOUTER, TOUTO, PTOUR, ISW, TOUT)

The subroutine CORTMP is used for storing/retrieving the temperature at outside of the cladding. Storing/retrieving is needed when the time step interval is halved due to the criteria.

Called from : ACTIVE
Commons required : ALWAYS, IOFIL
Variables required/changed:

ISW - indicator of store/retrieval.
TOUT - coolant outlet temperature. (°C)
TOUTER, TOUTO, PTOUR - refer to the Symbol Dictionary.

18) Subroutine CRACK (SIGR, SIGC, SIGZ, BM, G, TBF, TPR, TPC, TPZ, TSR, TSC, TSZ, PRS, LM, LMR, LMC, LMZ, DELT, NS, JCR)

The CRACK computes whether fuel cracking would occur in any of the three principal directions.

A crack occurs when the stress component exceeds the ultimate tensle stress. The changes in elastic strain due to cracks are compensated by equal changes in plastic strain and volumetric expansion.

(See chapter 3.8)

The cracks are assumed to heal when the stresses in the region become compressive.

If the fuel cracking or crack healing occurs during the time step, the Young's modulus and the equations using it are re-computed.

Called from : ACTIVE

Commons required : ALWAYS, XCRAK

Variables required:

SIGR, SIGC, SIGZ, BM, G, TBF, PRS - refer to the Symbol Dictionary.

DELT - time step interval. (hr)

NS - axial section number.

THLC, HEAL1, HEAL2, SIGF0, SIGF1 - refer to the Common XCRAK.

Variables changed :

TPR, TPC, TPZ, TSR, TSC, TSZ, LM, LMR, LMC, LMZ - refer to the Symbol Dictionary.

JCR - =1, if any crack occurred during the time step.
=0, otherwise.

Significant internal variables:

SIGF - pellet fracture strength (kg/cm^2) (ultimate tensile stress).

19) Subroutine CREEC (EQSIG, DEQP, TBC, FLNC, FLUX, DELT, N)

The equivalent creep-strain increments for the cladding regions are computed according to the creep law for cladding with associated constants.

Called from : ACTIVE

Functions used : EXP, SINH (Library)

Commons required : ALWAYS, OTHERS, XCREC, FBRATR

Variables required:

EQSIG, TBC, FLNC, FLUX - refer to the Symbol Dictionary.

DELT - time step interval. (hr)

N - axial section number.

A, Q, AE, BE, CE, FM, FN - refer to the Common XCREC.

NOC - option number for cladding creep formula. (input) (See chapter 5 for option)

JFA - option number for reactor model. (Refer to the Common FBRATR)

RKR - ideal gas constant. (cal/mol °R)

Variables changed :

DEQP - equivalent creep-strain increment.

20) Subroutine CREEF (EQSIG, DEQP, TBF, POW, FD, FLUX, FTHP, LM, DELT, N, G, EPSEQ)

The subroutine CREEF calculates the equivalent creep-strain increments for the fuel regions according to the creep law for fuel with associated constants.

Called from : ACTIVE

Functions used : EXP (Library)

Commons required : ALWAYS, OTHERS, XCREF

Variables required:

EQSIG, TBF, POW, FD, FLUX, LM, G, EPSEQ - refer to the Symbol Dictionary.

DELT - time step interval. (hr)

N - axial section number.

A0, A1, CFR, B, Q, Q1, FM, FN, GD - refer to the Common XCREF

Variables changed :

DEQP - equivalent creep-strain increment.

FTHP - thermal creep component.

21) Subroutine EDTGRS (IT, TIM, POWAV, TIN, TOUT, DELT, POW, FLUX, BU TNA, XCPO, XCPI, TCPI, TCPO, RCO, RCI, TCLO, TCL, FLZ, CLZ, FL, CL, OTOM, CPU, RHOT, RH, XM, RVB, RCB, REB, RUB, RF, RC, HG, HC, TOGAS, GR, TFR, TBFA, ATBFA, TBCLA, ATBLA, RCP, TCP, TBCPA, ATBPA, FLNC, GS, PLPR, CVPR, FLO, CLO, BUM, PO, FD, FDPM)

All variables in the argument are editted and printed to the printer before stress-strain matrix solving.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, CAPSUL, COMPLT, IOFIL

Variables required:

POW, FLUX, BU, TNA, XCPO, XCPI, TCPI, TCPO, RCO, RCI, TCLO, TCL, FLZ, CLZ, OTOM, CPU, RHOT, RH, XM, RVB, RCB, REB, RUB, HG, HC, GR, TFR, TBFA, ATBFA, TBCLA, ATBLA, RCP, TCP, TBCPA, ATBPA, FLNC, GS, BUM, FD, FDPM - refer to the Symbol Dictionary.

IT - time step number.

TIM - time (hr)

POWAV - average linear power (w/cm)

TIN - coolant inlet temperature (°C)

TOUT - coolant outlet temperature. (°C)

DELT - time interval. (hr)

FL - total fuel length. (cm)

CL - total cladding length. (cm)

RF - boundary radius of each fuel mesh. (cm)

RC - boundary radius of each cladding mesh. (cm)

TOGAS - total gas in plenum. (mol)

PLPR - plenum pressure. (kg/cm^2)

CVPR - central void pressure. (kg/cm^2)

FLO - initial total fuel length. (cm)

CLO - initial total cladding length. (cm)

PO - average external pressure on cladding. (kg/cm^2)

22) Subroutine EDTINP (RVB, RCB, REB, RUB, RCI, RCO, QSFF, QSFP, OTOM, CPU, RHOP, F, XCPI, XCPO, FLZ, CLZ, FL, CL, PTOP, PO)

Input variables are printed to the printer.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, CAPSUL, PLTOPT, IOFIL

Variables required:

RVB, RCB, REB, RUB, RCI, RCO, QSFF, QSFP, OTOM, CPU, RHOP, F, XCPI, XCPO, FLZ, CLZ - refer to the Symbol Dictionary.

FL - total fuel length. (cm)

CL - total cladding length. (cm)

PTOP - axial force on cladding. (kg)

PO - average external pressure on cladding. (kg/cm^2)

23) Subroutine EDTMTX (EPR, EPC, EPZ, SIGR, SIGC, SIGZ, TPR, TPC, TPZ, TSR, TSC, TSZ, RDIS, CRIF, RCP, TCP, TBCP, ATBCP, RCL, TCL, TBCL, ATBCL, YNG, POS, SHR, BM, RF, TF, TBF, ATBF, PRS, EHP, TPRF, TPRC, TPCF, TPCC, TPZF, TPZC, PRES, KFRIC, DPR, DPC, DPZ, DSR, DSC, DSZ, POSB, SHRB, SWG, SWS, N, RPM, TPM, GS, CPU, LM, LMR, LMC, LMZ, RCOO)

After the stress-strain matrix is solved, the stress, strain, displacement, and detailed informations are printed out.

Called from : ACTIVE

Functions used : TTIMER (Library)

Commons required : ALWAYS, XFRIC, CAPSUL, COMPLT, IOFIL

Variables required:

EPR, EPC, EPZ, SIGR, SIGC, SIGZ, TPR, TPC, TPZ, TSR, TSC, TSZ, RDIS, CRIF, RCP, TCP, TBCP, ATBCP, RCL, TCL, TBCL, ATBCL, BM, TBF, ATBF, PRS, EHP, KFRIC, SWG, SWS, GS, CPU, LM, LMR, LMC, LMZ, RCOO - refer to the Symbol Dictionary.

YNG - Young's modulus. (kg/cm^2)

POS - Poisson's ratio. (-)

SHR - shear modulus. (kg/cm^2)

RF - boundary radius of each fuel region. (cm)

TF - temperature at the boundary of each fuel region. (°C)

PRES - contact pressure at the boundary of each radial region. (kg/cm^2)

DPR - radial creep strain increment during the time interval. (-)

DPC - circumferential creep strain increment. (-)

DPZ - axial creep strain increment. (-)

POSB - average Poisson's ratio. (-)

SHRB - average shear modulus. (kg/cm^2)

RPM - same as RFRPM } in the Symbol Dictionary.

TPM - same as TFRPM

N - axial section number.

24) Subroutine EQUISI (SR, SC, SZ, ES, PRS, N)

This subroutine is provided to calculate the equivalent stress, and the hydro-static pressure in the N-th axial section and all the radial regions, using the corresponding stresses.

Called from : ACTIVE
Functions used : SQRT (Library)
Commons required : ALWAYS
Variables required:

- SR - the radial stress component. (kg/cm^2)
- SC - the circumferential stress component. (kg/cm^2)
- SZ - the axial stress component. (kg/cm^2)
- N - the axial section number.
- NTR - the total number of radial regions in the fuel and cladding.

Variables changed :

- ES - the equivalent stress. (kg/cm^2)
- PRS - the hydro-static pressure. (kg/cm^2)

25) Subroutine FLCOND (XK, XK1, ZH, FP, FP95, T)

The thermal conductivity of each fuel region is calculated in this subroutine.

Called from : FLTMP
Commons required : ALWAYS, CONST
Variables required:

- ZH - ρ/ρ_{th}, where ρ and ρ_{th} are the actual and theoretical densities of the fuel, respectively.
- FP - $(1-P)/(1+\beta P)$, when P=porosity, and $\beta \simeq 1$.
- FP95 - FP for P=0.05, that is FP95 = $(1-0.05)/(1+0.05 \cdot \beta)$
- T - temperature. (°C)
- NOK - option number for fuel conductivity formula.

Variables changed :

- XK - thermal conductivity of the fuel. (watt/cm.°C)
- XK1 - temperature differential of XK.

26) Subroutine FLTMP (POW, TCL, RCI, RUB, REB, RCB, RVB, RH, DXM, XM, R, T, TB, ATB, TBA, ATBA, FD, RPM, TPM, TBPM, ATBPM, FDPM, GDT, HG, F, OTOM, RHTHEO, DELT, RFO, IT, BU, RHOP, RB, EPR, EPC, EPZ, TPR, TPC, TPZ, TSR, TSC, TSZ, GS, SWG, SWS, EHP, WOLD, WNEW, XEPR, XEPC, XTPR, XTPC, XTPZ, XTSR, XTSC, XTSZ, XGS, XSWG, XSWS, XEHP, XSWS3, SWS3)

The temperature distribution and fuel restructuring are determined in the FLTMP.

The columnar grain region is equally divided into NCR meshes. The equiaxed grain region is equally divided into NER meshes, and for the undisturbed region, into NUR meshes.

The radial temperature distribution in the fuel is determined at the boundaries of NP regions of equal mass through integration of the one-dimensional heat-conduction equation. (See chapter 3.1-5)) The temperatures at the boundaries of NCR, NER, or NUR meshes are interpolated using those at the boundaries of NP regions of equal mass.

The temperature gradients at the interfaces of the NP regions of equal mass are stored for use in the subsequent calculation of plutonium migration.

After the boundary temperatures are determined, the average temperatures and thermal expansion of each region are computed.

The growth of the columnar grain region and the equiaxed grain region (i.e., fuel restructuring) are determined using the experimental formula. After the boundaries of the columnar and equiaxed regions have been determined, a final mass balance yields the radius of the central void

Some strains are modified considering the fuel restructuring.

Called from : ACTIVE
Subroutines called: COEF, FLCOND, OTMCAL
Functions used : ALOG, EXP (Library)

Variables required:

POW, TCL, RCI, RUB, REB, RCB, RVB, RH, XM, HG, F, BU, RHOP
- refer to the Symbol Dictionary.

RHTHEO- theoretical density. (g/cc) (same as RHOT in the Symbol Dictionary.

CM1, DHR, PI, BURH1, BURH2 - refer to the Common OTHERS.

IT - time step number.

DELT - time interval. (hr)

Variables changed :

DXM, FD, TBPM, ATBPM, FDPM, GDT, OTOM, RB, EPR, EPC, EPZ, TPR, TPC, TPZ, TSR, TSC, TSZ, GS, SWG, SWS, EHP, WOLD, WNEW, XEPR, XEPC, XTPR, XTPC, XTPZ, XTSR, XTSC, XTSZ, XGS, XSWG, XSWS, XEHP, XSWS3, SWS3 - refer to the Symbol Dictionary.

R-RFR
T-TFR
TB-TBF
ATB-ATBF
TBA-TBFA
ATBA-ATBFA
RPM-RFRPM
TPM-TFRPM
RFO-RFRO

in the Symbol Dictionary.

XML, CM - refer to the Common OTHERS.

27) Subroutine FLXCAL (POWAV, QSFF, ZFLX, FLNC, FLUX, DELT)

The FLXCAL computes the fast neutron flux and the fast neutron fluence.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS

Variables required:

POWAV - average linear power. (w/cm)

QSFF, ZFLX - refer to the Symbol Dictionary.

CFLX, SMQO, SMQ - refer to the Common OTHERS.

DELT - time interval. (hr)

Variables changed :

FLUX, FLNC - refer to the Symbol Dictionary.

28) Subroutine FRICHK (RUB, RCI, DISP, PRES, FRIC, CRIF, KFRIC, KONV, IFRIC, NS)

The fuel-clad gap is first assumed to be open. The frictional force is zero, and the interface pressure equals to the plenum pressure. If the computed displacements of fuel and cladding indicate that the gap is closed, then, frictional force is checked. If the computed frictional force is larger than can be justified by the contact pressure and the specified static coefficient of friction, slip occurs at the fuel-clad interface. Otherwise, stick condition is assumed.

If the assumed condition confilicts with the computed one, the matrix is solved again with the modified condition,

Called from : ACTIVE
Commons required : ALWAYS, OTHERS, XFRIC
Variables required:

RUB, RCI - refer to the Symbol Dictionary.
DISP - displacement computed through matrix solving. (cm)
PRES - contact pressure at the boundary of each region, computed through matrix solving. (kg/cm^2)
FRIC - frictional force computed through matrix solving.(kg/cm)
NS - axial section number.
FST, GAP - refer to the Common XFRIC.

Variables changed :

CRIF, KFRIC - refer to the Symbol Dictionary.
KONV - convergence indicator.
IFRIC - fuel-clad gap indicator.

29) Subroutine GAPCON (RCI, RUB, HG)

The fuel-clad gap heat-transfer coefficient is computed as a function of gap size. This value is used in the calculation of fuel outer temperature.

Called from : ACTIVE
Function used : EXP (Library)
Common required : ALWAYS

Variables required:

RCI, RUB - refer to the Symbol Dictionary.

Variables changed :

HG - gap heat-transfer coefficient. (cal/hr·cm^2·°F)

30) Subroutine GASOUT (RF, FLZ, POW, FD, TBF, VOL, EPR, EPC, EPZ, ATBF, SWS, GR, GS, GRT, DELT)

The subroutine GASOUT calculates the fractional gas-release rates, and gas-remaining rates for each of the fuel region. The release rate for any region is increased if the swelling of the region rises above a specified limit. (25% volume expansion)

The gas-release rate is multiplied by the amount of gas and by the time interval to determine the amount of gas released from, or that remaining in the region during the time step. The released gas is added to the previous total of released fission gas.

Called from : ACTIVE

Functions used : EXP (Library)

Commons required : ALWAYS, OTHERS, XGASO

Variables required:

POW, FD, TBF, VOL, EPR, EPC, EPZ, ATBF, SWS - refer to the Symbol Dictionary.

DELT - time interval. (hr)

AF, AN, CT, QR - refer to the Common XGASO.

A91 - refer to the Common OTHERS.

Variables changed :

GR - gas released. (mol)

GS - gas remaining. (mol)

GTR - total gas released. (mol)

31) Subroutine GOLDN (POWAV, QSF, TOUT, TIN, RCLO, POW, TCLO, RCPI, XCPO, AFNA, FL, HC, TNA, TCPO, TCPI, TOUTER, NSKIP, PO, FLZ)

The GOLDN computes the surface temperature at the outside of cladding. The coolant temperature at the middle of each axial section is determined by either Newton's interpolation method or linear interpolation between inlet and outlet temperature.

The linear power in each axial section is computed from the given average linear power for the fuel element. The outside temperature of the capsule (or cladding) is determined after the heat-transfer coefficient between the coolant and the fuel-element capsule (or cladding) is calculated.

Called from : ACTIVE
Functions used : ALOG, EXP (Library)
Commons required : ALWAYS, OTHERS, XGOLD, FBRATR, IOFIL
Variables required:

POWAV - average linear power for the fuel element. (w/cm)
QSF, TOUTER, FLZ - refer to the Symbol Dictionary.
TIN - coolant inlet temperature. (°C)
TOUT - coolant outlet temperature. (°C)
RCLO - outside radius of cladding. (cm)
RCPI - inside radius of capsule. (cm)
XCPO - outside radius of capsule. (cm)
AFNA - coolant flow area. (cm^2)
FL - fuel length. (cm)
NSKIP - =0, the coolant inlet, and outlet temperature is given (input).
=1, the cladding outer temperature is given (input).
PO - average external pressure on cladding. (kg/cm^2)
ASSO, ASS1, ANAO, ANA1, ANA2, CO, C1, C2 - refer to the Common XGOLD.

Variables changed :

POW, TCLO, HC, TNA, TCPO, TCPI - refer to the Symbol Dictionary.

32) Subroutine HEADER

The HEADER reads the title card until the *END card is encountered. The title card is printed soon after it is read before any other output is performed.

Called from : ACTIVE
Common required : ALWAYS, COMPLT, IOFIL

Variables changed :

TITLE - title

TITL - title for plotting.

33) Subroutine HKNS (TBCL, TSR, TSC, TSZ, FLX, DELT)

The HKNS uses the two formulae for the cladding swelling according to the option NOS to determine the cladding swelling.

Called from : ACTIVE

Functions used : ALOG, EXP (Library)

Commons required : ALWAYS, XHKNS, FBRATR

Variables required:

TBCL - average temperature of each region of the cladding. (°C)

FLX - neutron flux. (n/sec·cm^2)

DELT - time interval. (hr)

CS, BO, CO, C1, C2, C3 - refer to the Common XHKNS.

JFA - refer to the Common FBRATR.

Variables changed :

TSR, TSC, TSZ - refer to the Symbol Dictionary.

34) Subroutine INPUT (EPZ, TPR, TPC, TPZ, SIGR, SIGC, SIGZ, ADISP, RDISP, RVB, RCB, REB, RUB, RCI, RCO, QSFF, QSFP, F, EPR, EPC, TSR, TSC, TSZ, CPU, RHOP, OTOM, FL, CL, PTOP, PO, EPSIN, CLZO)

The subroutine INPUT reads the input cards. (See Chapter 6.3)

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, CAPSUL, PLTOPT, IOFIL

Variables changed :

Refer to the Chapter 6.

35) Subroutine IOSET (IO)

The IOSET determines the file reference number for use of temporary output. Presently, these variables are not used.

Called from : ACTIVE

Commons required : ALWAYS

Variables changed :

IO - output file reference number.

36) Subroutine ITERAT (PRS, VOL, TBF, ATBF, SWG, EHP, RH, RF, GS, SWS, BM, EPR, EPC, EPZ, SWS3, SIGR, SIGC, SIGZ, RHTHEO, DELT, N, RHOP, BU, TSR, TSC, TSZ, XEHP, XSWG, PLPR, CVPR)

The ITERAT computes the fuel swelling due to gaseous fission products and the hot-pressing. The gaseous fission products in the undisturbed region are assumed to contribute to swelling in that region in a manner similar to solid fission products.

Called from : ACTIVE

Functions used : EXP (Library)

Commons required : ALWAYS, OTHERS, XITER

Variables required:

PRS, VOL, TBF, GS, SWS3, RHOP, BU - refer to the Symbol Dictionary.

RHTHEO - theoretical density. (g/cc)

CVPR - central void pressure. (kg/cm^2)

DELT - time interval. (hr)

N - axial section number.

C,QRK - refer to the Common XITER.

PG, BURH1, BURH2 - refer to the Common OTHERS

Variables changed :

SWG, EHP, SWS, TSR, TSC, TSZ - refer to the Symbol Dictionary.

XEHP, XSWG - temporary use.

37) Subroutine LENGTH (FL, FLN, CL, CLN, FLZO, FLZ, CLZO, CLZ, EPZ)

The axial length of the fuel and the cladding is computed.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS

Variables required:

FLZO, CLZO, EPZ - refer to the Symbol Dictionary.

Variables changed :

FLZ,CLZ - refer to the Symbol Dictionary.

FL,FLN - total fuel length. (cm)

CL,CLN - total cladding length. (cm)

38) Subroutine MATRX (IA, JA, A, E, P, DEQP, EQSIG, RF, RC, G, PB, GB, FLZ, KFRIC, MOUNT, IFRIC, NS, RB)

The MATRX determines the vector A (vector containing the values of the elements of the given matrix of interest in row-wise order), the vector IA (vector containing the pointers to the first element in each row of the matrix in row-wise order), and the JA (vector containing the column indices of the matrix in row-wise order) for use in the subsequent matrix solving by SL-MATH. (See appendix A·2)

When the first call to the subroutine MATRX during the time step, the initial assumption of the boundary condition for the fuel-cladding interface is that the gap is open.

After the matrix is solved by SL-MATH, and if the results represent that the gap is closed, the parts of A, IA, and JA related to the boundary condition are modified at the second or the third call to the MATRX.

Called from : ACTIVE

Functions used : ALOG (Library)

Commons required : ALWAYS, OTHERS

Variables required:

E, P, DEQP, EQSIG, G, PB, GB, FLZ, KFRIC, RB - refer to the Symbol Dictionary.

RF - boundary radius of each fuel region.

RC - boundary radius of each cladding region.

MOUNT - count number that the MATRX is called.

IFRIC - indication of the fuel-cladding interface condition.

=0, the gap is open.

=1, stick condition.

=2, slip condition.

NS - axial section number.

Variables changed :

IA, JA, A - refer to the SL-MATH (See chapter A.2)

39) Subroutine MATRXP (IA, JA, A, E, P, DEQP, EQSIG, RC, G, PB, GB, CLZ, KFRIC, NS)

The subroutine MATRXP is used for the same purpose as the MATRX except that only the plenum section (i.e., only the cladding) is treated in this subroutine.

Called from : ACTIVE
Functions used : ALOG (Library)
Commons required : ALWAYS, OTHERS
Variables required:

E, P, DEQP, EQSIG, G, PB, GB, CLZ, KFRIC - refer to the Symbol Dictionary.

RC - boundary radius of each cladding region.

NS - axial section number.

Variables changed :

IA, JA, A - Same as those used in the MATRX, and refer to the SL-MATH.(See Chapter A.2)

40) Subroutine OTMCAL (OTOM, N)

The OTMCAL computes the burnup dependent O/M ratio. Presently, the values of OTOM are equal to the initial input value.

Called from : FLTMP
Commons required : ALWAYS

41) Subroutine PLENP (RVB, RCI, TFR, FLZ, GRT, FL, CL, TOUT, PLPR, CVPR, RUB, CLZ)

This subroutine is used to determine the central void pressure and the plenum pressure. The temperature of the gas in the central void and that of the gas in the plenum are volume averaged assuming the perfect gas law.

Called from : ACTIVE
Commons required : ALWAYS, OTHERS

Variables requied:

RVB, RCI, TFR, FLZ, RUB, CLZ - refer to the Symbol Dictionary.

GRT - total gas released in the plenum. (mol)

FL - total length of the fuel. (cm)

CL - total length of the cladding. (cm)

TOUT - coolant outlet temperature. (°C)

RK - ideal gas constant. (kg·cm/mol·°K)

Variables changed :

PLPR - plenum pressure. (kg/cm^2)

CVPR - central void pressure. (kg/cm^2)

42) Subroutine PLOT1 (XPOW, XRL, XTL, XT, XTLMX, XRLR, XTLR, XCPGP, XCPU, XBU, POW, RFRPM, TFRPM, RCB, REB, RUB, RCI, CPU, BU, PFR, TFR, IT, N)

This subroutine is used to store the information in the fuel region in preparation for plotting. The Unit conversion is also performed.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, XFRIC, PLTOPT

Variables required:

POW - average power. (w/cm)

RFRPM - radii of concentric cylindrical shells of equal mass into which the fuel region is subdivided. (cm)

TFRPM - temperature at the corresponding mesh point of RFRPM. (°C)

RCB - radius of columnar and equiaxed zone interface. (cm)

REB - radius of equiaxed and undisturbed interface. (cm)

RUB - outer radius of the fuel. (cm)

RCI - inner radius of the cladding. (cm)

CPU - plutonium concentration in each concentric cylindrical shell.

BU - burn-up. (atomic-percent)

RFR - radii of concentric cylindrical shells of equal mesh interval into which the fuel region is subdivided. But each region, i.e., columnar, equiaxed, undisturbed region, is subdivided independently. (cm)

TFR - temperature at the corresponding mesh point of RFR. (°C)

IT - counter for plotting.

N - axial section number to be plotted.

GAP - gap between fuel and cladding. (cm)

NCR - number of meshes into which the columnar region is subdivided.

NER - number of meshes into which the equiaxed region is subdivided.

NP1 - number of radial mesh points, including inner and outer boundaries, for plutonium migration.

Variables changed : all are used for plotting.

XPOW - same as POW at the specified axial section N. (w/cm)

XRL - same as RFRPM at the specified axial section N. (mm)

XTL - same as TFRPM at the specified axial section N. (°C)

XT - surface temperature of the fuel at the specified axial section N. (°C)

XTLMX - maximum temperature in the fuel at the specified axial section N. (°C)

XRLR - same as RCB at the specified axial section N. (mm)

XTLR(IT,1) - temperature at the columnar and equiaxed boundary at the specified axial section N. (°C)

XTLR(IT,2) - temperature at the equiaxed and undisturbed boundary at the specified axial section N. (°C)

XCPGP - same as GAP. (mm)

XCPU - same as CPU at the specified axial section N.

XBU - same as BU at the specified axial section N. (atomic-percent)

43) Subroutine PLOT2 (XSIGC, SIGC, VOL, IT, N)

The subroutine PLOT2 is used to store the average circumferential (hoop) stress in the cladding for plotting.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, PLTOPT

Variables required:

SIGC - circumferential stress component in the fuel and the cladding. (kg/cm^2)

VOL - volume of the each region in the fuel and the cladding. (cm^3)

IT - counter for plotting.

N - axial section number to be plotted.

NCL - total number of radial meshes in the cladding.

NFR - total number of radial meshes in the fuel.

Variables changed : this is used for plotting.

XSIGC - volume averaged hoop stress in the cladding at the specified axial section N. (kg/mm^2)

44) Subroutine PLOT3 (XCSTN, XTSC, XTPC, RCO, RCOO, TSC, TPC, VOL, RDIS, IT, N)

This is used to store the strain in the cladding for plotting.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, PLTOPT

Variables required:

RCO - outer radius of the cladding. (cm)

RCOO - initial outer radius of the cladding. (cm)

TSC - circumferential swelling strain component in the fuel and the cladding.

TPC - circumferential creep strain.

VOL - volume. (cm^3)

RDIS - displacements of the radial region boundaries. (cm)

IT - counter for plotting.

N - axial section number to be plotted.

Variables changed : all are used for plotting.

XCSTN - $\Delta D/D$ of the cladding. (%)

XTSC - volume averaged circumferential swelling strain in the cladding. (%)

XTPC - volume averaged circumferential creep strain the cladding. (%)

45) Subroutine PLOT4 (RFRPM, RCB, REB, LMR, LMC, LMZ, IT, N)

The PLOT4 is used to plot the fuel cracking which would occur in any of the three principal directions at the specified axial section. The cracks in the radial direction are drawn, if exist, as concentric circles in the circle representing the fuel pin. The concentric circle representing crack and the one representing region boundary are discriminated from each other by using a different colored pen.

The cracks in the circumferential direction are drawn, if exist, as arrowheads. These are also discriminated by using a different colored pen. The cracks in the axial direction are drawn, if exist, as lines in the rectangle which represents the fuel pin. The color is different from that of some boundaries.

Called from : ACTIVE
Functions used : SIN, COS (Library)
Subroutines called: PLOT, NEWPEN, SYMBOL, NUMBER, CIRCLE, AROHD, DASHPT (Library) AXIAL
Commons required : ALWAYS, OTHERS, PLTOPT, DAY
Variables required:

RFRPM - radii of concentric cylindrical shells of equal mass into which the fuel region is subdivided for plutonium migration. (cm)
RCB - radius of columnar and equiaxed region interface.(cm)
REB - radius of equiaxed and undisturbed region interface.(cm)
LMR - number of cracks in the radial direction.
LMC - number of cracks in the circumferential direction.
LMZ - number of cracks in the axial direction.
IT - time step counter.
N - axial section number to be plotted.

46) Subroutine PLOT5 (XTIME, XPLPR, XYP, ITP, IT, ITN, TIM, PLPR)

This subroutine is used to store the time step number and the real time to be plotted. The pressure in the plenum section and the gas release rate are also stored.

Called from : ACTIVE
Commons required : ALWAYS, OTHERS, PLTOPT, COMPLT
Variables required:

IT - counter for plotting.
ITN - time step counter.
TIM - time at the ITN time step. (hr)

PLPR - pressure in the plenum section. (kg/cm^2)

YP - gas release rate from the fuel pin (from the whole axial sections). (%)

Variables changed : all are used for plotting.

ITP - same as IT.

XTIME - same as TIM. (hr)

XPLPR - same as PLPR. (kg/cm^2)

XYP - same as YP. (%)

47) Subroutine PLTOR (XPOW, XRL, XTL, XT, XTLMX, XRLR, XTLR, XCPGAP, XCPU, XBU, XSIGC, XCSTN, XTSC, XTPC, XTIME, XPLPR, XYP, ITP, IT, ITN)

This is the plotting routine except for that of the fuel cracking.

Called from : ACTIVE

Subroutines called: PLOT, NEWPEN, SYMBOL, NUMBER (Library)
SUBTIT, AXIAL, AXISD, AXIST

Commons required : ALWAYS, OTHERS, DAY, COMPLT, PLTOPT

Variables required: all are used for plotting.

XPOW - average linear power. (w/cm)

XRL - radii of region boundaries in the fuel (mm)

XTL - fuel temperature at the corresponding point of XRL. (°C)

XT - fuel pellet surface temperature. (°C)

XTLMX - maximum temperature in the fuel. (°C)

XRLR(1) - radius of columnar and equiaxed region boundary. (mm)

XRLR(2) - radius of equiaxed and undistrubed region boundary. (mm)

XTLR(1) - temperature at the point XRLR(1). (°C)

XTLR(2) - temperature at the point XRLR(2). (°C)

XCPGAP- clad-pellet gap. (mm)

XCPU - plutonium concentration.

XBU - burn-up (a/o)

XSIGC - clad average hoop stress. (kg/mm^2)

XCSTN - clad strain-$\Delta D/D$. (%)

XTSC - average circumferential swelling strain in the cladding. (%)

XTPC - average circumfetential creep strain in the cladding. (%)

XTIME - time. (hr)
XPLPR - plenum pressure. (kg/cm^2)
XYP - gas release rate. (%)
ITP - time step number selected for plotting.
IT - counter for plotting.
ITN - time step counter.

48) Subroutine PLUTO (CP, RVB, RCB, REB, RP, TP, GDT, XM, RH, EPZ, DELT, D)

The PLUTO computes the radial migration of plutonium in the fuel by thermal and chemical diffusion.

The plutonium flux between each concentric region of equal mass is computed, and the result is used to compute the plutonium concentration in each region after the time interval.

Called from : ACTIVE
Functions used : EXP (Library)
Commons required : ALWAYS, OTHERS, IOFIL
Variables required:

RVB, RCB, REB, GDT, XM, RH, EPZ - refer to the Symbol Dictionary.
RP - radii of the boundaries of the NP equal mass radial regions. (cm)
TP - temperature at the boundary of each equal mass region. (°C)
DELT - time interval. (hr)

Variables changed :

CP - plutonium concentration in each equal mass region. (-)
D - plutonium concentration change during the time interval. (1/sec)

49) Subroutine POWCUT (TIM, TIP, TIN, PTIN, TOUT, PTOUT, REPOW, PPOWA, PTOUR, TOUTER, DELTOU, DELTIM, DELIN, DELOUT, DELPOW, KUT, NSK, KUTCNT)

The change of the average linear power which is given from the input card is cut up according to the option KUT. In keeping with the power cut, the time and the coolant inlet, outlet temperatures are also cut up.

Called from : ACTIVE

Commons required : ALWAYS

Variables required:

TIP - time of the previous time step. (hr)

PTIN - coolant inlet temperature of the previous time step. (°C)

PTOUT - coolant outlet temperature of the previous time step. (°C)

PPOWA - average power of the previous time step. (w/cm)

PTOUR - cladding outside temperature of the previous time step. (°C)

TOUTER- cladding outside temperature. (°C)

KUT - number to be cut up.

NSK - =0, coolant inlet, outlet temperature is given.
=1, cladding outside temperature is given.

KUTCNT- counter used for cutting.

Variables changed :

TIM - time after cut up. (hr)

TIN - coolant inlet temperature after cut up. (°C)

TOUT - coolant outlet temperature after cut up. (°C)

REPOW - average power. (w/cm)

DELTOU- cladding outer temperature increment. (°C)

DELTIM- time increment. (hr)

DELTIN- coolant inlet temperature increment. (°C)

DELOUT- coolant outlet temperature increment. (°C)

DELPOW- power increment. (w/cm)

50) Subroutine PROPTY (CPU, RHOP, RHOT)

This subroutine prepares for the numerical constants associated with the properties.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, CONST, XCRAK, XCREC, XCREF, XFRIC, XGASO, XGOLD, XHKNS, XITER, XSWEL, XXLAM, FBRATR

Variables required:

CPU, RHOP - refer to the Symbol Dictionary.

Variables changed :

RHOT - refer to the Symbol Dictionary.

Variables included in the labelled common are referred to the Common Dictionary.

51) Subroutine RHOMOD (RH, RHOP, EPZ, RVB, RCB, REB, RUB, BU, XM, RHTHEO, RDIS, DXM, RVBO, RCBO, REBO, RUBO, RCI, RCO, XMINT, EPC, RDIST, EPR, XEPR, XEPC, R, WOLD, WNEW, IT, DELT)

The RHOMOD computes the boundary radius of each region, and the burnup dependent density of each radial region.

The minimum mass per unit length in columnar region is bounded by 10^{-3} (g/cm) to avoid numerical problems.

Called from : ACTIVE

Functions used : SQRT (Library)

Commons required : ALWAYS, OTHERS, IOFIL

Variables required:

RHOP, EPZ, BU, RDIS, DXM, RVBO, RCBO, REBO, RUBO, XMINT, EPC

EPR - refer to the Symbol Dictionary.

RHTHEO- theoretical density. (g/cc)

DELT - time interval. (hr)

BURH1, BURH2 - refer to the Common OTHERS.

Variables changed :

RH, RVB, RCB, REB, RUB, XM, RCI, RCO, RDIST - refer to the Symbol Dictionary.

52) Subroutine SUBTIT (TITLE)

This subroutine draws the title above each plotted picture.

Called from : PLTOR

Variables required:

TITLE - title to be plotted.

53) Subroutine SWELL (POW, SWS, FD, SWS3, DELT)

The SWELL computes the fuel swelling due to solid fission products by specifying the net solid volume created per fission and by multiplying by the number of fissions that have occurred in each fuel region.

Called from : ACTIVE
Commons required : ALWAYS, OTHERS, XSWEL
Variables required:
POW,FD- refer to the Symbol Dictionary.
DELT - time interval. (hr)
Variables changed :
SWS, SWS3 - refer to the Symbol Dictionary.

54) Subroutine TIMCUT (CX, DELTO, EPSPOW)

This subroutine decides the criterion for the time step interval itself and for the average power. The current time step interval must not exceed the value of CX times DELTO. The absolute value of the average power decrement or the increment must not exceed EPSPOW.

In the code ACTIVE-II, CX, DELTO, and EPSPOW are fixed as follows.

CX = 2.0
DELTO = 240.0 (hrs)
EPSPOW = 50.0 (w/cm)

Called from : ACTIVE

55) Subroutine UPDATE (SIGR, SIGC, SIGZ, SIGRO, SIGCO, SIGZO, EPR, EPC, EPZF, EPZC, EPRO, EPCO, EPZO, TPR, TPC, TPZ, TPRO, TPCO, TPZO, DSR, DSC, DSZ, TSRO, TSCO, TSZO, DISP, RDIS, N, EPSEQ)

After the stress-strain matrix equations are solved by using SL-MATH, the results are stored into the corresponding variables in the subroutine.

Called from : ACTIVE

Functions used : SQRT (Library)

Commons required : ALWAYS

Variables required:

SIGR - radial stress component. (kg/cm^2)

SIGC - circumferential stress component. (kg/cm^2)

SIGZ - axial stress component. (kg/cm^2)

EPR - total radial strain. (-)

EPC - total circumferential strain. (-)

TPR - radial creep strain. (-)

TPC - circumferential creep strain. (-)

TPZ - axial creep strain. (-)

EPZF - total axial strain of the fuel. (-)

EPZC - total axial strain of the cladding. (-)

DISP - displacement. (cm)

All variables listed above are the results through matrix solving.

N - axial section number.

Variables changed :

SIGRO-SIGR
SIGCO-SIGC
SIGZO-SIGZ
EPRO-EPR
EPCO-EPC
EPZO-EPZ

} in the Symbol Dictionary.

EPSEZ, RDIS - refer to the Symbol Dictionary.

56) Subroutine UPDATP (SIGR, SIGC, SIGZ, SIGRO, SIGCO, SIGZO, EPR, EPC, EPZC, EPRO, EPCO, EPZO, TPR, TPC, TPZ, TPRO, TPCO, TPZO, TSR, TSC, TSZ, TSRO, TSCO, TSZO, DISP, RDIS, N)

The UPDATP is used for the same purpose as the subroutine UPDATE except that the UPDATP treats only the plenum section.

Called from : ACTIVE

Commons required : ALWAYS

Variables required:

SIGR, SIGC, SIGZ, EPR, EPC, TPR, TPC, TPZ, EPZC, DIST
- same as mentioned in the UPDATE.

N - axial section number.

Variables changed :

SIGRO, SIGCO, SIGZO, EPRO, EPCO, EPZO, RDIS
- same as mentioned in the UPDATE.

57) Subroutine UPDTE (RVB, RCB, REB, RUB, RCI, RCO, BU, RDIST, DXM, FLNC, WOLD, YRVB, YRCB, YREB, YRUB, YRCI, YRCO, YBU, YRDIST, YDXM, YFLNC, YWOLD, ISW)

The subroutine UPDTE is used to store or retrieve the values of several variables to maintain the system. The subroutine ACTIVE decides the time step interval, using the given time histories. If the criterion is not satisfied, a computational control is returned to the starting point of time step loop after retrieving the values of the variables changed between the starting point and the criterion checking.

Called from : ACTIVE

Commons required : ALWAYS

Variables required:

ISW - a switch for controlling store and retrieval. If ISW=1, store procedure is selected. If ISW=2, retrieval procedure is selected.

NN - the total number of axial sections, including a plenum section.

NFR - the total number of radial regions in the fuel.

Variables changed :

RVB - central void radius. (cm)

RCB - radius of columnar and equiaxed region interface.(cm)

REB - radius of equiaxed and undisturbed region interface.(cm)

RUB - outer radius of the fuel. (cm)

RCI - inner radius of the cladding. (cm)

RCO - outer radius of the cladding. (cm)

BU - burn-up (atomic-percent).

RDIST(N,1) - not used.

(N,2) - accumulated migration of columnar and equiaxed region interface caused by fuel restructuring. (cm)

(N,3) - accumulated migration of equiaxed and undisturbed region interface caused by fuel restructuring. (cm)

(N,4) - mass in unit length of the current time step. (g/cm)

Where N means the axial section number.

DXM - migration velocity (rate of restructuring) during the present time interval. (cm/sec)

FLNC - fluence. (n/cm^2)

WOLD - mass in unit length. (g/cm)

YRVB, YRCB, YREB, YRUB, YRCI, YRCO, YBU, and YRDIST have the same contents as those of corresponding variables without the initial Y, except that the variables with the initial Y have the starting value of the current time step.

58) Subroutine VOLUME (VOL, RFR, RCL, FLZ, CLZ)

The VOLUME computes the volume of each concentric cylinder for each axial section.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS

Variables required:

RFR, RCL, FLZ, CLZ - refer to the Symbol Dictionary.

Variables changed :

VOL - refer to the Symbol Dictionary.

59) Subroutine XLAME (TBC, TBF, E, P, G, RH, LM, BM, PB, GB, VOL, RHTHEO, N)

The XLAME computes the Poisson's ratio, Young's modulus, shear modulus, and bulk modulus of each region for the fuel and the cladding. The Poisson's ratio and the Young's modulus are modified in an attempt to represent the cracked material.

Called from : ACTIVE

Commons required : ALWAYS, OTHERS, XXLAM, FBRATR, IOFIL

Variables required:

TBF, RH, LM, VOL - refer to the Symbol Dictionary.

TBC - average temperature of each region in the cladding. (°C)

RHTHEO- theoretical density. (g/cc)

N - axial section number.

EO, EP, TLG, ONU, ONUP, A1, A2, A3, A4 - refer to the Common XXLAM.

Variables changed :

E, P. G, BM, PB, GB - refer to the Symbol Dictionary.

APPENDIX B: COMMON DICTIONARY

Appendix B: Common Dictionary

Some options, variables related to the dimensional size, variables which have fixed dimensions and some constants related to material properties are included in the following labelled commons.

B.1 Common list

Common Name	Explanation
ALWAYS	Variables related to the dimensional size and some options are included.
OTHERS	Some constants and variables having fixed dimensions are included.
CONST	Coefficients of thermal conductivity and of thermal expansion and of NEWTON-COTES-6 are included.
CAPSUL	Variables related to the capsule are included.
IOFIL	Definition of reference numbers of input/output files.
FBRATR	Reactor model (FBR, ATR, BWR, PWR) is specified.
DIM	Input sizes of variable dimensions are included.
PLTOPT	Options related to plotting are included.
DAY	Date is included.
COMPLT	Variables used for plotting are included.
XCRAK	Constants for fuel cracking are included.
XCREC	Constants for creep in the cladding are included.
XCREF	Constants for creep in the fuel are included.
XFRIC	Constants of friction are included.
XGASO	Constants for gas release are included.
XGOLD	Constants for coolant properties are included.
XHKNS	Constants for cladding swelling are included.
XITER	Constants for hot-pressing calculation are included.

Common Name	Explanation
XSWEL	Constant for solid volume per fission is included.
XXLAM	Constants for Poisson's ratio and Young's modulus are included.

Details are described in the section B.3.

B.2 Subroutines in which the commons are required.

The circles in the following table represent the correspondence between subroutines and commons.

Common / Sub-routine	ALWAYS	CAPSUL	COMPLT	CONST	DAY	DIM	FBRATR	IOFIL	OTHERS	PLTOPT	XCRAK	XCREC	XCREF	XFRIC	XGASO	XGOLD	XHKNS	XITER	XSWEL	XXLAM
MAIN																				
ACTIVE	○	○		○	○	○	○	○	○	○										
AREA																				
ATBFUN									○											
AXIAL																				
AXISD																				
AXIST																				
BCALC	○								○											
BCALCP	○								○											
BEGIN	○	○		○					○											
BURN	○								○											
CHECK	○																			
CHKSPD																				
CLEAR																				
CLTMP	○			○					○											
COEF	○			○			○													
CORTMP	○							○												
CRACK	○										○									
CREEC	○						○		○			○								
CREEF	○								○				○							
EDTGRS	○	○	○					○	○											
EDTINP	○	○						○	○	○										

Common / Sub-routine	ALWAYS	CAPSUL	COMPLT	CONST	DAY	DIM	FBRATR	IOFIL	OTHERS	PLTOPT	XCRAK	XCREC	XCREF	XFRIC	XGASO	XGOLD	XHKNS	XITER	XSWEL	XXLAM
EDTMTX	○	○	○					○						○						
FLCOND	○			○																
FLTMP	○			○			○	○	○											
FLXCAL	○								○											
FRICHK	○								○					○						
GAPCON	○																			
GASOUT	○								○						○					
GOLDN	○						○	○	○							○				
HEADER	○		○					○												
NKNS	○						○										○			
INPUT	○	○						○	○	○										
IOSET	○																			
ITERAT	○								○									○		
LENGTH	○								○											
MATRX	○							○	○					○						
MATRXP	○								○											
OTMCAL	○																			
PLENP	○								○											
PLOT1	○								○	○				○						
PLOT2	○								○	○										
PLOT3	○								○	○										
PLOT4	○				○				○	○										
PLOT5	○		○						○	○										
PLTOR	○		○		○				○	○										
PLUTO	○							○	○											
POWCUT	○							○												
PROPTY	○			○			○		○		○	○	○	○	○	○	○	○	○	○
RHOMOD	○							○	○											
SUBTIT																				
SWELL	○								○										○	
TIMCUT																				
UPDATE	○																			
UPDATP	○																			
UPDTE	○																			
VOLUME	○								○											
XLAME	○						○	○	○											○

B.3 Definition of Variables in Common

* The column 'Subroutine' specifies the subroutine in which the value is set.

1) Common ALWAYS

Variable	Definition	Value	Subroutine*
NN	Total number of axial sections, including a plenum section	input	ACTIVE
NCR	Number of meshes into which the columnar grain region is subdivided.	input	ACTIVE
NER	Number of meshes into which the equiaxed grain region is subdivided.	input	ACTIVE
NUR	Number of meshes into which the undisturbed region is subdivided.	input	ACTIVE
NCL	Number of meshes into which the cladding is subdivided.	input	ACTIVE
NP	Number of regions, i.e., concentric cylindrical shells, of equal mass into which the fuel is subdivided for plutonium migration.	input	ACTIVE
NOK	Option number for fuel conductivity formula. (Refer to the section 5 Material Properties.)	input	ACTIVE
NOS	Option number for cladding swelling formula. (Refer to the section 5.)	input	ACTIVE
NOC	Option number for cladding creep formula. (Refer to the section 5.)	input	ACTIVE
NPLOT	Option for plotting. =1 for plotting. =0 for no plotting.	input	ACTIVE

Variable	Definition	Value	Subroutine*
NCRK	Option for cracking calculation. =1 for cracking calculation. =0 for no cracking calculation.	input	ACTIVE.
NTR	Total number of radial meshes in the fuel and cladding.	NTR=NCR+NER +NUR+NCL	ACTIVE
NFR	Total number of radial meshes in the fuel.	NFR=NCR+NER +NUR	ACTIVE
NTR1	NTR+1	NTR1=NTR+1	ACTIVE
NTR2	NTR+2	NTR2=NTR+2	ACTIVE
NNTR2	NN×(NTR+2)	NNTR2= NN×(NTR+2)	ACTIVE
NNF	Total number of axial sections, excluding the plenum section.	NNF=NN-1	ACTIVE
NCL1	Number of mesh points in the cladding.	NCL1=NCL+1	ACTIVE
NFR1	Number of mesh points in the fuel.	NFR1=NFR+1	ACTIVE
NP1	Number of mesh points for plutonium migration in the fuel.	NP1=NP+1	ACTIVE
IGEN	Number of columns (or rows) of the stress-strain matrix.	IGEN=12 +10×NTR	ACTIVE
IGEN1	IGEN+1	IGEN1=IGEN+1	ACTIVE
NONZR	Number of non-zero elements in the matrix.	NONZR=24+ 56×NTR	ACTIVE
IGENP	Number of columns (or rows) of the matrix for the plenum section.	IGENP=7 +10×NCL	ACTIVE
IGENP1	IGENP+1	IGENP1= IGENP+1	ACTIVE
NONZRP	Number of non-zero elements in the matrix for the plenum section	NONZRP=12+ 56×NCL	ACTIVE
N1C	See the column 'Value'	NCL	ACTIVE
N2C		2×NCL	ACTIVE
N3C		3×NCL	ACTIVE

Variable	Definition	Value	Subroutine*
N4C		4×NCL	ACTIVE
N5C		5×NCL	ACTIVE
N6C		6×NCL	ACTIVE
N7C		7×NCL	ACTIVE
N8C		8×NCL	ACTIVE
N9C		9×NCL	ACTIVE
N10C		10×NCL	ACTIVE
N11C		11×NCL	ACTIVE
N12C		12×NCL	ACTIVE
N13C		13×NCL	ACTIVE
N14C		14×NCL	ACTIVE
N15C	See the column 'Value'	15×NCL	ACTIVE
N1		NTR	ACTIVE
N2		2×NTR	ACTIVE
N3		3×NTR	ACTIVE
N4		4×NTR	ACTIVE
N5		5×NTR	ACTIVE
N6		6×NTR	ACTIVE
N7		7×NTR	ACTIVE
N8		8×NTR	ACTIVE
N9		9×NTR	ACTIVE
N10		10×NTR	ACTIVE
N11		11×NTR	ACTIVE
N12		12×NTR	ACTIVE
N13		13×NTR	ACTIVE
N14		14×NTR	ACTIVE
N15		15×NTR	ACTIVE

2) Common OTHERS

Variable	Definition	Value	Subroutine
PG(3)	PG(1)=pressure due to surface tension of fission-gas bubbles for columnar grain region. PG(2)=pressure due to surface tension on-fission-gas bubbles for equiaxed grain region. PG(3)=pressure due to surface tension on fission-gas bubbles for undisturbed region. These are converted into the unit of kg/cm^2 in the subroutine PROPTY.	Input (psi.) PG(1)=3000., PG(2)=6000., PG(3)=1000000. are recommended.	INPUT
CM1(2)	CM1(1)=Constant C1 for restructuring velocity of columnar grain region outer boundary. CM1(2)=Constant C1 for restructuring velocity of equiaxed grain region outer boundary. (See Eq. (6) and (7))	Input CM1(1)=1.92×10^{7} (for FBR) CM1(1)=1.92×10^{8} (for thermal reactor) CM1(2)=1.2×10^{4} are recommended.	INPUT
DHR(2)	DHR(1)=Constant C2 for restructuring velocity of columnar grain region outer boundary. DHR(2)=Constant C2 for restructuring velocity of equiaxed grain region outer boundary. (See Eq. (6) and (7))	Input DHR(1)=64400. DHR(2)=44400. are the recommendations.	INPUT
ALFO(3)	Thermal coefficients of expansion for the columnar region (ALFO(1)), the equiaxed region (ALFO(2)), and the undisturbed region (ALFO(3)). (See section 5)	($°C^{-1}$)	BEGIN and PROPTY
ALF1(3)	Thermal coefficients of expansion for the columnar region (ALF1(1)), the equiaxed region (ALF1(2)) and the undisturbed region (ALF1(3)).	($°C^{-2}$)	BEGIN and PROPTY

Variable	Definition	Value	Subroutine
	(See section 5)		
XML(3)	XML(1)=0.0 XML(2)=mass in the colmnar region. XML(3)=mass in the columnar and the equiaxed region.	Calculated (g/cm)	FLTMP
CM(2)	CM(1)=mass increment of columnar region during the time interval. CM(2)=mass increment of equiaxed region during the time interval.	Calculated (g/sec)	FLTMP
PI	π	3.1415927	PROPTY
PI2	2π	6.2831853	PROPTY
DP	Constant	1.0	PROPTY
GRAV	Constant for unit conversion	10^{-3}	PROPTY
SMQ	Summation of axial power shapes multiplied by mass.	Calculation (g/cm)	BEGIN and BURN
SMQO	Initial value of SMQ	Calc.(g/cm)	BEGIN
RHOC	Density of cladding	Input (g/cm^3)	INPUT
BURH1	Modifier for burn-up dependent density in the columnar region.(a/o) (See Eq. (9))	Input Recommended value: 20.0	INPUT
BURH2	Modifier for burn-up dependent density in the equiaxed zone. (a/o) (See Eq. (11))	Input Recommended value: 20.0	INPUT
CFLX	Ratio of fast neutron flux to linear power. ($n/cm^2/sec$)/(watt/cm)	Input Default $= 3.17\times10^{12}$	PROPTY or INPUT
CPOW	Conversion ratio from input power to linear power.	1.0	ACTIVE
BUCNS	Constant for unit conversion into (atomic-percent) from (w/cm).	$3.1\times10^{10}\times3600/2.246\times10^{19}$	PROPTY
RK	Ideal gas constant	$8.31436\times1.0197\times10$ (kg·cm/mol·deg K)	PROPTY

Variable	Definition	Value	Subroutine
RKR	Ideal gas constant	8.31436/1.8/ 4.186 (cal/mol·deg R)	PROPTY
A91	Number of fissions/watt/hr.	1.116×10^{14}	PROPTY
GASI	Initial gas in the plenum section.	Input (moles)	INPUT
TCON	Constant of proportionality between the power and the outlet coolant temperature.	Input or calculated (degC/watt/cm)	INPUT or ACTIVE
PDBU	Decreasing rate of linear power due to burn-up.	Input (1/a/o)	INPUT
BUB	Average burn up of the fuel.	Calculated (atomic-percent)	BURN

3) Common CAPSUL

Variable	Definition	Value	Subroutine
RCPI	Inner radius of the capsule.	Inner (cm)	INPUT
RCPO	Outer radius of the capsule.	Input (cm)	INPUT
AFNA	Coolant flow area associated with the fuel element.	Input (cm^2)	INPUT

4) Common CONST

Variable	Definition	Value	Subroutine
FKA1 FKB1 FKC1	Numerical coefficients in the first (NOK=1) formula for fuel thermal conductivity. (See section 5)	Calculated	COEF
FKA2 FKB2 FKC2	Numerical coefficients in the second (NOK=2) formula for fuel conductivity. (See section 5)	Calculated	COEF
FKA3 FKB3 FKC3	Numerical coefficients in the third (NOK=3) formula for fuel conductivity. (See section 5)	Calculated	COEF
FA11	Coefficients of thermal expansion for the columnar region. (See section 5)	8.143×10^{-6}(FBR) 9.61×10^{-6}(ATR)	PROPTY
FA12		2.43×10^{-9}(FBR) 1.57×10^{-9}(ATR)	
FA21	Coefficients of thermal expansion for the equiaxed region. (See section 5)	8.143×10^{-6}(FBR) 9.61×10^{-6}(ATR)	PROPTY
FA22		2.43×10^{-9}(FBR) 1.57×10^{-9}(ATR)	
FA31	Coefficients of thermal expansion for the undisturbed region. (See section 5)	8.143×10^{-6}(FBR) 9.61×10^{-6}(ATR)	PROPTY
FA32		2.43×10^{-9}(FBR) 1.57×10^{-9}(ATR)	
CK1	Constants for cladding thermal conductivity. (See section 5)	1.130 (FBR) 0.121 (ATR)	PROPTY
CK2		1.36×10^{-4}(FBR) 1.24×10^{-4}(ATR)	
CA1	Constants for cladding thermal expansion. (See section 5)	17.0×10^{-6}(FBR) 5.67×10^{-6}(ATR)	PROPTY
CA2		4.25×10^{-9}(FBR) 1.7×10^{-9} (ATR)	
XEWTON(7)	Coefficients for integration by NEWTON-COTES-6	XEWTON(1)=41/140 (2)=216/140 (3)=27/140 (4)=272/140 (5)=27/140 (6)=216/140 (7)=41/140	BEGIN

5) Common COMPLT

Variable	Definition	Value	Subroutine
YP	Gas release rate	Calc. (%)	EDTGRS
DDOD	ΔD/D of the cladding. ΔD-diameter change. D-initial diameter.	Calc. (%)	EDTGRS
TITLE(20)	Title identifying the problem.	Input.	HEADER

6) Common DAY

Variable	Definition	Value	Subroutine
DAYDAY	Time of date.	Month/day/year	DATE

7) Common DIM

Variable	Definition	Value	Subroutine
IA	Dimension size in words (4 bytes) for all the variables excluding the matrix solving. IA must be set at least as large as the storage space required for the problem. (See section 6)	Input	MAIN
IAM	Dimension size in words (4 bytes) for the variables for matrix solving. (See section 6)	Input	MAIN

8) Common FBRATR

Variable	Definition	Value	Subroutine
JFA	Identification of the reactor model to be calculated.	Input =1 for FBR =2 for ATR =3 for PWR =4 for BWR	ACTIVE

9) Common IOFIL

Variable	Definition	Value	Subroutine
IO5	Logical card input device number.	5	ACTIVE
IO6	Logical printer output device number.	6	ACTIVE
IO7	Logical punch output device number.	7	ACTIVE

10) Common PLTOPT

Variable	Definition	Value	Subroutine
IPFUL	Axial section number for plotting the information associated with the fuel.	Input	INPUT
IPSTN	Axial section number at which the cladding hoop strain is plotted.	Input	INPUT
IPSTS	Axial section number at which the cladding hoop stress is plotted.	Input	INPUT
IPCRK	Axial section number at which the fuel cracking is plotted.	Input	INPUT
NBUF	Storage (words) required for plotting. The plotting informations are written to output logical unit (77) every NBUF steps.	Input If 0, NBUF is set to 50 in ACTIVE.	INPUT
IPCNT	Switch controlling the plotting.	Calc.	ACTIVE

11) Common XRACK

Variable	Definition	Value	Subroutine
THLC	Critical temperature at which cracks are assumed to heal. (See section 3)	1400 °C	PROPTY
HEAL1	Critical compressive stress by which cracks are assumed to be healed, when the temperature is less than, or equal to THLC. (See section 3)	-7000.0 (kg/cm^2)	PROPTY

Variable	Definition	Value	Subroutine
HEAL2	Critical compressive stress by which cracks are assumed to be healed, when the temperature is greater than THLC. (See section 3)	-70.0 (kg/cm^2)	PROPTY
SIGF0	Numerical coefficient in the expression for σ_f; fracture stress of pellet. (See section 3)	1000.0 (kg/cm^2)	PROPTY
SIGF1	Numerical coefficient in the expression for σ_f. (See section 3)	0.3 ($kg/cm^2/°K$)	PROPTY

12) Common XCREC

Variable	Definition		Value	Subroutine
CCA	A	in Eq. (17) for cladding creep rate. (See section 5)	2.7×10^{-11}	PROPTY
QCC	Q		95000(cal/mol)	
AE	Ae		10^{-8}	
BE	Be		2.2×10^{20}	
CE	Ce		4.3×10^{-30}	
FMC	m		7.0	
FNC	n		1.0	

13) Common XCREF

Variable	Definition		Value	Subroutine
CFA0	A_0	in Eq. (17) for creep rate for porous fuel. (See section 5)	1.83×10^{-3}	PROPTY
CFA1	A_1		1.07×10^{6}	
CFR	ϕ_0		3×10^{12} (fis/sec/cc)	
CFB	C		6×10^{-23}	
QCF	Q_0		14000 (cal/mol)	
QCF1	Q_1		11000 (cal/mol)	
FMF	m		4.5	
FNF	n		1.0	
GDF	d		10.0	

14) Common XFRIC

Variable	Definition	Value	Subroutine
FRICST	Static coefficient of friction. (See section 5)	0.8	PROPTY
FRICDY	Dynamic coefficient of friction. (See section 5)	0.6	PROPTY

15) Common XGASO

Variable	Definition	Value	Subroutine
AF	Number of gas atoms per fission.	0.246	PROPTY
AN	Avogadro's number.	6.02×10^{23}	PROPTY
CT	Constants for fission gas release rate. (See section 5)	4.48	PROPTY
QRG		8000	

16) Common XGOLD

Variable	Definition	Value	Subroutine
ASS0	Constants for thermal conductivity of capsule. (cal/hr/cm/°F)	75.9	PROPTY
ASS1		34.84×10^{-3}	
ANA0	Constants for thermal conductivity of sodium (cal/hr/cm/°F)	449	
ANA1		-0.155	
ANA2		17.29×10^{-6}	
AC0	Constants for thermal capacity of sodium. (cal/kg/°F)	192.053	
AC1		-440.061×10^{-4}	
AC2		1895.43×10^{-8}	

17) Common XHKNS

Variable	Definition	Value	Subroutine
HCS	Coefficients for cladding swelling. (See section 5)	10.0	PROPTY
CBO		9×10^{-35}	
HCO		4.028	
HC1		-3.712×10^{-2}	
HC2		1.0145×10^{-4}	
HC3		-7.879×10^{-8}	

18) Common XITER

Variable	Definition	Value	Subroutine
CI	Constants for hot-pressing calculation. (See section 5)	11.13×10^{6}	PROPTY
QRK		49600	

19) Common XSWEL

Variable	Definition	Value	Subroutine
SVPF	Net solid volume created per fission. (in^3)	0.7×10^{-24}	PROPTY

20) Common XXLAM

Variable	Definition	Value	Subroutine
EO	Constants in the formula for Young's modulus of fuel. (See section 5)	25.238×10^5	PROPTY
EP		-2.35	
TLG		22200	
ONU	Constants in the formula for Poisson's ratio at room temperature (25°C) (See section 5)	0.317	
ONUP		-0.46	
EA1	Constants in the formula for Young's modulus of cladding (See section 5)	2.0793×10^6 (FBR) $9.898\ 10^5$ (ATR)	
EA2		-8.8592×10^2 (FBR) -9.265×10^2 (ATR)	
EA3	Constants in the formula for Poisson's ratio of cladding. (See section 5)	0.3015 (FBR) 0.325 (ATR)	
EA4		0.85×10^{-4} (FBR) 2.14×10^{-4} (ATR)	

APPENDIX C: SYMBOL DICTIONARY

Appendix C: Symbol Dictionary

The variable names listed in the following symbol table are not always identical to those used in each subroutine. But these are related to each other through the column 'Subroutine' in the symbol table and through the description of the corresponding subroutine in section 'SUBROUTINE LIST'.

The parenthesis in the column 'Symbol' represents the axial (left-side) and the radial (right-side) dimensional sizes of the symbol. The variables in the parenthesis are listed in the common ALWAYS in the section 'Common Dictionary'.

The variable in the bracket in the column 'Symbol' represents the location of the allocated symbol in the variable 'A'.

The column 'Subroutine' specifies the subroutine in which the symbol is referred.

Symbol	Definition	Subroutine	Unit
ADISP(NN,NTR) A[NADIS]	Initial axial displacements (in each of NN axial sections and for each of NTR regions).	INPUT	cm
ATBCL(NN,NCL) A[NATBCL]	Cladding thermal expansion averaged in the radial region.	BCALC,BCALCP, CLTMP,EDTMTX	—
ATBCP(NN,NCL) A[NATBCP]	Capsule thermal expansion averaged in the radial region.	CLTMP,EDTMTX	—
ATBF (NN,NFR) A[NATBF]	Fuel thermal expansion averaged in the radial region.	BCALC,EDTMTX, FLTMP,GASOUT, ITERAT	—
ATBFA (NN,3) A[NATBFA]	Fuel thermal expansion averaged over the each region. ATBFA(NN,1)-over the columnar region. ATBFA(NN,2)-over the equiaxed region. ATBFA(NN,3)-over the undisturbed region.	EDTGRS,FLTMP	—
ATBLA (NN) A[NATBLA]	Cladding thermal expansion averaged over the cladding.	CLTMP,EDTGRS	—

Symbol	Definition	Subroutine	Unit
ATBPA (NN) A[NATBPA]	Thermal expansion averaged over the capsule.	CLTMP,EDTGRS	—
ATBPM (NN,NP) A[NATBPM]	Thermal expansions averaged in the regions into which the fuel is subdivided for plutonium migration.	FLTMP	—
BM (NN,NTR) A[NBM]	Bulk modulus in the radial region of the fuel and the cladding.	BCALC,CRACK EDTMTX,ITERAT, XLAME	kg/cm^2
BU (NN) A[NBU]	Burn-up.	BEGIN,BURN,EDTGRS, FLTMP,ITERAT, PLOT1,RHOMOD, UPDTE	atomic-percent
BUM (NN) A[NBUM]	Burn-up.	EDTGRS	MWD/T
CLZ (NN) A[NCLZ]	Cladding length of each axial section.	BEGIN,EDTGRS, EDTINP,LENGTH, MATRXP,PLENP, VOLUME	cm
CLZO (NN) [NCLZO]	Initial cladding length.	GEGIN,INPUT, LENGTH	cm
CPU (NN,NP) A[NCPU]	Plutonium concentration in the NP equal mass regions in the fuel.	BEGIN,EDTGRS, EDTINP,EDTMTX, INPUT,PLOT1,PLUTO, PROPTY	—
CPUD (NP) A[NCPUD]	Plutonium concentration change during the time interval.	PLUTO	1/sec
CRIF (NN) A[NCRIF]	Friction force between the fuel and the cladding.	BCALCP,EDTMTX, FRICHK	kg/cm
DELTOU (NN) A[NDELT]	Outer temperature increment of the cladding (used for power cut).	POWCUT	°C
DEQP (NN,NTR) A[NDEQP]	Equivalent creep-strain increment of each region.	CHECK,CREEC, CREEF,MATRX, MATRXP	—
DXM (NN,3) A[NDXM]	Migration velocities of zone boundaries, i.e., inner	BEGIN,CHKSPD, FLTMP,RHOMOD, UPDTE	cm/sec

Symbol	Definition	Subroutine	Unit
	radius, columnar-equiaxed boundary, and equiaxed-undisturbed region boundary.		—
E (NN, NTR) A[NYNG]	Young's modulus in the fuel and cladding.	EDTMTX, MATRX, MATRXP, XLAME	kg/cm^2
EHP (NN, NTR) A[NEHP]	Hot-pressing of the fuel(i.e., compaction by decreasing the fuel porosity) contributing to swelling.	BCALC, BEGIN, EDTMTX, FLTMP, ITERAT.	—
EPC (NN, NTR) A[NEPC]	Total circumfetential strain.	BCALC, BCALCP, EDTMTX, FLTMP, GASOUT, INPUT, ITERAT, RHOMOD, UPDATE, UPDATP	—
EPR (NN, NTR) A[NEPR]	Total radial strain.	BCALC, BCALCP, EDTMTX, FLTMP, GASOUT, INPUT, ITERAT, RHOMOD, UPDATE, UPDATP	—
EPSEQ(NN,NTR) A[NEPSEQ]	Equivalent creep-strain increment according to the Prandtl-Reuss flow laws.	BEGIN, CREEF, UPDATE	—
EPZ (NN, NTR) A[NEPZ]	Total axial strain.	BCALC, BCALCP, BURN, EDTMTX, FLTMP, GASOUT, INPUT, ITERAT, LENGTH, PLUTO, RHOMOD, UPDATE, UPDATP	—
EQSIG (NN,NTR) A[NEQSIG]	Equivalent stress.	CONVG, CREEC, CREEF, MATRX, MATRXP	kg/cm^2
F (NN,NP) A[NF]	Relative power density in radial shells of equal mass.	EDTINP, FLTMP, INPUT	—
FD (NN, NFR) A[NFD]	Fission density in the fuel.	CREEF, EDTGRS, FLTMP, GASOUT, SWELL	$1/cm^2$
FDPM (NN,NP) A[NFDPM]	Fission densities in the fuel regions into which the fuel is subdivided for plutonium migration.	EDTGRS, FLTMP	$1/cm^2$

Symbol	Definition	Subroutine	Unit
FLNC (NN) A[NFLNC]	Fast fluence.	BEGIN,CREEC, EDTGRS,FLXCAL, HKNS,UPDTE	n/cm
FLUX (NN) A[NFLUX]	Neutron flux.	BCALC,BCALCP, CREEC,CREEF, EDTGRS,FLXCAL	n/sec·cm^2
FLZ (NN) A[NFLZ]	Fuel length of each axial section.	BEGIN,EDTGRS, EDTINP,GASOUT, GOLDN,LENGTH, MATRX,PLENP, VOLUME	cm
FLZO (NN) A[NFLZO]	Initial fuel length.	BEGIN,LENGTH	cm
FTHP (NN,NTR) A[NFTHP]	Thermal creep component contributing to total creep of the fuel. (not used)	CREEF	—
G (NN,NTR) A[NSHR]	Shear modulus of each region of fuel and cladding.	BCALC,BCALCP, CRACK,CREEF, EDTMTX,MATRX, MATRXP,XLAME	kg/cm^2
GB (NN,2) A[NSHRB]	Shear modulus averaged over the fuel or cladding.	BCALC,BCALCP, EDTMTX,MATRX, MATRXP,XLAME	kg/cm^2
GDT (NN,NP1) A[NGDT]	Temperature derivative at each region boundary. (dT/dr)	FLTMP,PLUTO	°C/cm
GR (NN) A[NGR]	Gas release rate from each axial section during the time step.	BEGIN, EDTGRS, GASOUT	mole
GS (NN,NFR) A[NGS]	Gas remaining in each region.	BEGIN,EDTGRS, EDTMTX,FLTMP, GASOUT,ITERAT	mole
HC (NN) A[NHC]	Heat transfer coefficient between the coolant and the fuel-element capsule (or the cladding if the element is unencapsulated).	EDTGRS,GOLDN	cal/hr·cm^2·°F
HG (NN) A[NHG]	Gap heat-transfer coefficient between the fuel and the cladding.	BEGIN,EDTGRS, FLTMP,GAPCON	cal/hr·cm^2·°F

Symbol	Definition	Subroutine	Unit
IO (NN) A[NIO]	Device reference number used for temporary data output. (not used)	IOSET	—
ITP (NBUF) A[NPIT]	Time step number selected for plotting.	PLOT5, PLTOR	—
KFRIC (NN) A[NKFRIC]	Gap status between the fuel and the cladding of each axial section. (0-gap open, 1-stick, 2-slip).	BCALC,BCALCP, BEGIN,CLEAR, EDTMX,FRICHK, MATRX,MATRXP	—
LM (NN,NTR) A[NLM]	Number of the cracks summed up in the three principal directions.	BEGIN,CRACK,CREEF, EDTMTX,XLAME	—
LMC (NN,NTR) A[NLMC]	Number of cracks in the circumferential direction.	BEGIN,CRACK, EDTMTX,PLOT4	—
LMR (NN,NTR) A[NLMR]	Number of cracks in the radial direction.	BEGIN,CRACK, EDTMTX,PLOT4	—
LMZ (NN,NTR) A[NLMZ]	Number of cracks in the axial direction.	BEGIN,CRACK, EDTMTX,PLOT4	—
OTOM (NN) A[NOTOM]	O/M ratio(i.e., X in UOx) according to fuel stoichiometry.	EDTGRS,EDTINP, FLTMP,INPUT, OTMCAL	—
P (NN,NTR) A[NPOS]	Poisson's ratio of each region of the fuel and the cladding.	BCALC,BCALCP, EDTMTX,MATRX, MATRXP,XLAME	—
PB (NN,2) A[NPOSB]	Poisson's ratio average over the fuel or the cladding.	BCALC,BCALCP, EDTMTX,MATRX, MATRXP,XLAME	—
POW (NN) A[NPOW]	Linear power in each axial section	BEGIN,BURN,CLTMP, CREEF,EDTGRS, FLTMP,GASOUT, GOLDN,PLOT1,SWELL	watt/cm
PRS (NN,NTR) A[NPRS]	Hydro-static pressure.	BCALC,BEGIN,CRACK, EDTMTX,ITERAT	kg/cm^2
PTOUR (NN) A[NPTOU]	Cladding outer surface temperature of the previous time step.	CORTMP,POWCUT	°C

Symbol	Definition	Subroutine	Unit
QSF (NN) A[NQSF]	Axial power distribution factor.	BEGIN,BURN,GOLDN	—
QSFF (NN) A[NQSFF]	Axial fast flux shape.	BEGIN,EDTINP, FLXCAL,INPUT	—
QSFP (NN) A[NQSFP]	Nonnormalized axial power shape.	BEGIN,BURN,EDTINP, INPUT	—
RB (NN,NTR) A[NRBAR]	Relative radius representing the each radial region.	BCALC,CLTMP,FLTMP, MATRX	cm
RCB (NN) A[MRCB]	Radius of columnar and equiaxed region interface.	BEGIN,EDTGRS, EDTIMP,FLTMP, INPUT,PLOT1,PLOT4, PLUTO,RHOMOD, UPDTE	cm
RCBO (NN) A[NRCBO]	Initial radius of columnar and equiaxed region interface.	BEGIN,RHOMOD	cm
RCI (NN) A[NRCI]	Inner radius of the cladding.	BEGIN,CLTMP, EDTGRS,EDTINP, FLTMP,FRICHK, GAPCON,INPUT, PLENP,PLOT1, RHOMOD,UPDTE	cm
RCIO (NN) A[NRCIO]	Initial inner radius of the cladding.	BEGIN	cm
RCL (NN,NCL1) A[NRCL]	Radii of the region boundaries in the cladding.	BCALC,BCALCP, CLTMP,EDTGRS, EDTMIX,MATRX, MATRXP,VOLUME	cm
RCLO (NN,NCL1) A[NRCLO]	Initial radii of the region boundaries in the cladding.	CLTMP	cm
RCO (NN) A[NRCO]	Outer radius of the cladding.	BEGIN,EDTGRS, EDTINP,GOLDN, INPUT,PLOT3, RHOMOD,UPDTE	cm
RCOO (NN) A[NRCOO]	Initial outer radius of the cladding	BEGIN,EDTMTX, PLOT3	cm
RCP (NN) A[NRCP]	Radii of the region boundaries in the capsule.	CLTMP,EDTMTX	cm
RDIS (NN,NTR2) A[NRDIS]	Radial displacements of the region boundaries in the fuel and the cladding.	EDTMTX,INPUT, PLOT3,RHOMOD, UPDATE,UPDATP	cm

Symbol	Definition	Subroutine	Unit
RDIST (NN,4) A[NRDIST]	Used for checking. (not used for calculations) RDIST(NN,1)-not used. RDIST(NN,2)-restructuring accumulated into columnar form. RDIST(NN,3)-restructuring accumulated into equiaxed form. RDIST(NN,4)-mass	BEGIN,RHOMOD,UPDTE	—
REB (NN) A[NREB]	Radius of equiaxed and undisturbed region interface.	BEGIN,EDTGRS, EDTINP,FLTMP,INPUT, PLOT1,PLOT4,PLUTO, RHOMOD,UPDTE	cm
REBO (NN) A[NREBO]	Initial radius of equiaxed and undisturbed region interface.	BEGIN,RHOMOD	cm
RFR (NN,NFR1) A[NRFR]	Radii of the region boundaries in the fuel.	BCALC,EDTGRS, EDTMTX,FLTMP, GASOUT,ITERAT, MATRX,PLOT1, RHOMOD,VOLUME	cm
RFRO (NN,NFR1) A[NRFRO]	Initial radii of the region boundaries in the fuel.	FLTMP	cm
RFRPM(NN,NP1) A[NRFRPM]	Radii of the region boundaries of the NP equal mass radial shells.	EDTMTX,FLTMP,PLOT1, PLOT4,PLUTO	cm
RH (NN,NTR) A[NRH]	Density of the fuel and the cladding.	BCALC,BCALCP,BEGIN, EDTGRS,FLTMP, ITERAT,PLUTO, RHOMOD,XLAME	g/cc
RHOP (NN,3) A[NRHOP]	Initial density of the three region in the fuel.	BEGIN,EDTINP, FLTMP,INPUT,ITERAT, PROPTY,RHOMOD	g/cc
RHOT (NN) A[NRHOT]	Theoretical density of the fuel.	EDTGRS,FLTMP, ITERAT,PROPTY, RHOMOD,XLAME	g/cc
RUB (NN) A[NRUB]	Outer radius of the fuel.	BEGIN,EDTGRS,EDTINP, FLTMP,FRICHK,GAPCON, INPUT,PLENP,PLOT1, RHOMOD,UPDTE	cm
RUBO (NN) A[NRUBO]	Initial outer radius of the fuel.	BEGIN,RHOMOD	cm

Symbol	Definition	Subroutine	Unit
RVB (NN) A[NRVB]	Central void radius.	BEGIN,EDTGRS,EDTINP, FLTMP,INPUT,PLENP, PLUTO,RHOMOD,UPDTE,	cm
RVBO (NN) A[NRVBO]	Initial central void radius.	BEGIN,RHOMOD	cm
SIGC(NN,NTR) A[NSIGC]	Hoop stress component of the fuel and the cladding.	CONVG,EDTMTX,INPUT, PLOT2,UPDATE, ITERAT,UPDATP	kg/cm^2
SIGR (NN,NTR) A[NSIGR]	Radial stress component.	CONVG,EDTMTX,INPUT, ITERAT,UPDATE, UPDATP	kg/cm^2
SIGZ (NN,NTR) A[NSIGZ]	Axial stress component.	CONVG,EDTMTX,INPUT, ITERAT,UPDATE, UPDATP	kg/cm^2
SWG (NN,NFR) A[NSWG]	Swelling rates due to gaseous fission products.	BCALC,BEGIN,EDTMTX, FLTMP,ITERAT	—
SWS (NN,NTR) A[NSWS]	Swelling rates due to solid fission products.	BCALC,BEGIN,EDTMTX, FLTMP,GASOUT, ITERAT,SWELL	—
SWS3 (NN,NUR) A[NSWS3]	Swelling rates due to the solid fission products in the NUR radial meshes of the undisturbed region.	BEGIN,ITERAT, SWELL	—
TBCL (NN,NCL) A[NTBCL]	Average temperature of each region in the cladding.	BCALC,BCALCP,CLTMP, CREEC,EDTMTX, HKNS,XLAME	°C
TBCLA (NN) A[NTBCLA]	Average temperature of the cladding.	CLTMP,EDTGRS	°C
TBCP (NN,NCL) A[NTBCP]	Average temperature of each region in the capsule.	CLTMP,EDTMTX	°C
TBCPA (NN) A[NTBCPA]	Average temperature of the capsule.	CLTMP,EDTGRS	°C
TBF (NN,NFR) A[NTBF]	Average temperature of each region in the fuel.	CRACK,CREEF,EDTMTX, FLTMP,GASOUT, ITERAT,XLAME	°C
TBFA (NN,3) A[NTBFA]	Average temperature of each region in the fuel.	EDTGRS,FLTMP	°C
TBPM (NN,NP) A[NTBFPM]	Average temperature of each equal mass shell in the fuel.	FLTMP	°C

Symbol	Definition	Subroutine	Unit
TCL (NN,NCL1) A[NTCL]	Temperature at the boundaries of the NCL radial meshes of the cladding.	CLTMP,EDTGRS, EDTMTX,FLTMP	°C
TCLO (NN) A[NTCLO]	Outer temperature of the cladding.	CLTMP,EDTGRS,GOLDN	°C
TCP (NN,NCL1) A[NTCP]	Temperature at the boundaries of the NCL radial meshes of the capsule.	CLTMP,EDTGRS, EDTMTX	°C
TCPI (NN) A[NTCPI]	Innter temperature of the capsule.	EDTGRS,GOLDN	°C
TCPO (NN) A[NTCPO]	Outer temperature of the capsule.	CLTMP,EDTGRS,GOLDN	°C
TFR (NN,NFR1) A[NTFR]	Temperatures at the boundaries of the radial regions of the fuel.	EDTGRS,EDTMTX, FLTMP,PLENP,PLOT1,	°C
TFRPM(NN,NP1) A[NTFRPM]	Temperatures at the boundaries of the radial NP equal mass shells in the fuel.	EDTMTX,FLTMP, PLOT1,PLUTO	°C
TNA (NN) A[NTNA]	Coolant temperature.	EDTGRS,GOLDN	°C
TOUTER (NN) A[NTOUR]	Outer surface temperature of the cladding.	CORTMP,GOLDN, POWCUT	°C
TOUTO (NN) A[NTOUO]	Outer surface temperature of the cladding of the previous time step.	CORTMP	°C
TPC (NN,NTR) A[NTPC]	Cumulative total circumferential creep strain.	BCALC,BCALCP,CRACK, EDTMTX,FLTMP,INPUT, PLOT3,UPDATE,UPDATP	—
TPR (NN,NTR) A[NTPR]	Cumulative total radial creep strain.	BCALC,BCALCP,CRACK, EDTMTX,FLTMP,INPUT, UPDATE,UPDATP	—
TPZ (NN,NTR) A[NTPZ]	Cumulative total axial creep strain.	BCALC,BCALCP,CRACK, EDTMTX,FLTMP,INPUT, UPDATE,UPDATP	—

Symbol	Definition	Subroutine	Unit
TSC (NN,NTR) A[NTSC]	Cumulative total circumferential swelling strain.	BCALC,BCALCP,CRACK, EDTMTX,FLTMP,HKNS, INPUT,ITERAT,PLOT3, UPDATE,UPDATP	—
TSR (NN,NTR) A[NTSR]	Cumulative total radial swelling strain.	BCALC,BCALCP,CRACK, EDTMTX,FLTMP,HKNS, INPUT,ITERAT, UPDATE,UPDATP	—
TSZ (NN,NTR) A[NTSZ]	Cumulative total axial swelling strain.	BCALC,BCALCP,CRACK, EDTMTX,FLTMP,HKNS, INPUT,ITERAT, UPDATE,UPDATP	—
VOL (NN,NTR) A[NVOL]	Volume of each region.	BCALC,BCALCP, GASOUT,ITERAT, PLOT2,PLOT3, VOLUME,XLAME	cm^3
WNEW (NFR) A[NWNEW]	Mass in the each radial region of the fuel.	FLTMP,RHOMOD	g/cm
WOLD (NN,NFR) A[NWOLD]	Mass in the each radial region of the fuel at the previous time step.	FLTMP,RHOMOD, UPDTE	g/cm
XBU (NBUF) A[NPBU]	Same as BU (used for plotting)	PLOT1, PLTOR	a/o
XCPGP (NBUF) A[NPCPGP]	Clad-pellet gap (used for plotting)	PLOT1, PLTOR	mm
XCPI (NN) A[NXCPI]	Capsule inner radius.	BEGIN,CLTMP, EDTGRS,EDTINP	cm
XCPO (NN) A[NXCPO]	Capsule outer radius.	BEGIN,CLTMP, EDTGRS,EDTINP	cm
XCPU (NBUF,NP) A[NPCPU]	Same as CPU(used for plotting)	PLOT1, PLTOR	—
XCSTN (NBUF) A[NPCSTN]	Diameter change ($\Delta D/D$) of the cladding. (used for plotting)	PLOT3, PLTOR	%
XEHP (NFR) A[NXEHP]	Used temporarily for modifying EHP considering the restructuring.	FLTMP,ITERAT	—

Symbol	Definition	Subroutine	Unit
XEPC (NFR) A[NXEPC]	Used temporarily for modifying EPC considering the restructuring.	FLTMP,RHOMOD	—
XEPR (NFR) A[NXEPR]	Used for modifying EPR.	FLTMP,RHOMOD	—
XGS (NFR) A[NXGS]	Used for modifying GS	FLTMP	mole
XM (NN,3) A[NXM]	Mass of each region in the fuel.	BEGIN,BURN,EDTGRS, FLTMP,PLUTO,RHOMOD,	g/cm
XMINT (NN) A[NXMINT]	Initial mass of each axial section of the fuel.	BEGIN,RHOMOD	g/cm
XPLPR (NBUF) A[NPPLPR]	Plenum pressure (used for plotting)	PLOT5,PLTOR	kg/cm^2
XPOW (NBUF) A[NPPOW]	Same as POW(used for plotting)	PLOT1,PLTOR	watt/cm
XRL (NBUF,NP1) A[NPRL]	Same as RFRPM (used for plotting)	PLOT1,PLTOR	mm
XRLR (NBUF,2) A[NPRLR]	XRLR(NBUF,1)-same as RCB XRLR(NBUF,2)-same as REB (used for plotting)	PLOT1,PLTOR	mm
XSIGC (NBUF) A[NPSIGC]	Cladding average hoop stress. (used for plotting)	PLOT2,PLTOR	kg/mm^2
XSWG (NFR) A[NXSWG]	Used for modifying SWG considering the restructuring	FLTMP,ITERAT	—
XSWS (NFR) A[NXSWS]	Used for modifying SWS	FLTMP	—
SXWS3 (NUR) A[NXSWS3]	Used for modifying SWS3	FLTMP	—
XT (NBUF) A[NPT]	Fuel outer surface temperature.(used for plotting)	PLOT1,PLTOR	°C
XTIME (NBUF) A[NPTIME]	Time selected for plotting. (used for plotting)	PLOT5,PLTOR	hr

Symbol	Definition	Subroutine	Unit
XTL (NBUF,NP1) A[NPTL]	Same as TFRPM (used for plotting)	PLOT1,PLTOR	°C
XTLMX (NBUF) A[NPTLMX]	Maximum temperature in the fuel. (used for plotting)	PLOT1,PLTOR	°C
XTLR (NBUF,2) A[NPTLR]	Temperature at the radial point of radius XRLR. (used for plotting)	PLOT1,PLTOR	°C
XTPC (NFR) A[NXTPC]	Used for modifying TPC considering the restructuring	FLTMP	—
XTPC (NBUF) A[NPTPC]	Cladding average circumferential creep strain. (used for plotting)	PLOT3,PLTOR	—
XTPR (NFR) A[NXTPR]	Used for modifying TPR.	FLTMP	—
XTPZ (NFR) A[NXTPZ]	Used for modifying TPZ.	FLTMP	—
XTSC (NFR) A[NXTSC]	Used for modifying TSC.	FLTMP	—
XTSC (NBUF) A[NPTSC]	Cladding average circumferential swelling strain. (used for plotting)	PLOT3,PLTOR	—
XTSR (NFR) A[NXTSR]	Used for modifying TSR.	FLTMP	—
XTSZ (NFR) A[NXTSZ]	Used for modifying TSZ.	FLTMP	—
XYP (NBUF) A[NPYP]	Gas release rate. (used for plotting)	PLOT5,PLTOR	%
YBU (NN) A[LBU]	BU at the start of the	UPDTE	a/o
YDXM (NN,3) A[LDXM]	DXM time step.	CHKSPD,UPDTE	cm/sec

Symbol	Definition		Subroutine	Unit
YFLNC (NN) A[LFLNC]	FLNC		UPDTE	n/cm^2
YRCB (NN) A[LRCB]	RCB		UPDTE	cm
YRCI (NN) A[LRCI]	RCI	at the start of	UPDTE	cm
YRCO (NN) A[LRCO]	RCO	the time step.	UPDTE	cm
YRDIST(NN,4) A[LRDIST]	RDIST		UPDTE	—
YREB (NN) A[LREB]	REB		UPDTE	cm
YRUB (NN) A[LRUB]	RUB		UPDTE	cm
YRVB (NN) A[LRVB]	RVB		UPDTE	cm
YTOUR (NN) A[LTOUR]	TOUTER		CORTMP	°C
YWOLD(NN,NFR) A[LWOLD]	WOLD		UPDTE	g/cm
ZFLX (NN) A[NZFLX]	ratio of Frenkel pair-generation rate to flux.		BEGIN,FLXCAL	—

APPENDIX D: CONTROL CARDS AND MESSAGES

Appendix D: Control Cards and Messages

D.1 Control Cards

A typical set of control cards used for the ACTIVE-II code on the IBM 360/195 at Tokyo Data Center is shown in Table D.1. Included is a short FORTRAN program containing the variable 'A' and the variable 'AM'-these dimentions must be set at least as large as the core space reqüired for the problem data (excluding the data for the matrix) and for the data only for the matrix, respectively. (See the section 6.1 Core Storage.) And the variable 'IA' and the variable 'IAM' must also be the same number as the dimensions of the 'A' and the 'AM', respectively.

Logical unit requirements are as follows.

Logical Number	Device Type	Use
5		System input device.
6		System output device.
7		System punch device.
77	Tape	Plotter output device

Locigal unit requirements are described in the section I/O Unit Requirements.

On the stage of link-edit, four library data sets are required. SYS1.DCSLIB and SYS1.FORTLIB includes the functional subroutines such as SQRT, ABS, EXP, ALOG, DATE, STIMER, TTIMER, and so on, used in the ACTIVE-II.

SYS1.PLTLIB10 includes the subroutines used for plotting such as PLOTS, PLOT, SYMBOL, NUMBER, and so on.

FPP1.SLMATH includes the SL-MATH subroutines such as MSNR, MBSR, and so on.

```
//J1111AMK JOB (1111A,ACTIV),'KATSURAGAWA',TIME=1,REGION=384K
/*MAIN TYPE=(PLOT)
/*SETUP   UNIT=PLOT,ID=(PLOT,RING,SAVE,NL)
/*JOBPARM   LIN=20
//ACTIVE  EXEC PLTFGCLG
//FORT.SYSIN  DD  *
        *****************************************
        *****                               *****
        *****     SOURCE   PROGRAM   DECK   *****
        *****                               *****
        *****************************************
/*
//GO.SYSIN  DD  *
        *****************************************
        *****                               *****
        *****          DATA  CARD           *****
        *****                               *****
        *****************************************
/*
```

D.2 Error Stops

The following error stops have been programmed into ACTIVE-II.

	Cause	Subroutine
1.	Clad has hit capsule wall	GOLDN
2.	'*END' card missing (input miss)	HEADER
3.	Mass increment is too big	PLUTO
4.	Mass in columnar zone is negative	RHOMOD
5.	Clad temperature is too high	XLAME
6.	Refer to the MSNR	MSNR(SL-MATH)

APPENDIX E: SAMPLE INPUT

Appendix E: Sample Input

Sample input is shown in Table E.1. The sample pin is FlA fuel pin, whose abstract is shown in Chapter 6.2. For the detail explanation, refer to Chapter 6.3.

Table E.1 Sample Input.

```
*---.----1----.----2----.----3----.----4----.----5----.----6----.----7----.----8            PAGE    1

    1    0    1    0                                                                  REC NO.    1
  A C T I V E   S E N S I T I V I T Y   A N L Y S I S                                 REC NO.    2
      * * *   G . E . F - 1 A ( E B R - 2 )   * * *                                   REC NO.    3
          A X I A L   M E S H   =   4                                                 REC NO.    4
          R A D I A L   M E S H                                                       REC NO.    5
              C O L U M N A R   =   1                                                 REC NO.    6
              E Q U I A X E D   =   1                                                 REC NO.    7
              U N D I S T U R B E D     =   1                                         REC NO.    8
              C L A D D I N G   =   1                                                 REC NO.    9
*END                                                                                  REC NO.   10
    4    1    1    1    1   10    3   -2   -1    1    1    0    0                      REC NO.   11
    1    2    2    2    2                                                             REC NO.   12
0.001          0.001          0.001          0.001                                    REC NO.   13
0.002          0.002          0.002          0.002                                    REC NO.   14
0.003          0.003          0.003          0.003                                    REC NO.   15
0.2744         0.2744         0.2744         0.2744                                   REC NO.   16
0.2794         0.2794         0.2794         0.2794                                   REC NO.   17
0.3175         0.3175         0.3175         0.3175                                   REC NO.   18
14.45          3.61           18.06          30.06                                    REC NO.   19
0.9792         1.0723         0.9485         0.3                                      REC NO.   20
0.9792         1.0723         0.9485         0.0                                      REC NO.   21
1.998          1.998          1.998          1.998                                    REC NO.   22
0.2            0.2            0.2            0.2                                      REC NO.   23
10.54          10.54          10.54          10.54                                    REC NO.   24
1.0            1.0            1.0            1.0            1.0                       REC NO.   25
1.0            1.0            1.0            1.0            1.0                       REC NO.   26
0.4254         0.4762         0.822                                                   REC NO.   27
3000.0         6000.0         1000000.0                                               REC NO.   28
0.0            1.68                                                                   REC NO.   29
7.98           20.0           20.0                                                    REC NO.   30
       1.92E+07       1.2E+0464400.0       44400.0                                    REC NO.   31
      0.407E-030.0            0.0                                                     REC NO.   32
0.15           0.01                 3.48E+12371.1       372.0                         REC NO.   33
0.671          439.8                3.48E+12371.1       479.4          1              REC NO.   34
2.00           439.8                3.48E+12371.1       479.4                         REC NO.   35
2.125          0.01                 3.48E+12371.1       372.1                         REC NO.   36
2.5            0.01                 3.48E+12371.1       372.1                         REC NO.   37
2.921          439.8                3.48E+12371.1       479.4                         REC NO.   38
6.35           439.8                3.48E+12371.1       479.4                         REC NO.   39
6.643          195.5                3.48E+12371.1       419.2                         REC NO.   40
6.8            195.5                3.48E+12371.1       419.2                         REC NO.   41
7.142          439.8                3.48E+12371.1       479.4                         REC NO.   42
9.50           439.8                3.48E+12371.1       479.4                         REC NO.   43
10.0           0.01                 3.48E+12371.1       372.1                         REC NO.   44
11.0       1.0 0.01                 3.48E+1250.0        51.0                          REC NO.   45
12.0       1.0 0.01                 3.48E+1250.0        51.0           1              REC NO.   46

*---.----1----.----2----.----3----.----4----.----5----.----6----.----7----.----8

TOTAL RECORDS    46
```

APPENDIX F: SAMPLE OUTPUT

Appendix F: Sample Output

Sample output is shown here, calculated for F1A fuel pin as shown in Appendix E. For detail explanation of output, refer to Chapter 6.4, and 6.5.

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

========== A C T I V E - I I - F B R ==========

ACTIVE SENSITIVITY ANLYSIS

*** G.E.F-1A(EBR-2) ***

AXIAL MESH = 4

RADIAL MESH

COLUMNAR = 1

EQUIAXED = 1

UNDISTURBED = 1

CLADDING = 1

```
===== INPUT VALUES OUTPUT NEXT =====

---- CONTROL NUMBER  AND  OPTION

     NUMBER OF AXIAL SECTION  NN =  4
     NUMBER OF MESH FOR COLUMNAR GRAIN REGION  NCR =  1
     NUMBER OF MESH FOR EQUIAXED GRAIN REGION  NER =  1
     NUMBER OF MESH FOR UNDISTURBED REGION  NUR =  1
     NUMBER OF MESH FOR CLAD REGION  NCL =  1
     NUMBER OF MESH FOR PLUTONIUM MIGRATION  NP = 10
     FUEL THERMAL CONDUCTIVITY OPTION  NOK =  3      (1/2/3 = SAYLES/BAILY-ASAMOTO/BIANCHERIA)
     CLAD SWELLING OPTION  NOS = -2      (-1/-2 = ANNEALED 304,316S.S./20P.C.COLD WORK 316S.S.)
     CLAD CREEP OPTION  NOC = -1      (-1/-2 = BOLTAX,BIANCHERIA,GUHA/MODIFIED)
     CRACK OPTION  NCRK =  1      (0/1 = NO CRACK CALC./CRACK CALC.)
     PLOT OPTION  NPLOT =  1      (0/1 = NO PLOT/PLOT)
               AXIAL SECTION NUMBER FOR FUEL =  2
               AXIAL SECTION NUMBER FOR STRAIN =  2
               AXIAL SECTION NUMBER FOR STRESS =  2
               AXIAL SECTION NUMBER FOR CRACK =  2
               STORAGE(WORDS) REQUIRED FOR PLOTTING STEP =  50

---- CENTRAL VOID RADIUS ( CM )   1-NN

          0.0010       0.0010       0.0010       0.0010

---- COLUMNAR GRAIN REGION OUTER BOUNDARY ( CM )   1-NN

          0.0020       0.0020       0.0020       0.0020

---- EQUIAXED GRAIN REGION OUTER BOUNDARY ( CM )   1-NN

          0.0030       0.0030       0.0030       0.0030

---- FUEL OUTER RADIUS ( CM )   1-NN

          0.2744       0.2744       0.2744       0.2744

---- CLAD INNER RADIUS ( CM )   1-NN

          0.2794       0.2794       0.2794       0.2794

---- CLAD OUTER RADIUS ( CM )   1-NN

          0.3175       0.3175       0.3175       0.3175

---- CAPSULE INNER RADIUS ( CM )   1-NN

          0.4254       0.4254       0.4254       0.4254

---- CAPSULE OUTER RADIUS ( CM )   1-NN

          0.4762       0.4762       0.4762       0.4762

---- COOLANT FLOW AREA =      0.82199997   ( CM**2 )

---- FUEL AXIAL LENGTH ( CM )   1-NN      ... TOTAL AXIAL LENGTH =      36.1200 ( CM ) ...
```

```
          14.4500       3.6100      18.0600        0.0

 ---- CLAD AXIAL LENGTH ( CM )    1-NN       ... TOTAL AXIAL LENGTH =       66.1800 ( CM ) ...

          14.4500       3.6100      18.0600       30.0600

 ---- DENSITY OF FUEL ( 3 REGIONS ) AND CLAD ( G/CC )

      AXIAL SECTION =  1          10.6612      10.5400      10.5400       7.9800
      AXIAL SECTION =  2          10.6612      10.5400      10.5400       7.9800
      AXIAL SECTION =  3          10.6612      10.5400      10.5400       7.9800
      AXIAL SECTION =  4           0.0          0.0          0.0          7.9800

 ---- O/M OF FUEL    PU-O(X)    1-NN

          1.9980        1.9980       1.9980        1.9980

 ---- PLUTONIUM CONCENTRATION ( WEIGHT-FRACTION )   1-NP

      AXIAL SECTION =  1           0.2000       0.2000       0.2000       0.2000       0.2000       0.2000       0.2000
                                   0.2000       0.2000       0.2000
      AXIAL SECTION =  2           0.2000       0.2000       0.2000       0.2000       0.2000       0.2000       0.2000
                                   0.2000       0.2000       0.2000
      AXIAL SECTION =  3           0.2000       0.2000       0.2000       0.2000       0.2000       0.2000       0.2000
                                   0.2000       0.2000       0.2000

 ---- AXIAL FAST FLUX ( E .GE. 0.1MEV ) RATE    1-NN

          0.9792        1.0723       0.9485        0.3000

 ---- AXIAL HEAT GENERATION RATE    1-NN

          0.9792        1.0723       0.9485        0.0

 ---- RADIAL HEAT GENERATION RATE    1-NP

      AXIAL SECTION =  1           1.0000       1.0000       1.0000       1.0000       1.0000       1.0000       1.0000
                                   1.0000       1.0000       1.0000
      AXIAL SECTION =  2           1.0000       1.0000       1.0000       1.0000       1.0000       1.0000       1.0000
                                   1.0000       1.0000       1.0000
      AXIAL SECTION =  3           1.0000       1.0000       1.0000       1.0000       1.0000       1.0000       1.0000
                                   1.0000       1.0000       1.0000

 ---- INITIAL GAS IN PLENUM =       0.0004070  ( MOLES )

 ***  ACTUALLY USED CORE FOR ARRAY   =     3897  WORDS      ( INPUT =     10000 )

 ***  ACTUALLY USED CORE FOR MATRIX  =     5984  WORDS      ( INPUT =     20000 )

 ***  RECOMMENDED CORE SIZE = 256 K BYTE
```

```
************************************************************************************************************
 STEP NO.  1      TIME (HRS) =      3.600      POWAV (W/CM) =        0.010     TOUT (C) = 372.100  TIN (C) = 371.100
                  DELT (HRS) =      3.600      CFLX (N/W/CM) =  0.349E 13      PD (KG/CM**2) =    1.68
************************************************************************************************************
```

===== GROSS OUTPUT NEXT =====

+++ AXIAL DISTRIBUTION (N=1,NN) +++

POWER(W/CM)	0.010	0.011	0.009	0.0
FLUX(N/CM**2/SEC)	0.34076E 11	0.37316E 11	0.33008E 11	0.10440E 11
FLUENCE(N/CM**2)	0.44163E 15	0.48362E 15	0.42778E 15	0.13530E 15
BURN UP(A/O)	0.00000	0.00000	0.00000	0.0
BURN UP(MWD/T)	0.0	0.0	0.0	0.0
FUEL LENGTH(CM)	14.45000	3.61000	18.06000	0.0
CLAD LENGTH(CM)	14.45000	3.61000	18.06000	30.06000
THEO. DENS.(G/CC)	11.06000	11.06000	11.06000	0.0
COOLANT TEMP.(C)	371.301	371.557	371.856	372.100
FILM H.(W/CM**2/C)	3.33288	3.33231	3.33163	3.33109
GAP H.(W/CM**2/C)	0.81487	0.81487	0.81487	0.81487
DELT GAS REL.(MOLES)	0.0000000	0.0000000	0.0000000	0.0

TOTAL GAS IN PLENUM	0.00040700	(MOLES)		
CUMULATIVE GAS RELEASED	0.0	(MOLES)		
PERCENT OF TOTAL F.P. GAS	0.0	(O/O)		
PLENUM PRESSURE	2.896	(KG/CM**2)		
CENTRAL HOLE PRESSURE	2.894	(KG/CM**2)		
TOTAL FUEL LENGTH AND FL/FLO	36.12000	(CM)	0.0	(O/O)
TOTAL CLAD LENGTH AND CL/CLO	66.17999	(CM)	0.0	(O/O)

+++ PERFORMANCE +++

* AXIAL SECTION = 1 *

	FUEL CENTER	COLUMNAR GR.	EQUIAXED GR.	FUEL OUTER	CLAD INNER	CLAD OUTER	CAPSULE INNER	CAPSULE OUTER
RADIUS(CM)	0.00101	0.00200	0.00300	0.27440	0.27940	0.31750	0.42540	0.47620
TEMP.(C)	371.325	371.325	371.325	371.311	371.304	371.303	371.302	371.302
AV.TEMP(C)	371.325	371.325	371.318		371.303		371.302	
ALPHA-T(-)	0.00303	0.00303	0.00303		0.00626		0.00626	
DENS(G/CC)	10.66115	10.54000	10.54000		7.98000			
MASS(G/CM)	0.00010	0.00017	2.49291					

* AXIAL SECTION = 2 *

	FUEL CENTER	COLUMNAR GR.	EQUIAXED GR.	FUEL OUTER	CLAD INNER	CLAD OUTER	CAPSULE INNER	CAPSULE OUTER
RADIUS(CM)	0.00101	0.00200	0.00300	0.27440	0.27940	0.31750	0.42540	0.47620
TEMP.(C)	371.585	371.585	371.585	371.568	371.561	371.560	371.559	371.558
AV.TEMP(C)	371.585	371.585	371.576		371.560		371.558	
ALPHA-T(-)	0.00303	0.00303	0.00303		0.00627		0.00627	
DENS(G/CC)	10.66115	10.54000	10.54000		7.98000			
MASS(G/CM)	0.00010	0.00017	2.49291					

* AXIAL SECTION = 3 *

	FUEL CENTER	COLUMNAR GR.	EQUIAXED GR.	FUEL OUTER	CLAD INNER	CLAD OUTER	CAPSULE INNER	CAPSULE OUTER
RADIUS(CM)	0.00101	0.00200	0.00300	0.27440	0.27940	0.31750	0.42540	0.47620
TEMP.(C)	371.880	371.880	371.880	371.866	371.859	371.858	371.858	371.857
AV.TEMP(C)	371.880	371.880	371.873		371.858		371.857	
ALPHA-T(-)	0.00303	0.00303	0.00303		0.00627		0.00627	
DENS(G/CC)	10.66115	10.54000	10.54000		7.98000			
MASS(G/CM)	0.00010	0.00017	2.49291					

* AXIAL SECTION = 4 *

	FUEL CENTER	COLUMNAR GR.	EQUIAXED GR.	FUEL OUTER	CLAD INNER	CLAD OUTER	CAPSULE INNER	CAPSULE OUTER
RADIUS(CM)	0.0	0.0	0.0	0.0	0.27940	0.31750	0.42540	0.47620
TEMP.(C)	0.0	0.0	0.0	0.0	372.100	372.100	372.100	372.100
AV.TEMP(C)		0.0	0.0	0.0		372.100		372.100
ALPHA-T(-)		0.0	0.0	0.0		0.00628		0.00628
DENS(G/CC)		0.0	0.0	0.0		7.98000		

===== STRESS AND STRAIN OUTPUT NEXT =====

*** AXIAL SECTION = 1 ***

+++ CAPSULE (I=1,NCL) +++

RADIUS(CM)	0.42540		0.47620
TEMP.(C)	371.302		371.302
ALPHA-T(-)		0.00626	

+++ CLAD (I=1,NCL) +++

RADIUS(CM)	0.27940		0.31750
DISPLACEMENT(CM)	0.00175		0.00199
TEMP.(C)	371.304		371.303
PRESS.(KG/CM**2)	0.28960E 01		0.16800E 01
ALPHA-T(-)		0.00626	
POISSON RATIO(-)		0.33306	
YOUNG M.(KG/CM**2)		0.17504E 07	
STRESS(KG/CM**2)			
RADIAL		-0.22363E 01	
CIRCUMFERENTIAL		0.72244E 01	
AXIAL		0.20226E 01	
TOTAL STRAIN(-)			
RADIAL		0.62612E-02	
CIRCUMFERENTIAL		0.62684E-02	
AXIAL		0.62645E-02	
CREEP STRAIN(-)			
RADIAL		-0.59558E-15	
CIRCUMFERENTIAL		0.63651E-15	
AXIAL		-0.40928E-16	
CREEP IN DELT(-)			
RADIAL		-0.59558E-15	
CIRCUMFERENTIAL		0.63651E-15	
AXIAL		-0.40928E-16	
SWELLING STRAIN(-)			
TOTAL		0.55260E-15	

PERCENT CHANGE IN ORIGINAL O.D. = 0.0 (0/0)

AVERAGE POISSON RATIO(-) = 0.33306

AVERAGE YOUNG MODULUS(KG/CM**2) = 0.17504E 07

+++ FUEL (I=1,NFR) +++

RADIUS(CM)	0.00101		0.00200		0.00300		0.27440
DISPLACEMENT(CM)	0.00000		0.00001		0.00001		0.00083
TEMP.(C)	371.325		371.325		371.325		371.311
PRESS.(KG/CM**2)	0.28939E 01		0.29983E 01		0.29988E 01		0.28960E 01
ALPHA-T(-)		0.00303		0.00303		0.00303	
POISSON RATIO(-)		0.31705		0.31549		0.31549	
YOUNG M.(KG/CM**2)		0.22370E 07		0.21741E 07		0.21741E 07	
NUMBER OF CRACK							
RADIAL		0		0		0	
CIRCUMFERENTIAL		0		0		0	

AXIAL	0	0	0
STRESS(KG/CM**2)			
RADIAL	-0.29686E 01	-0.29986E 01	-0.28961E 01
CIRCUMFERENTIAL	-0.30990E 01	-0.29998E 01	-0.28959E 01
AXIAL	-0.34913E 01	-0.34155E 01	-0.32076E 01
TOTAL STRAIN(-)			
RADIAL	0.30275E-02	0.30274E-02	0.30274E-02
CIRCUMFERENTIAL	0.30274E-02	0.30274E-02	0.30274E-02
AXIAL	0.30272E-02	0.30272E-02	0.30272E-02
CREEP STRAIN(-)			
RADIAL	0.32659E-31	0.20904E-31	0.15566E-31
CIRCUMFERENTIAL	0.13095E-31	0.20731E-31	0.15600E-31
AXIAL	-0.45755E-31	-0.41635E-31	-0.31167E-31
CREEP IN DELT(-)			
RADIAL	0.32659E-31	0.20904E-31	0.15566E-31
CIRCUMFERENTIAL	0.13095E-31	0.20731E-31	0.15600E-31
AXIAL	-0.45755E-31	-0.41635E-31	-0.31167E-31
SWELLING STRAIN(-)			
TOTAL	0.65783E-09	0.36103E-09	0.15577E-09
SOLID COMPONENT	0.19297E-09	0.19294E-09	0.46196E-09
GAS COMPONENT	0.17805E-08	0.89015E-09	0.53401E-11
HOT PRESSING	0.0	0.0	0.0

PERCENT CHANGE IN ORIGINAL O.D. = 0.0 (O/O)

AVERAGE POISSON RATIO(-) = 0.31549

AVERAGE YOUNG MODULUS(KG/CM**2) = 0.21741E 07

GAP SIZE(CM) = 0.00592

FRICTION FORCE(KG/CM) = 0.0 (OPEN)

+++ PLUTONIUM MIGRATION (I=1,NP) +++

RADIUS(CM)	0.00101	0.08678	0.12272	0.15030	0.17355	0.19403	0.21255	0.22958	0.24543	0.26032	0.27440
TEMP.(C)	371.325	371.324	371.323	371.321	371.320	371.318	371.317	371.315	371.314	371.312	371.311
PU CONC.(-)	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	

*** AXIAL SECTION = 2 ***

+++ CAPSULE (I=1,NCL) +++

RADIUS(CM)	0.42540	0.47620
TEMP.(C)	371.559	371.558
ALPHA-T(-)	0.00627	

+++ CLAD (I=1,NCL) +++

RADIUS(CM)	0.27940	0.31750
DISPLACEMENT(CM)	0.00175	0.00199
TEMP.(C)	371.561	371.560
PRESS.(KG/CM**2)	0.28960E 01	0.16800E 01
ALPHA-T(-)	0.00627	
POISSON RATIO(-)	0.33308	
YOUNG M.(KG/CM**2)	0.17501E 07	
STRESS(KG/CM**2)		
RADIAL	-0.22363E 01	
CIRCUMFERENTIAL	0.72244E 01	
AXIAL	0.20984E 01	
TOTAL STRAIN(-)		
RADIAL	0.62660E-02	

CIRCUMFERENTIAL	0.62732E-02
AXIAL	0.62693E-02
CREEP STRAIN(-)	
RADIAL	-0.65581E-15
CIRCUMFERENTIAL	0.69342E-15
AXIAL	-0.37613E-16
CREEP IN DELT(-)	
RADIAL	-0.65581E-15
CIRCUMFERENTIAL	0.69342E-15
AXIAL	-0.37613E-16
SWELLING STRAIN(-)	
TOTAL	0.63787E-15

PERCENT CHANGE IN ORIGINAL O.D. = 0.0 (0/0)

AVERAGE POISSON RATIO(-) = 0.33308

AVERAGE YOUNG MODULUS(KG/CM**2) = 0.17501E 07

+++ FUEL (I=1,NFR) +++

RADIUS(CM)	0.00101	0.00200	0.00300	0.27440
DISPLACEMENT(CM)	0.00000	0.00001	0.00001	0.00083
TEMP.(C)	371.585	371.585	371.585	371.568
PRESS.(KG/CM**2)	0.28939E 01	0.30113E 01	0.30144E 01	0.28960E 01
ALPHA-T(-)	0.00303	0.00303	0.00303	
POISSON RATIO(-)	0.31705	0.31550	0.31550	
YOUNG M.(KG/CM**2)	0.22370E 07	0.21741E 07	0.21741E 07	
NUMBER OF CRACK				
RADIAL	0	0	0	
CIRCUMFERENTIAL	0	0	0	
AXIAL	0	0	0	
STRESS(KG/CM**2)				
RADIAL	-0.29779E 01	-0.30132E 01	-0.28961E 01	
CIRCUMFERENTIAL	-0.31246E 01	-0.30203E 01	-0.28958E 01	
AXIAL	-0.34251E 01	-0.33525E 01	-0.31100E 01	
TOTAL STRAIN(-)				
RADIAL	0.30298E-02	0.30298E-02	0.30297E-02	
CIRCUMFERENTIAL	0.30297E-02	0.30298E-02	0.30297E-02	
AXIAL	0.30296E-02	0.30296E-02	0.30296E-02	
CREEP STRAIN(-)				
RADIAL	0.29693E-31	0.17317E-31	0.10684E-31	
CIRCUMFERENTIAL	0.76894E-32	0.16254E-31	0.10723E-31	
AXIAL	-0.37383E-31	-0.33571E-31	-0.21407E-31	
CREEP IN DELT(-)				
RADIAL	0.29693E-31	0.17317E-31	0.10684E-31	
CIRCUMFERENTIAL	0.76894E-32	0.16254E-31	0.10723E-31	
AXIAL	-0.37383E-31	-0.33571E-31	-0.21407E-31	
SWELLING STRAIN(-)				
TOTAL	0.72064E-09	0.39549E-09	0.17058E-09	
SOLID COMPONENT	0.21132E-09	0.21129E-09	0.50588E-09	
GAS COMPONENT	0.19506E-08	0.97517E-09	0.58502E-11	
HOT PRESSING	0.0	0.0	0.0	

PERCENT CHANGE IN ORIGINAL O.D. = 0.0 (0/0)

AVERAGE POISSON RATIO(-) = 0.31550

AVERAGE YOUNG MODULUS(KG/CM**2) = 0.21741E 07

GAP SIZE(CM) = 0.00592

FRICTION FORCE(KG/CM) = 0.0 (OPEN)

+++ PLUTONIUM MIGRATION (I=1,NP) +++

RADIUS(CM)	0.00101	0.08678	0.12272	0.15030	0.17355	0.19403	0.21255	0.22958	0.24543	0.26032	0.27440
TEMP.(C)	371.585	371.583	371.582	371.580	371.578	371.576	371.575	371.573	371.571	371.570	371.568
PU CONC.(-)	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	

*** AXIAL SECTION = 3 ***

+++ CAPSULE (I=1,NCL) +++

RADIUS(CM)	0.42540	0.47620
TEMP.(C)	371.858	371.857
ALPHA-T(-)	0.00627	

+++ CLAD (I=1,NCL) +++

RADIUS(CM)	0.27940	0.31750
DISPLACEMENT(CM)	0.00175	0.00199
TEMP.(C)	371.859	371.858
PRESS.(KG/CM**2)	0.28960E 01	0.16800E 01
ALPHA-T(-)	0.00627	
POISSON RATIO(-)	0.33311	
YOUNG M.(KG/CM**2)	0.17499E 07	
STRESS(KG/CM**2)		
RADIAL	-0.22363E 01	
CIRCUMFERENTIAL	0.72244E 01	
AXIAL	0.21869E 01	
TOTAL STRAIN(-)		
RADIAL	0.62715E-02	
CIRCUMFERENTIAL	0.62787E-02	
AXIAL	0.62749E-02	
CREEP STRAIN(-)		
RADIAL	-0.58381E-15	
CIRCUMFERENTIAL	0.60964E-15	
AXIAL	-0.25828E-16	
CREEP IN DELT(-)		
RADIAL	-0.58381E-15	
CIRCUMFERENTIAL	0.60964E-15	
AXIAL	-0.25828E-16	
SWELLING STRAIN(-)		
TOTAL	0.53512E-15	

PERCENT CHANGE IN ORIGINAL O.D. = 0.0 (O/O)

AVERAGE POISSON RATIO(-) = 0.33311

AVERAGE YOUNG MODULUS(KG/CM**2) = 0.17499E 07

+++ FUEL (I=1,NFR) +++

RADIUS(CM)	0.00101	0.00200	0.00300	0.27440
DISPLACEMENT(CM)	0.00000	0.00001	0.00001	0.00083
TEMP.(C)	371.880	371.880	371.880	371.866
PRESS.(KG/CM**2)	0.28939E 01	0.29989E 01	0.30016E 01	0.28960E 01
ALPHA-T(-)	0.00303	0.00303	0.00303	
POISSON RATIO(-)	0.31705	0.31550	0.31550	
YOUNG M.(KG/CM**2)	0.22369E 07	0.21740E 07	0.21740E 07	
NUMBER OF CRACK				
RADIAL	0	0	0	
CIRCUMFERENTIAL	0	0	0	
AXIAL	0	0	0	
STRESS(KG/CM**2)				

RADIAL	-0.29690E 01	-0.30006E 01	-0.28961E 01
CIRCUMFERENTIAL	-0.31001E 01	-0.30070E 01	-0.28959E 01
AXIAL	-0.32741E 01	-0.32059E 01	-0.29942E 01
TOTAL STRAIN(-)			
RADIAL	0.30325E-02	0.30324E-02	0.30324E-02
CIRCUMFERENTIAL	0.30324E-02	0.30324E-02	0.30324E-02
AXIAL	0.30323E-02	0.30323E-02	0.30323E-02
CREEP STRAIN(-)			
RADIAL	0.21816E-31	0.10586E-31	0.48960E-32
CIRCUMFERENTIAL	0.21448E-32	0.96273E-32	0.49309E-32
AXIAL	-0.23961E-31	-0.20214E-31	-0.98274E-32
CREEP IN DELT(-)			
RADIAL	0.21816E-31	0.10586E-31	0.48960E-32
CIRCUMFERENTIAL	0.21448E-32	0.96273E-32	0.49309E-32
AXIAL	-0.23961E-31	-0.20214E-31	-0.98274E-32
SWELLING STRAIN(-)			
TOTAL	0.63770E-09	0.34996E-09	0.15088E-09
SOLID COMPONENT	0.18692E-09	0.18689E-09	0.44748E-09
GAS COMPONENT	0.17262E-08	0.86298E-09	0.51771E-11
HOT PRESSING	0.0	0.0	0.0

PERCENT CHANGE IN ORIGINAL O.D. = 0.0 (O/O)

AVERAGE POISSON RATIO(-) = 0.31550

AVERAGE YOUNG MODULUS(KG/CM**2) = 0.21740E 07

GAP SIZE(CM) = 0.00592
FRICTION FORCE(KG/CM) = 0.0 (OPEN)

+++ PLUTONIUM MIGRATION (I=1,NP) +++

RADIUS(CM)	0.00101	0.08678	0.12272	0.15030	0.17355	0.19403	0.21255	0.22958	0.24543	0.26032	0.27440
TEMP.(C)	371.880	371.879	371.877	371.876	371.875	371.873	371.872	371.870	371.869	371.867	371.866
PU CONC.(-)	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	0.20000	

*** AXIAL SECTION = 4 ***

+++ CAPSULE (I=1,NCL) +++

RADIUS(CM)	0.42540	0.47620
TEMP.(C)	372.100	372.100
ALPHA-T(-)	0.00628	

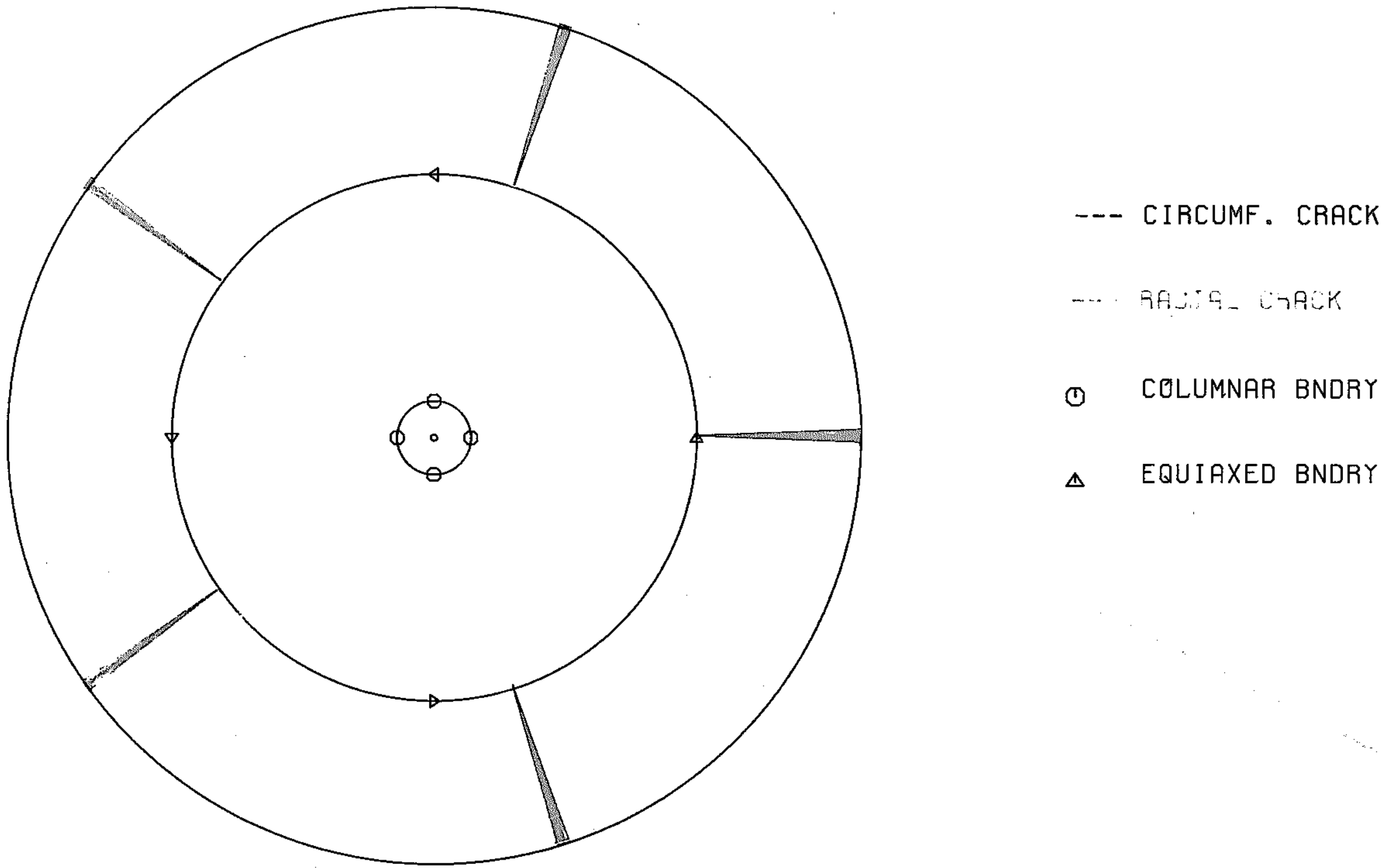
+++ CLAD (I=1,NCL) +++

RADIUS(CM)	0.27940	0.31750
DISPLACEMENT(CM)	0.00176	0.00199
TEMP.(C)	372.100	372.100
PRESS.(KG/CM**2)	0.28960E 01	0.16800E 01
ALPHA-T(-)	0.00628	
POISSON RATIO(-)	0.33313	
YOUNG M.(KG/CM**2)	0.17496E 07	
STRESS(KG/CM**2)		
RADIAL	-0.22363E 01	
CIRCUMFERENTIAL	0.72244E 01	
AXIAL	0.23743E 01	
TOTAL STRAIN(-)		
RADIAL	0.62759E-02	
CIRCUMFERENTIAL	0.62831E-02	
AXIAL	0.62795E-02	

```
CREEP STRAIN(-)
  RADIAL                          -0.18715E-15
  CIRCUMFERENTIAL                  0.19033E-15
  AXIAL                           -0.31848E-17
CREEP IN DELT(-)
  RADIAL                          -0.18715E-15
  CIRCUMFERENTIAL                  0.19033E-15
  AXIAL                           -0.31848E-17
SWELLING STRAIN(-)
  TOTAL                            0.95831E-16

PERCENT CHANGE IN ORIGINAL O.D. =         0.0       (0/0)

AVERAGE POISSON RATIO(-)        =         0.33313

AVERAGE YOUNG MODULUS(KG/CM**2) =     0.17496E 07

CPU TIME =    0.40 (SEC)
```

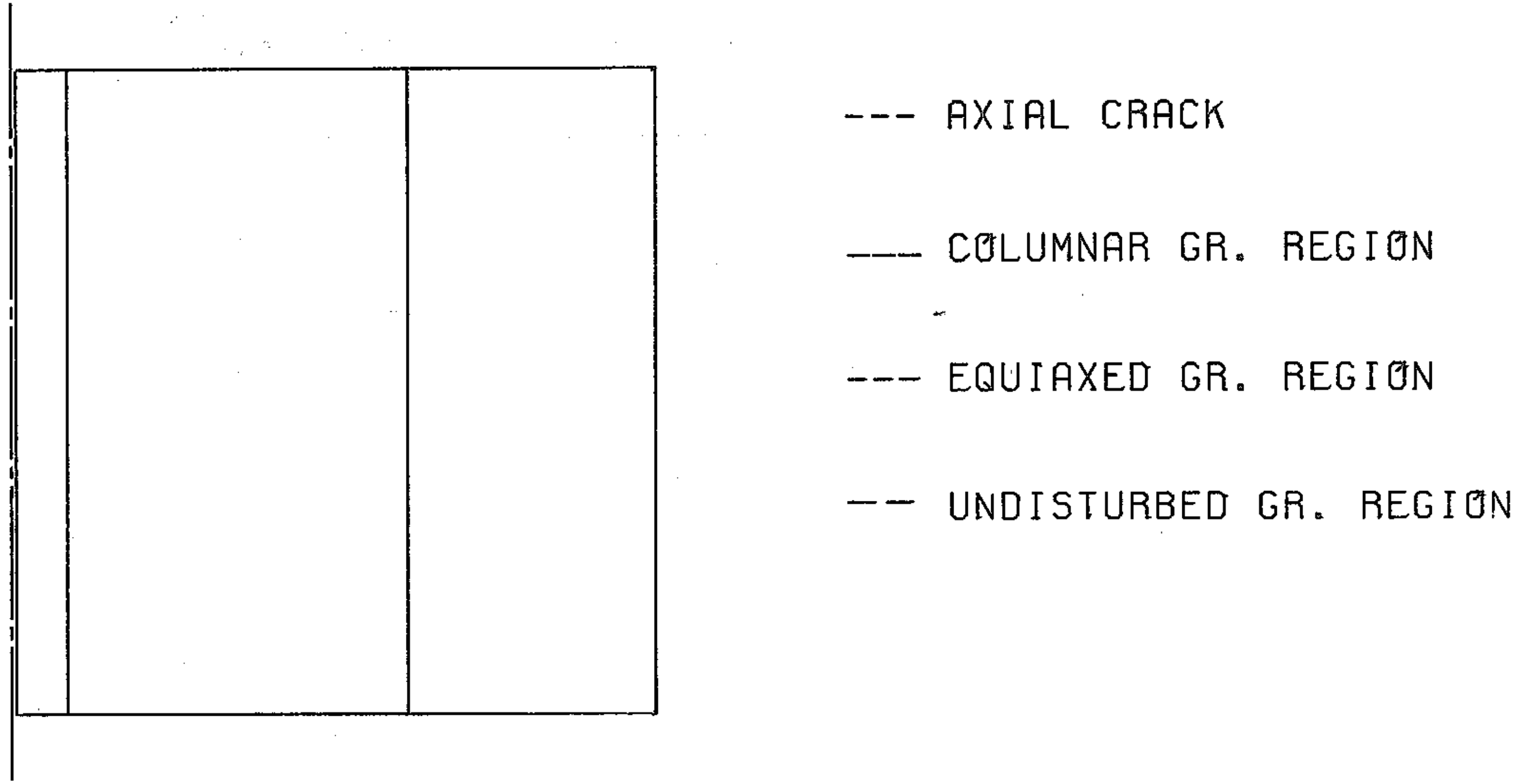
** RUN DATE **
7/31/75

ACTIVE-II



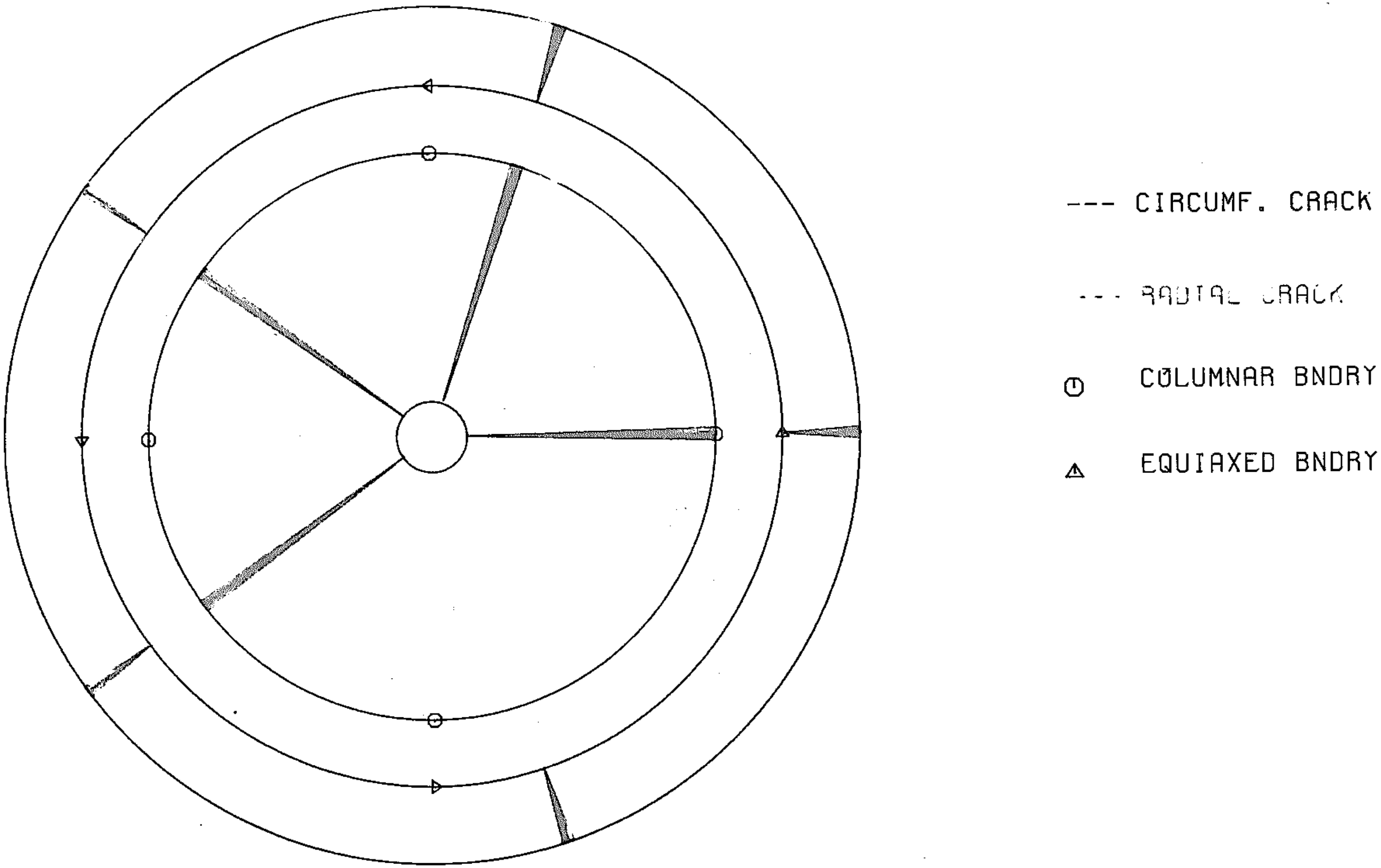
R-C-DIRECTION CRACK

AXIAL SECTION = 2   STEP = 36



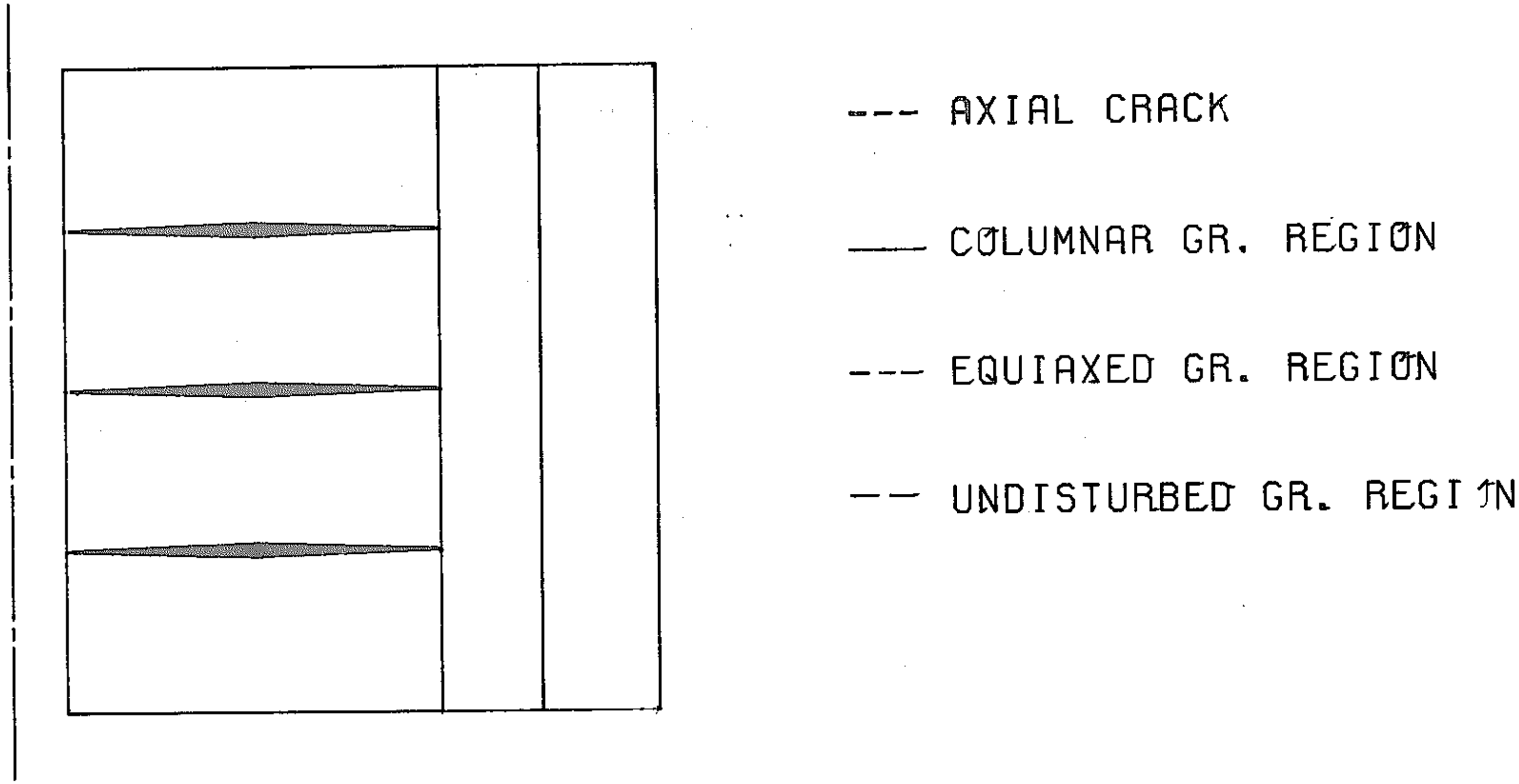
Z-DIRECTION CRACK

AXIAL SECTION = 2   STEP = 36



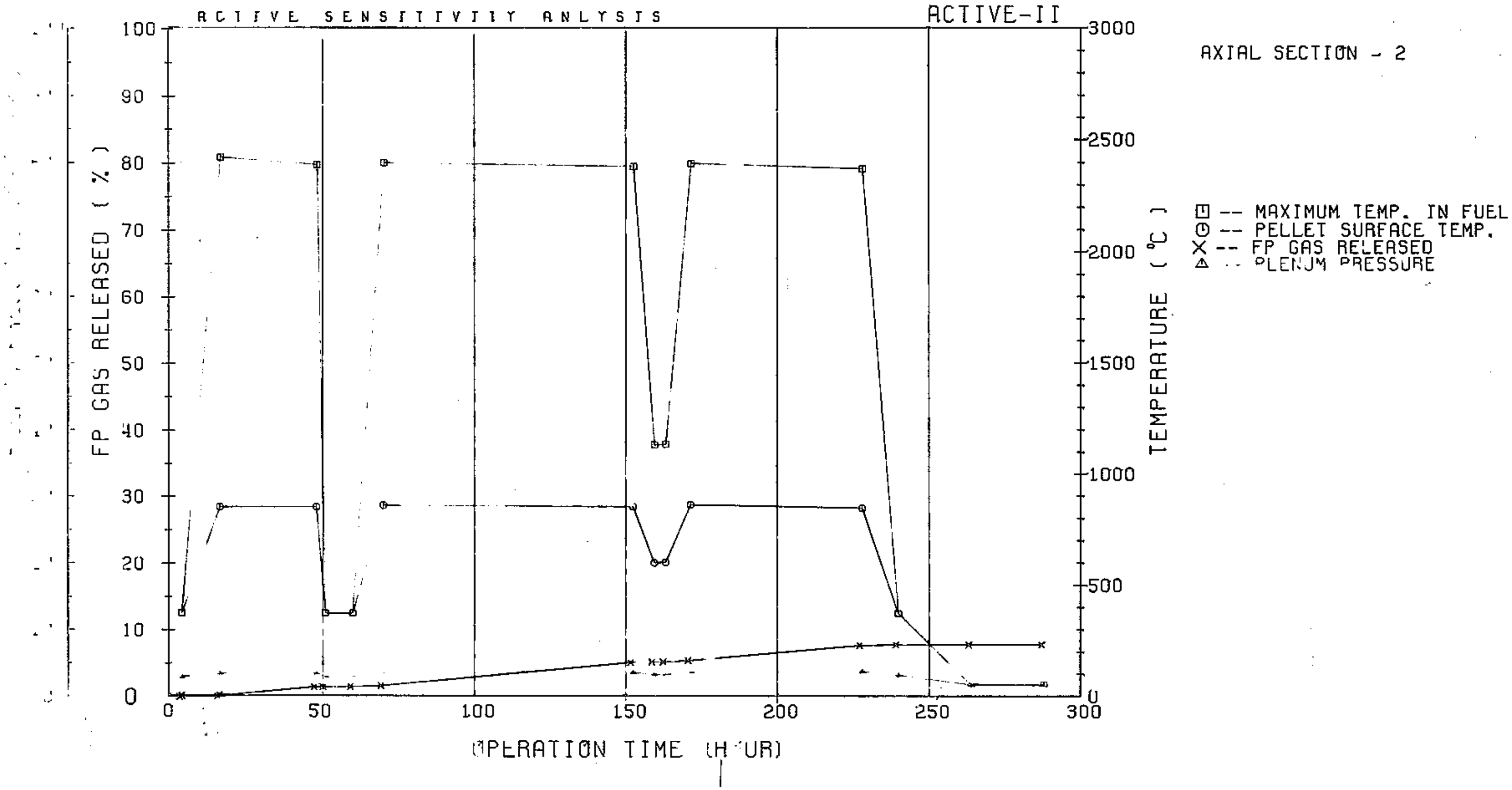
R-C-DIRECTION CRACK

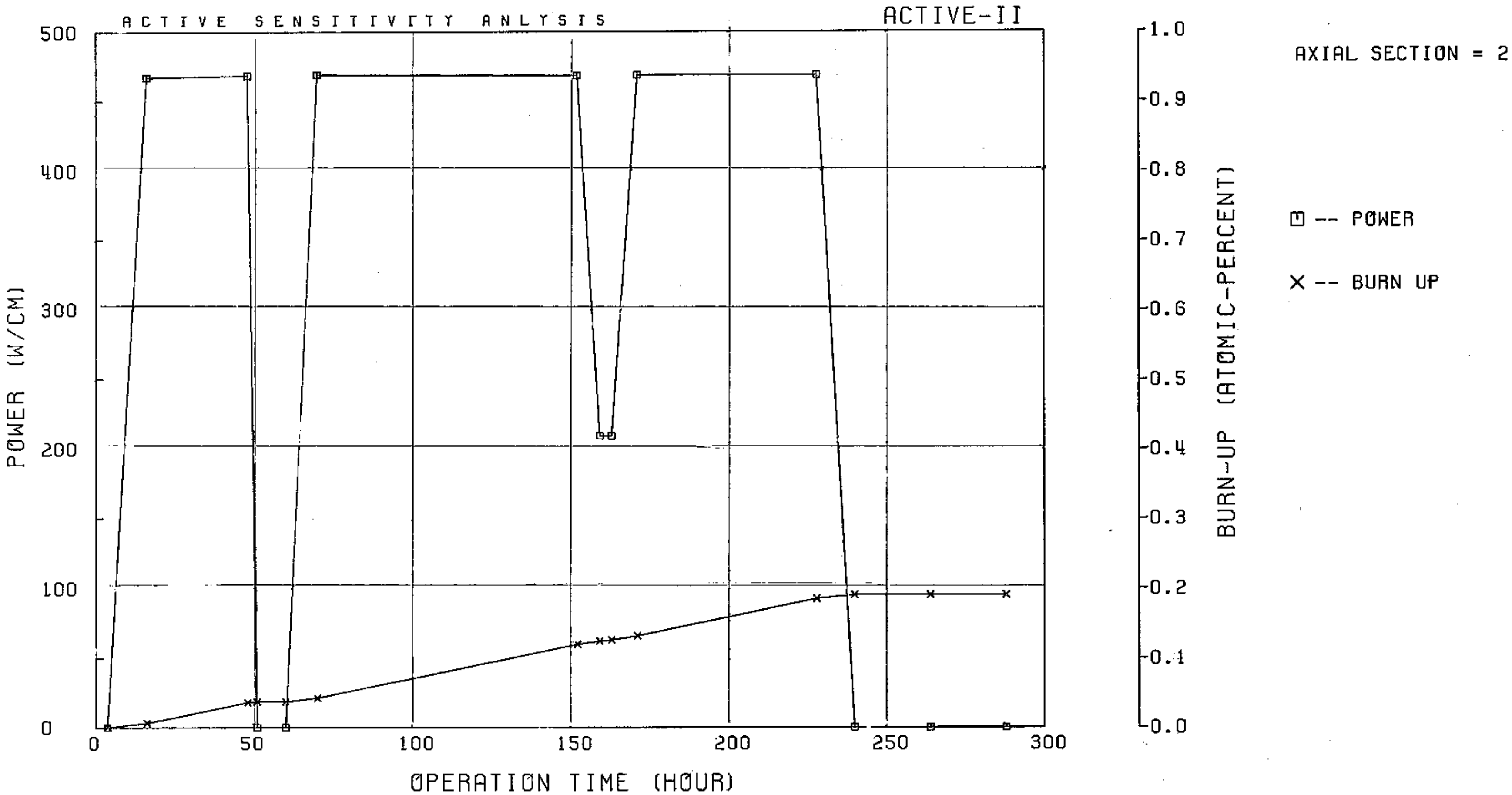
AXIAL SECTION = 2 STEP - 122

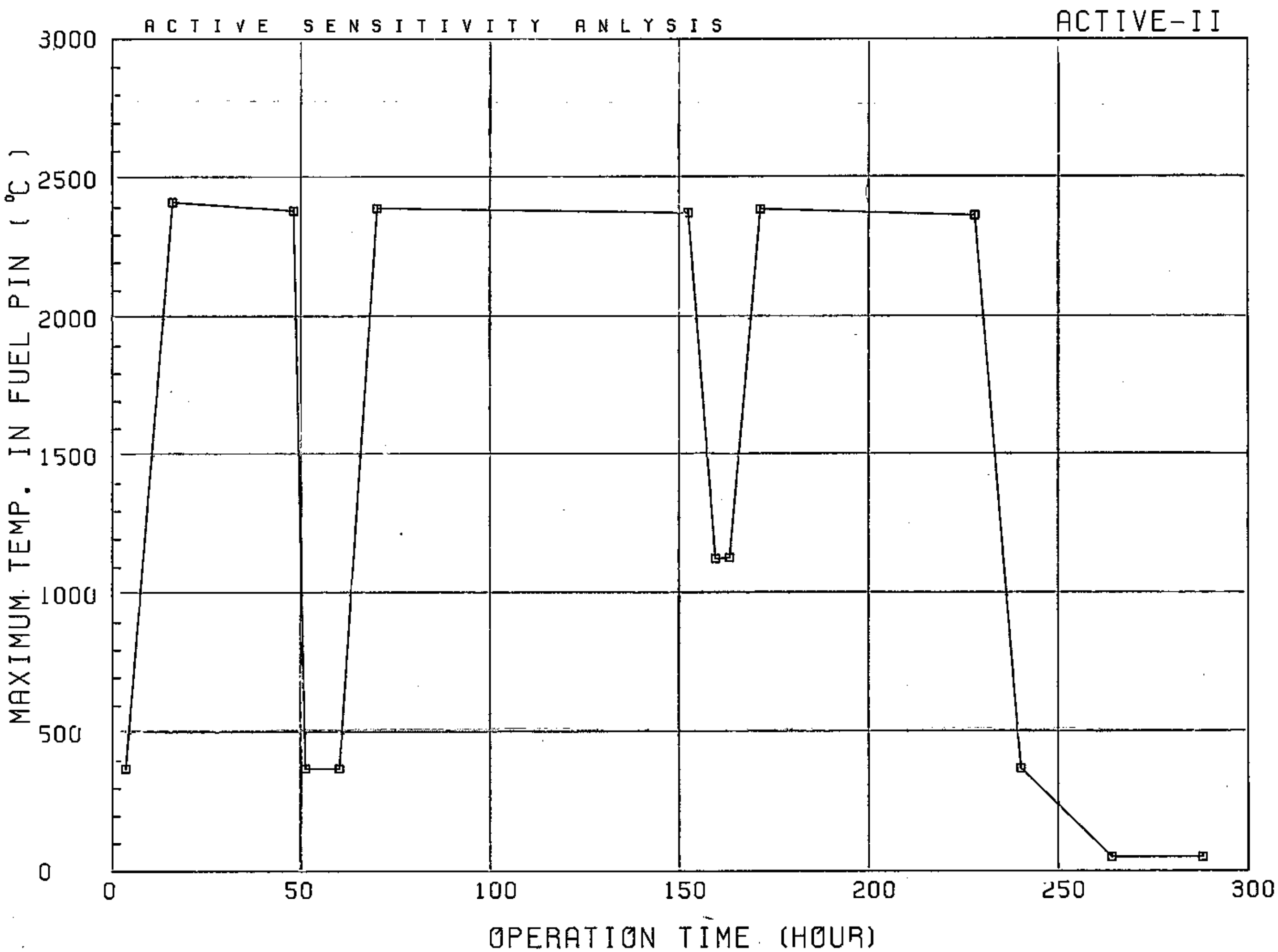


Z-DIRECTION CRACK

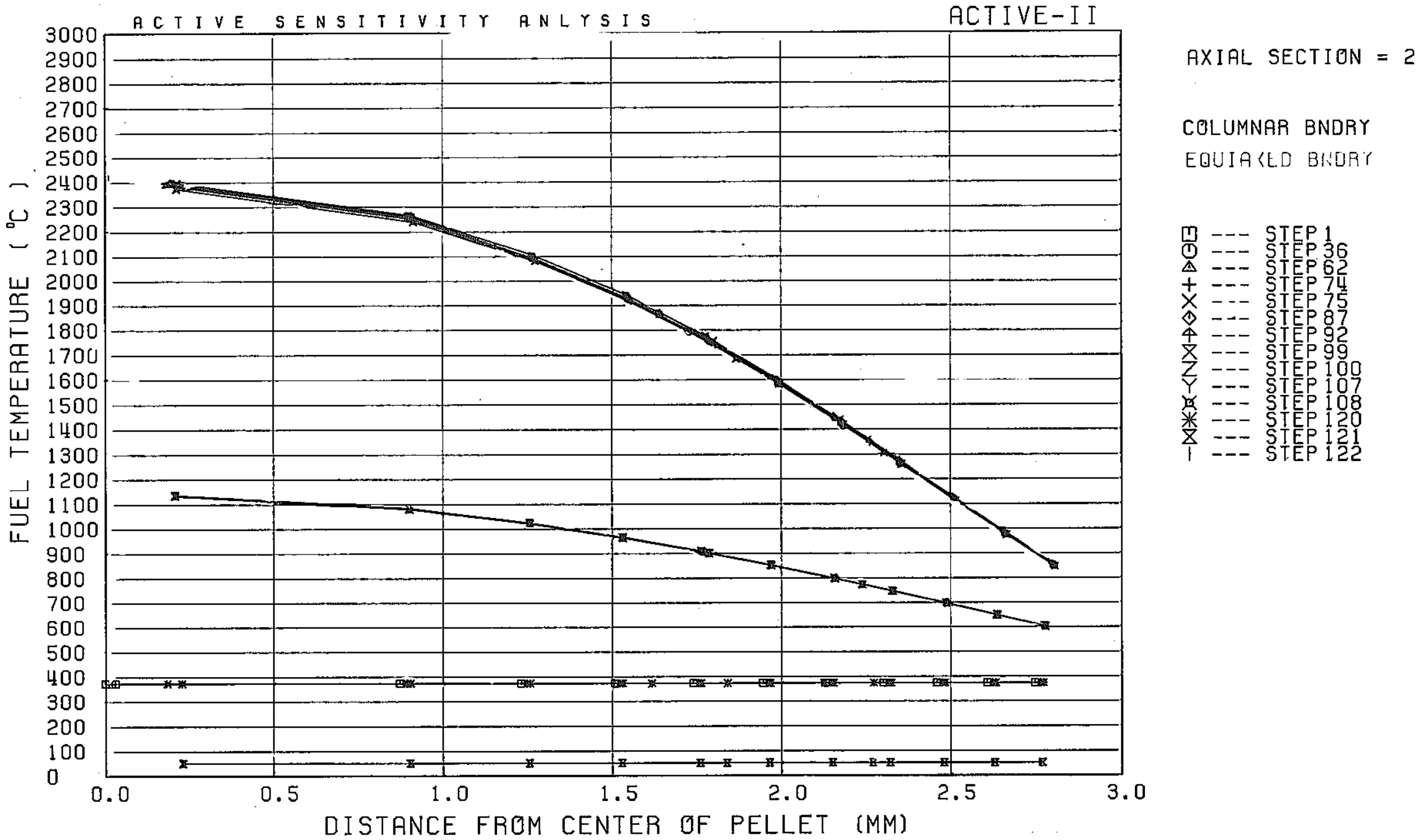
AXIAL SECTION = 2   STEP - 122

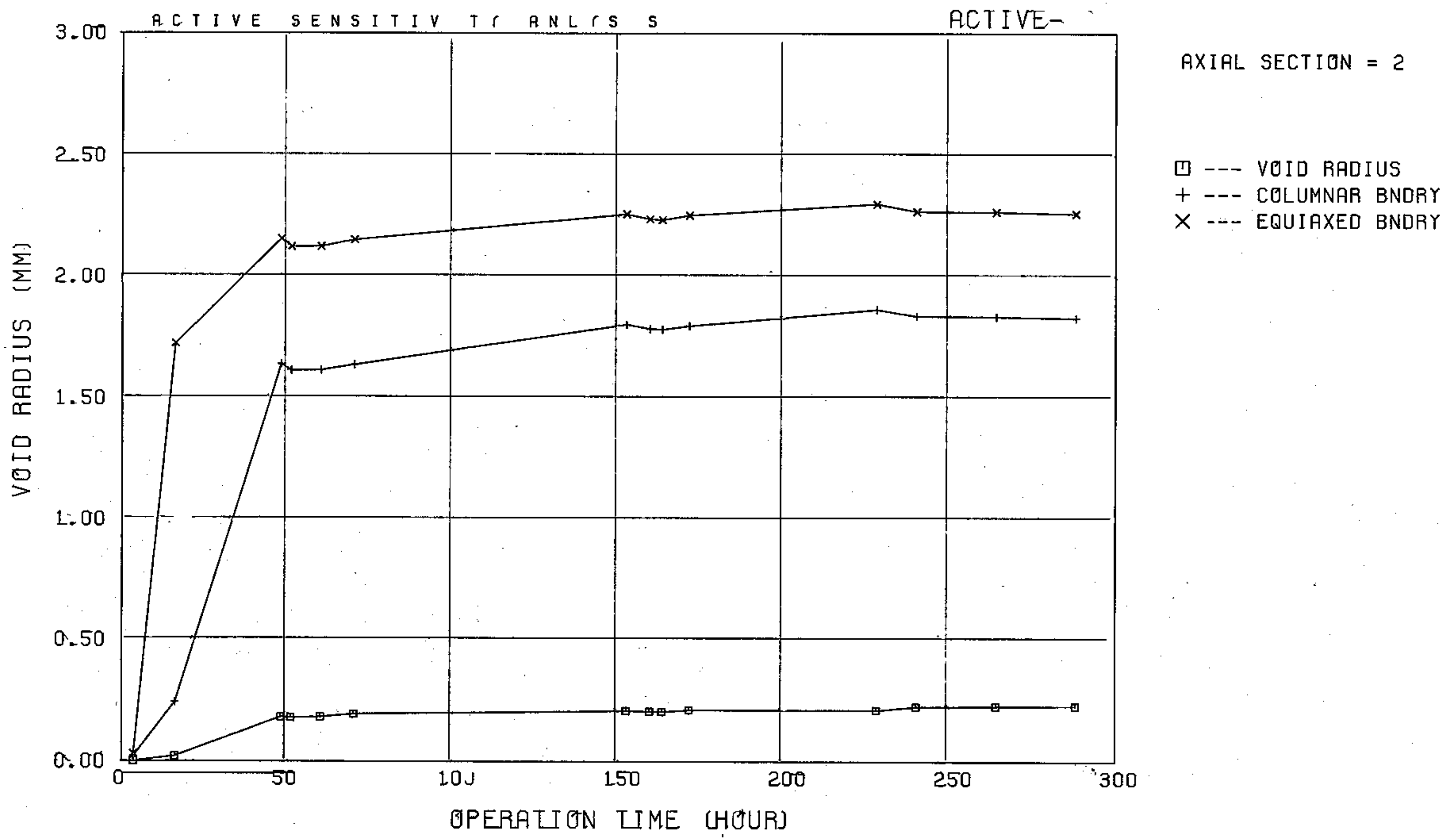

ACTIVE SENSITIVITY ANLYSIS
ACTIVE-II
AXIAL SECTION - 2
FP GAS RELEASED (%)
TEMPERATURE (°C)
OPERATION TIME (HOUR)
0
10
20
30
40
50
60
70
80
90
100
0
500
1000
1500
2000
2500
3000
0
50
100
150
200
250
300
-- MAXIMUM TEMP. IN FUEL
-- PELLET SURFACE TEMP.
-- FP GAS RELEASED
.. PLENUM PRESSURE

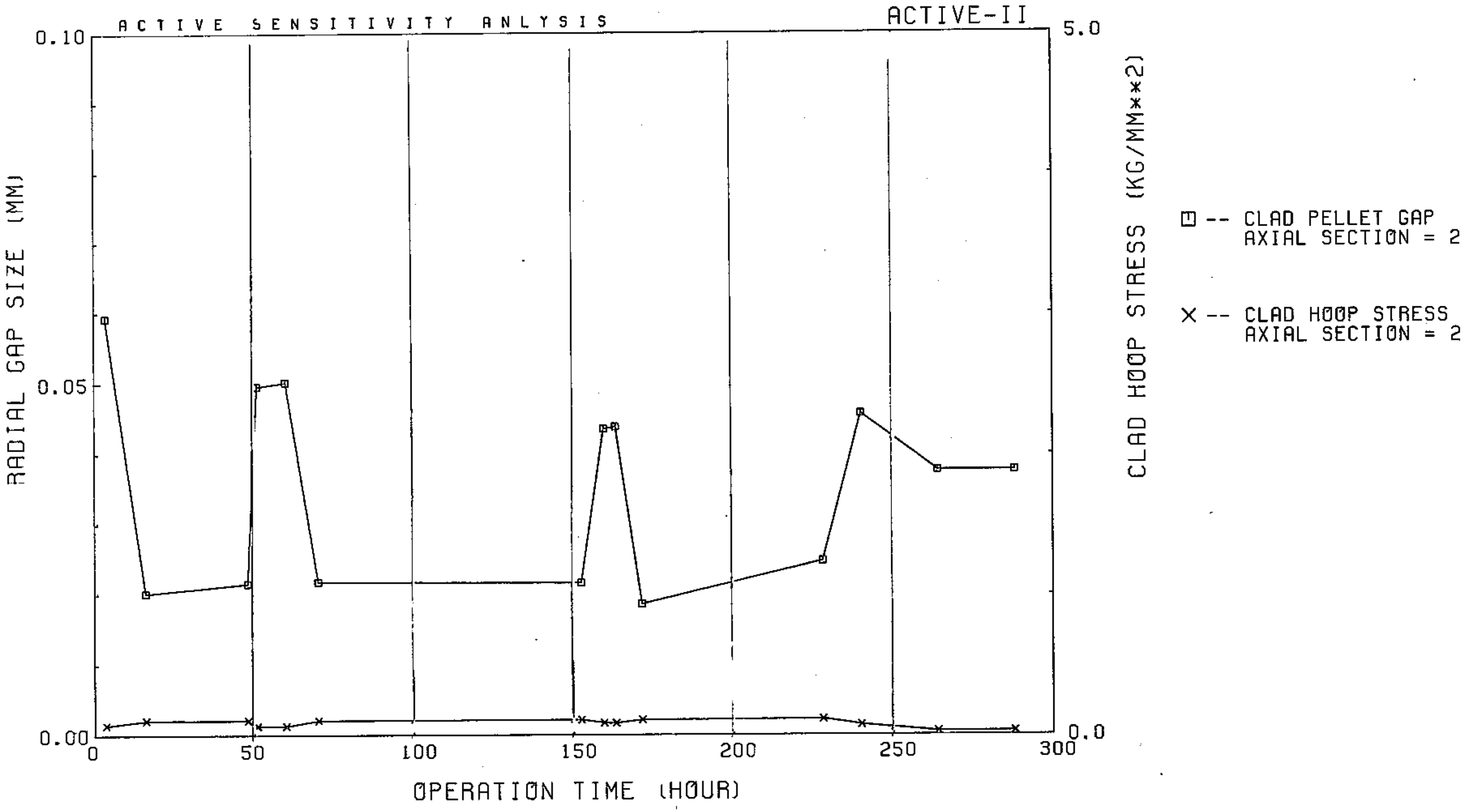

ACTIVE SENSITIVITY ANLYSIS
ACTIVE-II
AXIAL SECTION = 2
POWER
BURN UP
POWER (W/CM)
BURN-UP (ATOMIC-PERCENT)
OPERATION TIME (HOUR)
0
100
200
300
400
500
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
50
150
250

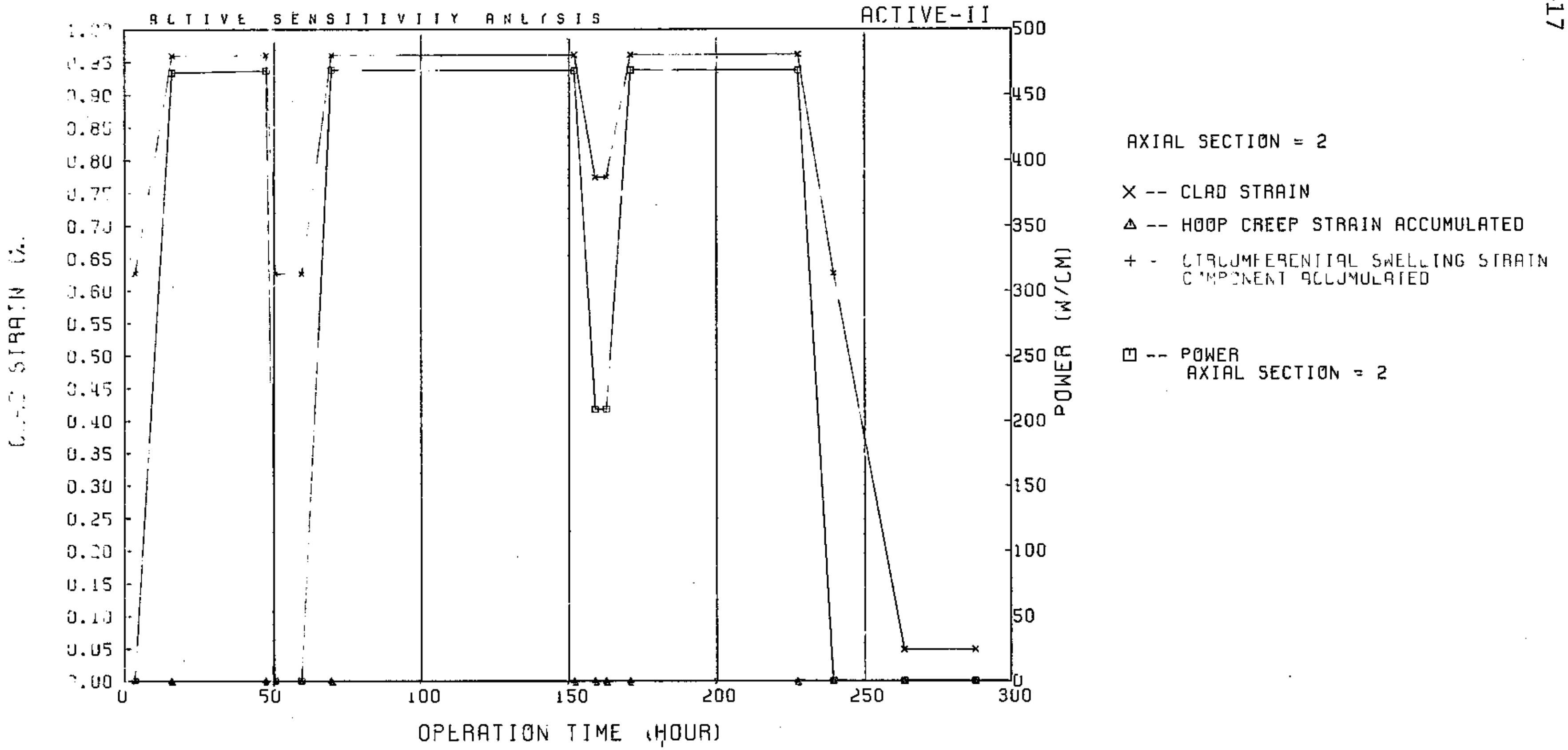


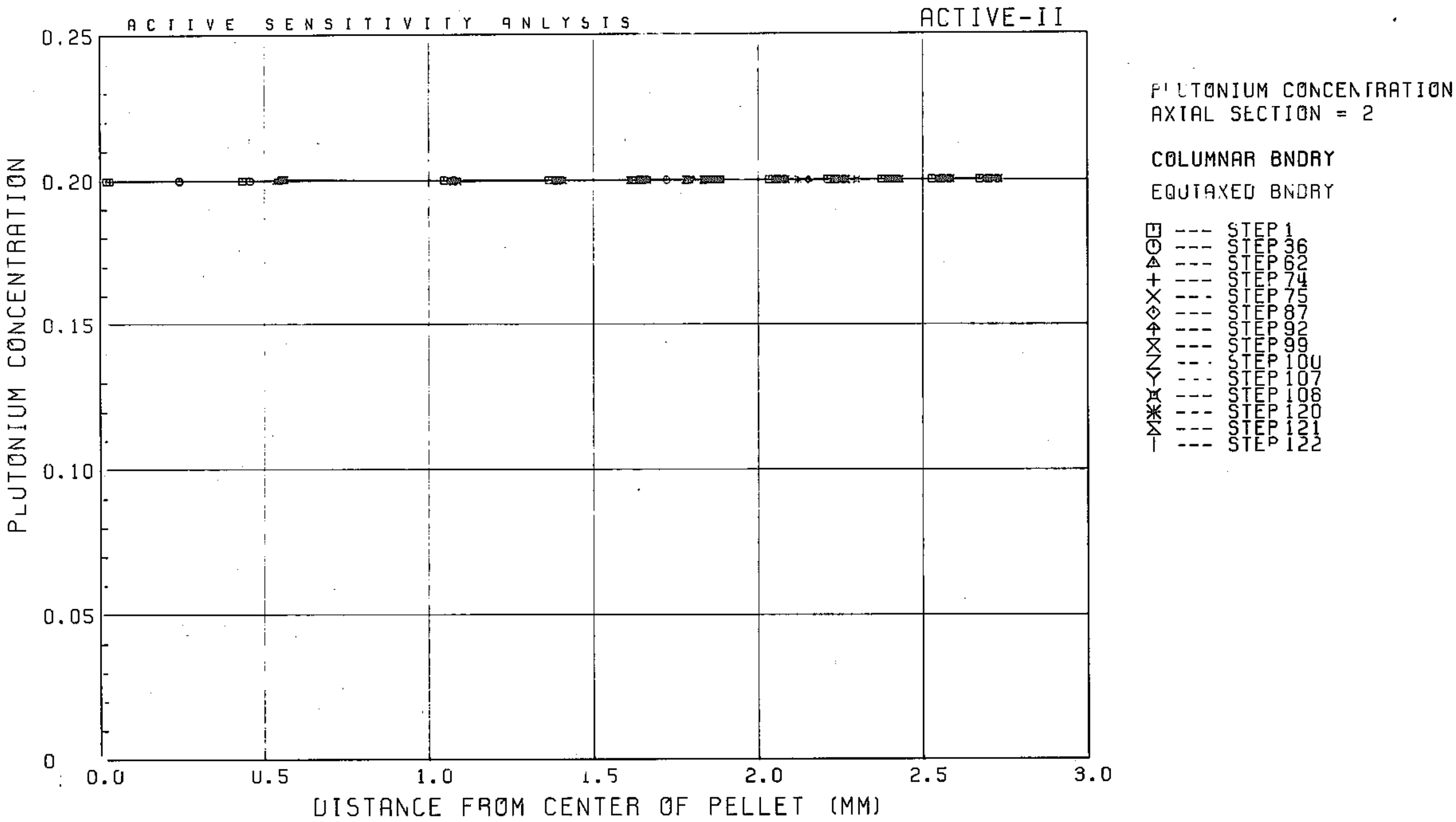
AXIAL SECTION = 2


ACTIVE SENSITIVITY ANLYSIS
ACTIVE-II
AXIAL SECTION = 2
COLUMNAR BNDRY
EQUIAXED BNDRY
STEP 1
STEP 36
STEP 62
STEP 74
STEP 75
STEP 87
STEP 92
STEP 99
STEP 100
STEP 107
STEP 108
STEP 120
STEP 121
STEP 122
FUEL TEMPERATURE (°C)
DISTANCE FROM CENTER OF PELLET (MM)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000


ACTIVE SENSITIV T ANL S S
ACTIVE-
AXIAL SECTION = 2
VOID RADIUS
COLUMNAR BNDRY
EQUIAXED BNDRY
VOID RADIUS (MM)
OPERATION TIME (HOUR)
0.00
0.50
1.00
1.50
2.00
2.50
3.00
0
50
100
150
200
250
300


ACTIVE SENSITIVITY ANLYSIS
ACTIVE-II
RADIAL GAP SIZE (MM)
0.10
0.05
0.00
CLAD HOOP STRESS (KG/MM**2)
5.0
0.0
0
50
100
150
200
250
300
OPERATION TIME (HOUR)
CLAD PELLET GAP AXIAL SECTION = 2
CLAD HOOP STRESS AXIAL SECTION = 2


ACTIVE SENSITIVITY ANALYSIS
ACTIVE-II
OPERATION TIME (HOUR)
POWER (W/CM)
AXIAL SECTION = 2
X -- CLAD STRAIN
Δ -- HOOP CREEP STRAIN ACCUMULATED
+ - CIRCUMFERENTIAL SWELLING STRAIN COMPONENT ACCUMULATED
□ -- POWER AXIAL SECTION = 2


ACTIVE SENSITIVITY ANLYSIS
ACTIVE-II
PLUTONIUM CONCENTRATION
AXIAL SECTION = 2
COLUMNAR BNDRY
EQUIAXED BNDRY
--- STEP 1
--- STEP 36
--- STEP 62
--- STEP 74
--- STEP 75
--- STEP 87
--- STEP 92
--- STEP 99
--- STEP 100
--- STEP 107
--- STEP 108
--- STEP 120
--- STEP 121
--- STEP 122
PLUTONIUM CONCENTRATION
0.25
0.20
0.15
0.10
0.05
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
DISTANCE FROM CENTER OF PELLET (MM)

APPENDIX G: PROGRAM LIST

Appendix G: Program List

The program is available from;
Department of Nuclear Fuels Development,
Power Reactor and Nuclear Fuel Development Corporation
1-9-13 Akasaka, Minato-ku, Tokyo, JAPAN

```
      COMMON /DIM/ IA,IAM                                              00001300
      DIMENSION A(4000),AM(6000)                                       00001400
      IA=4000                                                          00001500
      IAM=6000                                                         00001600
      CALL ACTIVE(A,AM)                                                00001700
      STOP                                                             00001800
      END                                                              00001900
```

```
      SUBROUTINE ACTIVE ( A,AM )
C
C
C=======================================================================
C=============================                   =======================
C=============================  ACTIVE-II  =============================
C=============================                   =======================
C=======================================================================
C==                                                                  ==
C== A COMPUTER ANALYSIS OF FAST/THERMAL-REACTOR FUEL-ELEMENT BEHAVIOR ==
C== AS A FUNCTION OF REACTOR OPERATING HISTORY .                      ==
C==                                                                  ==
C=======================================================================
C
C
      REAL*8 DAYDAY
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /CONST/ FKA1,FKB1,FKC1,FKA2,FKB2,FKC2,FKA3,FKB3,FKC3,
     *              FA11,FA12,FA21,FA22,FA31,FA32,
     *              CK1,CK2,CA1,CA2,XEWTON(7)
      COMMON /CAPSUL/ RCPI,RCPO,AFNA
      COMMON /IOFIL/ I05,I06,I07
      COMMON /FBRATR/ JFA
      COMMON /DIM/  IA,IAM
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      COMMON /DAY/ DAYDAY
C
      DIMENSION  A( 1 ) , AM( 1 )
C
      I05 = 5
      I06 = 6
      I07 = 7
      ISPLOT = 0
C
C---- TO GET THE DATE
C
      CALL DATE ( DAYDAY )
C
 6000 CONTINUE
C
C---- TIMER ZERO CLEAR
C
      CALL STIMER(2)
C
C
C---- CARD  1
C
      READ(I05,9006,END=998) JFA,JOUT,EPSSPD,EPSCRF,EPSCRC
C
C
```

```
      IF ( EPSSPD.EQ.0. ) EPSSPD=0.01
      IF ( EPSCRF.EQ.0. ) EPSCRF=0.05
      IF ( EPSCRC.EQ.0. ) EPSCRC=0.01
C
C     JFA = 1 ... FBR
C     JFA = 2 ... ATR
C     JFA = 3 ... PWR
C     JFA = 4 ... BWR
C
C     JOUT = 0 ... OUTPUT EVERY STEP OF CARD INPUT
C     JOUT = 1 ... OUTPUT EVERY STEP CALCULATED
C
C     EPSSPD = LIMIT FOR RESTRUCTURING VELOCITY (CM/SEC)
C     EPSCRF = LIMIT FOR FUEL CREEP STRAIN (-)
C     EPSCRC = LIMIT FOR CLADDING CREEP STRAIN (-)
C
      GO TO ( 10,20,30,40 ) , JFA
   10 WRITE(I06,9004)
      GO TO 50
   20 WRITE(I06,9005)
      GO TO 50
   30 WRITE(I06,9007)
      GO TO 50
   40 WRITE(I06,9008)
   50 CONTINUE
C
C---- CARD 2    ...    TITLE CARD  (  CONTINUED TILL *END CARD  )
C
      CALL HEADER
C
C
C---- CARD 3    ...    CONTROL CARD
C
C
      READ(I05,9000) NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *               NSKIP,NBUF
C
C
C     NN = NUMBER OF AXIAL SECTION
C     NCR = NUMBER OF MESH FOR COLUMNAR GRAIN REGION
C     NER = NUMBER OF MESH FOR EQUIAXED GRAIN REGION
C     NUR = NUMBER OF MESH FOR UNDISTURBED REGION
C     NCL = NUMBER OF MESH FOR CLAD REGION
C     NP = NUMBER OF MESH FOR PLUTONIUM MIGRATION
C     NOK = FUEL THERMAL CONDUCTIVITY OPTION
C     1/2/3 = SAYLES/BAILY-ASAMOTO/BIANCHERIA
C     NOS = CLAD SWELLING OPTION
C     -1/-2 = ANNEALED 304,316S.S./20P.C.COLD WORK 316S.S.
C     NOC = CLAD CREEP OPTION
C     -1/-2 = BOLTAX,BIANCHERIA,GUHA/MODIFIED
C     NPLOT = PLOT OPTION
C     0/1 = NO PLOT/PLOT
C     NCRK = 0   ...   NO CRACK CALC. AND 2 SIGE LOOPS
C     NCRK = 1   ...   CRACK CALC. AND 2 SIGE LOOPS
C     NSKIP = COOLANT TEMPERATURE INPUT OPTION
C     0/1 = NO INPUT/INPUT TEMPERATURE DISTRIBUTION
C     NBUF = STORAGE(WORDS) REQUIRED FOR PLOTTER
C     IF NBUF=0,THEN CODE SETS NBUF 50
```

```
C                                                                       00011700
C                                                                       00011800
C===========  LOCATION SET FOR DIMENSION A  ===========================00011900
C                                                                       00012000
      NFR = NCR+NER+NUR                                                 00012100
      NFR1 = NFR+1                                                      00012200
      NCL1 = NCL + 1                                                    00012300
      NTR= NFR+NCL                                                      00012400
      NTR1=NTR+1                                                        00012500
      NTR2 = NTR+2                                                      00012600
      NNTR = NN*NTR                                                     00012700
      NNTR2 = NN*NTR2                                                   00012800
      NNFR = NN*NFR                                                     00012900
      NNFR1 = NN*NFR1                                                   00013000
      NNCL = NN*NCL                                                     00013100
      NNCL1 = NN*NCL1                                                   00013200
      NP1 = NP + 1                                                      00013300
      NNP = NN*NP                                                       00013400
      NNP1 = NN*NP1                                                     00013500
      NNF = NN-1                                                        00013600
      N1 = NTR                                                          00013700
      N2 = NTR*2                                                        00013800
      N3 = NTR*3                                                        00013900
      N4 = NTR*4                                                        00014000
      N5 = NTR*5                                                        00014100
      N6 = NTR*6                                                        00014200
      N7 = NTR*7                                                        00014300
      N8 = NTR*8                                                        00014400
      N9 = NTR*9                                                        00014500
      N10 = NTR*10                                                      00014600
      N11 = NTR*11                                                      00014700
      N12 = NTR*12                                                      00014800
      N13 = NTR*13                                                      00014900
      N14 = NTR*14                                                      00015000
      N15 = NTR*15                                                      00015100
      N1C = NCL                                                         00015200
      N2C = NCL*2                                                       00015300
      N3C = NCL*3                                                       00015400
      N4C = NCL*4                                                       00015500
      N5C = NCL*5                                                       00015600
      N6C = NCL*6                                                       00015700
      N7C = NCL*7                                                       00015800
      N8C = NCL*8                                                       00015900
      N9C = NCL*9                                                       00016000
      N10C = NCL*10                                                     00016100
      N11C = NCL*11                                                     00016200
      N12C = NCL*12                                                     00016300
      N13C = NCL*13                                                     00016400
      N14C = NCL*14                                                     00016500
      N15C = NCL*15                                                     00016600
C                                                                       00016700
      NIO = 1                                                           00016800
      NEPR = NIO + NN                                                   00016900
      NEPC = NEPR + NNTR                                                00017000
      NEPZ = NEPC + NNTR                                                00017100
      NTPR = NEPZ + NNTR                                                00017200
      NTPC = NTPR + NNTR                                                00017300
      NTPZ = NTPC + NNTR                                                00017400
```

```
      NTSR = NTPZ + NNTR
      NTSC = NTSR + NNTR
      NTSZ = NTSC + NNTR
      NSIGR = NTSZ + NNTR
      NSIGC = NSIGR + NNTR
      NSIGZ = NSIGC + NNTR
      NADIS = NSIGZ + NNTR
      NRDIS = NADIS + NNTR
      NRVB  = NRDIS + NNTR2
      NRCB  = NRVB + NN
      NREB  = NRCB + NN
      NRUB  = NREB + NN
      NRCI  = NRUB + NN
      NRCO  = NRCI + NN
      NQSFF = NRCO + NN
      NQSFP = NQSFF + NN
      NF = NQSFP + NN
      NOTOM = NF + NNP
      NDXM = NOTOM + NN
      NXM = NDXM + 3*NN
      NZFLX = NXM + 3*NN
      NRH = NZFLX + NN
      NRHOP = NRH + NNTR
      NCPU = NRHOP + 3*NN
      NLM = NCPU + NNP
      NEHP = NLM + NNTR
      NSWS = NEHP + NNTR
      NGS = NSWS + NNTR
      NGR = NGS + NNFR
      NQSF = NGR + NN
      NPOW = NQSF + NN
      NXCPI = NPOW + NN
      NXCPO = NXCPI + NN
      NBU = NXCPO + NN
      NBUM = NBU + NN
      NHG = NBUM + NN
      NTCLO = NHG + NN
      NHC = NTCLO + NN
      NTNA = NHC + NN
      NTCPO = NTNA + NN
      NTCPI = NTCPO + NN
      NRCP = NTCPI + NN
      NTCP = NRCP + NNCL1
      NTBCP = NTCP + NNCL1
      NTBCPA = NTBCP + NNCL
      NATBCP = NTBCPA + NN
      NATBPA = NATBCP + NNCL
      NRCL = NATBPA + NN
      NTCL = NRCL + NNCL1
      NTBCL = NTCL + NNCL1
      NTBCLA = NTBCL + NNCL
      NATBCL = NTBCLA + NN
      NATBLA = NATBCL + NNCL
      NYNG = NATBLA + NN
      NPOS = NYNG + NNTR
      NPOSB = NPOS + NNTR
      NSHR = NPOSB + 2*NN
      NSHRB = NSHR + NNTR
```

```
      NBM = NSHRB + 2*NN                                                00023300
      NRFR = NBM + NNTR                                                 00023400
      NTFR = NRFR + NNFR1                                               00023500
      NTBF = NTFR + NNFR1                                               00023600
      NTBFA = NTBF + NNFR                                               00023700
      NATBF = NTBFA + 3*NN                                              00023800
      NATBFA = NATBF + NNFR                                             00023900
      NRFRPM = NATBFA + 3*NN                                            00024000
      NTFRPM = NRFRPM + NNP1                                            00024100
      NGDT = NTFRPM + NNP1                                              00024200
      NTBFPM = NGDT + NNP1                                              00024300
      NATBPM = NTBFPM + NNP                                             00024400
      NFDPM = NATBPM + NNP                                              00024500
      NPRS = NFDPM + NNP                                                00024600
      NFLZO = NPRS + NNTR                                               00024700
      NFLZ = NFLZO + NN                                                 00024800
      NCLZO = NFLZ + NN                                                 00024900
      NCLZ = NCLZO + NN                                                 00025000
      NZL = NCLZ + NN                                                   00025100
      NFD = NZL + NN                                                    00025200
      NSWG = NFD + NNFR                                                 00025300
      NSWS3 = NSWG + NNFR                                               00025400
      NFLNC = NSWS3 + NN*NUR                                            00025500
      NFLUX = NFLNC + NN                                                00025600
      NDEQP = NFLUX + NN                                                00025700
      NEQSIG = NDEQP + NNTR                                             00025800
      NFTHP = NEQSIG + NNTR                                             00025900
      NVOL = NFTHP + NNTR                                               00026000
      NKFRIC = NVOL + NNTR                                              00026100
      NCRIF = NKFRIC + NN                                               00026200
      NRHOT = NCRIF + NN                                                00026300
      NRVBO = NRHOT + NN                                                00026400
      NRCBO = NRVBO + NN                                                00026500
      NREBO = NRCBO + NN                                                00026600
      NRUBO = NREBO + NN                                                00026700
      NRCIO = NRUBO + NN                                                00026800
      NRCOO = NRCIO + NN                                                00026900
      NRFRO = NRCOO + NN                                                00027000
      NRCLO = NRFRO + NNFR1                                             00027100
      NCPUD = NRCLO + NNCL1                                             00027200
      NTOUR = NCPUD + NP                                                00027300
      NTOUO = NTOUR + NN                                                00027400
      NPTOU = NTOUO + NN                                                00027500
      NDELT = NPTOU + NN                                                00027600
      NXMINT = NDELT + NN                                               00027700
      NRDIST = NXMINT +    NN                                           00027800
      NEPSEQ = NRDIST + 4*NN                                            00027900
      NRBAR = NEPSEQ + NN*NTR                                           00028000
      NWOLD = NRBAR + NNTR                                              00028100
      NWNEW = NWOLD + NNFR                                              00028200
      NXEPR = NWNEW + NFR                                               00028300
      NXEPC = NXEPR + NFR                                               00028400
      NXTPR = NXEPC + NFR                                               00028500
      NXTPC = NXTPR + NFR                                               00028600
      NXTPZ = NXTPC + NFR                                               00028700
      NXTSR = NXTPZ + NFR                                               00028800
      NXTSC = NXTSR + NFR                                               00028900
      NXTSZ = NXTSC + NFR                                               00029000
```

```
      NXGS   = NXTSZ + NFR
      NXSWG  = NXGS  + NFR
      NXSWS  = NXSWG + NFR
      NXSWS3 = NXSWS + NFR
      NXEHP  = NXSWS3 + NUR
      NLMR   = NXEHP + NFR
      NLMC  = NLMR + NNTR
      NLMZ  = NLMC + NNTR
      MEMO  = NLMZ + NNTR
      IPCNT = 0
      IF ( NPLOT.EQ.0 ) GO TO 110
      IF ( NBUF.EQ.0 ) NBUF = 50
      NPIT  = NLMZ + NNTR
      NPPOW  = NPIT + NBUF
      NPTIME  = NPPOW + NBUF
      NPPLPR  = NPTIME + NBUF
      NPYP  = NPPLPR + NBUF
      NPRL  = NPYP + NBUF
      NPTL  = NPRL  + NP1*NBUF
      NPT  = NPTL  + NP1*NBUF
      NPTLMX  = NPT  + NBUF
      NPRLR  = NPTLMX + NBUF
      NPTLR  = NPRLR + NBUF*2
      NPCPGP  = NPTLR + NBUF*2
      NPCPU  = NPCPGP + NBUF
      NPBU  = NPCPU + NP*NBUF
      NPSIGC  = NPBU + NBUF
      NPCSTN  = NPSIGC + NBUF
      NPTSC  = NPCSTN + NBUF
      NPTPC  = NPTSC + NBUF
      MEMO  = NPTPC + NBUF
  110 CONTINUE
C
C
      LRVB  = MEMO
      LRCB  = LRVB + NN
      LREB  = LRCB + NN
      LRUB  = LREB + NN
      LRCI  = LRUB + NN
      LRCO  = LRCI + NN
      LBU  = LRCO + NN
      LRDIST  = LBU + NN
      LDXM  = LRDIST + 4*NN
      LFLNC  = LDXM + 3*NN
      LWOLD  = LFLNC + NN
      LTOUR  = LWOLD + NNFR
      MEMO  = LTOUR + NN
C
C
C==========  LOCATION SET FOR MATRIX ( DIMENSION AM )
C
      IGEN  = N1O + 12
      NONZR  = 56*NTR + 24
C
C
      IGUESS = 10
C
C
```

```
C       IBOT = NUMBER OF NON-ZERO ELEMENT AFTER L / U RESOLUTION
C
C
        IBOT = IGUESS * NONZR
C
        IGEN1 = IGEN + 1
        IGENP = 10*NCL + 7
        IGENP1 = IGENP + 1
        NONZRP = 54*NCL + 12
C
        MRES = 1
        MAUX = MRES + IGEN
        MIA = MAUX + IGEN
        MJA = MIA + IGEN1
        MA = MJA + NONZR
        MIC = MA + NONZR
        MJC = MIC + IGEN1
        MIU = MJC + IBOT
        MDI = MIU + IGEN
        MC = MDI + IGEN
        MIPR = MC + IBOT
        MIPC = MIPR + IGEN
        MKA = MIPC + IGEN
        MIER = MKA + IGEN
        MB = MIER + 5
        MEMOMX = MB + IGEN
C
C
        MEPR = MRES
        MEPC = MEPR + NTR
        MSIGR = MEPC + NTR
        MSIGC = MSIGR + NTR
        MSIGZ = MSIGC + NTR
        MTPR = MSIGZ + NTR
        MTPC = MTPR + NTR
        MTPZ = MTPC + NTR
        MTPRF = MTPZ + NTR
        MTPRC = MTPRF + 1
        MTPCF = MTPRC + 1
        MTPCC = MTPCF + 1
        MTPZF = MTPCC + 1
        MTPZC = MTPZF + 1
        MEPZF = MTPZC + 1
        MEPZC = MEPZF + 1
        MDISP = MEPZC + 1
        MPRES = MDISP + NTR2
        MFRIC = MPRES + NTR1
        MDSR = MSIGR
        MDSC = MSIGC
        MDSZ = MSIGZ
C
C       FOR PLENUM SECTION ONLY
C
        MEPRP = MRES
        MEPCP = MEPRP + NCL
        MSIGRP = MEPCP + NCL
        MSIGCP = MSIGRP + NCL
        MSIGZP = MSIGCP + NCL
```

```
      MTPRP  = MSIGZP + NCL
      MTPCP  = MTPRP  + NCL
      MTPZP  = MTPCP  + NCL
      MTPRCP = MTPZP  + NCL
      MTPCCP = MTPRCP + 1
      MTPZCP = MTPCCP + 1
      MEPZCP = MTPZCP + 1
      MDISPP = MEPZCP + 1
      MPRESP = MDISPP + NCL1
      MFRICP = MPRESP + NCL1
      MDSRP  = MSIGRP
      MDSCP  = MSIGCP
      MDSZP  = MSIGZP
C
C
C=========================================================================
C
C
C---- READ CARD 4   TO  CARD 23
C
C
C
      CALL INPUT  ( A(NEPZ),A(NTPR),A(NTPC),A(NTPZ),A(NSIGR),A(NSIGC),
     *              A(NSIGZ),A(NADIS),A(NRDIS),A(NRVB),A(NRCB),A(NREB),
     *              A(NRUB),A(NRCI),A(NRCO),A(NQSFF),A(NQSFP),A(NF),
     *              A(NEPR),A(NEPC),A(NTSR),A(NTSC),A(NTSZ),A(NCPU),
     *              A(NRHOP),A(NOTOM),FL,CL,PTOP,PO,EPSIN,A(NCLZO) )
C
C---- MATERIAL PROPERTIES  AND CONSTANTS
C
      CALL PROPTY ( A(NCPU),A(NRHOP),A(NRHOT) )
C
C
C=========================================================================
C==================   INITIAL SET   =====================================
C=========================================================================
C
C
C
C---- SET I/O DEVICE FOR WORK FILE
C
      CALL IOSET ( A(NIO) )
C
C---- INITIAL VALUES SET
C
      CALL BEGIN ( A(NXM),A(NZFLX),A(NRH),A(NRHOP),A(NCPU),
     *            A(NLM),A(NEHP),A(NSWS),A(NGS),A(NGR),A(NRVB),
     *            A(NRCB),A(NREB),A(NRUB),A(NQSF),A(NQSFF),A(NQSFP),
     *            A(NPOW),A(NXCPI),A(NXCPO),A(NBU),A(NHG),A(NKFRIC),
     *            A(NFLZO),A(NFLZ),A(NPRS),A(NFLNC),
     *            A(NCLZO),A(NCLZ),FL,CL,A(NDXM),A(NSWG),A(NRCI),
     *            A(NRCO),A(NRVBO),A(NRCBO),A(NREBO),A(NRUBO),
     *            A(NRCIO),A(NRCOO),A(NXMINT),A(NRDIST),A(NEPSEQ),
     *            A(NSWS3),A(NLMR),A(NLMC),A(NLMZ) )
C
C---- INPUT VALUES EDIT
C
      CALL EDTINP ( A(NRVB),A(NRCB),A(NREB),A(NRUB),A(NRCI),A(NRCO),
```

```
     *              A(NQSFF),A(NQSFP),A(NOTOM),A(NCPU),A(NRHOP),A(NF),
     *              A(NXCPI),A(NXCPO),A(NFLZ),A(NCLZ),FL,CL,PTOP,PO )
C
      MEMOC = (MEMO+MEMOMX+53000)*4/128000
      MEMOC = (MEMOC+1)*128
      WRITE(IO6,9002)  MEMO , IA , MEMOMX , IAM , MEMOC
C
      IT = 0
      ITPLOT = 0
      TIM = 0.
      TIMO = 0.
      TOGAS = GASI
      POWAO = 0.
      PPOWA = 0.
      PPO = 1.033
      PTOUT = 20.
      PTIN = 20.
      JKL = 1
      DELTO = 1.E+10
      CPOW = 1.0
      FLO = FL
      FLN = FL
      CLO = CL
      CLN = CL
      KJI = 0
      KJIK = 0
C
C
C======================================================================
C=================   TIME STEP LOOP   =================================
C======================================================================
C
C
  140 CONTINUE
C
      IT=IT+1
C
      CALL UPDTE ( A(NRVB),A(NRCB),A(NREB),A(NRUB),A(NRCI),A(NRCO),
     *             A(NBU),A(NRDIST),A(NDXM),A(NFLNC),A(NWOLD),
     *             A(LRVB),A(LRCB),A(LREB),A(LRUB),A(LRCI),A(LRCO),
     *             A(LBU),A(LRDIST),A(LDXM),A(LFLNC),A(LWOLD),1 )
      IF ( KJI.EQ.0 ) GO TO 141
      TIP = TIM
      TIM = YTIM
      TOUT = YTOUT
      TIN = YTIN
      PO = YPO
      POWAV = YPOWAV
      IF ( NSKIP.GT.0 ) CALL CORTMP ( A(NTOUR),A(LTOUR),A(NPTOU),1,TOUT)
      DELT = TIM-TIP
      KJI = 0
      JJOUT = 1
      GO TO 390
  141 CONTINUE
      KJIK = 0
      IF (JKL-1) 160,160,150
150   TIP=TIM
      TIM=TIMO
```

```
      IF ( NSKIP.GT.0 ) CALL CORTMP(A(NTOUR),A(NTOUO),A(NPTOU),1,TOUT)
      TOUT=TOUTO
      TIN=TINO
      PO = POO
      POWAV=POWAO
      JKL=1
      GO TO 360
  160 CONTINUE
      TIP=TIM
C
C=======================================================================
C
C
C
C---- CARD  24   ...   OPERATING HISTORY
C
C
  163 READ(IO5,9001,END=999)  TIM,POI,REPOW,CF,TIN,TOUT,NPLOTC
C
C
      TIM=TIM*24.0
      IF ( TIM-TIMO ) 170,180,180
  170 IF(NPLOT.EQ.0) GO TO 6000
C.........
      CALL PLTOR ( A(NPPOW),A(NPRL),A(NPTL),A(NPT),A(NPTLMX),
     *            A(NPRLR),A(NPTLR),A(NPCPGP),A(NPCPU),A(NPBU),
     *            A(NPSIGC),A(NPCSTN),A(NPTSC),A(NPTPC),
     *            A(NPTIME),A(NPPLPR),A(NPYP),A(NPIT),ITPLOT,IT )
C.........
      GO TO 6000
  180 CONTINUE
      IF ( NSKIP.GT.0 ) CALL CORTMP(A(NTOUR),A(NTOUO),A(NPTOU),0,TOUT)
      IF(POI.GT.0.) PO=POI
      JJOUT = 1
      IF (CF) 210,210,200
  200 CFLX=CF
  210 CONTINUE
      POWAV=REPOW*CPOW
      IF (TOUT) 230,230,240
  230 TOUT=TIN+TCON*POWAV*SMQ/SMQO*FLN/FLO
      GO TO 250
  240 TCON=(TOUT-TIN)/POWAV/SMQ*SMQO/FLN*FLO
  250 IF (TIM-TIP-0.01) 260,270,270
  260 TIM=TIP+0.01
  270 IF (POWAV-0.01) 280,290,290
  280 POWAV=0.01
  290 IF (TOUT-TIN-1.) 300,310,310
  300 TOUT=TIN+1.
C
  310 TIMO = TIM
      TOUTO=TOUT
      POO=PO
      TINO=TIN
      POWAO = POWAV
      IF ( NSKIP.GT.0 ) CALL CORTMP(A(NTOUR),A(NTOUO),A(NPTOU),2,TOUT)
C
C
  360 DELT=TIM-TIP
```

```
C
C
      CALL TIMCUT(CX,DELTO,EPSPOW)
C
C---- TIME INTERVAL CHANGE CHECK
C
  370 IF ( DELT-CX*DELTO ) 385,380,380
C
C---- POWER CHANGE CHECK
C
  385 IF ( ABS(POWAV-PPOWA)-EPSPOW ) 390,380,380
C
C
  380 CONTINUE
C
C---- TIME INTERVAL CUT
C
      TIM=TIP+DELT/2.
      POWAV=PPOWA+(POWAV-PPOWA)/2.
      TOUT=PTOUT+(TOUT-PTOUT)/2.
      PO=PPO+(PO-PPO)/2.
      TIN=PTIN+(TIN-PTIN)/2.
      IF ( NSKIP.GT.0 ) CALL CORTMP(A(NTOUR),A(NTOUO),A(NPTOU),3,TOUT)
      DELT=DELT/2.
      JKL=2
      GO TO 370
C
C     PREVIOUS POWAV PRESUMES NO BURNUP OF THE FUEL ELEMENT.
C
  390 CONTINUE
      YSMQ = SMQ/SMQO
  391 CONTINUE
      POWERC = POWAV
C     POWERC = COLD POWER
      POWAV = POWAV*YSMQ
C
C---- RESTRUCTURING VELOCITY CHECK
C
      CALL CHKSPD ( A(NDXM),EPSSPD,NN,ICHK,DELT )
      IF ( ICHK.GT.0 ) GO TO 417
C
C---- REGION BOUNDARY MOVEMENT
C
      CALL RHOMOD ( A(NRH),A(NRHOP),A(NEPZ),A(NRVB),A(NRCB),
     *              A(NREB),A(NRUB),A(NBU),A(NXM),A(NRHOT),A(NRDIS),
     *              A(NDXM),A(NRVBO),A(NRCBO),A(NREBO),A(NRUBO),
     *              A(NRCI),A(NRCO),A(NXMINT),A(NEPC),A(NRDIST),A(NEPR),
     *              A(NXEPR),A(NXEPC),A(NRFR),A(NWOLD),A(NWNEW),IT,
     *              DELT )
C
C---- CLADDING OUTER SURFACE TEMPERATURE
C
      CALL GOLDN ( POWAV,A(NQSF),TOUT,TIN,A(NRCO),A(NPOW),A(NTCLO),
     *            RCPI,RCPO,AFNA,FL,A(NHC),A(NTNA),A(NTCPO),A(NTCPI),
     *            A(NTOUR),NSKIP,PO,A(NFLZ) )
C
      IF(RCPO-RCPI) 400,400,410
C
```

```
C---- CAPSULE TEMPERATURE DISTRIBUTION
C
  410 CALL CLTMP ( A(NPOW),A(NTCPO),A(NXCPO),A(NXCPI),A(NRCP),A(NTCP),
     *                A(NTBCP),A(NTBCPA),A(NATBCP),A(NATBPA),A(NRCLO),IT,
     *                A(NRBAR) )
C
C---- CLADDING TEMPERATURE DISTRIBUTION
C
  400 CALL CLTMP ( A(NPOW),A(NTCLO),A(NRCO),A(NRCI),A(NRCL),A(NTCL),
     *                A(NTBCL),A(NTBCLA),A(NATBCL),A(NATBLA),A(NRCLO),IT,
     *                A(NRBAR) )
C
C---- FUEL-CLADDING GAP CONDUCTANCE
C
      CALL GAPCON ( A(NRCI),A(NRUB),A(NHG) )
C
C---- FUEL BURN-UP
C
      CALL BURN ( A(NBU),A(NPOW),A(NXM),A(NEPZ),A(NQSF),A(NQSFP),
     *               DELT )
C
C---- FLUX , FLUENCE
C
      CALL FLXCAL ( POWAV,A(NQSFF),A(NZFLX),A(NFLNC),A(NFLUX),DELT )
C
C---- FUEL TEMPERATURE DISTRIBUTION , RESTRUCTURING
C
      CALL FLTMP
     *     ( A(NPOW),A(NTCL),A(NRCI),A(NRUB),A(NREB),A(NRCB),A(NRVB),
     *       A(NRH),A(NDXM),A(NXM),A(NRFR),A(NTFR),A(NTBF),A(NATBF),
     *       A(NTBFA),A(NATBFA),A(NFD),A(NRFRPM),A(NTFRPM),A(NTBFPM),
     *       A(NATBPM),A(NFDPM),A(NGDT),A(NHG),A(NF),A(NOTOM),A(NRHOT),
     *       DELT,A(NRFRO),IT,A(NBU),A(NRHOP),A(NRBAR),
     *       A(NEPR),A(NEPC),A(NEPZ),A(NTPR),A(NTPC),A(NTPZ),A(NTSR),
     *       A(NTSC),A(NTSZ),A(NGS),A(NSWG),A(NSWS),A(NEHP),
     *       A(NWOLD),A(NWNEW),A(NXEPR),A(NXEPC),A(NXTPR),A(NXTPC),
     *       A(NXTPZ),A(NXTSR),A(NXTSC),A(NXTSZ),A(NXGS),A(NXSWG),
     *       A(NXSWS),A(NXEHP),A(NXSWS3),A(NSWS3) )
C
C---- AXIAL LENGTH
C
      CALL LENGTH ( FL,FLN,CL,CLN,A(NFLZO),A(NFLZ),A(NCLZO),A(NCLZ),
     *                A(NEPZ) )
C
C---- VOLUME
C
      CALL VOLUME ( A(NVOL),A(NRFR),A(NRCL),A(NFLZ),A(NCLZ) )
C
      DO 415 NS=1,NN
C
C---- YOUNG'S MODULUS , POISSON'S RATIO
C
      CALL XLAME( A(NTBCL),A(NTBF),A(NYNG),A(NPOS),A(NSHR),A(NRH),
     *                A(NLM),A(NBM),A(NPOSB),A(NSHRB),A(NVOL),A(NRHOT),
     *                NS )
C
C---- EQUIVALENT STRESS
C
```

```
      CALL EQUISI ( A(NSIGR),A(NSIGC),A(NSIGZ),A(NEQSIG),
     *              A(NPRS),NS )
C
C---- CLADDING CREEP STRAIN
C
      CALL CREEC ( A(NEQSIG),A(NDEQP),A(NTBCL),A(NFLNC),A(NFLUX),
     *             DELT,NS )
C
C
      IF ( NS.EQ.NN ) GO TO 415
C
C---- FUEL CREEP STRAIN
C
      CALL CREEF ( A(NEQSIG),A(NDEQP),A(NTBF),A(NPOW),A(NFD),A(NFLUX),
     *             A(NFTHP),A(NLM),DELT,NS,A(NSHR),A(NEPSEQ) )
  415 CONTINUE
C
C---- EQUIVALENT CREEP STRAIN INCREMENT CHECK
C
      CALL CHECK ( A(NDEQP),EPSCRF,EPSCRC,ICHK )
      IF ( ICHK.EQ.0 ) GO TO 425
  417 CONTINUE
      CALL UPDTE ( A(NRVB),A(NRCB),A(NREB),A(NRUB),A(NRCI),A(NRCO),
     *             A(NBU),A(NRDIST),A(NDXM),A(NFLNC),A(NWOLD),
     *             A(LRVB),A(LRCB),A(LREB),A(LRUB),A(LRCI),A(LRCO),
     *             A(LBU),A(LRDIST),A(LDXM),A(LFLNC),A(LWOLD),2 )
      POWAV = POWERC
      IF ( KJIK.NE.0 ) GO TO 420
      YTIM = TIM
      YTOUT = TOUT
      YTIN = TIN
      YPO = PO
      YPOWAV = POWAV
      IF ( NSKIP.GT.0 ) CALL CORTMP(A(NTOUR),A(LTOUR),A(NPTOU),2,TOUT)
      KJIK = 1
  420 CONTINUE
      KJI = 1
      IF ( JOUT.EQ.0 ) JJOUT=0
C
C---- TIME INTERVAL CUT
C
      TIM = ( TIM+TIP ) / 2.0
      POWAV = ( POWAV+PPOWA ) / 2.0
      TOUT = ( TOUT+PTOUT ) / 2.0
      TIN = ( TIN+PTIN ) / 2.0
      PO = ( PO+PPO ) / 2.0
      DELT = DELT / 2.0
      IF ( NSKIP.GT.0 ) CALL CORTMP (A(NTOUR),A(LTOUR),A(NPTOU),3,TOUT)
      GO TO 391
  425 CONTINUE
C
C---- PLUTONIUM MIGRATION
C
      CALL PLUTO ( A(NCPU ),A(NRVB),A(NRCB),A(NREB),A(NRFRPM),A(NTFRPM),
     *             A(NGDT),A(NXM),A(NRH),A(NEPZ),DELT,A(NCPUD) )
C
C---- GAS RELEASE RATE
C
```

```
      CALL GASOUT ( A(NRFR),A(NFLZ),A(NPOW),A(NFD),A(NTBF),A(NVOL),
     *              A(NEPR),A(NEPC),A(NEPZ),A(NATBF),A(NSWS),A(NGR),
     *              A(NGS),TOGAS,DELT )
C
C---- PLENUM PRESSURE
C
      CALL PLENP ( A(NRVB),A(NRCI),A(NTFR),A(NFLZ),TOGAS,FL,CL,TOUT,
     *             PLPR,CVPR,A(NRUB),A(NCLZ) )
C
C---- CLADDING SWELLING STRAIN
C
      CALL HKNS ( A(NTBCL),A(NTSR),A(NTSC),A(NTSZ),A(NFLUX),DELT )
C
C---- FUEL SWELLING STRAIN DUE TO SOLID FISSION PRODUCTS
C
      CALL SWELL ( A(NPOW),A(NSWS),A(NFD),A(NSWS3),DELT )
C
      ISP = 0
C
      IF(JOUT.EQ.1) GO TO 440
      IF( JKL.NE.1 ) GO TO 450
      IF(JJOUT.LE.0) GO TO 450
C
  440 CONTINUE
C
C
      CALL EDTGRS ( IT,TIM,POWAV,TIN,TOUT,DELT,A(NPOW),A(NFLUX),A(NBU),
     *              A(NTNA),A(NXCPO),A(NXCPI),A(NTCPI),A(NTCPO),
     *              A(NRCO),A(NRCI),A(NTCLO),A(NTCL),A(NFLZ),A(NCLZ),
     *              FL,CL,A(NOTOM),A(NCPU),A(NRHOT),A(NRH),A(NXM),
     *              A(NRVB),A(NRCB),A(NREB),A(NRUB),A(NRFR),
     *              A(NRCL),A(NHG),A(NHC),TOGAS,A(NGR),A(NTFR),
     *              A(NTBFA),A(NATBFA),A(NTBCLA),A(NATBLA),A(NRCP),
     *              A(NTCP),A(NTBCPA),A(NATBPA),A(NFLNC),A(NGS),
     *              PLPR,CVPR,FLO,CLO ,A(NBUM),PO,A(NFD),A(NFDPM) )
      ISP = 1
      ITPLOT = ITPLOT + 1
C
  450 CONTINUE
C
C
C==== AXIAL SECTION LOOP    GOING FROM LOWER TO UPPER SECTION ==========
C
C
      DO 1000 NS=1,NN
      IF(NS.EQ.NN) GO TO 600
C
      JCR = 0
C
  490 CONTINUE
      MOUNT = 0
      KONV = 0
C     IFRIC = 0
      CALL CLEAR(IFRIC,A(NKFRIC),NS)
C
C     IFRIC = 0  ...  GAP OPEN
C     IFRIC = 1  ...  STICK
C     IFRIC = 2  ...  SLIP
```

```
C
      IF ( JCR.GT.0 ) GO TO 500
C
C---- FUEL SWELLING STRAIN , HOT-PRESSING
C
      CALL ITERAT ( A(NPRS),A(NVOL),A(NTBF),A(NATBF),A(NSWG),
     *              A(NEHP),A(NRH),A(NRFR),A(NGS),A(NSWS),A(NBM),
     *              A(NEPR),A(NEPC),A(NEPZ),A(NSWS3),A(NSIGR),A(NSIGC),
     *              A(NSIGZ),A(NRHOT),DELT,NS,A(NRHOP),A(NBU),
     *              A(NTSR),A(NTSC),A(NTSZ),A(NXEHP),A(NXSWG) ,
     *              PLPR,CVPR )
C
  500 CONTINUE
C
C---- DECIDE IA , JA , A   ...   ARGUMENT TO MSNR SUBROUTINE ( SL/MATH )
C
      CALL MATRX  ( AM(MIA),AM(MJA),AM(MA),A(NYNG),A(NPOS),A(NDEQP),
     *              A(NEQSIG),A(NRFR),A(NRCL),A(NSHR),A(NPOSB),A(NSHRB),
     *              A(NFLZ),A(NKFRIC),MOUNT,IFRIC,NS,A(NRBAR) )
C
C
C---- PIVOTING AND L / U RESOLUTION
C
      CALL MSNR ( IGEN,AM(MIA),AM(MJA),AM(MA),IBOT,AM(MIC),AM(MJC),
     *            AM(MIU),AM(MDI),AM(MC),AM(MIPR),AM(MIPC),AM(MIER),
     *            AM(MKA),AM(MRES) )
C
C
C---- B ARRAY CALCULATION OF AX=B
C
C
      CALL BCALC ( AM(   MB   ),A(NATBF),A(NATBCL),A(NTPR),A(NTPC),
     *             A(NTPZ),A(NPRS),A(NRFR),A(NRCL),A(NPOS),A(NBM),
     *             A(NSHR),A(NPOSB),A(NSHRB),A(NVOL),A(NRH),PLPR,CVPR,
     *             PO,PTOP,DELT,A(NTSR),A(NTSC),A(NTSZ),A(NFLUX),
     *             A(NTBCL),A(NKFRIC),A(NSWG),A(NSWS),A(NEHP),A(NEPR),
     *             A(NEPC),A(NEPZ),MOUNT,IFRIC,NS,A(NRBAR) )
C
C---- SOLVE AX=B
C
      CALL MBSR ( IGEN,AM(MIPR),AM(MIPC),AM(MIC),AM(MIU),AM(MJC),
     *            AM(MDI),AM(MC),AM(MB),AM(MRES),AM(MAUX) )
C
C     AM ( MRES ) ARE FINAL RESULTS
C
C
C---- FUEL-CLADDING GAP ASSUMPTION CHECK
C
      CALL FRICHK ( A(NRUB),A(NRCI),AM(MDISP),AM(MPRES),AM(MFRIC),
     *              A(NCRIF),A(NKFRIC),KONV,IFRIC,NS )
C
      MOUNT = MOUNT + 1
C
      IF ( KONV.EQ.0 ) GO TO 500
      IF ( NCRK.LE.0 ) GO TO 550
C
C---- FUEL CRACKING
C
```

```
      CALL CRACK
     *        ( AM(MSIGR),AM(MSIGC),AM(MSIGZ ),A(NBM),A(NSHR),A(NTBF),
     *               A(NTPR),A(NTPC),A(NTPZ),A(NTSR),A(NTSC),A(NTSZ),
     *               A(NPRS),A(NLM),A(NLMR),A(NLMC),A(NLMZ),DELT,NS,JCR )
      IF ( JCR.EQ.0 ) GO TO 550
C
C---- RE-COMPUTE YOUNG'S MODULUS , POISSON'S RATIO
C
      CALL XLAME( A(NTBCL),A(NTBF),A(NYNG),A(NPOS),A(NSHR),A(NRH),
     *               A(NLM),A(NBM),A(NPOSB),A(NSHRB),A(NVOL),A(NRHOT),
     *               NS )
      GO TO 490
  550 CONTINUE
C
C
C
C     IF CONVERGED , THEN UPDATE VALUES
C
C
      CALL UPDATE ( AM(MSIGR),AM(MSIGC),AM(MSIGZ),A(NSIGR),A(NSIGC),
     *               A(NSIGZ),AM(MEPR),AM(MEPC),AM(MEPZF),AM(MEPZC),
     *               A(NEPR),A(NEPC),A(NEPZ),AM(MTPR),AM(MTPC),AM(MTPZ),
     *               A(NTPR),A(NTPC),A(NTPZ),AM(MDSR),AM(MDSC),AM(MDSZ),
     *               A(NTSR),A(NTSC),A(NTSZ),AM(MDISP),A(NRDIS),NS,
     *               A(NEPSEQ) )
C
      IF(JOUT.EQ.1) GO TO 590
      IF ( JKL.NE.1 ) GO TO 650
      IF(JJOUT.LE.0) GO TO 650
C
  590 CONTINUE
C
C---- WRITE RESULTS OF EACH AXIAL SECTION
C
      CALL EDTMTX ( A(NEPR),A(NEPC),A(NEPZ),A(NSIGR),A(NSIGC),A(NSIGZ),
     *               A(NTPR),A(NTPC),A(NTPZ),A(NTSR),A(NTSC),A(NTSZ),
     *               A(NRDIS),A(NCRIF),A(NRCP),A(NTCP),A(NTBCP),
     *               A(NATBCP),A(NRCL),A(NTCL),A(NTBCL),A(NATBCL),
     *               A(NYNG),A(NPOS),A(NSHR),A(NBM),A(NRFR),A(NTFR),
     *               A(NTBF),A(NATBF),A(NPRS),A(NEHP),AM(MTPRF),
     *               AM(MTPRC),AM(MTPCF),AM(MTPCC),AM(MTPZF),AM(MTPZC),
     *                          AM(MPRES),A(NKFRIC),AM(MTPR),AM(MTPC),
     *               AM(MTPZ),AM(MDSR),AM(MDSC),AM(MDSZ),A(NPOSB),
     *               A(NSHRB),A(NSWG),A(NSWS),NS,A(NRFRPM),A(NTFRPM),
     *               A(NGS),A(NCPU),A(NLM),A(NLMR),A(NLMC),A(NLMZ),
     *               A(NRCOO) )
C
      GO TO 650
C
  600 CONTINUE
C
C---- FOLLOWING ARE FOR ONLY PLENUM SECTION
C
      CALL MATRXP ( AM(MIA),AM(MJA),AM(MA),A(NYNG),A(NPOS),A(NDEQP),
     *               A(NEQSIG),A(NRCL ),A(NSHR),A(NPOSB),A(NSHRB),
     *               A(NCLZ),A(NKFRIC),NS )
C
      CALL MSNR ( IGENP,AM(MIA),AM(MJA),AM(MA),IBOT,AM(MIC),AM(MJC),
```

```
     *              AM(MIU),AM(MDI),AM(MC),AM(MIPR),AM(MIPC),AM(MIER),
     *              AM(MKA),AM(MRES) )
C
      CALL BCALCP ( AM(MB),A(NATBCL),A(NTPR),A(NTPC),A(NTPZ),A(NRCL),
     *               A(NPOS),A(NSHR),A(NPOSB),A(NSHRB),A(NVOL),A(NRH),
     *               A(NTBCL),A(NFLUX),A(NKFRIC),A(NCRIF),DELT,A(NTSR),
     *               A(NTSC),A(NTSZ),PLPR,PTOP,PO,A(NEPR),A(NEPC),
     *               A(NEPZ),NS )
      CALL MBSR ( IGENP,AM(MIPR),AM(MIPC),AM(MIC),AM(MIU),AM(MJC),
     *              AM(MDI),AM(MC),AM(MB),AM(MRES),AM(MAUX) )
      CALL UPDATP ( AM(MSIGRP),AM(MSIGCP),AM(MSIGZP),A(NSIGR),A(NSIGC),
     *               A(NSIGZ),AM(MEPRP),AM(MEPCP),AM(MEPZCP),A(NEPR),
     *               A(NEPC),A(NEPZ),AM(MTPRP),AM(MTPCP),AM(MTPZP),
     *               A(NTPR),A(NTPC),A(NTPZ),AM(MDSRP),AM(MDSCP),
     *               AM(MDSZP),A(NTSR),A(NTSC),A(NTSZ),AM(MDISPP),
     *               A(NRDIS),NS )
C
      IF(JOUT.EQ.1) GO TO 640
      IF ( JKL.NE.1 ) GO TO 650
      IF(JJOUT.LE.0) GO TO 650
C
  640 CONTINUE
      CALL EDTMTX ( A(NEPR),A(NEPC),A(NEPZ),A(NSIGR),A(NSIGC),A(NSIGZ),
     *               A(NTPR),A(NTPC),A(NTPZ),A(NTSR),A(NTSC),A(NTSZ),
     *               A(NRDIS),A(NCRIF),A(NRCP),A(NTCP),A(NTBCP),
     *               A(NATBCP),A(NRCL),A(NTCL),A(NTBCL),A(NATBCL),
     *               A(NYNG),A(NPOS),A(NSHR),A(NBM),A(NRFR),A(NTFR),
     *               A(NTBF),A(NATBF),A(NPRS),A(NEHP),AM(MTPRF),
     *               AM(MTPRCP),AM(MTPCF),AM(MTPCCP),AM(MTPZF),
     *               AM(MTPZCP),
     *                                                 AM(MPRESP),
     *               A(NKFRIC),AM(MTPRP),AM(MTPCP),AM(MTPZP),AM(MDSRP),
     *               AM(MDSCP),AM(MDSZP),A(NPOSB),A(NSHRB),A(NSWG),
     *               A(NSWS),NS,A(NRFRPM),A(NTFRPM),A(NGS),A(NCPU),
     *               A(NLM),A(NLMR),A(NLMC),A(NLMZ),A(NRCOO) )
      IF(JOUT.LE.0) JJOUT=0
C
  650 CONTINUE
C
C
      IF ( NPLOT.EQ.0 ) GO TO 730
      IF ( ISP.EQ.0 ) GO TO 730
C
C
C==== PREPARATION FOR PLOTTER ===========================================
C
C
      IF ( ISPLOT.EQ.0 ) CALL PLOTS(ACTIVE,ACTIVE,77)
C     77 IN ARGUMENTS MEANS LOGICAL UNIT NUMBER FOR PLOT
      IF ( ISPLOT.EQ.0 ) CALL PLOT(0.,20.,-3)
      ISPLOT = 1
      IF ( NS.NE.IPFUL ) GO TO 710
C
C---- FUEL
C
      CALL PLOT1 ( A(NPPOW),A(NPRL),A(NPTL),A(NPT),A(NPTLMX),
     *              A(NPRLR),A(NPTLR),A(NPCPGP),A(NPCPU),A(NPBU),
     *              A(NPOW),A(NRFRPM),A(NTFRPM),A(NRCB),A(NREB),
```

```
     *              A(NRUB),A(NRCI),A(NCPU),A(NBU),A(NRFR),A(NTFR),
     *              ITPLOT,NS )
  710 CONTINUE
      IF ( NS.NE.IPSTS ) GO TO 720
C
C---- CLAD STRESS
C
      CALL PLOT2 ( A(NPSIGC),A(NSIGC),A(NVOL),ITPLOT,NS )
  720 CONTINUE
      IF ( NS.NE.IPSTN ) GO TO 730
C
C---- CLAD STRAIN
C
      CALL PLOT3 ( A(NPCSTN),A(NPTSC),A(NPTPC),A(NRCO),A(NRCOO),
     *              A(NTSC),A(NTPC),A(NVOL),A(NRDIS),ITPLOT,NS )
  730 CONTINUE
      IF ( NCRK.LE.0 ) GO TO 1000
      IF ( NPLOTC.EQ.0 ) GO TO 1000
      IF ( ISP.EQ.0 ) GO TO 1000
      IF ( NS.NE.IPCRK ) GO TO 1000
C
      IF ( ISPLOT.EQ.0 ) CALL PLOTS(ACTIVE,ACTIVE,77)
      IF ( ISPLOT.EQ.0 ) CALL PLOT(0.,20.,-3)
      ISPLOT = 1
C
C---- FUEL CRACK
C
      CALL PLOT4 ( A(NRFRPM),A(NRCB),A(NREB),A(NLMR),A(NLMC),
     *              A(NLMZ),IT,NS )
      IPCNT = IPCNT + 1
C
 1000 CONTINUE
C
C
C
      IF ( NPLOT.NE.0 .AND. ISP.EQ.1 )
     *  CALL PLOT5 ( A(NPTIME),A(NPPLPR),A(NPYP),A(NPIT),ITPLOT,IT,
     *                TIM,PLPR )
C
C
C==== GOING TO NEXT TIME STEP ==========================================
C
C
      PPOWA = POWERC
      PPO=PO
      PTIN = TIN
      PTOUT = TOUT
      IF ( NSKIP.GT.0 ) CALL CORTMP(A(NTOUR),A(NTOUO),A(NPTOU),4,TOUT)
      IF ( KJIK.NE.0 ) GO TO 140
      DELTO = DELT
C
      GO TO 140
C
999   CONTINUE
      IF ( NPLOT.EQ.0 ) GO TO 998
      CALL PLTOR ( A(NPPOW),A(NPRL),A(NPTL),A(NPT),A(NPTLMX),
     *              A(NPRLR),A(NPTLR),A(NPCPGP),A(NPCPU),A(NPBU),
     *              A(NPSIGC),A(NPCSTN),A(NPTSC),A(NPTPC),
```

```
     *                A(NPTIME),A(NPPLPR),A(NPYP),A(NPIT),ITPLOT,IT )
  998 CONTINUE
      IF ( ISPLOT.NE.0 ) CALL PLOT ( 0.,0.,999 )
      WRITE(IO6,9003)   DAYDAY
      RETURN
C
C     FORMAT
C
 9000 FORMAT(16I5)
 9001 FORMAT(E10.3,F5.1,3E15.5,E10.5,2I5)
 9002 FORMAT(/1H0,'  ***  ACTUALLY USED CORE FOR ARRAY  =',I8,'  WORDS',
     *5X,'( INPUT =',I8,' )'  /
     *          1H0,'  ***  ACTUALLY USED CORE FOR MATRIX =',I8,'  WORDS',
     *5X,'( INPUT =',I8,' )'  /
     *          1H0,'  ***  RECOMMENDED CORE SIZE =',I4,' K BYTE' )
 9003 FORMAT( 1H1,' *****  JOB ENDED NORMALLY   *****  ACTIVE-II   *****',
     *          60X,'DATE',2X,A8 )
 9004 FORMAT( 1H1, 9(/////, 1H0, 35X, '==========  A C T I V E - I I - F
     * B R  =========='), /, 1H1)
 9005 FORMAT( 1H1, 9(/////, 1H0, 35X, '==========  A C T I V E - I I - A
     * T R  =========='), /, 1H1)
 9006 FORMAT(2I5,10X,3E10.5 )
 9007 FORMAT( 1H1, 9(/////, 1H0, 35X, '==========  A C T I V E - I I - P
     * W R  =========='), /, 1H1)
 9008 FORMAT( 1H1, 9(/////, 1H0, 35X, '==========  A C T I V E - I I - B
     * W R  =========='), /, 1H1)
C
      END
```

```
      FUNCTION AREA ( R,DR )
C
C     AREA CALC. BETWEEN (R+DR) AND (R)
C
      AREA = DR*( 2.*R+DR )
C
      RETURN
      END
```

```
      FUNCTION ATBFUN ( T,DT,I )
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
C
C     FUEL THERMAL EXPANSION CALC.
C
      Y = 20.
      ATBFUN = ALFO(I)*T+ALF1(I)*T*T/2.+(ALFO(I)+ALF1(I)*T)*DT/2.-
     *         (ALFO(I)*Y+ALF1(I)*Y*Y/2.)
      RETURN
      END
```

```
      SUBROUTINE AXIAL (XO,YO,ID)                                      00002100
      CALL SYMBOL(XO,YO,4.,'AXIAL SECTION =',0.,15)                     00002200
      CALL NUMBER(XO+64.,YO,4.,FLOAT(ID),0.,-1)                         00002300
      RETURN                                                            00002400
      END                                                               00002500
```

```
      SUBROUTINE AXISD(MX,XMN,DX)
      DO 100 I=1,MX
      X= XMN*(MX-I)
      CALL NEWPEN(3)
      IF(I.EQ.1.OR.I.EQ.MX) GO TO 200
      CALL PLOT (X/DX,220.,3)
      CALL PLOT (X/DX,0.,2)
  200 CONTINUE
      CALL NEWPEN(1)
      CALL NUMBER (X/DX-4.,-7.0,4.0,X,0.,1)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE AXIST(MX,XMN,DX)
      DO 100 I=1,MX
      J=MX-I+1
      X = XMN*J
      CALL NEWPEN(3)
      IF(I.EQ.1) GO TO 200
      CALL PLOT (J/DX,220.,3)
      CALL PLOT (J/DX,0.0,2)
  200 CONTINUE
      CALL NEWPEN(1)
      CALL NUMBER (J/DX-4.0,-7.0,4.0,X,0.,-1)
  100 CONTINUE
      CALL NUMBER (-2.0,-7.0,4.0,0.,0.,-1)
      RETURN
      END
```

```
      SUBROUTINE BCALC ( B,ATBF,ATBCL,TPR,TPC,TPZ,PRS,RF,RC,P,BM,G,PB,  00005400
     *                   GB,VOL,RH,PLPR,CVPR,PO,PTOP,DELT,TSR,TSC,TSZ,  00005500
     *                   FLUX,TBCL,KFRIC,SWG,SWS,EHP,EPR,EPC,EPZ,       00005600
     *                   MOUNT,IFRIC,NS,RB )                            00005700
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,     00005800
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,        00005900
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,             00006000
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,    00006100
     *                N12C,N13C,N14C,N15C,                              00006200
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15      00006300
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2), 00006400
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,    00006500
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB          00006600
C                                                                       00006700
      DIMENSION  B(IGEN) , ATBF(NN,NFR) , ATBCL(NN,NCL) , TPR(NN,NTR) , 00006800
     *           TPC(NN,NTR) , TPZ(NN,NTR) , PRS(NN,NTR) , RF(NN,NFR1) ,00006900
     *           RC(NN,NCL1) , P(NN,NTR) , BM(NN,NTR) , G(NN,NTR) ,     00007000
     *           PB(NN,2) , GB(NN,2) , VOL(NN,NTR) , RH(NN,NTR) ,       00007100
     *           TSR(NN,NTR) , TSC(NN,NTR) , TSZ(NN,NTR) , FLUX(NN) ,   00007200
     *           TBCL(NN,NCL) , KFRIC(NN) , SWG(NN,NFR) ,               00007300
     *           SWS(NN,NTR) , EHP(NN,NTR) , EPR(NN,NTR) , EPC(NN,NTR) ,00007400
     *           EPZ(NN,NTR) , RB(NN,NTR)                               00007500
C                                                                       00007600
C     RIGHT-HAND VECTORS B OF AX=B                                      00007700
C                                                                       00007800
      IF ( MOUNT ) 10,10,3000                                           00007900
C                                                                       00008000
   10 CONTINUE                                                          00008100
C                                                                       00008200
      DO 999 I=1,IGEN                                                   00008300
  999 B(I) = 0.                                                         00008400
C                                                                       00008500
      DO 1000 I=1,NFR                                                   00008600
      B(I) = ATBF(NS,I)  + TSR(NS,I)                                    00008700
 1000 CONTINUE                                                          00008800
C                                                                       00008900
      DO 1100 I=1,NCL                                                   00009000
      B(NFR+I) = ATBCL(NS,I) + TSR(NS,NFR+I)                            00009100
 1100 CONTINUE                                                          00009200
C                                                                       00009300
      DO 1200 I=1,NFR                                                   00009400
      B(N1+I) = ATBF(NS,I) + TSC(NS,I)                                  00009500
 1200 CONTINUE                                                          00009600
C                                                                       00009700
      DO 1300 I=1,NCL                                                   00009800
      B(N1+NFR+I) = ATBCL(NS,I) + TSC(NS,NFR+I)                         00009900
 1300 CONTINUE                                                          00010000
C                                                                       00010100
      DO 1400 I=1,N1                                                    00010200
      B(N2+I) = -TPR(NS,I)                                              00010300
 1400 CONTINUE                                                          00010400
C                                                                       00010500
      DO 1500 I=1,N1                                                    00010600
      B(N3+I) = -TPC(NS,I)                                              00010700
 1500 CONTINUE                                                          00010800
C                                                                       00010900
      DO 1600 I=1,N1                                                    00011000
      B(N4+I) = -TPZ(NS,I)                                              00011100
```

```
 1600 CONTINUE
C
C
      DO 2000 I=1,NFR
      RI = RF(NS,I)
      RO = RF(NS,I+1)
      RR = (RO+RI)*(RO-RI)
C     RA = RB(NS,I)
      RA = RI
      XY1 = (1.-2.*P(NS,I))*RA*ALOG(RA/RI) / (2.*(1.-P(NS,I)))
      XY2 = (RI*RI+(1.-2.*P(NS,I))*RA*RA)*RO*RO*ALOG(RO/RI) /
     *   (2.*(1.-P(NS,I))*RA*RR)
      OM = 1.-2.*RO*RO*ALOG(RO/RI) / RR
      B(N5+6+I)   = RA*(1.+P(NS,I))*ATBF(NS,I) + EPR(NS,I)*(-RI+RA/2.-
     *   XY1+XY2) + EPC(NS,I)*(-RA/2.+XY1-XY2)
     *   + TSR(NS,I)*( 0.5*RA+XY1-XY2 )
     *   + TSC(NS,I)*( 0.5*RA-XY1+XY2 )
     *   + TSZ(NS,I)*RA*P(NS,I)
 2000 CONTINUE
C
      DO 2100 I=1,NCL
      RI = RC(NS,I)
      RO = RC(NS,I+1)
      RR = (RO+RI)*(RO-RI)
C     RA = RB(NS,NFR+I)
      RA = RI
      XY1 = (1.-2.*P(NS,NFR+I))*RA*ALOG(RA/RI) / (2.*(1.-P(NS,NFR+I)))
      XY2 = (RI*RI+(1.-2.*P(NS,NFR+I))*RA*RA)*RO*RO*ALOG(RO/RI) /
     *   (2.*(1.-P(NS,NFR+I))*RA*RR)
      OM = 1.-2.*RO*RO*ALOG(RO/RI) / RR
      B(N5+6+NFR+I)   = RA*(1.+P(NS,NFR+I))*ATBCL(NS,I) +
     *  EPR(NS,NFR+I)*(-RI+RA/2.-XY1+XY2)+EPC(NS,NFR+I)*(-RA/2.+XY1-XY2)
     *   + TSR(NS,NFR+I)*( 0.5*RA+XY1-XY2 )
     *   + TSC(NS,NFR+I)*( 0.5*RA-XY1+XY2 )
     *   + TSZ(NS,NFR+I)*RA*P(NS,NFR+I)
 2100 CONTINUE
      DO 2200 I=1,NFR
      RI = RF(NS,I)
      RO = RF(NS,I+1)
      RR = (RO+RI)*(RO-RI)
C     RA = RB(NS,I)
      RA = RO
      XY1 = (1.-2.*P(NS,I))*RA*ALOG(RA/RI) / (2.*(1.-P(NS,I)))
      XY2 = (RI*RI+(1.-2.*P(NS,I))*RA*RA)*RO*RO*ALOG(RO/RI) /
     *   (2.*(1.-P(NS,I))*RA*RR)
      TH = 1.-2.*RI*RI*ALOG(RO/RI) / RR
      B(N6+6+I)   = RA*(1.+P(NS,I))*ATBF(NS,I) + EPR(NS,I)*(-RO+RA/2.-
     *   XY1+XY2) + EPC(NS,I)*(-RA/2.+XY1-XY2)
     *   + TSR(NS,I)*( 0.5*RA+XY1-XY2 )
     *   + TSC(NS,I)*( 0.5*RA-XY1+XY2 )
     *   + TSZ(NS,I)*RA*P(NS,I)
 2200 CONTINUE
C
      DO 2300 I=1,NCL
      RI = RC(NS,I)
      RO = RC(NS,I+1)
      RR = (RO+RI)*(RO-RI)
C     RA = RB(NS,NFR+I)
```

```
      RA = RO
      XY1 = (1.-2.*P(NS,NFR+I))*RA*ALOG(RA/RI) / (2.*(1.-P(NS,NFR+I)))
      XY2 = (RI*RI+(1.-2.*P(NS,NFR+I))*RA*RA)*RO*RO*ALOG(RO/RI) /
     *   (2.*(1.-P(NS,NFR+I))*RA*RR)
      TH = 1.-2.*RI*RI*ALOG(RO/RI) / RR
      B(N6+6+NFR+I)  = RA*(1.+P(NS,NFR+I))*ATBCL(NS,I) + EPR(NS,NFR+I)*
     *  (-RO+RA/2.-XY1+XY2) + EPC(NS,NFR+I)*(-RA/2.+XY1-XY2)
     *   + TSR(NS,NFR+I)*( 0.5*RA+XY1-XY2 )
     *   + TSC(NS,NFR+I)*( 0.5*RA-XY1+XY2 )
     *   + TSZ(NS,NFR+I)*RA*P(NS,NFR+I)
 2300 CONTINUE
C
      B(N7+7)  = CVPR
C
      B(N7+8)  = PO
C
      SUMMA = 0.
      SUMMB = 0.
      DO 2400 N=NS,NNF
      DO 2400 I=1,NFR
 2400 SUMMA = SUMMA + RH(N,I)*VOL(N,I)*GRAV
C
      DO 2450 I=1,NFR
 2450 SUMMB = SUMMB + RH(NS,I)*VOL(NS,I)*GRAV / 2.0
      SUMMAS = SUMMA-SUMMB
      RO = RF(NS,NFR1)
      RI = RF(NS,1)
      RR = (RO+RI) * (RO-RI)
      XG = GB(NS,1)
      XP = PB(NS,1)
C
      SUMRAT = 0.
      SUMEPR = 0.
      SUMTSZ = 0.
      DO 2500 I=1,NFR
      SUMRAT = SUMRAT + (RF(NS,I+1)+RF(NS,I))*(RF(NS,I+1)-RF(NS,I))*
     *                   ATBF(NS,I)
      SUMEPR = SUMEPR + (RF(NS,I+1)+RF(NS,I))*(RF(NS,I+1)-RF(NS,I))*
     *    EPR(NS,I)
      SUMTSZ = SUMTSZ + (RF(NS,I+1)+RF(NS,I))*(RF(NS,I+1)-RF(NS,I))*
     *    TSZ(NS,I)
 2500 CONTINUE
C
C
      B(N7+9)    = -(PLPR+SUMMAS/(PI*RR))/(2.*XG) + (1.+XP)*SUMRAT/
     *                RR          - XP*RI*RI*CVPR/(XG*RR)
     *    + SUMTSZ*( 1.+XP ) / RR
C
      SUMMA = 0.
      SUMMB = 0.
C
      DO 2600 N=NS,NN
      DO 2600 I=1,NCL
 2600 SUMMA = SUMMA + RH(N,NFR+I)*VOL(N,NFR+I)*GRAV
C
      DO 2650 I=1,NCL
 2650 SUMMB = SUMMB + RH(NS,NFR+I)*VOL(NS,NFR+I)*GRAV/2.0
C
```

```
      SUMMAS = SUMMA-SUMMB
C
      SUMRAT = 0.
      SUMEPR = 0.
      SUMTSZ = 0.
      DO 2700 I=1,NCL
      SUMRAT = SUMRAT + (RC(NS,I+1)**2-RC(NS,I)**2)*ATBCL(NS,I)
      SUMTSZ = SUMTSZ + (RC(NS,I+1)**2-RC(NS,I)**2)*TSZ(NS,NFR+I)
 2700 SUMEPR = SUMEPR + (RC(NS,I+1)**2-RC(NS,I)**2)*EPR(NS,NFR+I)
C
      RO = RC(NS,NCL1)
      RI = RC(NS,1)
      RR = (RO+RI) * (RO-RI)
      XP = PB(NS,2)
      XG = GB(NS,2)
C
      B(N7+10)  = (-PTOP-PI*(RO**2*PO-RI**2*PLPR)-SUMMAS)/(2.*PI*XG*RR)+
     *            (1.+XP)*SUMRAT/RR                + XP*RO*RO*PO / (XG*RR)
     *   + SUMTSZ*( 1.+XP ) / RR
C
      J = 1
      DO 2800 I=1,N1
      IF(I-NFR) 500,500,600
  500 RO = RF(NS,I+1)
      RI = RF(NS,I)
      ATB = ATBF(NS,I)
      GO TO 700
  600 RO = RC(NS,J+1)
      RI = RC(NS,J)
      ATB = ATBCL(NS,J)
      J = J+1
  700 CONTINUE
      XP = P(NS,I)
      XG = G(NS,I)
      RR = (RO+RI) * (RO-RI)
      TH = 1.-2.*RI*RI*ALOG(RO/RI)/RR
      OM = 1.-2.*RO*RO*ALOG(RO/RI) / RR
      B(N7+10+I) =
     *   XG*( EPR(NS,I)-EPC(NS,I) ) * ( TH+OM-TH*OM ) / (2.*(1.-XP))
     *      - XG*( OM+TH-TH*OM ) / (2.*(1.-XP)) * TSR(NS,I)
     *      - XG*( TH*OM-    TH-OM ) / (2.*(1.-XP))  * TSC(NS,I)
      B(N8+10+I) =
     *   XG*( EPR(NS,I)-EPC(NS,I) )*( TH*OM-TH-OM ) / (2.*(1.-XP))
     *      - XG*( TH*OM-TH-OM     ) / (2.*(1.-XP))  * TSR(NS,I)
     *      - XG*( OM+    TH-TH*OM     ) / (2.*(1.-XP)) * TSC(NS,I)
      B(N9+10+I)  = -2.*XG*(1.+XP)*ATB
     *      - 2.*XG*(1.+XP) * TSZ(NS,I)
 2800 CONTINUE
C
C
C     VARIABLE CONDITIONS   GAP OPEN/STICK/SLIP
C
 3000 CONTINUE
C
      IF( IFRIC-1 ) 800,900,920
C
  800 CONTINUE
C
```

```
C       OPEN
C
        B(N10+11) = PLPR
        B(N10+12) = 0.
C
        GO TO 990
C
  900 CONTINUE
C
C       STICK OR SLIP
C
        B(N10+11) = EPZ(NS,1)-EPZ(NS,NTR)
        GO TO 930
C
  920 CONTINUE
        B(N10+11) = 0.
C
  930 CONTINUE
        B(N10+12) = RC(NS,1)-RF(NS,NFR1)
C
  990 CONTINUE
C
        RETURN
        END
```

```
      SUBROUTINE BCALCP ( B,ATBCL,TPR,TPC,TPZ,RC,P,G,PB,GB,VOL,RH,TBCL,
     *                    FLUX,KFRIC,CRIF,DELT,TSR,TSC,TSZ,PLPR,PTOP,
     *                    PO,EPR,EPC,EPZ,NS )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      DIMENSION  B(IGENP) , ATBCL(NN,NCL) , TPR(NN,NTR) , TPC(NN,NTR) ,
     *           TPZ(NN,NTR) , RC(NN,NCL1) , P(NN,NTR) , G(NN,NTR) ,
     *           PB(NN,2) , GB(NN,2) , VOL(NN,NTR) , RH(NN,NTR) ,
     *           TBCL(NN,NCL) , FLUX(NN), KFRIC(NN) , CRIF(NN) ,
     *           TSR(NN,NTR) , TSC(NN,NTR) , TSZ(NN,NTR) , EPR(NN,NTR) ,
     *           EPC(NN,NTR) , EPZ(NN,NTR)
C
C
C     RIGHT-HAND VECTORS B OF AX=B
C
C     FOR PLENUM SECTION ONLY
C
C
      DO 999 I=1,IGENP
  999 B(I) = 0.
C
      DO 1000 I=1,NCL
      B(I) = ATBCL(NS,I) + TSR(NS,NFR+I)
 1000 CONTINUE
C
      DO 1100 I=1,NCL
      B(N1C+I) = ATBCL(NS,I) + TSC(NS,NFR+I)
 1100 CONTINUE
C
      DO 1200 I=1,NCL
      B(N2C+I) = -TPR(NS,NFR+I)
 1200 CONTINUE
C
      DO 1300 I=1,NCL
      B(N3C+I) = -TPC(NS,NFR+I)
 1300 CONTINUE
C
      DO 1400 I=1,NCL
      B(N4C+I) = -TPZ(NS,NFR+I)
 1400 CONTINUE
C
      DO 1700 I=1,NCL
      XP = P(NS,NFR+I)
      RI = RC(NS,I)
      RO = RC(NS,I+1)
      RR = (RO+RI)*(RO-RI)
      OM = 1.-2.*RO*RO*ALOG(RO/RI) / RR
      B(N5C+3+I) = RC(NS,I)*(1.+P(NS,NFR+I))*ATBCL(NS,I)
     4  -RI*( OM*(EPR(NS,NFR+I)-EPC(NS,NFR+I) ) / 2. + EPC(NS,NFR+I) )
     * -    ( -0.5*RI+RI*RO*RO*ALOG(RO/RI) / RR ) * TSR(NS,NFR+I)
     * -    ( -0.5*RI-RI*RO*RO*ALOG(RO/RI) / RR ) * TSC(NS,NFR+I)
```

```
     * + RI*XP*TSZ(NS,NFR+I)
 1700 CONTINUE
C
      DO 1800 I=1,NCL
      XP = P(NS,NFR+I)
      RI = RC(NS,I)
      RO = RC(NS,I+1)
      RR = (RO+RI)*(RO-RI)
      TH = 1.-2.*RI*RI*ALOG(RO/RI) / RR
      B(N6C+3+I) = RC(NS,I+1)*(1.+P(NS,NFR+I))*ATBCL(NS,I)
     * -RO*( TH*(EPR(NS,NFR+I)-EPC(NS,NFR+I))/2. + EPC(NS,NFR+I) )
     * -   ( -0.5*RO+RO*RI*RI*ALOG(RO/RI) / RR ) * TSR(NS,NFR+I)
     * -   ( -0.5*RO-RO*RI*RI*ALOG(RO/RI) / RR ) * TSC(NS,NFR+I)
     * + RO*XP*TSZ(NS,NFR+I)
 1800 CONTINUE
C
      B(N7C+4) = PLPR
C
      B(N7C+5)  = PO
C
C     CLAD OF PLENUM SECTION AXIAL DISPLACEMENT
      SUMRAT = 0.
      SUMMAS = 0.
      SUMEPR = 0.
      SUMTSZ = 0.
      DO 1850 I=1,NCL
      SUMMAS = SUMMAS + RH(NS,NFR+I)*VOL(NS,NFR+I)*GRAV / 2.0
      SUMRAT = SUMRAT + (RC(NS,I+1)**2-RC(NS,I)**2)*ATBCL(NS,I)
      SUMEPR = SUMEPR + (RC(NS,I+1)**2-RC(NS,I)**2)*EPR(NS,NFR+I)
      SUMTSZ = SUMTSZ + (RC(NS,I+1)**2-RC(NS,I)**2)*TSZ(NS,NFR+I)
 1850 CONTINUE
C
      RO = RC(NS,NCL1)
      RI = RC(NS,1)
      RR = (RO+RI) * (RO-RI)
      XP = PB(NS,2)
      XG = GB(NS,2)
C
      B(N7C+6)  = (-PTOP-PI*(RO**2*PO-RI**2*PLPR)-SUMMAS)/(2.*PI*XG*RR)+
     *            (1.+XP)*SUMRAT/RR                + XP*RO*RO*PO / (XG*RR)
     * + SUMTSZ*(1.+XP)/RR
C
C
C     STRESS
C
C
      DO 1900 I=1,NCL
      RO = RC(NS,I+1)
      RI = RC(NS,I)
      XP = P(NS,NFR+I)
      XG = G(NS,NFR+I)
      RR = (RO+RI) * (RO-RI)
      ATB = ATBCL(NS,I)
      TH = 1.-2.*RI*RI*ALOG(RO/RI) / RR
      OM = 1.-2.*RO*RO*ALOG(RO/RI) / RR
C
      B(N7C+6+I) =
     *   XG*( EPR(NS,NFR+I)-EPC(NS,NFR+I) )*( TH+OM-TH*OM )/(2.*(1.-XP))
```

```
     *        - XG*( OM+TH-TH*OM ) / (2.*(1.-XP)) * TSR(NS,NFR+I)
     *        - XG*( TH*OM-   TH-OM ) / (2.*(1.-XP)) * TSC(NS,NFR+I)
      B(N8C+6+I) =
     *     XG*( EPR(NS,NFR+I)-EPC(NS,NFR+I) )*( TH*OM-TH-OM )/(2.*(1.-XP))
     *        - XG*( TH*OM-TH-OM     ) / (2.*(1.-XP)) * TSR(NS,NFR+I)
     *        - XG*( OM+    TH-TH*OM     ) / (2.*(1.-XP)) * TSC(NS,NFR+I)
      B(N9C+6+I)  = -2.*XG*(1.+XP)*ATB
     *        - 2.*XG*(1.+XP) * TSZ(NS,NFR+I)
 1900 CONTINUE
C
      B(N10C+7) = 0.
C
C
C     FRICTION FORCE
C
C
C
      RETURN
      END
```

```
      SUBROUTINE BEGIN ( XM,ZFLX,RH,RHOP,CPU,LM,EHP,SWS,GS,GR,
     *                   RVB,RCB,REB,RUB,QSF,QSFF,QSFP,POW,XCPI,XCPO,
     4                   BU,HG,KFRIC,FLZO,FLZ,PRS,FLNC,CLZO,CLZ,
     *                   FL,CL,DXM,SWG,RCI,RCO,RVBO,RCBO,REBO,RUBO,
     *                   RCIO,RCOO,XMINT,RDIST,EPSEQ,SWS3,LMR,LMC,LMZ )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /CONST/ FKA1,FKB1,FKC1,FKA2,FKB2,FKC2,FKA3,FKB3,FKC3,
     *               FA11,FA12,FA21,FA22,FA31,FA32,
     *               CK1,CK2,CA1,CA2,XEWTON(7)
      COMMON /CAPSUL/ RCPI,RCPO,AFNA
      DIMENSION  XM(NN, 3 ) , ZFLX(NN) , RH(NN,NTR) , RHOP(NN,3) ,
     *           EHP(NN,NTR) , SWS(NN,NTR) , GS(NN,NFR) , RVB(NN) ,
     *           RCB(NN) , REB(NN) , RUB(NN) , QSF(NN) , QSFF(NN) ,
     *           QSFP(NN) , POW(NN) , XCPI(NN) , XCPO(NN) , BU(NN) ,
     *           HG(NN) , KFRIC(NN) , FLZO(NN) , FLZ(NN) ,
     *           SWG(NN,NFR)   , PRS(NN,NTR) , FLNC(NN) , CLZO(NN) ,
     *           CPU(NN,NP) , LM(NN,NTR) , CLZ(NN) , GR(NN) , DXM(NN,3),
     *           RCI(NN) , RCO(NN) , RVBO(NN) , RCBO(NN) , REBO(NN) ,
     *           RUBO(NN) , RCIO(NN) , RCOO(NN) , XMINT(NN) ,
     *           RDIST(NN,4) , EPSEQ(NN,NTR) , SWS3(NN,NUR) ,
     *           LMR(NN,NTR) , LMC(NN,NTR) , LMZ(NN,NTR)
C
C
C     INITIAL VALUES SET
C
C
      FL = 0.0
      CL = 0.0
      DO 1000 N=1,NN
      DO 3000 I=1,4
 3000 RDIST(N,I)=0.
C
      IF ( N.EQ.NN ) GO TO 100
C
      ZFLX(N) = 1.053E-22/0.85
      FLZO(N) = CLZO(N)
      FLZ(N) = FLZO(N)
      CLZ(N) = CLZO(N)
      FL = FL+FLZO(N)
      CL = CL+CLZO(N)
      DXM(N,1) = 0.
      DXM(N,2) = 0.
      DXM(N,3) = 0.
      KFRIC(N) = 0
      DO 1010 K=1,NCR
      RH(N,K)=RHOP(N,1)
 1010 CONTINUE
      XM(N,1) = PI*RHOP(N,1)*(RCB(N)-RVB(N))*(RCB(N)+RVB(N))
      KK=NCR+1
      KKK=NCR+NER
```

```
      DO 1020 K=KK,KKK                                                  00050300
      RH(N,K)=RHOP(N,2)                                                 00050400
 1020 CONTINUE                                                          00050500
      XM(N,2) = PI*RHOP(N,2)*(REB(N)-RCB(N))*(REB(N)+RCB(N))            00050600
      KK = NCR+NER+1                                                    00050700
      KKK=NFR                                                           00050800
      DO 1030 K=KK,KKK                                                  00050900
      RH(N,K)=RHOP(N,3)                                                 00051000
 1030 CONTINUE                                                          00051100
      XM(N,3) = PI*RHOP(N,3)*(RUB(N)-REB(N))*(RUB(N)+REB(N))            00051200
C                                                                       00051300
  100 CONTINUE                                                          00051400
C                                                                       00051500
      KK = NFR+1                                                        00051600
      KKK = NTR                                                         00051700
      DO 1035 K=KK,KKK                                                  00051800
      RH(N,K) = RHOC                                                    00051900
 1035 CONTINUE                                                          00052000
C                                                                       00052100
      RVBO(N) = RVB(N)                                                  00052200
      RCBO(N) = RCB(N)                                                  00052300
      REBO(N) = REB(N)                                                  00052400
      RUBO(N) = RUB(N)                                                  00052500
      RCIO(N) = RCI(N)                                                  00052600
      RCOO(N) = RCO(N)                                                  00052700
C                                                                       00052800
      DO 1040 K=1,NFR                                                   00052900
      SWG(N,K) = 0.                                                     00053000
      SWS(N,K) = 0.0                                                    00053100
 1040 GS(N,K) = 0.0                                                     00053200
C                                                                       00053300
      DO 1050 K=1,NUR                                                   00053400
 1050 SWS3(N,K) = 0.                                                    00053500
      DO 1000 K=1,NTR                                                   00053600
      LM(N,K) = 0                                                       00053700
      LMR(N,K) = 0                                                      00053800
      LMC(N,K) = 0                                                      00053900
      LMZ(N,K) = 0                                                      00054000
      EHP(N,K) = 0.                                                     00054100
      PRS(N,K) = 0.                                                     00054200
      EPSEQ(N,K) = 0.                                                   00054300
 1000 CONTINUE                                                          00054400
C                                                                       00054500
      DO 1400 I=1,NFR                                                   00054600
 1400 RH(NN,I) = 0.                                                     00054700
      DO 1500 I=1,NCL                                                   00054800
 1500 RH(NN,NFR+I) = RHOC                                               00054900
C                                                                       00055000
      RHOP(NN,1) = 0.                                                   00055100
      RHOP(NN,2) = 0.                                                   00055200
      RHOP(NN,3) = 0.                                                   00055300
      GR(NN) = 0.                                                       00055400
      FLZO(NN) = 0.0                                                    00055500
      FLZ(NN) = FLZO(NN)                                                00055600
      CLZ(NN) = CLZO(NN)                                                00055700
      FL = FL+FLZO(NN)                                                  00055800
      CL = CL+CLZO(NN)                                                  00055900
C                                                                       00056000
```

```
      ALFO(1) = FA11
      ALF1(1) = FA12
      ALFO(2) = FA21
      ALF1(2) = FA22
      ALFO(3) = FA31
      ALF1(3) = FA32
      QSF(NN) = 0.
      POW(NN) = 0.
      SMQ = 0.
      X = 0.
      DO 2000 N=1,NNF
      SM = 0.
      DO 2010 K=1,3
 2010 SM = SM + XM(N,K)
      XMINT(N) = SM
      QSF(N) = SM * QSFP(N)
      SMQ = SMQ + QSF(N)
 2000 X = X + QSFF(N)
      DO 2020 N=1,NN
      XCPO(N) = RCPO
      XCPI(N) = RCPI
      FLNC(N) = 0.
      BU(N) = 0.0
      HG(N) = 4.0
      QSF(N) = QSF(N)*NNF / SMQ
 2020 QSFF(N) = QSFF(N)*NNF / X
      SMQO = SMQ
C
C=========  INTEGRATION BY NEWTON-COTES-6
C
      XEWTON( 1 ) = 41./140.
      XEWTON( 2 ) = 216./140.
      XEWTON( 3 ) = 27./140.
      XEWTON( 4 ) = 272./140.
      XEWTON( 5 ) = 27./140.
      XEWTON( 6 ) = 216./140.
      XEWTON( 7 ) = 41./140.
C
      RETURN
      END
```

```
      SUBROUTINE BURN ( BU,POW,XM,EPZ,QSF,QSFP,DELT )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      DIMENSION  BU(NN) , POW(NN) , XM(NN,3) , EPZ(NN,NTR) ,
     *          QSF(NN) , QSFP(NN)
C
C
C     BURN UP CALCULATION
C
C
      SMQ = 0.
      BUB = 0.
C
      DO 1000 N=1,NNF
      SM = XM(N,1)+XM(N,2)+XM(N,3)
      BU(N) = BU(N) + BUCNS*POW(N)*DELT/SM
      BUB = BUB + BU(N)
      QSF(N) = SM*QSFP(N)*(1.-PDBU*BU(N))
 1000 SMQ = SMQ + QSF(N)
C
      BUB = BUB/NNF
C
C
      RETURN
      END
```

```
      SUBROUTINE CHECK ( DEQP,EF,EC,ISW )
      DIMENSION  DEQP(NN,NTR)
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C
C     CREEP STRAIN INCREMENT CHECK
C
      ISW = 0
      DO 100 N=1,NN
      IF(N.EQ.NN) GO TO 250
      DO 200 J=1,NFR
      IF ( DEQP(N,J).GT.EF ) GO TO 400
  200 CONTINUE
C
  250 CONTINUE
      DO 300 J=1,NCL
      IF ( DEQP(N,NFR+J).GT.EC ) GO TO 400
  300 CONTINUE
C
  100 CONTINUE
      RETURN
C
  400 CONTINUE
      ISW = 1
      RETURN
      END
```

```
      SUBROUTINE CHKSPD ( DXM,EPS,NN,ISW,DELT )
      DIMENSION  DXM(NN,3)
C
C     RESTRUCTURING VELOCITY CHECK
C
      ISW = 0
      DO 100 N=1,NN
      IF ( N.EQ.NN ) GO TO 100
      DO 100 J=2,3
      X = DXM(N,J)*DELT*3600.0
      IF ( X.GT.EPS ) GO TO 200
  100 CONTINUE
      RETURN
  200 CONTINUE
      ISW = 1
      RETURN
      END
```

```
      SUBROUTINE CLEAR(IFRIC,JFRIC,NS)                                  00063300
      DIMENSION JFRIC(1)                                                00063400
C                                                                       00063500
C     ASSUME , GAP IS OPEN                                              00063600
C                                                                       00063700
C     IFRIC = JFRIC(NS)                                                 00063800
      IFRIC = 0                                                         00063900
      RETURN                                                            00064000
      END                                                               00064100
```

```
      SUBROUTINE CLTMP ( POW,TO,RO,RI,R,T,TB,TBA,ATB,ATBA,RCLO,IT,RB )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /CONST/ FKA1,FKB1,FKC1,FKA2,FKB2,FKC2,FKA3,FKB3,FKC3,
     *              FA11,FA12,FA21,FA22,FA31,FA32,
     *              CK1,CK2,CA1,CA2,A(7)
      DIMENSION POW(NN) , TO(NN) , RO(NN) , RI(NN) , R(NN,NCL1) ,
     *          T(NN,NCL1) , TB(NN,NCL) , TBA(NN) , ATB(NN,NCL) ,
     *          ATBA(NN) , RCLO(NN,NCL1) , RB(NN,NTR)
C
      DIMENSION  RCL(7) , TCL(7) , ATCL(7)
C
C
C     CLADDING/CAPSULE TEMPERATURE DISTRIBUTION CALC.
C
C
      Y = 20.0
      DO 1000 N=1,MN
      P = POW(N) / PI
      DR = ( RO(N)-RI(N) ) / NCL
      DO 1010 K=1,NCL1
 1010 R(N,K) = RI(N) + DR*(K-1)
      IF(IT.NE.1) GO TO 1016
      DO 1015 K=1,NCL1
 1015 RCLO(N,K) = R(N,K)
 1016 CONTINUE
C
      DO 1017 K=1,NCL
      RB(N,NFR+K) = 2.*(R(N,K+1)*R(N,K+1)+R(N,K+1)*R(N,K)+R(N,K)*R(N,K))
     *   / (3.*(R(N,K+1)+R(N,K)))
 1017 CONTINUE
C
      TOF = TO(N)
      AK = ( CK1+CK2*TOF )
      AK2 = AK*AK
      DRR = ( RO(N)-RI(N) ) / 6.
      S = 0.
      S1 = 0.
      DO 1020 L=1,7
      RR = RI(N) + DRR*(L-1)
      RCL(L) = RR
      XL = ALOG( RO(N) / RR )
      TT = TOF   + P*XL / ( AK + SQRT( AK2+CK2*P*XL ) )
      TCL(L) = TT
C     TCL NOW IN DEG C
      ATCL(L) = CA1*TT + CA2*TT**2/2. - CA1*Y - CA2*Y**2/2.
      S1 = S1 + A(L)*RR*ATCL(L)
      S = S + A(L)*RR*TCL(L)
C
 1020 CONTINUE
C
C
      IF ( N.EQ.NN ) GO TO 90
```

```
      TBA(N) = S / ( RO(N)+RI(N) ) / 3.
      GO TO 95
   90 TBA(N) = TO(N)
   95 CONTINUE
      ATBA(N) = S1 / ( RO(N)+RI(N) ) / 3.
C
      IF( NCL.NE.1 ) GO TO 100
C
C
C     NCL = 1
C
      T(N,1) = TCL(1)
      T(N,2) = TO(N)
      TB(N,1) = TBA(N)
      ATB(N,1) = ATBA(N)
      GO TO 1000
C
C
  100 CONTINUE
C
C     NCL . GE . 2
C
      L = 2
      T(N,1) = TCL(1)
      S1 = RCL(1)
      S2 = 0.
      DO 1030 K=1,NCL
C
  150 CONTINUE
      IF ( R(N,K+1)-RCL(L) ) 300,300,200
  200 CONTINUE
      DRDR = RCL(L)-S1
      ARA = AREA( S1,DRDR )
      ARA = ARA*ATCL(L-1)
      S2 = S2 + ARA
      S1 = RCL(L)
      L = L+1
      IF ( L.GT.7 ) GO TO 310
      GO TO 150
  300 CONTINUE
      DRDR = DRR-(RCL(L)-R(N,K+1))
      ARA = AREA( S1,DRDR )
      ARA = ARA*ATCL(L-1)
      S2 = S2 + ARA
  310 CONTINUE
      IF ( L.GT.7 ) L=7
      RAT = ( RCL(L)-R(N,K+1) ) / DRR
      T(N,K+1) = TCL(L) + RAT*(TCL(L-1)-TCL(L))
      TB(N,K) = ( T(N,K)+T(N,K+1) ) / 2.
      ATB(N,K) = S2 / ( DR*(2.*R(N, K )+DR) )
      S1 = R(N,K+1)
      S2 = 0.0
C
 1030 CONTINUE
C
      T(N,NCL1) = TO(N)
C
 1000 CONTINUE
```

```
C                                                                       00075800
      RETURN                                                            00075900
      END                                                               00076000
```

```
      SUBROUTINE COEF ( OTOM,N )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /CONST/ FKA1,FKB1,FKC1,FKA2,FKB2,FKC2,FKA3,FKB3,FKC3,
     *               FA11,FA12,FA21,FA22,FA31,FA32,
     *               CK1,CK2,CA1,CA2,XEWTON(7)
      COMMON /FBRATR/ JFA
C
      DIMENSION  OTOM(NN)
C
C
C     COEFFICIENTS FOR FUEL THERMAL CONDUCTIVITY
C
C
      OTOMX = 2.00 - OTOM(N)
      IF(OTOMX) 100,100,200
  100 OTOMX = -2.*OTOMX
  200 DELKF = 5.0*OTOMX**2 - 3.7*OTOMX + 1.0
      FKA1 = -0.018*DELKF
      FKB1 = 54.0*DELKF
      FKC1 = 1.75E-12 * DELKF
      FKA2 = 0.011*DELKF
      FKB2 = 0.485 / DELKF
      FKC2 = 0.4465 / DELKF
      IF(JFA-2) 10,20,20
   10 CONTINUE
      FKA3 = 3.11 + 355.0*OTOMX
      FKB3 = 0.0272
      FKC3 = 5.39E-13
      GO TO 30
   20 CONTINUE
      FKA3 = 3.38
      FKB3 = 0.0262
      FKC3 = 4.79E-13
      IF(NOK.EQ.4) FKC3 = 6.1256E-13
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CORTMP ( TOUTER,TOUTO,PTOUTR,ISW,TOUT )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /IOFIL/ I05,I06,I07
      DIMENSION  TOUTER(NN) , TOUTO(NN) , PTOUTR(NN)
C
C     COOLANT TEMPERATURE ADJUSTMENT DUE TO TIME CUT
C
      IF(ISW.EQ.0) GO TO 50
      GO TO (10,20,30,40) , ISW
   10 DO 100 I=1,NN
  100 TOUTER(I) = TOUTO(I)
      GO TO 60
   20 DO 200 I=1,NN
  200 TOUTO(I) = TOUTER(I)
      GO TO 60
   30 DO 300 I=1,NN
  300 TOUTER(I) = PTOUTR(I)+( TOUTER(I)-PTOUTR(I) ) / 2.0
      GO TO 60
   40 DO 400 I=1,NN
  400 PTOUTR(I) = TOUTER(I)
      GO TO 60
   50 READ(I05,9000) ( TOUTER(I),I=1,NN )
      TOUT = TOUTER(NN)
   60 RETURN
 9000 FORMAT(5E15.3)
      END
```

```
      SUBROUTINE CRACK
     *          ( SIGR,SIGC,SIGZ,BM,G,TBF,TPR,TPC,TPZ,TSR,TSC,TSZ,
     *            PRS,LM,LMR,LMC,LMZ,DELT,NS,JCR )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      DIMENSION  SIGR(NTR) , SIGC(NTR) , SIGZ(NTR) ,
     *           BM(NN,NTR) , G(NN,NTR) , TBF(NN,NFR) , TPR(NN,NTR) ,
     *           TPC(NN,NTR) , TPZ(NN,NTR) , TSR(NN,NTR) ,
     *           TSC(NN,NTR) , TSZ(NN,NTR) , PRS(NN,NTR) ,
     *           LM(NN,NTR) , LMR(NN,NTR) , LMC(NN,NTR) ,
     *           LMZ(NN,NTR)
      COMMON /XCRAK/ THLC,HEAL1,HEAL2,SIGF0,SIGF1
C
C
C     CRACKING CHECK                J.T.A ROBERTS AND B.J. WRONA
C
C
      JCR = 0
C
      IF ( DELT.LE.0.001 )  RETURN
C
      DO 1000 I=1,NFR
      TS = TBF(NS,I) + 273.
C
      SIG = SIGR( I )
      L = 1
C
C     L = 1  RADIAL DIRECTION
C     L = 2  THETA  DIRECTION
C     L = 3  AXIAL  DIRECTION
C
      IF ( SIG.GT.SIGC(I) ) GO TO 10
      SIG = SIGC( I )
      L = 2
   10 CONTINUE
      IF ( SIG.GT.SIGZ(I) ) GO TO 20
      SIG = SIGZ( I )
      L = 3
   20 CONTINUE
C
C
C     CRACKING CHECK
C
      SIGF = SIGF0 + SIGF1*TS
C
C     SIGF MEANS CRITICAL STRESS OF BRITTLE-TO-DUCTILE TRANSITION
C
      IF ( SIG.LT.SIGF ) GO TO 500
C
      JCR = 1
      TSR(NS,I) = TSR(NS,I) + SIG/ ( 9.*BM(NS,I) )
      TSC(NS,I) = TSR(NS,I)
      TSZ(NS,I) = TSR(NS,I)
      X = SIG / ( 6.*G(NS,I) )
```

```
      IF ( L-2 ) 30,40,50
   30 CONTINUE
      LMR(NS,I) = LMR(NS,I) + 1
      TPR(NS,I) = TPR(NS,I) + 2.*X
      TPC(NS,I) = TPC(NS,I) - X
      TPZ(NS,I) = TPZ(NS,I) - X
      GO TO 60
C
   40 CONTINUE
      LMC(NS,I) = LMC(NS,I) + 1
      TPC(NS,I) = TPC(NS,I) + 2.*X
      TPR(NS,I) = TPR(NS,I) - X
      TPZ(NS,I) = TPZ(NS,I) - X
      GO TO 60
C
   50 CONTINUE
      LMZ(NS,I) = LMZ(NS,I) + 1
      TPZ(NS,I) = TPZ(NS,I) + 2.*X
      TPR(NS,I) = TPR(NS,I) - X
      TPC(NS,I) = TPC(NS,I) - X
   60 CONTINUE
C
      LM(NS,I) = LMR(NS,I) + LMC(NS,I) + LMZ(NS,I)
C
      GO TO 1000
C
C
C     HEALING CHECK
C
C
  500 CONTINUE
      IF ( LM(NS,I).EQ.0 ) GO TO 1000
      HEAL = HEAL1
      THL = THLC + 273.
      IF ( TS.GT.THL ) HEAL = HEAL2
      IF ( -(SIGR(I)+SIGC(I)+SIGZ(I))/3. .LE. -HEAL )
     *    GO TO 1000
      LMR(NS,I) = 0
      LMC(NS,I) = 0
      LMZ(NS,I) = 0
      LM(NS,I) = 0
C
      JCR = 2
C
C
 1000 CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE CREEC ( EQSIG,DEQP,TBC,FLNC,FLUX,DELT,N )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XCREC/ A,Q,AE,BE,CE,FM,FN
      COMMON /FBRATR/ JFA
C
      DIMENSION  EQSIG(NN,NTR) , DEQP(NN,NTR) , TBC(NN,NCL) ,
     *           FLNC(NN) , FLUX(NN)
C
C
C     CLADDING CREEP STRAIN CALC.
C
C
      DO 1000 I=1,NCL
C
      J = NFR + I
      T = TBC(N,I)*1.8 + 32. + 460.
C     T NOW IN DEG R
      IF( EQSIG(N,J).LT.10.0 ) EQSIG(N,J)=10.0
C     PREVENTING ZERO DIVIDE
      ES = EQSIG(N,J) / 7.03E-02
C
      IF(JFA-2) 50,500,500
   50 CONTINUE
      IF( NOC+1 ) 100,200,200
  100 C = T/910.
      GO TO 300
  200 C = 1.0
  300 CONTINUE
C
      AEC = AE/C
      BEC = BE*C
      CEC = CE/C
C
      B = CEC + AEC/BEC * EXP( -FLNC(N)/BEC )
C
C
C     RKR IS IDEAL GAS CONSTANT IN CAL/MOLE/DEG R
C
      X1 = A * EXP( -Q/(RKR*T) ) * ES**FM
      X2 = B * FLUX(N) * ES**FN
C
      DEP = (X1+X2) * DELT
      GO TO 900
  500 IF(NOC+1) 700,600,600
  600 DEP = 4.0E-27*ES*FLUX(N)*(TBC(N,I)-160.)*DELT
      GO TO 900
  700 TEMP = TBC(N,I)+273.
      ES = 1.63E-03*EQSIG(N,J)
      XFLX = FLUX(N)
```

```
      DEP = 1.02E-13*EXP(-7053./TEMP)*XFLX**0.85*SINH(ES)*DELT
  900 IF(DEP.LE.0.) DEP = 1.0E-30
      DEQP(N,J) = DEP
C
 1000 CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE CREEF ( EQSIG,DEQP,TBF,POW,FD,FLUX,FTHP,LM,DELT,N,
     *                   G,EPSEQ )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XCREF/ A0,A1,CFR,B,Q,Q1,FM,FN,GD
C
      DIMENSION  EQSIG(NN,NTR) , DEQP(NN,NTR) , TBF(NN,NFR) , POW(NN) ,
     *           FD(NN,NFR) , FLUX(NN)  , FTHP(NN,NTR) , LM(NN,NTR) ,
     *           G(NN,NTR) , EPSEQ(NN,NTR)
C
C
C     FUEL CREEP STRAIN CALC.
C
C
      DO 1000 I=1,NFR
      FSRT = POW(N)*FD(N,I) * 3.1E+10
      TX = TBF(N,I)*1.8 + 32. + 460.
C     TX NOW IN DEG R
C
C     RKR IS IDEAL GAS CONSTANT IN CAL/MOLE/DEG R
C     FSRT IS NO. OF FISSIONS/ SEC / CC
C     TX NOW IN DEG R
C
      IF( EQSIG(N,I).LT.10.0 ) EQSIG(N,I)=10.0
C     PREVENTING ZERO DIVIDE
      IF ( EPSEQ(N,I).GT.0.0) GO TO 100
      EPSEQ(N,I) = 1.E-30
      DEQP(N,I) = EPSEQ(N,I)
      GO TO 1000
  100 CONTINUE
      ES = EQSIG(N,I) / 7.03E-02
      LML = LM(N,I)
C     A = A0*1.5**(LML*(FM-1.))
      A = A0
      X0 = 0.0
      IT = 0
  200 CONTINUE
      X1 = A*EXP(-Q/(RKR*TX))*(1.+FSRT/CFR)*ES**FM
C     X2 = A1*EXP(-Q1/(RKR*TX))/(GD**2) + B*FLUX(N)
      X2 = A1*EXP(-Q1/(RKR*TX))/(GD**2) + B*FSRT
      X2 = X2*ES**FN
      X3 = ( FM*X1+FN*X2 )*DELT
      X2 = (X1+X2)*DELT
      IF(X2.LE.0.) GO TO 500
      X4 = ABS(X2-X0)/X2
      X0 = X2
      X = ES
      ES = ES*( EPSEQ(N,I)-X2+X3 ) / ( X3+ES/G(N,I)/3. )
      IF(ES.LT.0.) GO TO 500
      IT = IT + 1
      IF (IT.GT.100) GO TO 500
```

```
      IF (X4-0.0001) 300,200,200
  300 IF (X4-0.0000001) 500,400,400
  400 X5 = ABS(ES-X)
      IF ( X5-0.5 ) 500,200,200
  500 CONTINUE
      DEP = X2
      IF( DEP.LE.0. ) DEP=1.E-30
      DEQP(N,I) = DEP
C
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE EDTGRS ( IT,TIM,POWAV,TIN,TOUT,DELT,POW,FLUX,BU,TNA,
     *                    XCPO,XCPI,TCPI,TCPO,RCO,RCI,TCLO,TCL,FLZ,CLZ,
     *                    FL,CL,OTOM,CPU,RHOT,RH,XM,RVB,RCB,REB,
     *                    RUB,RF,RC,HG,HC,TOGAS,GR,TFR,TBFA,ATBFA,
     *                    TBCLA,ATBLA,RCP,TCP,TBCPA,ATBPA,FLNC,GS,
     *                    PLPR,CVPR,FLO,CLO ,BUM,PO,FD,FDPM )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /CAPSUL/ RCPI,RCPO,AFNA
      COMMON /COMPLT/ YP,DDOD,TITLE(20)
      COMMON /IOFIL/ I05,I06,I07
C
      DIMENSION  POW(NN) , FLUX(NN) , BU(NN) , TNA(NN) , XCPO(NN) ,
     *           XCPI(NN) , TCPI(NN) , TCPO(NN) , RCO(NN) , RCI(NN) ,
     *           TCLO(NN) , TCL(NN,NCL1) , FLZ(NN) , CLZ(NN) ,
     *           OTOM(NN) , RHOT(NN) , RH(NN,NTR) , XM(NN,3) ,
     *           RVB(NN) , RCB(NN) , REB(NN) , CPU(NN,NP) ,
     *           RUB(NN) , RF(NN,NFR1) , RC(NN,NCL1) , HG(NN) ,
     *           HC(NN) , GR(NN) , TFR(NN,NFR1) , TBFA(NN,3) ,
     *           ATBFA(NN,3) , TBCLA(NN) , ATBLA(NN) , RCP(NN,NCL1) ,
     *           TCP(NN,NCL1) , TBCPA(NN) , ATBPA(NN) , FLNC(NN) ,
     *           GS(NN,NFR),BUM(NN) , FD(NN,NFR) , FDPM(NN,NP)
C
C
C     GROSS OUTPUT OF EACH REGION FOR EACH STEP
C
C
      DO 50 K=1,NN
   50 BUM(K)=BU(K)*8386.0
      WRITE(I06,9000) IT,TIM,POWAV,TOUT,TIN,DELT,CFLX,PO
      WRITE(I06,9001)
      WRITE(I06,9002) POW
      WRITE(I06,9003) FLUX
      WRITE(I06,9004) FLNC
      WRITE(I06,9005) BU
      WRITE(I06,9006) BUM
      WRITE(I06,9007) FLZ
      WRITE(I06,9008) CLZ
      WRITE(I06,9009) RHOT
      WRITE(I06,9010) TNA
      DO 100 K=1,NN
      HC(K) = HC(K) * 0.002093
  100 HG(K) = HG(K) * 0.002093
      WRITE(I06,9011) HC
      WRITE(I06,9012) HG
      DO 200 K=1,NN
      HC(K) = HC(K) / 0.002093
  200 HG(K) = HG(K) / 0.002093
      WRITE(I06,9013) GR
      TGS = 0.
      DO 900 N=1,NN
```

```
      DO 900 K=1,NFR
      TGS = TGS + GS(N,K)
  900 CONTINUE
      X = TOGAS - GASI
      YP = 100.*X/(TGS+X)
      WRITE(I06,9014) TOGAS,X,YP
      WRITE(I06,9015) PLPR,CVPR
      FDLBL = 100.*(FL-FLO)/FLO
      CDLBL = 100.*(CL-CLO)/CLO
      WRITE(I06,9016) FL,FDLBL,CL,CDLBL
      WRITE(I06,9101)
      DO 1000 N=1,NN
      K = N
      WRITE(I06,9102) K
      IF(K.EQ.NN) GO TO 950
      IF(XCPI(K).GE.XCPO(K)) GO TO 910
      WRITE(I06,9103) RVB(K),RCB(K),REB(K),RUB(K),RCI(K),RCO(K),XCPI(K),
     *                XCPO(K)
      WRITE(I06,9104) TFR(K,1),TFR(K,NCR+1),TFR(K,NCR+NER+1),TFR(K,NFR1)
     *               ,TCL(K,1),TCL(K,NCL1),TCP(K,1),TCP(K,NCL1)
      WRITE(I06,9105) TBFA(K,1),TBFA(K,2),TBFA(K,3),TBCLA(K),TBCPA(K)
      WRITE(I06,9106) ATBFA(K,1),ATBFA(K,2),ATBFA(K,3),ATBLA(K),ATBPA(K)
      WRITE(I06,9107) RH(K,1),RH(K,NCR+1),RH(K,NCR+NER+1),RH(K,NFR+1)
      WRITE(I06,9108) XM(K,1),XM(K,2),XM(K,3)
      GO TO 920
  910 WRITE(I06,9103) RVB(K),RCB(K),REB(K),RUB(K),RCI(K),RCO(K)
      WRITE(I06,9104) TFR(K,1),TFR(K,NCR+1),TFR(K,NCR+NER+1),TFR(K,NFR1)
     *               ,TCL(K,1),TCL(K,NCL1)
      WRITE(I06,9105) TBFA(K,1),TBFA(K,2),TBFA(K,3),TBCLA(K)
      WRITE(I06,9106) ATBFA(K,1),ATBFA(K,2),ATBFA(K,3),ATBLA(K)
      WRITE(I06,9107) RH(K,1),RH(K,NCR+1),RH(K,NCR+NER+1),RH(K,NFR+1)
      WRITE(I06,9108) XM(K,1),XM(K,2),XM(K,3)
  920 CONTINUE
      GO TO 1000
  950 X=0.0
      IF(XCPI(K).GE.XCPO(K)) GO TO 960
      WRITE(I06,9103) X,X,X,X,RCI(K),RCO(K),XCPI(K),XCPO(K)
      WRITE(I06,9104) X,X,X,X,TCL(K,1),TCL(K,NCL1),TCP(K,1),TCP(K,NCL1)
      WRITE(I06,9105) X,X,X,TBCLA(K),TBCPA(K)
      WRITE(I06,9106) X,X,X,ATBLA(K),ATBPA(K)
      WRITE(I06,9107) X,X,X,RH(K,NFR+1)
      GO TO 970
  960 WRITE(I06,9103) X,X,X,X,RCI(K),RCO(K)
      WRITE(I06,9104) X,X,X,X,TCL(K,1),TCL(K,NCL1)
      WRITE(I06,9105) X,X,X,TBCLA(K)
      WRITE(I06,9106) X,X,X,ATBLA(K)
      WRITE(I06,9107) X,X,X,RH(K,NFR+1)
  970 CONTINUE
 1000 CONTINUE
      WRITE(I06,9050)
C
C
      RETURN
C
 9000 FORMAT ( 1H1, / ' ' , 114('*') ,  /
     *              '  STEP NO.',I3,5X,'TIME (HRS) =',F11.3,7X,'POWAV (W/
     *CM) =',F11.3,5X,'TOUT (C) =',F8.3,2X,'TIN (C) =',F8.3, / 18X,'DELT
     * (HRS) =',F11.3,7X,'CFLX (N/W/CM) =',E11.3 ,4X,'PO (KG/CM**2) =' ,
```

```
     *F7.2, / ' ',114('*') )
 9001 FORMAT(  / 35X,'=====  GROSS OUTPUT NEXT  =====',//,1H ,'+++  AXIA
     *L DISTRIBUTION (N=1,NN)  +++' , / )
 9002 FORMAT(1H ,'    POWER(W/CM)          '  ,7F15.3,(/,1H ,23X,7F15.3) )
 9003 FORMAT(1H ,'    FLUX(N/CM**2/SEC)     ',7E15.5,(/,1H ,25X,7E15.5) )
 9004 FORMAT(1H ,'    FLUENCE(N/CM**2)      ',7E15.5,(/,1H ,25X,7E15.5) )
 9005 FORMAT(1H ,'    BURN UP(A/O)          ',7F15.5,(/,1H ,25X,7F15.5) )
 9006 FORMAT(1H ,'    BURN UP(MWD/T)     '  ,7F15.1,(/,1H ,21X,7F15.1) )
 9007 FORMAT(1H ,'    FUEL LENGTH(CM)       ',7F15.5,(/,1H ,25X,7F15.5) )
 9008 FORMAT(1H ,'    CLAD LENGTH(CM)       ',7F15.5,(/,1H ,25X,7F15.5) )
 9009 FORMAT(1H ,'    THEO. DENS.(G/CC)     ',7F15.5,(/,1H ,25X,7F15.5) )
 9010 FORMAT(1H ,'    COOLANT TEMP.(C)     '  ,7F15.3,(/,1H ,23X,7F15.3) )
 9011 FORMAT(1H ,'    FILM H.(W/CM**2/C)    ',7F15.5,(/,1H ,25X,7F15.5) )
 9012 FORMAT(1H ,'    GAP H.(W/CM**2/C)     ',7F15.5,(/,1H ,25X,7F15.5) )
 9013 FORMAT(1H ,'    DELT GAS REL.(MOLES)',4X,7F15.7,(/,1H ,27X,7F15.7))
 9014 FORMAT( 1H0, '    TOTAL GAS IN PLENUM', 16X, F15.8, ' (MOLES)',  /
     *     ,1H ,    '    CUMULATIVE GAS RELEASED', 12X, F15.8, ' (MOLES)',
     */    ,1H ,    '    PERCENT OF TOTAL F.P. GAS', 7X, F15.5, '     (0/0)
     *'   )
 9015 FORMAT( 1H , '    PLENUM PRESSURE', 15X, F15.3, 6X, '(KG/CM**2)', /
     *     ,1H ,    '    CENTRAL HOLE PRESSURE', 9X, F15.3, 6X, '(KG/CM**2
     *)'   )
 9016 FORMAT( 1H , '    TOTAL FUEL LENGTH AND FL/FLO', 4X, F15.5, '     (C
     *M)', 10X, F15.5, 2X, '(0/0)', /
     *     ,1H ,    '    TOTAL CLAD LENGTH AND CL/CLO', 4X, F15.5, '     (C
     *M)', 10X, F15.5, 2X, '(0/0)' )
 9101 FORMAT( 1H0,'  +++  PERFORMANCE  +++' )
 9102 FORMAT( 1H0,'    *  AXIAL SECTION = ', I2, '  *  ' / 14X, '  FUEL C
     *ENTER     COLUMNAR GR.   EQUIAXED GR.   FUEL OUTER      CLAD INNER
     *   CLAD OUTER    CAPSULE INNER  CAPSULE OUTER' )
 9103 FORMAT( 1H ,' RADIUS(CM)', 8F15.5 )
 9104 FORMAT( 1H ,' TEMP.(C)', 8F15.3  )
 9105 FORMAT( 1H ,' AV.TEMP(C)', 5X, 3F15.3, 15X, F15.3, 15X, F15.3 )
 9106 FORMAT( 1H ,' ALPHA-T(-)', 7X, 3F15.5, 15X, F15.5, 15X, F15.5 )
 9107 FORMAT( 1H ,' DENS(G/CC)', 7X, 3F15.5, 15X, F15.5  )
 9108 FORMAT( 1H ,' MASS(G/CM)', 7X, 3F15.5  )
 9050 FORMAT( 1H0,/ , 35X, '=====  STRESS AND STRAIN OUTPUT NEXT  ====='
     *)
C
      END
```

```
      SUBROUTINE EDTINP ( RVB,RCB,REB,RUB,RCI,RCO,QSFF,QSFP,OTOM,CPU,
     *                    RHOP,F,XCPI,XCPO,FLZ,CLZ,FL,CL,PTOP,PO )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /CAPSUL/ RCPI,RCPO,AFNA
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      COMMON /IOFIL/ I05,I06,I07
C
      DIMENSION  RVB(NN) , RCB(NN) , REB(NN) , RUB(NN) , RCI(NN) ,
     *           RCO(NN) , QSFF(NN) , QSFP(NN) , OTOM(NN) , CPU(NN,NP) ,
     *           RHOP(NN,3) , F(NN,NP) , XCPI(NN) , XCPO(NN) , FLZ(NN) ,
     *           CLZ(NN)
C
C
C     OUTPUT OF INPUT VALUES
C
C
C
      WRITE(I06,9000)
C
      WRITE(I06,9001)  NN,NCR,NER,NUR,NCL,NP,NOK
      WRITE(I06,9022) NOS,NOC,NCRK,NPLOT,IPFUL,IPSTN,IPSTS,IPCRK,NBUF
C     RVB
      WRITE(I06,9002)  RVB
C     RCB
      WRITE(I06,9003)  RCB
C     REB
      WRITE(I06,9004)  REB
C     RUB
      WRITE(I06,9005)  RUB
C     RCI
      WRITE(I06,9006)  RCI
C     RCO
      WRITE(I06,9007)  RCO
C
      IF ( RCPO-RCPI ) 200,200,100
  100 CONTINUE
      WRITE(I06,9008) XCPI
      WRITE(I06,9009) XCPO
      GO TO 300
C
  200 CONTINUE
      WRITE(I06,9010)
C
  300 CONTINUE
      WRITE(I06,9025)  AFNA
      WRITE(I06,9011)  FL , FLZ
C
      WRITE(I06,9012)  CL , CLZ
C
```

```
      WRITE(I06,9013)
C
      DO 1000 I=1,NN
      WRITE(I06,9014) I , ( RHOP(I,J),J=1,3 ) , RHOC
 1000 CONTINUE
C
      WRITE(I06,9015)  OTOM
      WRITE(I06,9016)
      DO 2000 I=1,NNF
      WRITE(I06,9017) I , ( CPU(I,J),J=1,NP )
 2000 CONTINUE
C
      WRITE(I06,9018)  QSFF
      WRITE(I06,9019)  QSFP
      WRITE(I06,9020)
C
      DO 3000 I=1,NNF
      WRITE(I06,9021) I , ( F(I,J),J=1,NP )
 3000 CONTINUE
C
C
      WRITE(I06,9024)  GASI
C
      RETURN
C
C
 9000 FORMAT (1H1 , // 25X,'=====  INPUT VALUES OUTPUT NEXT  =====',// )
 9001 FORMAT (1H0,'  ---- CONTROL NUMBER  AND  OPTION  ' , //
     *         10X,'NUMBER OF AXIAL SECTION  NN =' , I3 , /
     *         10X,'NUMBER OF MESH FOR COLUMNAR GRAIN REGION  NCR =',
     *         I3 , /
     *         10X,'NUMBER OF MESH FOR EQUIAXED GRAIN REGION  NER =',
     *         I3 , /
     *         10X,'NUMBER OF MESH FOR UNDISTURBED REGION  NUR =',
     *         I3 , /
     *         10X,'NUMBER OF MESH FOR CLAD REGION  NCL =',I3 , /
     *         10X,'NUMBER OF MESH FOR PLUTONIUM MIGRATION  NP =',I3 , /
     *         10X,'FUEL THERMAL CONDUCTIVITY OPTION  NOK =',I3,
     *         5X,'(1/2/3 = SAYLES/BAILY-ASAMOTO/BIANCHERIA)'    )
 9002 FORMAT (1H0,'  ---- CENTRAL VOID RADIUS ( CM )   1-NN ' , //
     *        (10X,9F12.4) )
 9003 FORMAT (1H0,'  ---- COLUMNAR GRAIN REGION OUTER BOUNDARY ( CM )
     *1-NN ' , // (10X,9F12.4) )
 9004 FORMAT (1H0,'  ---- EQUIAXED GRAIN REGION OUTER BOUNDARY ( CM )
     *1-NN ' , // (10X,9F12.4) )
 9005 FORMAT (1H0,'  ---- FUEL OUTER RADIUS ( CM )   1-NN ' , //
     *        (10X,9F12.4) )
 9006 FORMAT (1H0,'  ---- CLAD INNER RADIUS ( CM )   1-NN ' , //
     *        (10X,9F12.4) )
 9007 FORMAT (1H0,'  ---- CLAD OUTER RADIUS ( CM )   1-NN ' , //
     *        (10X,9F12.4) )
 9008 FORMAT (1H0,'  ---- CAPSULE INNER RADIUS ( CM )   1-NN ' , //
     *        (10X,9F12.4) )
 9009 FORMAT (1H0,'  ---- CAPSULE OUTER RADIUS ( CM )   1-NN ' , //
     *        (10X,9F12.4) )
 9010 FORMAT (1H0,'  **** THIS FUEL IS UN-ENCAPSULATED **** ' )
 9011 FORMAT (1H0,'  ---- FUEL AXIAL LENGTH ( CM )   1-NN      ... TOTAL
     * AXIAL LENGTH =' , F12.4 , ' ( CM ) ...' , //  (10X,9F12.4) )
```

```
 9012 FORMAT (1HO,'  ---- CLAD AXIAL LENGTH ( CM )   1-NN        ... TOTAL
     * AXIAL LENGTH =' , F12.4 , ' ( CM ) ...' , //  (10X,9F12.4) )
 9013 FORMAT (1HO,'  ---- DENSITY OF FUEL ( 3 REGIONS ) AND CLAD ( G/CC
     *) ' / )
 9014 FORMAT (1H ,10X,'AXIAL SECTION =',I3,5X,4F12.4 )
 9015 FORMAT (1HO,'  ---- O/M OF FUEL    PU-O(X)    1-NN ' , //
     *        (10X,9F12.4) )
 9016 FORMAT (1HO,'  ---- PLUTONIUM CONCENTRATION ( WEIGHT-FRACTION )
     *1-NP ' /  )
 9017 FORMAT (1H ,10X,'AXIAL SECTION =',I3,5X,7F12.4, / (34X,7F12.4) )
 9018 FORMAT (1HO,'  ---- AXIAL FAST FLUX ( E .GE. 0.1MEV ) RATE    1-NN
     *' , // (10X,9F12.4) )
 9019 FORMAT (1HO,'  ---- AXIAL HEAT GENERATION RATE    1-NN ' , //
     *        (10X,9F12.4) )
 9020 FORMAT (1HO,'  ---- RADIAL HEAT GENERATION RATE    1-NP ' , / )
 9021 FORMAT (1H ,10X,'AXIAL SECTION =',I3,5X,7F12.4, / (34X,7F12.4) )
 9022 FORMAT (10X,'CLAD SWELLING OPTION  NOS =',I3,
     *         5X,'(-1/-2 = ANNEALED 304,316S.S./20P.C.COLD WORK 316S.S.)
     *         ' , /
     *         10X,'CLAD CREEP OPTION  NOC =',I3,
     *         5X,'(-1/-2 = BOLTAX,BIANCHERIA,GUHA/MODIFIED)' , /
     *         10X,'CRACK OPTION  NCRK =',I3 , 5X , '(0/1 = NO CRACK CALC
     *./CRACK CALC.)' , /
     *         10X,'PLOT OPTION  NPLOT =',I3,5X,'(0/1 = NO PLOT/PLOT)',/
     *         20X,'AXIAL SECTION NUMBER FOR FUEL =',I3, /
     *         20X,'AXIAL SECTION NUMBER FOR STRAIN =',I3 , /
     *         20X,'AXIAL SECTION NUMBER FOR STRESS =',I3 , /
     * 20X,'AXIAL SECTION NUMBER FOR CRACK =',I3, /
     * 20X,'STORAGE(WORDS) REQUIRED FOR PLOTTING STEP =',I4, / )
 9024 FORMAT( 1HO,'  ---- INITIAL GAS IN PLENUM =',F15.7,'  ( MOLES )' )
 9025 FORMAT( 1HO,'  ---- COOLANT FLOW AREA =' , F15.8 , '  ( CM**2 )' )
C
      END
```

```
      SUBROUTINE EDTMTX ( EPR,EPC,EPZ,SIGR,SIGC,SIGZ,TPR,TPC,TPZ,TSR,
     *                    TSC,TSZ,RDIS,CRIF,RCP,TCP,TBCP,ATBCP,RCL,TCL,
     *                    TBCL,ATBCL,YNG,POS,SHR,BM,RF,TF,TBF,ATBF,PRS,
     *                    EHP,TPRF,TPRC,TPCF,TPCC,TPZF,TPZC,
     *                                      PRES,KFRIC,DPR,DPC,DPZ,
     *                    DSR,DSC,DSZ,POSB,SHRB,SWG,SWS,N,RPM,TPM,GS,
     *                    CPU,LM,LMR,LMC,LMZ,RCOO )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /XFRIC/ FST,FDY,GAP
      COMMON /CAPSUL/ RCPI,RCPO,AFNA
      COMMON /COMPLT/ YP,DDOD,TITLE(20)
      COMMON /IOFIL/ I05,I06,I07
      DIMENSION EPR(NN,NTR) , EPC(NN,NTR) , EPZ(NN,NTR) , SIGR(NN,NTR) ,
     *          SIGC(NN,NTR) , SIGZ(NN,NTR) , TPR(NN,NTR) , TPC(NN,NTR),
     *          TPZ(NN,NTR) , TSR(NN,NTR) , TSC(NN,NTR) , TSZ(NN,NTR) ,
     *          RDIS(NN,NTR2) , CRIF(NN) , RCP(NN,NCL1) , TCP(NN,NCL1) ,
     *          TBCP(NN,NCL) , ATBCP(NN,NCL) , RCL(NN,NCL1) ,
     *          TCL(NN,NCL1) , TBCL(NN,NCL) , ATBCL(NN,NCL) ,
     *          YNG(NN,NTR) , POS(NN,NTR) , SHR(NN,NTR) , BM(NN,NTR) ,
     *          RF(NN,NFR1) , TF(NN,NFR1) , TBF(NN,NFR) , ATBF(NN,NFR) ,
     *          PRS(NN,NTR) , EHP(NN,NTR) , PRES(NTR1) , KFRIC(NN) ,
     *          DPR(NTR) , DPC(NTR) , DPZ(NTR) , DSR(NTR) , DSC(NTR) ,
     *          DSZ(NTR) , POSB(NN,2) , SHRB(NN,2) , SWG(NN,NFR) ,
     *          SWS(NN,NTR) , RPM(NN,NP1) , TPM(NN,NP1) , GS(NN,NFR)
      DIMENSION CPU(NN,NP) , LM(NN,NTR) , LMR(NN,NTR) , LMC(NN,NTR) ,
     *          LMZ(NN,NTR) , RCOO(NN)
C
C
C     DETAILED AND MATRIX RESULTANT OUTPUT
C
C
      WRITE(I06,9000) N
      IF (RCPO-RCPI) 200,200,100
  100 CONTINUE
C     CAPSULE OUTPUT
      WRITE(I06,9110)
      WRITE(I06,9210) ( RCP(N,I),I=1,NCL1 )
      WRITE(I06,9220) ( TCP(N,I),I=1,NCL1 )
      WRITE(I06,9310) ( ATBCP(N,I),I=1,NCL )
  200 CONTINUE
C     CLAD OUTPUT
      WRITE(I06,9120)
      WRITE(I06,9210) ( RCL(N,I),I=1,NCL1 )
      WRITE(I06,9240) ( RDIS(N,I+1),I=NFR1,NTR1 )
      WRITE(I06,9220) ( TCL(N,I),I=1,NCL1 )
      J1 = NFR
      IF(N.EQ.NN) J1=0
      WRITE(I06,9230) ( PRES(J1+I),I=1,NCL1 )
      WRITE(I06,9310) (ATBCL(N,I),I=1,NCL)
      JCOUNT = 0
  400 IF(JCOUNT.GE.1) GO TO 500
      JJJ = 2
```

```
      KK = NFR1
      KKK = NTR
      KKKK = NCL
      GO TO 600
  500 KK = 1
      KKK = NFR
      KKKK = NFR
      J1 = 0
      JJJ = 1
  600 CONTINUE
      WRITE(I06,9320) (POS(N,I),I=KK,KKK)
      WRITE(I06,9330) (YNG(N,I),I=KK,KKK)
      IF ( KK.GT.NFR ) GO TO 650
      WRITE(I06,9600)
      WRITE(I06,9601) ( LMR(N,I),I=KK,KKK )
      WRITE(I06,9602) ( LMC(N,I),I=KK,KKK )
      WRITE(I06,9603) ( LMZ(N,I),I=KK,KKK )
C
  650 CONTINUE
      WRITE(I06,9340)
      WRITE(I06,9391) (SIGR(N,I),I=KK,KKK)
      WRITE(I06,9392) (SIGC(N,I),I=KK,KKK)
      WRITE(I06,9393) (SIGZ(N,I),I=KK,KKK)
      WRITE(I06,9350)
      WRITE(I06,9391) (EPR(N,I),I=KK,KKK)
      WRITE(I06,9392) (EPC(N,I),I=KK,KKK)
      WRITE(I06,9393) (EPZ(N,I),I=KK,KKK)
      WRITE(I06,9360)
      WRITE(I06,9391) (TPR(N,I),I=KK,KKK)
      WRITE(I06,9392) (TPC(N,I),I=KK,KKK)
      WRITE(I06,9393) (TPZ(N,I),I=KK,KKK)
      WRITE(I06,9370)
      WRITE(I06,9391) (DPR(J1+I),I=1,KKKK)
      WRITE(I06,9392) (DPC(J1+I),I=1,KKKK)
      WRITE(I06,9393) (DPZ(J1+I),I=1,KKKK)
      WRITE(I06,9380)
      WRITE(I06,9381) (TSR(N,I),I=KK,KKK)
      IF(JCOUNT.EQ.0) GO TO 700
      WRITE(I06,9382) (SWS(N,I),I=1,NFR)
      WRITE(I06,9383) (SWG(N,I),I=1,NFR)
      WRITE(I06,9384) (EHP(N,I),I=1,NFR)
C
  700 CONTINUE
C     WRITE(I06,9385) (DSR(J1+I),I=1,KKKK)
      DDOD = ( RCL(N,NCL1)-RCOO(N) ) / RCOO(N) * 100.
      WRITE(I06,9250) DDOD
      WRITE(I06,9410) POSB(N,JJJ)
      YNGB = SHRB(N,JJJ)*2.*(1.+POSB(N,JJJ))
      WRITE(I06,9420) YNGB
      IF(N.EQ.NN) GO TO 900
      IF(JCOUNT.GE.1) GO TO 800
      WRITE(I06,9130)
      WRITE(I06,9210) (RF(N,I),I=1,NFR1)
      WRITE(I06,9240) (RDIS(N,I),I=1,NFR1)
      WRITE(I06,9220) (TF(N,I),I=1,NFR1)
      WRITE(I06,9230) ( PRES(I),I=1,NFR1 )
      WRITE(I06,9310) (ATBF(N,I),I=1,NFR)
      JCOUNT = 1
```

```
      GO TO 400
  800 WRITE(I06,9430) GAP
      IF ( KFRIC(N)-1 ) 801,802,803
  801 WRITE(I06,9700) CRIF(N)
      GO TO 804
  802 WRITE(I06,9701) CRIF(N)
      GO TO 804
  803 WRITE(I06,9702) CRIF(N)
  804 CONTINUE
      WRITE(I06,9140)
      WRITE(I06,9510) (RPM(N,I),I=1,NP1)
      WRITE(I06,9520) (TPM(N,I),I=1,NP1)
      WRITE(I06,9530) (CPU(N,I),I=1,NP)
C
      RETURN
  900 CONTINUE
C
C     TO GET THE CPU TIME ACCUMULATED  ( SEC )
C
      CALL  TTIMER(TIMER)
C
      WRITE(I06,12321) TIMER
      RETURN
C
12321 FORMAT(1H0,'CPU TIME =',F7.2,' (SEC)' )
 9000 FORMAT(22H0***  AXIAL SECTION = , I2, 5H  *** )
 9110 FORMAT(29H0 +++  CAPSULE (I=1,NCL)  +++ )
 9120 FORMAT(29H0 +++  CLAD (I=1,NCL)  +++    )
 9130 FORMAT(29H0 +++  FUEL (I=1,NFR)  +++    )
 9140 FORMAT(40H0 +++  PLUTONIUM MIGRATION (I=1,NP)  +++)
 9210 FORMAT(21H     RADIUS(CM)         , 7F15.5, (/, 1H ,20X, 7F15.5 ) )
 9220 FORMAT(19H     TEMP.(C)           , 7F15.3, (/, 1H ,18X, 7F15.3 ) )
 9230 FORMAT(21H     PRESS.(KG/CM**2)   , 7E15.5, (/, 1H ,20X, 7E15.5 ) )
 9240 FORMAT(21H     DISPLACEMENT(CM)   , 7F15.5, (/, 1H ,20X, 7F15.5 ) )
 9250 FORMAT(/4X,'PERCENT CHANGE IN ORIGINAL O.D. =',F15.5,' (O/O)' / )
 9310 FORMAT(27H     ALPHA-T(-)               , 7F15.5,(/,1H ,26X,7F15.5) )
 9320 FORMAT(27H     POISSON RATIO(-)         , 7F15.5,(/,1H ,26X,7F15.5) )
 9330 FORMAT(27H     YOUNG M.(KG/CM**2)       , 7E15.5,(/,1H ,26X,7E15.5) )
 9340 FORMAT(23H     STRESS(KG/CM**2)     )
 9350 FORMAT(23H     TOTAL STRAIN(-)      )
 9360 FORMAT(23H     CREEP STRAIN(-)      )
 9370 FORMAT(23H     CREEP IN DELT(-)     )
 9380 FORMAT(23H     SWELLING STRAIN(-)   )
 9381 FORMAT(27H       TOTAL                  , 7E15.5,(/,1H ,26X,7E15.5) )
 9382 FORMAT(27H         SOLID COMPONENT      , 7E15.5,(/,1H ,26X,7E15.5) )
 9383 FORMAT(27H         GAS COMPONENT        , 7E15.5,(/,1H ,26X,7E15.5) )
 9384 FORMAT(27H         HOT PRESSING         , 7E15.5,(/,1H ,26X,7E15.5) )
 9385 FORMAT(27H       INCLEMENT IN DELT      , 7E15.5,(/,1H ,26X,7E15.5) )
 9391 FORMAT(27H       RADIAL                 , 7E15.5,(/,1H ,26X,7E15.5) )
 9392 FORMAT(27H       CIRCUMFERENTIAL        , 7E15.5,(/,1H ,26X,7E15.5) )
 9393 FORMAT(27H       AXIAL                  , 7E15.5,(/,1H ,26X,7E15.5) )
 9410 FORMAT(37H     AVERAGE POISSON RATIO(-)          = , F15.5 )
 9420 FORMAT(37H0    AVERAGE YOUNG MODULUS(KG/CM**2) = , E15.5 )
 9430 FORMAT(37H0    GAP SIZE(CM)                      = , F15.5 )
 9440 FORMAT(37H     FRICTION FORCE(KG/CM)             = , E15.5 )
 9510 FORMAT(20H     RADIUS(CM)      , 11F10.5, (/, 1H , 19X, 11F10.5) )
 9520 FORMAT(18H     TEMP.(C)        , 11F10.3, (/, 1H , 17X, 11F10.3) )
 9530 FORMAT(15H     PU CONC.(-) ,10X, 10F10.5, (/, 1H , 24X, 10F10.5) )
```

```
9600 FORMAT( 1H ,'    NUMBER OF CRACK ' )                                    00048000
9601 FORMAT(27H        RADIAL                ,I10,6I15,(/,1H ,21X,          00048100
    * 7I15) )                                                                00048200
9602 FORMAT(27H        CIRCUMFERENTIAL       ,I10,6I15,(/,1H ,21X,          00048300
    * 7I15) )                                                                00048400
9603 FORMAT(27H        AXIAL                 ,I10,6I15,(/,1H ,21X,          00048500
    * 7I15) )                                                                00048600
9700 FORMAT(37H      FRICTION FORCE(KG/CM)          = , E15.5 ,              00048700
    * '      ( OPEN )' )                                                     00048800
9701 FORMAT(37H      FRICTION FORCE(KG/CM)          = , E15.5 ,              00048900
    * '      ( SLIP )' )                                                     00049000
9702 FORMAT(37H      FRICTION FORCE(KG/CM)          = , E15.5 ,              00049100
    * '      ( STICK )' )                                                    00049200
     END                                                                     00049300
```

```
      SUBROUTINE EQUISI ( SR,SC,SZ,ES,PRS,N )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      DIMENSION  SR(NN,NTR) , SC(NN,NTR) , SZ(NN,NTR) , ES(NN,NTR)
     *           , PRS(NN,NTR)
C
C     EQUIVALENT STRESS CALC.
C
      DO 1000 I=1,NTR
      ES(N,I) = SQRT( ((SR(N,I)-SC(N,I))**2 + (SR(N,I)-SZ(N,I))**2
     *                + (SC(N,I)-SZ(N,I))**2) / 2.0 )
      PRS(N,I) = - ( SR(N,I)+SC(N,I)+SZ(N,I) ) / 3.0
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FLCOND ( XK,XK1,ZH,FP,FP95,T )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /CONST/ FKA1,FKB1,FKC1,FKA2,FKB2,FKC2,FKA3,FKB3,FKC3,
     *               FA11,FA12,FA21,FA22,FA31,FA32,
     *               CK1,CK2,CA1,CA2,XEWTON(7)
C
C
C     FUEL THERMAL CONDUCTIVITY CALC.
C
C
      IF( NOK-2 ) 100,200,300
C
  100 Z = T+273.
      XK = (FKB1/Z + FKA1 + FKC1*Z**3) * FP
      XK1 = (-FKB1/Z**2 + 3.*FKC1*Z**2) * FP
      RETURN
C
  200 XK = FKA2 + 1./((FKB2-FKC2*ZH)*T)
      XK1 = -1./((FKB2-FKC2*ZH)*T*T)
      RETURN
C
  300 Z = T+273.
      A3 = FKA3*FP95/FP
      B3 = FKB3*FP95/FP
      C3 = FKC3*FP/FP95
C
C     BIANCHERIA OPTION
C
      XK = 1./(A3+B3*Z) + C3*Z**3
      XK1 = -B3/(A3+B3*Z)**2 + 3.*C3*Z**2
      RETURN
C     XK IN W/CM/DEG C
C     XK1 IN W/CM/DEG C /DEG C
C
      END
```

```
      SUBROUTINE FLTMP ( POW,TCL,RCI,RUB,REB,RCB,RVB,RH,DXM,XM,R,T,       00053500
     *                    TB,ATB,TBA,ATBA,FD,RPM,TPM,TBPM,ATBPM,FDPM,GDT,00053600
     *                    HG,F,OTOM,RHTHEO,DELT,RFO,IT,BU,RHOP,RB,         00053700
     *                    EPR,EPC,EPZ,TPR,TPC,TPZ,TSR,TSC,TSZ,GS,SWG,SWS,00053800
     *                      EHP,WOLD,WNEW,XEPR,XEPC,XTPR,XTPC,XTPZ,XTSR, 00053900
     *                      XTSC,XTSZ,XGS,XSWG,XSWS,XEHP,XSWS3,SWS3  )   00054000
C                                                                         00054100
      REAL*8 R1,R2,R3,R4,R1R,R2R,R3R,RHD1,RHD2,RHD3,XMD1,XMD2,XMD3       00054200
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,      00054300
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,         00054400
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,              00054500
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,     00054600
     *                N12C,N13C,N14C,N15C,                               00054700
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15      00054800
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),  00054900
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,     00055000
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB           00055100
      COMMON /CONST/ FKA1,FKB1,FKC1,FKA2,FKB2,FKC2,FKA3,FKB3,FKC3,       00055200
     *               FA11,FA12,FA21,FA22,FA31,FA32,                      00055300
     *               CK1,CK2,CA1,CA2,XEWTON(7)                           00055400
      COMMON /IOFIL/ I05,I06,I07                                         00055500
      COMMON /FBRATR/ JFA                                                00055600
C                                                                         00055700
      REAL*8  XAR                                                        00055800
      DIMENSION  POW(NN) , TCL(NN,NCL1) , RCI(NN) , RUB(NN) ,            00055900
     *           REB(NN) , RCB(NN) , RVB(NN) , RH(NN,NTR) , DXM(NN,3) ,  00056000
     *           XM(NN,3) , R(NN,NFR1) , T(NN,NFR1) , TB(NN,NFR) ,       00056100
     *           ATB(NN,NFR) , TBA(NN,3) , ATBA(NN,3) , FD(NN,NFR) ,     00056200
     *           RPM(NN,NP1) , TPM(NN,NP1) , TBPM(NN,NP) , FDPM(NN,NP) ,00056300
     *           GDT(NN,NP1) , HG(NN) , ATBPM(NN,NP) , F(NN,NP) ,        00056400
     *           OTOM(NN) , RHTHEO(NN) , RFO(NN,NFR1) , BU(NN),          00056500
     *           RHOP(NN,3) , RB(NN,NTR)                                 00056600
      DIMENSION EPR(NN,NTR),EPC(NN,NTR),EPZ(NN,NTR),TPR(NN,NTR),         00056700
     *          TPC(NN,NTR),TPZ(NN,NTR),TSR(NN,NTR),TSC(NN,NTR),         00056800
     *          TSZ(NN,NTR),GS(NN,NFR),SWG(NN,NFR),SWS(NN,NTR),          00056900
     *          EHP(NN,NTR),SWS3(NN,NUR)                                 00057000
      DIMENSION  WOLD(NN,NFR),WNEW(NFR),XEPR(NFR),XEPC(NFR),XTPR(NFR),   00057100
     *           XTPC(NFR),XTPZ(NFR),XTSR(NFR),XTSC(NFR),XTSZ(NFR),      00057200
     *           XGS(NFR),XSWG(NFR),XSWS(NFR),XEHP(NFR),XSWS3(NUR)       00057300
C                                                                         00057400
C                                                                         00057500
C     FUEL TEMPERATURE DISTRIBUTION , RESTRUCTURING CALC.                00057600
C                                                                         00057700
C                                                                         00057800
      BT = 1.0                                                           00057900
      IF(JFA.GE.2) BT = 0.5                                              00058000
C                                                                         00058100
      DO 1000 N=1,NNF                                                    00058200
C                                                                         00058300
      P = POW(N)*8.6011E+02                                              00058400
C     P NOW IN CAL/CM/HR                                                 00058500
      TO = TCL(N,1) + P/(HG(N)*2.*PI*RCI(N)) / 1.8                       00058600
C     TO NOW IN DEG C                                                    00058700
      W = POW(N)                                                         00058800
C     W NOW IN WATTS/CM                                                  00058900
C     RUB(N) NOW IN CM                                                   00059000
C                                                                         00059100
      RHOT = RHTHEO( N )                                                 00059200
```

```
C
  700 RO = RUB(N)
C
C       RO NOW IN CM
C
        R(N,NFR1) = RO
        T(N,NFR1) = TO
        RPM(N,NP1) = RO
        TPM(N,NP1) = TO
C
C       SUBDIVIDE EACH ZONE
C
        DR = (RCB(N)-RVB(N)) / NCR
C
        R(N,1) = RVB(N)
        DO 3100 I=2,NCR
 3100   R(N,I) = R(N,I-1) + DR
C
        DR = (REB(N)-RCB(N)) / NER
C
        K = NCR+2
        KK = NCR+NER
        R(N,NCR+1) = RCB(N)
        DO 3200 I=K,KK
 3200   R(N,I) = R(N,I-1) + DR
C
        DR = (RUB(N)-REB(N)) / NUR
C
        K = NCR+NER+2
        KK = NFR
        R(N,NCR+NER+1) = REB(N)
        DO 3300 I=K,KK
 3300   R(N,I) = R(N,I-1) + DR
C
        IF(IT.NE.1) GO TO 3330
C
        DO 3333 I=1,NFR1
 3333   RFO(N,I) = R(N,I)
 3330   CONTINUE
C
        DO 3400 I=1,NFR
        RB(N,I) = 2.*(R(N,I+1)*R(N,I+1)+R(N,I+1)*R(N,I)+R(N,I)*R(N,I)) /
     *            (3.*(R(N,I+1)+R(N,I)))
 3400   CONTINUE
C
        XML(1) = 0.
        XML(2) = XM(N,1)
        XML(3) = XM(N,1)+XM(N,2)
        SM = XM(N,1)+XM(N,2)+XM(N,3)
        SF = 0.
C
        DO 1100 J=1,NP
 1100   SF = SF+F(N,J)
C
        NPNP = NP * 2
        DMO = SM/NP
C       DMO = SM/NPNP
        XM1 = SM
```

```
      RR = RO
      TT = TO
      WW = W
C
      I = 3
C
      NM = NP
      DDM = DMO
C
  300 DM1 = XM(N,I)
      SAT = 0.
C
      GO TO ( 310,320,330 ) , I
  310 J = 1
      P = 1. - (RHOP(N,I) + (0.99*RHOT - RHOP(N,I)) *BU(N)/BURH1)/RHOT
      GO TO 340
  320 J = NCR+1
      P = 1. - (RHOP(N,I) +(0.965*RHOT - RHOP(N,I)) *BU(N)/BURH2)/RHOT
      GO TO 340
  330 J = NCR+NER+1
      P = 1. - RHOP(N,I)/RHOT
  340 CONTINUE
C
      FP = (1.-P)/(1.+BT*P)
      FP95 = (1.-0.05)/(1.+BT*0.05)
C
  400 CONTINUE
      IF(DDM-DM1) 200,100,100
  100 IF(I-2) 110,120,130
  110 RI = RVB(N)
      GO TO 140
  120 RI = RCB(N)
      GO TO 140
  130 RI = REB(N)
  140 DM = DM1
      ICHECK = 1
      GO TO 500
C
  200 DM = DDM
      XAR = RR**2 - DM/(PI*RH(N,J))
      IF(XAR.LT.0.) XAR = 0.
      RI = DSQRT(XAR)
      ICHECK = 0
C
  500 DR = RR - RI
      W1 = W*NP*F(N,NM)/(SF*SM)
      W3 = W1*RH(N,J)
      DW = W1*DM
      WW = WW - DW
C
C     FUEL THERMAL CONDUCTIVITY CALCULATION
C
C
C     OTOM MODIFY
C
      CALL OTMCAL ( OTOM,N )
C
      CALL COEF ( OTOM,N )
```

```
C
C     THERMAL CONDUCTIVITY ITERATION LOOP
C
      JTEMP = 0
      TTT = TT
  540 CONTINUE
C
      TCR = 1.+ALFO(I)*(TTT-20.)+ALF1(I)*(TTT*TTT-400.)/2.
C     TCR = 1.+ALFO(I)*(TT-20.)+ALF1(I)*(TT*TT-400.)/2.
      ZH = RH(N,J)/(RHOT*TCR)
C
      CALL FLCOND(XK,XK1,ZH,FP,FP95,TTT)
C     CALL FLCOND ( XK,XK1,ZH,FP,FP95,TT )
C
      IF(WW.LE.0.) WW=0.0
C     IF(WW.LE.0.) GO TO 550
      IF(RI.LE.0.001) GO TO 550
      DT = W3*(RR**2-RI**2)/(4.*XK)-(W3*RI**2/(2.*XK)-WW/(PI2*XK))*
     *     ALOG(RR/RI)
      GO TO 560
  550 DT = W3*RR**2/(4.*XK)
  560 CONTINUE
      IF(JTEMP.EQ.2) GO TO 570
      TTT = TT+DT/2.
      JTEMP = JTEMP + 1
      GO TO 540
  570 CONTINUE
C
C     END OF THERMAL CONDUCTIVITY LOOP
C
      TBPM(N,NM) = TT + DT/2.
      ATBPM(N,NM) = ATBFUN ( TT,DT,I )
      SAT = SAT + ATBPM(N,NM)*DM
      FDPM(N,NM) = F(N,NM)*NP*RH(N,J) / (SM*SF)
C     FDPM NOW IN FISSIONS*CM/FISSIONS/CM**3
C
      XM1 = XM1 - DM
      RR = RI
      TT = TT + DT
C
      IF(ICHECK.EQ.0) GO TO 600
      ATBA(N,I) = SAT/XM(N,I)
      IF(RR.LE.0.) GO TO 825
      I = I - 1
      IF(I.GE.1) GO TO 831
      GO TO 829
  825 DO 826 JJ=1,I
      ATBA(N,JJ) = ATBA(N,I)
  826 CONTINUE
  829 CONTINUE
      DO 830 JJ=1,NM
      TPM(N,JJ) = TT
      RPM(N,JJ) = RR
      TBPM(N,JJ) = TBPM(N,NM)
      ATBPM(N,JJ) = ATBPM(N,NM)
      FDPM(N,JJ) = FDPM(N,NM)
  830 GDT(N,JJ) = -DT/DR
      GO TO 860
```

```
C
  831 GRDT = DT/DR
      Y = 0.1*(TT-TO)/RO
      IF( GRDT-Y ) 835,840,840
  835 GRDT = Y
  840 TK = TT+273.
      IF(GRDT.LE.0.) GRDT=1.E-06
C
      GO TO ( 841,842 ) , I
  841 CONTINUE
      DRCB = CM1(I)*(TK**(-1.5)) * EXP(-DHR(I)/TK) * GRDT
      CM(I) = PI2*RR*RH(N,J)*DRCB
      GO TO 843
  842 CONTINUE
      DREB = CM1(I)*EXP(-DHR(I)/TK)
      CM(I) = PI2*RR*RH(N,J)*DREB
  843 CONTINUE
      DDM = DDM-DM
      IF(DDM.GT.0.) GO TO 850
      TPM(N,NM) = TT
      RPM(N,NM) = RR
      GDT(N,NM) = -DT/DR
      NM = NM-1
      IF(NM.GE.1) GO TO 300
      GO TO 7700
  850 IF(I.GE.1) GO TO 300
      GO TO 860
  600 TPM(N,NM) = TT
      RPM(N,NM) = RR
      GDT(N,NM) = -DT/DR
      IF(RR.LE.0.) GO TO 650
      NM = NM - 1
      DM1 = DM1 - DM
      DDM = DM0
      IF(NM.GE.1) GO TO 400
      GO TO 7700
  650 DO 660 JJ=1,NM
      TPM(N,JJ) = TT
      RPM(N,JJ) = RR
      GDT(N,JJ) = -DT/DR
  660 CONTINUE
 7700 GRDT = DT/DR
      Y = 0.1*(TT-TO)/RO
      IF(GRDT-Y) 7835,7840,7840
 7835 GRDT = Y
 7840 TK = TT+273.
      IF(GRDT.LE.0.) GRDT=1.E-06
      IF(I-2) 790,720,710
  710 DREB = CM1(2)*EXP(-DHR(2)/TK)
      CM(2) = PI2*RR*RH(N,NCR+1)*DREB
      ATBA(N,I)=SAT/(XM(N,I)-DM1)
      DM1=0.
      SAT=ATBPM(N,I-1)
      I=I-1
  720 DRCB = CM1(1)*(TK**(-1.5))*EXP(-DHR(1)/TK)*GRDT
      CM(1) = PI2*RR*RH(N,1)*DRCB
      ATBA(N,I)=SAT/(XM(N,I)-DM1)
      DM1=0.
```

```
      SAT=ATBPM(N,I-1)
      I=I-1
  790 CONTINUE
      ATBA(N,I)=SAT/(XM(N,I)-DM1)
C
C
  860 CONTINUE
      T(N,1) = TT
      FD(N,1) = FDPM(N,1)
C
C
C     INTERPOLATION
C
      IJ = 2
      DO 1400 K=2,NFR
 1401 IF( R(N,K)-RPM(N,IJ) ) 1403,1403,1402
 1402 IJ = IJ + 1
      IF(IJ.GE.NPI) GO TO 1404
      GO TO 1401
 1403 RAT = ( RPM(N,IJ)-R(N,K) ) / ( RPM(N,IJ)-RPM(N,IJ-1) )
      T(N,K) = TPM(N,IJ) + ( TPM(N,IJ-1)-TPM(N,IJ) ) * RAT
      FD(N,K) = FDPM(N,IJ) + ( FDPM(N,IJ-1)-FDPM(N,IJ) ) * RAT
      GO TO 1400
 1404 FD(N,K) = FDPM(N,NP)
      T(N,K) = TPM(N,NP1) + ( TPM(N,NP)-TPM(N,NP1) ) * ( RPM(N,NP1)-
     *           R(N,K) ) / ( RPM(N,NP1)-RPM(N,NP) )
      IJ = NP
 1400 CONTINUE
C
      DO 1405 K=1,NFR
      TB(N,K) = ( T(N,K)+T(N,K+1) ) / 2.0
 1405 CONTINUE
C
      DO 1500 IJ=1,3
      ST = 0.
      IF ( IJ-2 ) 1501,1502,1503
 1501 IJK1 = 1
      IJK2 = NCR
      GO TO 1504
 1502 IJK1 = NCR + 1
      IJK2 = NCR + NER
      GO TO 1504
 1503 IJK1 = NCR + NER + 1
      IJK2 = NFR
 1504 CONTINUE
C
      DO 1550 IJK=IJK1,IJK2
      DDT = ( T(N,IJK)-T(N,IJK+1) )
      ATB(N,IJK) = ATBFUN( T(N,IJK+1),DDT,IJ )
 1550 ST = ST + TB(N,IJK)
      TBA(N,IJ) = ST / (IJK2-IJK1+1)
 1500 CONTINUE
C
C
      DXM(N,3) = -CM(2)
      DXM(N,2) = CM(2)-CM(1)
      DXM(N,1) = CM(1)
C
```

```
C
C
C     REGION MASS CALC.  AND  MODIFICATION
C
C
C
 4100 CONTINUE
C
      DXM(N,1) = DXM(N,1) / ( PI2*RH(N,1)*RVB(N) )
      DXM(N,2) = DRCB
      DXM(N,3) = DREB
C
      DO 6100 I=1,NFR
 6100 WNEW(I) = PI*RH(N,I)*(R(N,I+1)**2-R(N,I)**2)*(1.+EPZ(N,I))
      IF (IT.LE.1) GO TO 6900
      J = NFR
      I = NFR
      XEPR(J) = 0.
      XEPC(J) = 0.
      XTPR(J) = 0.
      XTPC(J) = 0.
      XTPZ(J) = 0.
      XTSR(J) = 0.
      XTSC(J) = 0.
      XTSZ(J) = 0.
      XGS(J)  = 0.
      XSWG(J) = 0.
      XSWS(J) = 0.
      XEHP(J) = 0.
 6200 CONTINUE
      W = WNEW(J)
 6250 IF(WOLD(N,I)-W) 6400,6300,6300
 6300 DW = W/WNEW(J)
      XEPR(J) = XEPR(J) + DW*EPR(N,I)
      XEPC(J) = XEPC(J) + DW*EPC(N,I)
      XTPR(J) = XTPR(J) + DW*TPR(N,I)
      XTPC(J) = XTPC(J) + DW*TPC(N,I)
      XTPZ(J) = XTPZ(J) + DW*TPZ(N,I)
      XTSR(J) = XTSR(J) + DW*TSR(N,I)
      XTSC(J) = XTSC(J) + DW*TSC(N,I)
      XTSZ(J) = XTSZ(J) + DW*TSZ(N,I)
      XGS(J) = XGS(J) + DW *GS(N,I)
      XSWG(J) = XSWG(J) + DW*SWG(N,I)
      XSWS(J) = XSWS(J) + DW*SWS(N,I)
      XEHP(J) = XEHP(J) + DW*EHP(N,I)
      WOLD(N,I) = WOLD(N,I)-W
      J = J-1
      IF(J.LE.0) GO TO 6850
      XEPR(J) = 0.
      XEPC(J) = 0.
      XTPR(J) = 0.
      XTPC(J) = 0.
      XTPZ(J) = 0.
      XTSR(J) = 0.
      XTSC(J) = 0.
      XTSZ(J) = 0.
      XGS(J)  = 0.
      XSWG(J) = 0.
```

```
      XSWS(J) = 0.
      XEHP(J) = 0.
      GO TO 6200
 6400 DW = WOLD(N,I)/WNEW(J)
      XEPR(J) = XEPR(J) + DW*EPR(N,I)
      XEPC(J) = XEPC(J) + DW*EPC(N,I)
      XTPR(J) = XTPR(J) + DW*TPR(N,I)
      XTPC(J) = XTPC(J) + DW*TPC(N,I)
      XTPZ(J) = XTPZ(J) + DW*TPZ(N,I)
      XTSR(J) = XTSR(J) + DW*TSR(N,I)
      XTSC(J) = XTSC(J) + DW*TSC(N,I)
      XTSZ(J) = XTSZ(J) + DW*TSZ(N,I)
      XGS(J) = XGS(J) + DW *GS(N,I)
      XSWG(J) = XSWG(J) + DW*SWG(N,I)
      XSWS(J) = XSWS(J) + DW*SWS(N,I)
      XEHP(J) = XEHP(J) + DW*EHP(N,I)
      W = W - WOLD(N,I)
      I = I - 1
      IF(I.GE.1) GO TO 6250
      DW = W/WNEW(J)
      XEPR(J) = XEPR(J)+DW*EPR(N,1)
      XEPC(J) = XEPC(J)+DW*EPC(N,1)
      XTPR(J) = XTPR(J)+DW*TPR(N,1)
      XTPC(J) = XTPC(J)+DW*TPC(N,1)
      XTPZ(J) = XTPZ(J)+DW*TPZ(N,1)
      XTSR(J) = XTSR(J)+DW*TSR(N,1)
      XTSC(J) = XTSC(J)+DW*TSC(N,1)
      XTSZ(J) = XTSZ(J)+DW*TSZ(N,1)
      XGS(J) = XGS(J)+DW*GS(N,1)
      XSWG(J) = XSWG(J)+DW*SWG(N,1)
      XSWS(J) = XSWS(J)+DW*SWS(N,1)
      XEHP(J) = XEHP(J)+DW*EHP(N,1)
      J = J - 1
      IF(J.LE.0).GO TO 6850
      DO 6450 I = 1,J
      XEPR(I) =  EPR(N,I)
      XEPC(I) = EPC(N,I)
      XTPR(I) = TPR(N,I)
      XTPC(I) = TPC(N,I)
      XTPZ(I) = TPZ(N,I)
      XTSR(I) = TSR(N,I)
      XTSC(I) = TSC(N,I)
      XTSZ(I) = TSZ(N,I)
      XGS(I) = GS(N,I)
      XSWG(I) = SWG(N,I)
      XSWS(I) = SWS(N,I)
      XEHP(I) = EHP(N,I)
 6450 CONTINUE
 6850 DO 6550 I=1,NFR
      EPR(N,I) = XEPR(I)
      EPC(N,I) = XEPC(I)
      TPR(N,I) = XTPR(I)
      TPC(N,I) = XTPC(I)
      TPZ(N,I) = XTPZ(I)
C     TSR(N,I) = XTSR(I)
C     TSC(N,I) = XTSC(I)
C     TSZ(N,I) = XTSZ(I)
C     GS(N,I) = XGS(I)
```

```
C      SWG(N,I) = XSWG(I)
C      SWS(N,I) = XSWS(I)
C      EHP(N,I) = XEHP(I)
 6550 CONTINUE
 6900 DO 6950 I=1,NFR
 6950 WOLD(N,I) = WNEW(I)
 1000 CONTINUE
C
 4200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FLXCAL ( POWAV,QSFF,ZFLX,FLNC,FLUX,DELT )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
C
      DIMENSION  QSFF(NN) , ZFLX(NN) , FLNC(NN) , FLUX(NN)
C
C     FLUX , FLUENCE CALC.
C
      DO 1000 N=1,NN
      FLUX(N) = CFLX*QSFF(N)*POWAV*SMQ0/SMQ
      FLNC(N) = FLNC(N) + FLUX(N)*DELT*3600.
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FRICHK ( RUB,RCI,DISP,PRES,FRIC,CRIF,KFRIC,KONV,IFRIC,
     *                    NS )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XFRIC/ FST,FDY,GAP
C
      DIMENSION  RUB(NN) , RCI(NN) , DISP(NTR2) , PRES(NTR1) ,
     *           CRIF(NN) , KFRIC(NN)
C
C
C
C     FRICTION FORCE CHECK
C
C
      IF ( IFRIC-1 ) 100,200,300
C
  100 CONTINUE
C
      GAP = (RCI(NS)+DISP( NFR+2 ))-(RUB(NS)+DISP(NFR+1))
C
      IF ( GAP ) 102,102,101
  101 CONTINUE
C
C     GAP OPEN
C
      IFRIC = 0
      GO TO 300
C
  102 CONTINUE
      GAP = 0.
      IFRIC = 2
C     IFRIC = 1
      KONV = 0
      RETURN
C
  200 CONTINUE
C
      GAP = 0.
      GRIC = PI2*(RUB(NS)+DISP(NFR+1))*PRES(NFR+1)*FST
      IF(GRIC.LE.0.) GO TO 202
      IF ( FRIC-GRIC ) 201,201,202
C
  201 CONTINUE
C
C     STICK
C
      IFRIC = 1
      GO TO 300
  202 CONTINUE
C
```

```
C       SLIP
C
        IFRIC = 2
        KONV = 0
        RETURN
  300 CONTINUE
        KONV = 1
        KFRIC(NS) = IFRIC
        CRIF(NS) = FRIC
C
        RETURN
        END
```

```
      SUBROUTINE GAPCON ( RCI,RUB,HG )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      DIMENSION  RCI(NN) , RUB(NN) , HG(NN)
C
C     FUEL-CLADDING GAP CONDUCTANCE CALC.
C
      DO 1000 N=1,NN
      RO = RCI(N)
      RI = RUB(N)
      X = RO-RI
      IF(X.LT.0.) X=0.0
C     HG(N) = 1873.*EXP(-135.2*X) / 144.
C     HG NOW IN BTU/HR/IN/IN/DEGF
C
C
      HG( N ) = 508.05 * EXP( -53.23*X )
C
C     HG NOW IN CAL/HR/CM/CM/DEG F
C
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GASOUT ( RF,FLZ,POW,FD,TBF,VOL,EPR,EPC,EPZ,ATBF,SWS,
     *                    GR,GS,GRT,DELT )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XGASO/ AF,AN,CT,QR
C
      DIMENSION  RF(NN,NFR1) , FLZ(NN) , POW(NN) , FD(NN,NFR) ,
     *           TBF(NN,NFR) , VOL(NN,NTR) , EPR(NN,NTR) , EPC(NN,NTR) ,
     *           EPZ(NN,NTR) , ATBF(NN,NFR) , SWS(NN,NTR) , GR(NN) ,
     *           GS(NN,NFR)
C
C     GAS RELEASE RATE CALC.
C
C
C     AF IS ATOMS OF GAS PER FISSION
C     AN IS AVOGADRO'S NUMBER
C
C     FRPT IS FRACTION OF GAS RELEASED PER HOUR
C
C     GR(N) = GAS RELEASED FROM AXIAL SECTION N  IN DELT
C     GS(N,I) = GAS REMAINING IN AXIAL SECTION N , RADIAL REGION I
C     GRT = TOTAL GAS RELEASED
C
      DO 1000 N=1,NNF
      DFN = A91*POW(N)*DELT
      GR(N) = 0.
C
      DO 1100 I=1,NFR
      T = TBF(N,I)*1.8+32.0 + 460.0
      FRPT = EXP( -CT-QR/T )
      DGS = AF*VOL(N,I)*FD(N,I)*DFN/AN
      GO = GS(N,I)
      EG = EPR(N,I)+EPC(N,I)+EPZ(N,I)-3.*ATBF(N,I)-SWS(N,I)
      EHR = 0.
      IF(EG-0.25) 200,200,100
  100 EHR = (EG-0.25)*100.
  200 F = FRPT + EHR*0.0005
      X = F*DELT
      IF(X-0.1) 300,400,400
  300 Y = (1.-X/3.+X*X/12.)*X/2.
      Z = X-X*Y
      REL = GO*Z + DGS*Y
      GS(N,I) = GO+DGS-REL
      GO TO 1100
  400 IF(X-170.) 600,500,500
  500 GS(N,I) = DGS/X
      GO TO 700
  600 GS(N,I) = DGS/X + (GO-DGS/X)*EXP(-X)
  700 REL = GO + DGS - GS(N,I)
 1100 GR(N) = GR(N) + REL
```

```
C
      GRT = GRT + GR(N)
C
 1000 CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE GOLDN ( POWAV,QSF,TOUT,TIN,RCLO,POW,TCLO,RCPI,
     *                   XCPO,AFNA,FL,HC,TNA,TCPO,TCPI,TOUTER,NSKIP,PO,
     *                   FLZ )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XGOLD/ ASS0,ASS1,ANA0,ANA1,ANA2,C0,C1,C2
      COMMON /FBRATR/ JFA
      COMMON /IOFIL/ I05,I06,I07
      DIMENSION QSF(NN) , TCLO(NN) , POW(NN) , RCLO(NN) , HC(NN) ,
     *          TNA(NN) , TCPO(NN) , TCPI(NN) , TOUTER(NN) , FLZ(NN)
C
C
C     CLADDING OUTER SURFACE TEMPERATURE CALC.
C
C
C========================================================================
C      CALCULATES CLAD OD TEMPERATURE FOR ENCAPSULATED (RCPI.LT.XCPO),
C     AND UNENCAPSULATED (RCPI.GE.XCPO) FUEL ELEMENTS.
C     TEMPERATURE DROPS IN NA ARE DETERMINED BY NEWTONIAN EXTRAPOLATION
C     (2-3 AND 8-9). ***
C      CAPSULE IS  ASSUMED TO BE STAINLESS STEEL
C========================================================================
C
      AF=AFNA
      WB=POWAV*8.6011E+02
C
C     WB IS AVERAGE POWER IN CAL/HR/CM
C
      TINF = 1.8*TIN + 32.
      TOUTF = 1.8*TOUT + 32.
      B0=C0+(C1+C2*TINF)*TINF
      B1=C1+2.*C2*TINF
      B2=C2
      DTMP=TOUTF-TINF
      CPDT=(B0+(B1/2.+B2*DTMP/3.)*DTMP)*DTMP
      WDOT=WB*FL/CPDT
      JNX=0
      IF (XCPO-RCPI) 110,110,120
  110 JNX=1
  120 RCPO=XCPO
      IF( NSKIP ) 121,121,122
  121 CONTINUE
      TNA(NN)=TOUT
      TCPO(NN)=TOUT
      TCPI(NN)=TOUT
      TCLO(NN)=TOUT
      GO TO 123
  122 CONTINUE
      TNA(NN) = TOUTER(NN)
      TCPO(NN) = TOUTER(NN)
      TCPI(NN) = TOUTER(NN)
```

```
      TCLO(NN) = TOUTER(NN)
  123 CONTINUE
      POW(NN)=0.
      XNF=NNF
      S=0.
      SPW = 0.
C=======================================================================
      DO 130 N=1,NNF
      SPW = SPW + QSF(N)
130   S=S+QSF(N)*FLZ(N)
      IF( NSKIP.GT.0 ) GO TO 135
      X=0.
      SW=0.
C=======================================================================
C
  135 CONTINUE
      DO 1000 N=1,NN
C
      IF ( NSKIP.GT.0 ) TNA(N) = N*( TOUT-TIN ) / NN + TIN
      IF( NSKIP.GT.0 ) GO TO 155
      SW=SW+QSF(N)*FLZ(N)
      C=CPDT*(SW-QSF(N)*FLZ(N)/2.)/S
140   CP=B0+(B1+B2*X)*X
      X1=(C+(B1/2.+B2*X/1.5)*X*X)/CP
      EPS=ABS(X1-X)/X1
      X=X1
      IF (EPS-.00001) 150,140,140
150   T=TINF+X
      TNA(N)=( T-32. ) / 1.8
      IF(JFA.GE.2)  TNA(N) = N *(TOUT-TIN)/NN+TIN
C==   THIS IS NA TEMPERATURE AT AXIAL NODE N  =========================
  155 CONTINUE
      W=WB*QSF(N)*XNF/SPW
      POW(N)=W / 8.6011E+02
C
C     POW(N) NOW IN WATT/CM
C
      IF (JNX) 170,170,160
160   RCPO=RCLO(N)
170   P=PI2*RCPO
      IF( NSKIP.GT.0 ) GO TO 195
      GO TO (2100,2200,2300,2200),JFA
 2100 DE = 4.*AF/P
      CNST=4.*WDOT/P
      CNST1=W/RCPO/PI2
      TB=T
      Y=0.
180   XK=ANA0+(ANA1+ANA2*TB)*TB
      CP=C0+(C1+C2*TB)*TB
      PE=CNST*CP/XK
C==   THIS IS PECLET NO.  =============================================
      XNU=5.0+0.025*(PE**0.8)
C==   THIS IS NUSSELT NO.  ============================================
      HC( N ) = XNU*XK/DE
C     HC NOW IN CAL/HR/CM**2/DEG F
      Y1=CNST1/HC(N)
      IF(Y1.LE.0.) GO TO 190
      TB=T+Y1/2.
```

```
      EPS=ABS(Y1-Y)/Y1
      Y=Y1
      IF (EPS-.0001) 190,180,180
  190 T5=T+Y
      GO TO 196
 2200 HFLUX = POW(N)/P
      IF(HFLUX.LE.0.) GO TO 2210
      DTFILM = 7.93*(HFLUX**0.25)*EXP(-PO/63.2)
C     DTFILM = HFLUX/1.1443
      T5 = (TNA(N)+DTFILM)*1.8+32.0
      HC(N) = HFLUX/DTFILM*1762.3*252.0/929.03
C     HC NOW IN CAL/HR/CM**2/DEG F
      GO TO 196
 2210 HC(N) = 0.0
      T5 = TNA(N)*1.8+32.0
      GO TO 196
C
C $$$$$   F I L M   C O E F I C I E N T   O F   P W R  ( JFA = 3 )   $$$
C     * * * ( DITTUS - BOETLER EQ. ) * * *
 2300 CONTINUE
C *** INTEGRATED HEAT CAPACITY IS IN KCAL/KG
      DTMP = TOUT - TIN
      CPDT = 2.623*DTMP - 0.01473*DTMP**2/2. + 3.528E-05*DTMP**3/3.
C *** FLP IS IN M
      FLP = FL/100.0
C *** WBP IS IN KCAL/HR/M
      WBP = WB/10.
C *** AFNAP IS IN M**2
      AFNAP = AFNA/10000.0
C *** GAMMAP IS IN KG/M**3
      GAMMAP = 860.8 + 1.042*TNA(N) - 0.00514*TNA(N)**2
C *** WP IS IN M/HR
      WP = FLP*WBP/(CPDT*AFNAP*GAMMAP)
C *** DEP IS IN M
      DEP = 4.*AFNA/(PI2*RCPO*100.)
C *** ENUP IS IN M**2/HR
      ENUP =(0.4015E-06 - 1.791E-09*TNA(N) + 2.946E-12*TNA(N)**2)*3600.
C *** REYNOL'S NUMBER
      REP = WP*DEP/ENUP
C *** PRANDTLE'S NUMBER
      PRP = 5.491 - 0.0381*TNA(N) + 0.777E-04*TNA(N)**2
C *** THERMAL CONDUCTIVITY IS IN KCAL/M/HR/C
      RAMDAP = 0.4847 + 1.596E-03*TNA(N) - 5.497E-06*TNA(N)**2
C *** FILM COEFICIENT IS IN KCAL/HR/M**2/C
      HCP = 0.023*RAMDAP/DEP*REP**0.8*PRP**0.4
      HC(N) = HCP/18.
C *** HC IS IN CAL/HR/CM**2/F
      HCP = HCP*1.1622E-04
C *** HCP IS IN W/CM**2/C
      T5 = (TNA(N)+POW(N)/(PI2*RCPO*HCP))*1.8 + 32.0
      GO TO 196
C
  195 CONTINUE
      T5 = TOUTER(N)*1.8 + 32.0
      HC( N ) = 0.0
  196 CONTINUE
      TCPO(N)=( T5-32. ) / 1.8
C==   THIS IS CAPSULE OD TEMPERATURE   ==================================
```

```
      IF (JNX) 210,210,200
200   TCLO(N)=TCPO(N)
      GO TO 1000
210   W1=2.*W/PI2
      XL=ALOG(RCPO/RCPI)
      XKO=(ASS0+ASS1*T5)
      T4=T5+W1*XL/(XKO+SQRT(XKO*XKO+ASS1*W1*XL))
      TCPI(N)=( T4-32. ) / 1.8
C==   THIS IS CAPSULE ID TEMPERATURE  ==================================
      IF (RCPI-RCLO(N)) 220,220,230
220   WRITE (I06,270) N
C
      STOP
C
230   C=W*ALOG(RCPI/RCLO(N))/PI2
      D0=ANA0+(ANA1+ANA2*T4)*T4
      D1=ANA1+2.*ANA2*T4
      D2=ANA2
      Y=0.
240   XK=D0+(D1+D2*Y)*Y
      Y1=(C+(D1/2.+D2*Y/1.5)*Y*Y)/XK
      IF(Y1.LE.0.) GO TO 250
      EPS=ABS(Y1-Y)/Y1
      Y=Y1
      IF (EPS-.0001) 250,240,240
250   TCLO(N)=( T4+Y-32. ) / 1.8
C==   THIS IS CLAD OD TEMPERATURE  =====================================
C
 1000 CONTINUE
C
C
  270 FORMAT( 1H1,'----- SUBROUTINE GOLDN -----', / ,
     * 1H0,' FUEL-CLAD HAS HIT CAPSULE WALL IN SECTION',I2,'.'/,
     *2X,'EXECUTION CONTINUATION IS NONSENSE')
      RETURN
      END
```

```
      SUBROUTINE HEADER
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK
      COMMON /COMPLT/ YP,DDOD,TITL(20)
      COMMON /IOFIL/ I05,I06,I07
      DIMENSION TITLE(20)
      DATA CEND/'*END'/
C
C     TITLE CARD READ TILL FOUND *END CARD
C
      ISW = 0
  100 READ(I05,9000,END=300) TITLE
      IF ( TITLE(1) . EQ . CEND ) GO TO 200
      WRITE(I06,9001) TITLE
      IF ( ISW.NE.0 ) GO TO 100
      DO 101 I=1,20
      TITL(I) = TITLE(I)
  101 CONTINUE
      ISW = 1
      GO TO 100
C
  300 WRITE(I06,9002)
      STOP
C
  200 CONTINUE
C
      RETURN
C
C     FORMAT
C
 9000 FORMAT(20A4)
 9001 FORMAT(1H0,20X,20A4)
 9002 FORMAT(1H1,'----- SUBROUTINE HEADER -----', / ,
     * 1H0,'*END CARD MISSING ( *END CARD MEANS TITLE CARD END )' )
C
      END
```

```
      SUBROUTINE HKNS ( TBCL,TSR,TSC,TSZ,FLX,DELT )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C
      COMMON /XHKNS/ CS,B0,C0,C1,C2,C3
      COMMON /FBRATR/ JFA
      DIMENSION  TBCL(NN,NCL) , TSR(NN,NTR) , TSC(NN,NTR) ,
     *           TSZ(NN,NTR) , FLX(NN)
C
C
C     CLADDING SWELLING STRAIN CALC.
C
C
      DO 1000 N=1,NN
      FLUX = FLX(N)
C
      DO 2000 I=1,NCL
      TX = TBCL(N,I)
      SW = TSR(N,NFR+I)
C
      IF(JFA-2) 10,20,20
   10 CONTINUE
      IF(NOS+2) 200,200,100
  100 TK = TX + 273.
C     TX IN DEG C   ,   TK NOW IN DEG K
      FX = FLUX/1.2E+22
      SITA = TK-623.+1.E-50
      IF(SITA.LT.CS) SITA = CS
      XN = 2.05-27./SITA+78./SITA**2
      SDH = (FX*3600.)**XN*((TK-40.)*1.E-10)*EXP(-0.015*TK-5100./TK+
     *        32.6) / 300.
      GO TO 300
C
  200 XN = 1.50
      TC = TX
C     TX IN DEG C   ,   TC NOW IN DEG C
      X = ALOG(B0/300.) + XN*ALOG(FLUX*3600.)
      SDH = EXP( X ) * (C0+TC*(C1+TC*(C2+TC*C3)))
      IF(SDH.LT.0.) SDH = 0.
  300 CONTINUE
C
      TSR(N,NFR+I) = (SW**(1./XN) + SDH**(1./XN)*DELT)**XN
      GO TO 30
C
   20 TSR(N,NFR+I) = SW
   30 CONTINUE
      TSC(N,NFR+I) = TSR(N,NFR+I)
      TSZ(N,NFR+I) = TSR(N,NFR+I)
 2000 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INPUT  ( EPZ,TPR,TPC,TPZ,SIGR,SIGC,SIGZ,ADISP,RDISP,
     *                    RVB,RCB,REB,RUB,RCI,RCO,QSFF,QSFP,F,EPR,EPC,
     *                    TSR,TSC,TSZ,CPU,RHOP,OTOM,FL,CL,PTOP,PO,EPSIN,
     *                    CLZO )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *            N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /CAPSUL/ RCPI,RCPO,AFNA
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      COMMON /IOFIL/ I05,I06,I07
      DIMENSION TPR(NN,NTR),TPC(NN,NTR),TPZ(NN,NTR),SIGR(NN,NTR),
     *          SIGC(NN,NTR),SIGZ(NN,NTR),ADISP(NN,NTR),RDISP(NN,NTR2),
     *          RVB(NN),RCB(NN),REB(NN),RUB(NN),RCI(NN),RCO(NN),
     *          QSFF(NN),QSFP(NN),F(NN,NP),EPZ(NN,NTR),EPR(NN,NTR),
     *          EPC(NN,NTR) , TSR(NN,NTR) , TSC(NN,NTR) , TSZ(NN,NTR) ,
     *          CPU(NN,NP) , RHOP(NN,3) , OTOM(NN) , CLZO(NN)
C
C
C     READ CARD 4  TO CARD 23
C
C
C
C---- CARD 4   ...    SKIP AND PLOT OPTION
C
      READ(I05,9000) IDLE,IPFUL,IPSTN,IPSTS,IPCRK
C
C
C     IDLE = INITIAL VALUES SET OPTION
C     0/1 = CARD INPUT/ZERO SET BY CODE
C     IPFUL = AXIAL SECTION NUMBER FOR FUEL PLOT
C     IPSTN = AXIAL SECTION NUMBER FOR STRAIN PLOT
C     IPSTS = AXIAL SECTION NUMBER FOR STRESS PLOT
C     IPCRK = AXIAL SECTION NUMBER FOR CRACK PLOT
C
      IF(IDLE) 100,100,200
  100 CONTINUE
      READ(I05,9001)  EPR
      READ(I05,9001)  EPC
      READ(I05,9001)  EPZ
      READ(I05,9001)  TPR
      READ(I05,9001)  TPC
      READ(I05,9001)  TPZ
      READ(I05,9001)  TSR
      READ(I05,9001)  TSC
      READ(I05,9001)  TSZ
      READ(I05,9001)  SIGR
      READ(I05,9001)  SIGC
      READ(I05,9001)  SIGZ
      READ(I05,9001)  ADISP
      READ(I05,9001)  RDISP
      GO TO 300
```

```
C
  200 CONTINUE
C
      DO 1000 N=1,NN
      DO 1050 M=1,NTR
      EPR(N,M) = 0.0
      EPC(N,M) = 0.0
      EPZ(N,M) = 0.0
      TPR(N,M) = 0.0
      TPC(N,M) = 0.0
      TPZ(N,M) = 0.0
      TSR(N,M) = 0.0
      TSC(N,M) = 0.0
      TSZ(N,M) = 0.0
      SIGR(N,M) = 0.0
      SIGC(N,M) = 0.0
      SIGZ(N,M) = 0.0
 1050 ADISP(N,M) = 0.0
      DO 1000 M=1,NTR2
      RDISP(N,M) = 0.0
 1000 CONTINUE
C
  300 CONTINUE
C
C---- CARD 5    ...   CENTRAL VOID RADIUS ( CM )  1 - NN
      READ(I05,9001) RVB
C---- CARD 6    ...   COLUMNAR GRAIN REGION BOUNDARY ( CM ) 1 - NN
      READ(I05,9001) RCB
C---- CARD 7    ...   EQUIAXED GRAIN REGION BOUNDARY ( CM ) 1 - NN
      READ(I05,9001) REB
C---- CARD 8    ...   FUEL OUTER RADIUS ( CM ) 1 - NN
      READ(I05,9001) RUB
C---- CARD 9    ...   CLAD INNER RADIUS ( CM ) 1 - NN
      READ(I05,9001) RCI
C---- CARD 10   ...    CLAD OUTER RADIUS ( CM ) 1 - NN
      READ(I05,9001) RCO
C---- CARD 11   ...    CLAD AXIAL LENGTH ( CM ) 1 - NN
      READ(I05,9001) CLZO
C---- CARD 12   ...    AXIAL FLUX RATE ( - ) 1 - NN
      READ(I05,9001) QSFF
C---- CARD 13   ...   AXIAL POWER RATE ( - ) 1 - NN
      READ(I05,9001) QSFP
C---- CARD 14   ...   O / M ( INPUT X  OF  PU-O(X) ) 1 - NNF
      READ(I05,9001) OTOM
C
C---- CARD 15   ...   PULUTONIUM CONCENTRATION ( WEIGHT RATE ) 1 - NNF
      READ(I05,9001) ( CPU(N,1),N=1,NNF )
C
C---- CARD 16   ...   DENSITY OF UNDISTURBED REGION ( G/CC ) 1 - NNF
      READ(I05,9001) ( RHOP(N,3),N=1,NNF )
C
C---- CARD 17   ...   RADIAL HEAT GENERATION RATE ( - ) 1 - NP , 1 - NNF
      READ(I05,9001)  ( F(1,I),I=1,NP )
      IF(NNF.LE.1) GO TO 2001
      DO 2000 N=2,NNF
      DO 1999 I=1,NP
      F(N,I) = F(1,I)
 1999 CONTINUE
```

```
 2000 CONTINUE
 2001 CONTINUE
C
C
C---- CARD 18  ...    CAPSULE INNER RADIUS,OUTER RADIUS,COOLANT FLOW AREA
      READ(I05,9001) RCPI,RCPO,AFNA
C---- CARD 19  ...    SURFACE TENSION OF VOID IN FUEL 3 REGIONS
C                    FROM 1 TO 3 ( KG/CM**2 )
      READ(I05,9001) (PG(I),I=1,3)
C---- CARD 20  ...    AXIAL FORCE ON CLADDING ( KG ) ,
C                    COOLANT PRESSURE ( KG/CM**2 )
      READ(I05,9001) PTOP,PO
C---- CARD 21  ...    CLAD DENSITY ( G/CC ) , DENSITY MODIFIER BY BURN-UP
      READ(I05,9001) RHOC,BURH1,BURH2
C---- CARD 22  ...    CONSTANTS FOR BOUNDARY VELOCITY
      READ(I05,9001) (CM1(L),L=1,2),(DHR(L),L=1,2)
C---- CARD 23  ...    INITIAL GAS MOLE IN PLENUM ( MOLE )
      READ(I05,9001) GASI,TCON,PDBU
C
      RETURN
C
C     FORMAT
C
 9001 FORMAT(5E15.8)
 9000 FORMAT(16I5)
C
      END
```

```
      SUBROUTINE IOSET(IO)
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C
C     LOGICAL UNIT SET FOR WORK FILE
C
      DIMENSION IO(NN)
      DO 100 I=1,NN
      IO(I) = I+10
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ITERAT ( PRS,VOL,TBF,ATBF,SWG ,EHP,RH,RF,GS,SWS,
     *                  BM,EPR,EPC,EPZ,SWS3,SIGR,SIGC,SIGZ,RHTHEO,
     *                  DELT,N,RHOP,BU,TSR,TSC,TSZ,XEHP,XSWG ,
     *                  PLPR,CVPR )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *              NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *              IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *              N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *              N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *              PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *              CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XITER/ C,QRK
      DIMENSION  PRS(NN,NTR) , VOL(NN,NTR) , TBF(NN,NFR) ,
     *          ATBF(NN,NFR) , SWG(NN,NFR)  , EHP(NN,NTR) , RH(NN,NFR),
     *          RF(NN,NFR1) , GS(NN,NFR) , SWS(NN,NTR) , BM(NN,NTR) ,
     *          EPR(NN,NTR) , EPC(NN,NTR) , EPZ(NN,NTR) ,
     *          SWS3(NN,NUR) , SIGR(NN,NTR) , SIGC(NN,NTR) ,
     *          SIGZ(NN,NTR) , RHTHEO(NN) , RHOP(NN,3) , BU(NN) ,
     *          TSR(NN,NTR) , TSC(NN,NTR) , TSZ(NN,NTR) , XEHP(NFR) ,
     *          XSWG(NFR)
C
C
C     FUEL SWELLING STRAIN , HOT-PRESSING CALC.
C
C
C     RK IS IDEAL GAS CONSTANT IN KG*CM/MOLE/DEG K
C     GS IS FISSION GAS IN MOLES CALCULATED IN GASOUT SUBROUTINE
C     BM IS BULK MODULUS
C
C
      K = 1
      J = 1
C
      RHOT = RHTHEO( N )
C
      DO 1100 I=1,NFR
      T = TBF(N,I) + 273.
C
C
      IF ( I.GT.NCR ) K=2
      IF ( I.GT.NCR+NER ) K=3
C
      IF(K-2) 100,100,110
C
C     SWELLING CAUSED FROM FISSION SOLID
C
  110 SWS(N,I) = SWS3(N,J) + 2.389*GS(N,I)/VOL(N,I)*2.54*2.54*2.54
      J = J+1
  100 CONTINUE
C
C
C     SWELLING CAUSED FROM HOT PRESSING
C
      GO TO (210,220,230) , K
  210 P = 1. - (RHOP(N,I) + (0.99*RHOT - RHOP(N,I)) *BU(N)/BURH1)/RHOT
      GO TO 240
```

```
  220 P = 1. - (RHOP(N,I) +(0.965*RHOT - RHOP(N,I)) *BU(N)/BURH2)/RHOT
      GO TO 240
  230 P = 1. - RHOP(N,I)/RHOT
  240 CONTINUE
      XEHP(I) = EHP(N,I)
      XSWG(I) = SWG(N,I)
C     TEMP      = -( C/T )*PRS(N,I)*( P          + XEHP(I)       )*DELT*
C     TEMP      = -( C/T )*PRS(N,I)*( P+ XSWG(I)+XEHP(I)         )*DELT*
C    *               EXP( -QRK/T )
      TEMP = - C/T * PRS(N,I) * P * DELT * EXP(-QRK/T)
      IF(TEMP.GT.0.) TEMP=0.
      EHP(N,I) = XEHP(I) + TEMP
      IF(EHP(N,I).LT.-P) EHP(N,I)=-P
C
C
C     SWELLING CAUSED FROM FISSION GAS
C
C
      P = PG(K)
      TEMP = PRS(N,I)
      IF(TEMP.LT.0.0) TEMP=0.0
      PPP = P + TEMP
      IF( PPP.LE.CVPR )  PPP = CVPR
      SWG(N,I) = GS(N,I)*RK*T / ( VOL(N,I)*(PPP) )
C
C     SWG(N,I) = GS(N,I)*RK*T / ( VOL(N,I)*(P+TEMP) )
C     SWG(N,I) = GS(N,I)*RK*T / ( VOL(N,I)*P )
C
      TSR(N,I) = ( SWS(N,I)+SWG(N,I)+EHP(N,I) ) / 3.
      TSC(N,I) = TSR(N,I)
      TSZ(N,I) = TSR(N,I)
C
 1100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE LENGTH ( FL,FLN,CL,CLN,FLZO,FLZ,CLZO,CLZ,EPZ )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
C
      DIMENSION  EPZ(NN,NTR) , FLZ(NN) , FLZO(NN) , CLZ(NN) ,
     *           CLZO(NN)
C
C     AXIAL LENGTH CALC.
C
      FL = 0.
      CL = 0.
C
      DO 1000 N=1,NNF
      FLZ(N) = FLZO(N)*(1.+EPZ(N,1))
      FL = FL+FLZ(N)
      CLZ(N) = CLZO(N)*(1.+EPZ(N,NFR+1))
      CL = CL+CLZ(N)
C     CLZ(N) = FLZ(N)
 1000 CONTINUE
C
      CL = CL + CLZO(NN)*(1.+EPZ(NN,NFR+1))
      CLZ(NN) = CLZO(NN)*(1.+EPZ(NN,NFR+1))
      FLN = FL
      CLN = CL
C
C
      RETURN
      END
```

```
      SUBROUTINE MATRX  ( IA,JA,A,E,P,DEQP,EQSIG,RF,RC,G,PB,GB,FLZ,
     *                    KFRIC,MOUNT,IFRIC,NS,RB )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XFRIC/ FST,FDY
      COMMON /IOFIL/ IO5,IO6,IO7
      DIMENSION   IA(IGEN1) , JA(NONZR) , A(NONZR) , E(NN,NTR) ,
     *            P(NN,NTR) , DEQP(NN,NTR) , EQSIG(NN,NTR) ,
     *            RF(NN,NFR1) , RC(NN,NCL1) , G(NN,NTR) , PB(NN,2) ,
     *            GB(NN,2) , FLZ(NN) , KFRIC(NN) , RB(NN,NTR)
C
C
C     ELEMENTS A OF AX=B , AND IA,JA FOR USE IN THE SUBSEQUENT
C     CALL TO THE SL-MATH SUBROUTINES
C
C
      IF ( MOUNT ) 20,20,10
C
   10 J = 56*NTR + 20
      GO TO 3000
C
   20 CONTINUE
C
      J = 0
      K=1
C     1 - N
      DO 1000 I=K,N1
      XE = E(NS,I)
      XP = P(NS,I)
      J = J + 1
      IA(I) = J
      JA(J ) = I
      A(J ) = 1.0
      J = J + 1
      JA(J    ) = I + N2
      A(J    ) = -1./XE
      J = J + 1
      JA(J    ) = I + N3
      A(J    ) = XP/XE
      J = J + 1
      JA(J    ) = I + N4
      A(J    ) = XP/XE
      J = J + 1
      JA( J ) = I + N5
      A( J ) = -1.0
 1000 CONTINUE
C
C     N+1 - 2N
C
      DO 1100 I=K,N1
      XE = E(NS,I)
```

```
      XP = P(NS,I)
      J = J + 1
      IA(I+N1) = J
      JA(J) = I + N1
      A(J) = 1.0
      J = J + 1
      JA(J) = I + N2
      A(J) = XP/XE
      J = J + 1
      JA(J) = I + N3
      A(J) = -1./XE
      J = J + 1
      JA(J) = I + N4
      A(J) = XP/XE
      J = J + 1
      JA(J) = I + N6
      A(J) = -1.0
 1100 CONTINUE
C
C     2N+1 - 3N
C
      DO 1200 I=K,N1
      XEP = DEQP(NS,I)
      XES = EQSIG(NS,I)
      J = J + 1
      IA(I+N2) = J
      JA(J) = I + N2
      A(J) = XEP/XES
      J = J + 1
      JA(J) = I + N3
      A(J)  = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N4
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N5
      A(J) = -1.0
 1200 CONTINUE
C
C     3N+1 - 4N
C
      DO 1300 I=K,N1
      XEP = DEQP(NS,I)
      XES = EQSIG(NS,I)
      J = J + 1
      IA(I+N3) = J
      JA(J) = I + N2
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N3
      A(J) = XEP/XES
      J = J + 1
      JA(J) = I + N4
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N6
      A(J) = -1.0
 1300 CONTINUE
```

```
C
C       4N+1 - 5N
C
        DO 1400 I=K,NI
        XEP = DEQP(NS,I)
        XES = EQSIG(NS,I)
        J = J + 1
        IA(I+N4) = J
        JA(J) = I + N2
        A(J) = -XEP/(2.*XES)
        J = J + 1
        JA(J) = I + N3
        A(J) = -XEP/(2.*XES)
        J = J + 1
        JA(J) = I + N4
        A(J) = XEP/XES
        J = J + 1
        JA(J) = I + N7
        A(J) = -1.0
  1400  CONTINUE
C
C
C       5N+1 - 5N+3
C
        DO 1800 I=1,3
        J = J + 1
        IA(I+N5) = J
        DO 1850 IJ=1,NFR
        RO = RF(NS,IJ+1)
        RI = RF(NS,IJ)
        JA(J) = N5+(I-1)*NTR + IJ
        A( J ) = (RO+RI)*(RO-RI)
        J = J + 1
  1850  CONTINUE
        JA(J) = N8  + 1 + 2*(I-1)
        RO = RF(NS,NFR1)
        RI = RF(NS,1)
        A( J ) = -(RO+RI)*(RO-RI)
  1800  CONTINUE
C
C       5N+4 - 5N+6
C
        DO 1900 I=1,3
        J = J + 1
        IA(I+N5+3) = J
        DO 1950 IJ=1,NCL
        RO = RC(NS,IJ+1)
        RI = RC(NS,IJ)
        JA(J) = N5 + (I-1)*NTR + NFR + IJ
        A( J ) = (RO+RI)*(RO-RI)
        J = J + 1
  1950  CONTINUE
        JA(J) = N8  + 2 + 2*(I-1)
        RO = RC(NS,NCL1)
        RI = RC(NS,1)
        A( J ) = -(RO+RI)*(RO-RI)
  1900  CONTINUE
C
```

```
C       5N+7 - 7N+6                                                        00017500
C                                                                          00017600
        K=0                                                                00017700
C       INNER DISPLACEMENT                                                 00017800
        DO 2000 I=1,N1                                                     00017900
        J = J + 1                                                          00018000
        IA(I+N5+6) = J                                                     00018100
        IF(I-NFR) 640,640,650                                              00018200
   640  RO = RF(NS,I+1)                                                    00018300
        RI = RF(NS,I)                                                      00018400
        GO TO 660                                                          00018500
   650  K = K+1                                                            00018600
        RO = RC(NS,K+1)                                                    00018700
        RI = RC(NS,K)                                                      00018800
   660  CONTINUE                                                           00018900
        XP = P(NS,I)                                                       00019000
        XG = G(NS,I)                                                       00019100
        RR =(RO+RI)*(RO-RI)                                                00019200
C       RA = RB(NS,I)                                                      00019300
        RA = RI                                                            00019400
        XY1 = (1.-2.*XP)*RA*ALOG(RA/RI) / (2.*(1.-XP))                     00019500
        XY2 = RO*RO*(RI*RI+(1.-2.*XP)*RA*RA)*ALOG(RO/RI) /                 00019600
     *        (2.*RR*(1.-XP)*RA)                                           00019700
        JA(J) = I                                                          00019800
        A(J) = RA-RI                                                       00019900
        J = J+1                                                            00020000
        JA(J) = I + N5                                                     00020100
        A(J) = -RA*(1.-2.*XP)/2. -XY1+XY2                                  00020200
        J = J + 1                                                          00020300
        JA(J) = I + N6                                                     00020400
        A(J) = -RA*(1.-2.*XP)/2. +XY1-XY2                                  00020500
        J = J + 1                                                          00020600
        A(J) = RA*XP                                                       00020700
        IF(I-NFR) 670,670,680                                              00020800
   670  JA(J) = N8 + 7                                                     00020900
        J = J + 1                                                          00021000
        JA(J) = N8 + 8 + I                                                 00021100
        A(J) = 1.                                                          00021200
        GO TO 690                                                          00021300
   680  JA(J) = N8 + 8                                                     00021400
        J = J + 1                                                          00021500
        JA(J) = N8 + 9 + I                                                 00021600
        A(J) = 1.                                                          00021700
   690  CONTINUE                                                           00021800
        J = J + 1                                                          00021900
        JA(J) = N9 + 10 + I                                                00022000
        A(J) = -RI*RI*(RO*RO/RA+(1.-2.*XP)*RA) / (2.*XG*RR)                00022100
        J = J + 1                                                          00022200
        JA(J) = N9 + 11 + I                                                00022300
        A(J) = RO*RO*(RI*RI/RA+(1.-2.*XP)*RA) / (2.*XG*RR)                 00022400
  2000  CONTINUE                                                           00022500
C                                                                          00022600
C       OUTER DISPLACEMENT                                                 00022700
C                                                                          00022800
        K=0                                                                00022900
        DO 2100 I=1,N1                                                     00023000
        J = J + 1                                                          00023100
        IA(I+N6+6) = J                                                     00023200
```

```
      IF(I-NFR) 700,700,710
  700 RO = RF(NS,I+1)
      RI = RF(NS,I)
      GO TO 720
  710 K = K + 1
      RO = RC(NS,K+1)
      RI = RC(NS,K)
  720 CONTINUE
      XP = P(NS,I)
      XG = G(NS,I)
      RR = (RO+RI)*(RO-RI)
C     RA = RB(NS,I)
      RA = RO
      XY1 = (1.-2.*XP)*RA*ALOG(RA/RI) / (2.*(1.-XP))
      XY2 = RO*RO*(RI*RI+(1.-2.*XP)*RA*RA)*ALOG(RO/RI) /
     *         (2.*RR*RA*(1.-XP))
      JA(J) = I
      A(J) = RA-RO
      J = J+1
      JA(J) = I + N5
      A(J) = -RA*(1.-2.*XP)/2. -XY1+XY2
      J = J + 1
      JA(J) = I + N6
      A(J) = -RA*(1.-2.*XP)/2. +XY1-XY2
      J = J + 1
      A(J) = RA*XP
      IF(I-NFR) 730,730,740
  730 JA(J) = N8 + 7
      J = J + 1
      JA(J) = N8 + 9 + I
      A(J) = 1.
      GO TO 750
  740 JA(J) = N8 + 8
      J = J + 1
      JA(J) = N8 + 10 + I
      A(J) = 1.
  750 CONTINUE
      J = J + 1
      JA(J) = N9 + 10 + I
      A(J) = -RI*RI*(RO*RO/RA+(1.-2.*XP)*RA) / (2.*XG*RR)
      J = J + 1
      JA(J) = N9 + 11 + I
      A(J) = RO*RO*(RI*RI/RA+(1.-2.*XP)*RA) / (2.*XG*RR)
 2100 CONTINUE
C
C     BOUNDARY CONDITION  FOR  PRESSURE
C     FUEL
      J = J + 1
      IA(N7+7) = J
      JA(J) = N9 + 11
      A(J) = 1.0
C
C
      J = J + 1
      IA(N7+8) = J
      JA(J) = N10 + 11
      A(J) = 1.
C
```

```
C       FUEL AXIAL DISPLACEMENT
C
        J = J + 1
        XGB = GB(NS,1)
        XPB = PB(NS,1)
        IA(N7+9) = J
        JA(J) = N8  + 1
        A(J) = XPB
        J = J + 1
        JA(J) = N8  + 3
        A(J) = XPB
        J = J + 1
        JA(J) = N8  + 5
        A(J) = -1.0
        J = J + 1
        JA(J) = N8 + 7
        A(J) = 1.+XPB
        J = J + 1
        RO = RF(NS,NFR1)
        RI = RF(NS,1)
        RR = (RO+RI)*(RO-RI)
        JA(J) = N9 + 11 + NFR
        A(J) = -XPB*RO*RO /(XGB*RR)
        J = J + 1
        DL = FLZ( NS )
        JA(J) = N10 + 12
C==== REMARK
        A(J) = DL/(2.*XGB*PI*RR)
C==== REMARK
C
C       CLAD AXIAL DISPLACEMENT
C
        J = J + 1
        IA(N7+10) = J
        RO = RC(NS,NCL1)
        RI = RC(NS,1)
        XPB = PB(NS,2)
        XGB = GB(NS,2)
        RR = (RO+RI)*(RO-RI)
        JA(J) = N8  + 2
        A(J) = XPB
        J = J + 1
        JA(J) = N8  + 4
        A(J) = XPB
        J = J + 1
        JA(J) = N8  + 6
        A(J) = -1.0
        J = J + 1
        JA(J) = N8 + 8
        A(J) = 1.+XPB
        J = J + 1
        JA(J) = N9 + 11 + NFR
        A(J) = XPB*RI*RI / (XGB*RR)
        J = J + 1
        JA(J) = N10 + 12
C==== REMARK
        A(J) = -DL /(2.*XGB*PI*RR)
C==== REMARK
```

```
C
C      STRESS      ===    SIGR   ===
C
       K = 0
       DO 2600 I=1,N1
       J = J + 1
       IA(N7+10+I) = J
       JA(J) = N2 + I
       A(J) = 1.0
       J = J + 1
       IF(I-NFR) 770,770,780
   770 RO = RF(NS,I+1)
       RI = RF(NS,I)
       GO TO 790
   780 K = K + 1
       RO = RC(NS,K+1)
       RI = RC(NS,K)
   790 CONTINUE
       XP = P(NS,I)
       XG = G(NS,I)
       RR = (RO+RI) * (RO-RI)
       TH = 1.-2.*RI*RI*ALOG(RO/RI)/RR
       OM = 1.-2.*RO*RO*ALOG(RO/RI)/RR
       JA(J) = N5 + I
       A(J) = XG*((1.-TH)*OM + TH ) / (2.*(1.-XP))
       J = J + 1
       JA(J) = N6 + I
       A(J) = XG*((TH-1.)*OM - TH ) / (2.*(1.-XP))
       J = J + 1
       JA(J) = N9 + 10 + I
       A(J) = -RI*RI*OM / RR
       J = J + 1
       JA(J) = N9 + 11 + I
       A(J) = RO*RO*TH / RR
  2600 CONTINUE
C
C      STRESS      ===    SIGC   ===
C
       K = 0
       DO 2700 I=1,N1
       J = J + 1
       IA(N8+10+I) = J
       JA(J) = N3 + I
       A(J) = 1.0
       J = J + 1
C
       IF(I-NFR) 830,830,840
   830 RO = RF(NS,I+1)
       RI = RF(NS,I)
       GO TO 850
   840 K = K + 1
       RO = RC(NS,K+1)
       RI = RC(NS,K)
   850 CONTINUE
       XP = P(NS,I)
       XG = G(NS,I)
       RR = (RO+RI) * (RO-RI)
C
```

```
      TH = 1.-2.*RI*RI*ALOG(RO/RI)/RR
      OM = 1.-2.*RO*RO*ALOG(RO/RI)/RR
C
      JA(J) = N5 + I
      A(J) = XG*((OM-1.)*TH-    OM ) / (2.*(1.-XP))
      J = J + 1
      JA(J) = N6 + I
      A(J) = XG*((1.-OM)*TH+    OM ) / (2.*(1.-XP))
      J = J + 1
      JA(J) = N9 + 10 + I
      A(J) = -RI*RI*(2.-OM)/RR
      J = J + 1
      JA(J) = N9 + 11 + I
      A(J) = RO*RO*(2.-TH) / RR
 2700 CONTINUE
C
C     STRESS      ===  SIGZ  ===
C
      K = 0
      DO 2800 I=1,N1
      J = J + 1
      IA(N9+10+I) = J
      JA(J) = N4 + I
      A(J) = 1.0
      J = J + 1
C
      IF(I-NFR) 890,890,900
  890 RO = RF(NS,I+1)
      RI = RF(NS,I)
      GO TO 910
  900 K = K + 1
      RO = RC(NS,K+1)
      RI = RC(NS,K)
  910 CONTINUE
      XP = P(NS,I)
      XG = G(NS,I)
      RR = (RO+RI)*(RO-RI)
C
      TH = 1.-2.*RI*RI*ALOG(RO/RI)/RR
      OM = 1.-2.*RO*RO*ALOG(RO/RI)/RR
C
      JA(J) = N5 + I
      A(J) =-2.*XG*XP
      J = J + 1
      JA(J) = N6 + I
      A(J) =-2.*XG*XP
      J = J + 1
      JA(J) = N7 + I
      A(J) = 2.*XG
      J = J + 1
      A(J) = -2.*XG*(1.+XP)
      IF(I-NFR) 920,920,930
  920 JA(J) = N8 + 7
      GO TO 940
  930 JA(J) = N8 + 8
  940 CONTINUE
      J = J + 1
      JA(J) = N9 + 10 + I
```

```
      A(J) = -2.*XP*RI*RI/RR
      J = J + 1
      JA(J) = N9 + 11 + I
      A(J) =  2.*XP*RO*RO/RR
 2800 CONTINUE
C
C
 3000 CONTINUE
C
      IF ( IFRIC-1 ) 950,960,970
C
C
  950 CONTINUE
C
C
C     GAP OPEN
C
C
      J = J + 1
      IA(N10+11) = J
      JA(J) = N9+11 + NFR
      A(J) = 1.
      J = J+ 1
      IA(N10+12) = J
      JA(J) = N10 + 10
      A(J) = 0.
      J = J + 1
      JA(J) = N10 + 11
      A(J) = 0.
      J = J + 1
      JA(J) = N10 + 12
      A(J) = 1.
C
      GO TO 990
C
C
  960 CONTINUE
C
C     STICK ASSUMPTION
C
      J = J + 1
      IA(N10+11) = J
      JA(J) = N8 + 7
      A(J) = 1.
      J = J + 1
      JA(J) = N8 + 8
      A(J) = -1.
C
      GO TO 980
C
C
  970 CONTINUE
C
C     SLIP ASSUMPTION
C
      J = J + 1
      IA(N10+11) = J
      JA(J) = N9 + 11 + NFR
```

```
      A(J) = PI2*RF(NS,NFR1)*FDY
      J = J + 1
      JA(J) = N10 + 12
      A(J) = -1.
C
  980 CONTINUE
C
      J = J + 1
      IA(N10+12) = J
      JA(J) = N8 + 9 + NFR
      A(J) = 1.
      J = J + 1
      JA(J) = N8 + 10 + NFR
      A(J) = -1.
  990 CONTINUE
      J = J+1
      IA(N10+13) = J
C
C
      RETURN
      END
```

```
      SUBROUTINE MATRXP ( IA,JA,A,E,P,DEQP,EQSIG,RC,G,PB,GB,CLZ,KFRIC,
     *                    NS )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
C
      DIMENSION  IA(IGENP1) , JA(NONZRP) , A(NONZRP) , E(NN,NTR) ,
     *           P(NN,NTR) , DEQP(NN,NTR) , EQSIG(NN,NTR) , RC(NN,NCL1),
     *           G(NN,NTR) , PB(NN,2) , GB(NN,2) , CLZ(NN) , KFRIC(NN)
C
C
C     ELEMENTS A OF AX=B , AND IA,JA FOR USE IN THE SUBSEQUENT
C     CALL TO THE SL-MATH SUBROUTINES
C
C     FOR PLENUM SECTION ONLY
C
C
      J = 0
C     1 - N
      DO 1000 I=1,NCL
      XE = E(NS,NFR+I)
      XP = P(NS,NFR+I)
      J = J + 1
      IA(I) = J
      JA(J) = I
      A(J) = 1.0
      J = J + 1
      JA(J) = I + N2C
      A(J) = -1./XE
      J = J + 1
      JA(J) = I + N3C
      A(J) = XP/XE
      J = J + 1
      JA(J) = I + N4C
      A(J) = XP/XE
      J = J + 1
      JA(J) = I + N5C
      A(J) = -1.0
 1000 CONTINUE
C
C     N+1 - 2N
C
      DO 1100 I=1,NCL
      XE = E(NS,NFR+I)
      XP = P(NS,NFR+I)
      J = J + 1
      IA(N1C+I) = J
      JA(J) = I + N1C
      A(J) = 1.0
      J = J + 1
      JA(J) = I + N2C
```

```
      A(J) = XP/XE
      J = J + 1
      JA(J) = I + N3C
      A(J) = -1./XE
      J = J + 1
      JA(J) = I + N4C
      A(J) = XP/XE
      J = J + 1
      JA(J) = I + N6C
      A(J) = -1.0
 1100 CONTINUE
C
C     2N+1 - 3N
C
      DO 1200 I=1,NCL
      XEP = DEQP(NS,NFR+I)
      XES = EQSIG(NS,NFR+I)
      J = J + 1
      IA(N2C+I) = J
      JA(J) = I + N2C
      A(J) = XEP/XES
      J = J + 1
      JA(J) = I + N3C
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N4C
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N5C
      A(J) = -1.0
 1200 CONTINUE
C
C     3N+1 - 4N
C
      DO 1300 I=1,NCL
      XEP = DEQP(NS,NFR+I)
      XES = EQSIG(NS,NFR+I)
      J = J + 1
      IA(I+N3C) = J
      JA(J) = I + N2C
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N3C
      A(J) = XEP/XES
      J = J + 1
      JA(J) = I + N4C
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N6C
      A(J) = -1.0
 1300 CONTINUE
C
C     4N+1 - 5N
C
      DO 1400 I=1,NCL
      XEP = DEQP(NS,NFR+I)
      XES = EQSIG(NS,NFR+I)
      J = J + 1
```

```
      IA(I+N4C) = J
      JA(J) = I + N2C
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N3C
      A(J) = -XEP/(2.*XES)
      J = J + 1
      JA(J) = I + N4C
      A(J) = XEP/XES
      J = J + 1
      JA(J) = I + N7C
      A(J) = -1.0
 1400 CONTINUE
C
C
C
C     5N+1 - 5N+3
C
      DO 1800 I=1,3
      J = J + 1
      IA(I+N5C) = J
C
      DO 1850 IJ=1,NCL
      RO = RC(NS,IJ+1)
      RI = RC(NS,IJ)
      JA(J) = N5C + (I-1)*NCL + IJ
      A(J) = (RO+RI)*(RO-RI)
      J = J + 1
 1850 CONTINUE
C
      JA(J) = N8C  + I
      RO = RC(NS,NCL1)
      RI = RC(NS,1)
      A(J) = -(RO+RI)*(RO-RI)
 1800 CONTINUE
C
C     5N+4 - 7N+3
C
C     INNER DISPLACEMENT        5N+4 - 6N+3
C
      DO 1900 I=1,NCL
      J = J + 1
      IA(I+N5C+3) = J
      RO = RC(NS,I+1)
      RI = RC(NS,I)
      RR = (RO+RI)*(RO-RI)
      XP = P(NS,NFR+I)
      XG = G(NS,NFR+I)
      JA(J) = I + N5C
      A(J) = -RI*(1.-2.*XP)/2. + RI*RO*RO*ALOG(RO/RI) / RR
      J = J + 1
      JA(J) = I + N6C
      A(J) = -RI*(1.-2.*XP)/2. - RI*RO*RO*ALOG(RO/RI) / RR
      J = J + 1
      JA(J) = N8C + 4
      A(J) = RI*XP
      J = J + 1
      JA(J) = N8C + 4 + I
```

```
      A(J) = 1.0
      J = J + 1
      JA(J) = N9C + 5 + I
      A(J) = -RI*((1.-2.*XP)*RI*RI+RO*RO) / (2.*XG*RR)
      J = J + 1
      JA(J) = N9C + 6 + I
      A(J) = (1.-XP)*RI*RO*RO / (XG*RR)
 1900 CONTINUE
C
C     OUTER DISPLACEMENT       6N+4 - 7N+3
C
      DO 2000 I=1,NCL
      J = J + 1
      IA(I+N6C+3) = J
      RO = RC(NS,I+1)
      RI = RC(NS,I)
      RR = (RO+RI)*(RO-RI)
      XP = P(NS,NFR+I)
      XG = G(NS,NFR+I)
      JA(J) = I + N5C
      A(J) = -RO*(1.-2.*XP)/2. + RO*RI*RI*ALOG(RO/RI)/RR
      J = J + 1
      JA(J) = I + N6C
      A(J) = -RO*(1.-2.*XP)/2. - RO*RI*RI*ALOG(RO/RI)/RR
      J = J + 1
      JA(J) = N8C + 4
      A(J) = RO*XP
      J = J + 1
      JA(J) = N8C + 5 + I
      A(J) = 1.0
      J = J + 1
      JA(J) = N9C + 5 + I
      A(J) = -RO*RI*RI*(1.-XP) / (XG*RR)
      J = J + 1
      JA(J) = N9C + 6 + I
      A(J) =  RO*(RI*RI+(1.-2.*XP)*RO*RO) / (2.*XG*RR)
 2000 CONTINUE
C
C     BOUNDARY CONDITION FOR PRESSURE
C
      J = J + 1
      IA(N7C+4) = J
      JA(J) = N9C + 6
      A(J) = 1.0
      J = J + 1
      IA(N7C+5) = J
      JA(J) = N10C + 6
      A(J) = 1.
C
C
C
C     CLAD AXIAL DISPLACEMENT             7N+6
C
      J = J + 1
      IA(N7C+6) = J
      RO = RC(NS,NCL1)
      RI = RC(NS,1)
      XPB = PB(NS,2)
```

```
      XGB = GB(NS,2)
      RR = (RO+RI)*(RO-RI)
      JA(J) = N8C + 1
      A(J) = XPB
      J = J + 1
      JA(J) = N8C + 2
      A(J) = XPB
      J = J + 1
      JA(J) = N8C + 3
      A(J) = -1.0
      J = J + 1
      JA(J) = N8C + 4
      A(J) = 1.+XPB
      J = J + 1
      JA(J) = N9C + 6
      A(J) = XPB*RI*RI / (XGB*RR)
      J = J + 1
      JA(J) = N10C + 7
C==== REMARK
      A(J) = -CLZ(NS) / (2.*PI*XGB*RR)
C==== REMARK
C
C     STRESS     ===  SIGR  ===                          7N+7 - 8N+6
C
      DO 2300 I=1,NCL
      J = J + 1
      IA(I+N7C+6) = J
      JA(J) = I + N2C
      A(J) = 1.0
      J = J + 1
      RO = RC(NS,I+1)
      RI = RC(NS,I)
      RR = (RO+RI)*(RO-RI)
      XP = P(NS,NFR+I)
      XG = G(NS,NFR+I)
      TH = 1.-2.*RI*RI*ALOG(RO/RI)/RR
      OM = 1.-2.*RO*RO*ALOG(RO/RI)/RR
      JA(J) = I + N5C
      A(J) = XG*((1.-TH)*OM + TH ) / (2.*(1.-XP))
      J = J + 1
      JA(J) = I + N6C
      A(J) = XG*((TH-1.)*OM - TH ) / (2.*(1.-XP))
      J = J + 1
      JA(J) = N9C + 5 + I
      A(J) = -RI*RI*OM / RR
      J = J + 1
      JA(J) = N9C + 6 + I
      A(J) = RO*RO*TH / RR
 2300 CONTINUE
C
C     STRESS     ===  SIGC  ===                          8N+7 - 9N+6
C
      DO 2400 I=1,NCL
      J = J + 1
      IA(I+N8C+6) = J
      JA(J) = I + N3C
      A(J) = 1.0
      J = J + 1
```

```
C
      RO = RC(NS,I+1)
      RI = RC(NS,I)
      RR = (RO+RI)*(RO-RI)
      XP = P(NS,NFR+I)
      XG = G(NS,NFR+I)
      TH = 1.-2.*RI*RI*ALOG(RO/RI) / RR
      OM = 1.-2.*RO*RO*ALOG(RO/RI) / RR
C
      JA(J) = I + N5C
      A(J) = XG*((OM-1.)*TH-   OM) / (2.*(1.-XP))
      J = J + 1
      JA(J) = I + N6C
      A(J) = XG*((1.-OM)*TH+   OM) / (2.*(1.-XP))
      J = J + 1
      JA(J) = N9C + 5 + I
      A(J) = -RI*RI*(2.-OM) / RR
      J = J + 1
      JA(J) = N9C + 6 + I
      A(J) = RO*RO*(2.-TH) / RR
 2400 CONTINUE
C
C     STRESS     ===  SIGZ  ===                        9N+7 - 10N+6
C
      DO 2500 I=1,NCL
      J = J + 1
      IA(I+N9C+6) = J
      JA(J) = I + N4C
      A(J) = 1.0
      J = J + 1
      RO = RC(NS,I+1)
      RI = RC(NS,I)
      XP = P(NS,NFR+I)
      XG = G(NS,NFR+I)
      RR = (RO+RI)*(RO-RI)
C
      TH = 1.-2.*RI*RI*ALOG(RO/RI) / RR
      OM = 1.-2.*RO*RO*ALOG(RO/RI) / RR
C
      JA(J) = I + N5C
      A(J) =-2.*XG*XP
      J = J + 1
      JA(J) = I + N6C
      A(J) =-2.*XG*XP
      J = J + 1
      JA(J) = I + N7C
      A(J) = 2.*XG
      J = J + 1
      JA(J) = N8C + 4
      A(J) = -2.*XG*(1.+XP)
      J = J + 1
      JA(J) = N9C + 5 + I
      A(J) = -2.*XP*RI*RI / RR
      J = J + 1
      JA(J) = N9C + 6 + I
      A(J) =  2.*XP*RO*RO / RR
 2500 CONTINUE
C
```

```
C     FRICTION FORCE             10N+7                               00089200
C                                                                    00089300
      J = J + 1                                                      00089400
      IA(N10C+7) = J                                                 00089500
      JA(J) = N10C + 7                                               00089600
      A(J) = 1.0                                                     00089700
C                                                                    00089800
      J = J + 1                                                      00089900
      IA(N10C+8) = J                                                 00090000
C                                                                    00090100
      RETURN                                                         00090200
      END                                                            00090300
```

```
      SUBROUTINE OTMCAL ( OTOM,N )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      DIMENSION  OTOM(NN)
C
C     OTOM (O/M) MODIFY
C
      RETURN
      END
```

```
      SUBROUTINE PLENP ( RVB,RCI,TFR,FLZ,GRT,FL,CL,TOUT,PLPR,CVPR,RUB,
     *                   CLZ )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *            N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
C
      DIMENSION  RVB(NN) , RCI(NN) , TFR(NN,NFR1) , FLZ(NN) ,RUB(NN) ,
     *           CLZ(NN)
C
C
C     PLENUM PRESSURE , AND CENTRAL VOID PRESSURE CALC.
C
C
      VT = 0.
      VTT = 0.
      TO = TOUT + 273.
C     VPL = PI*RCI(NN)*RCI(NN)*CLZ(NN)
      VPL = PI*RCI(NN)*RCI(NN)*( CL-FL )
C
      DO 1000 N=1,NNF
      VVO = PI*RVB(N)*RVB(N)*FLZ(N)
      VT = VT + VVO
      TX = TFR(N,1) + 273.
      VTT = VTT + VVO/TX
C
      VGAP = PI*( RCI(N)+RUB(N))*(RCI(N)-RUB(N) ) * FLZ(N)
      VPL = VPL + VGAP
C
 1000 CONTINUE
C
C
      VTOT = VPL + VT
C
C     TAV = VTOT/(VPL/TO + VTT)
C
C     RK IS IDEAL GAS CONSTANT IN KG*CM/MOLE/DEG K
C
C     PLPR = GRT*RK*TAV / VTOT
C
      PLPR = GRT*RK*TO / VTOT
      CVPR = GRT*RK*VT / (VTOT*VTT)
C
C     PLPR,CVPR NOW IN KG/CM**2
C
C
      RETURN
      END
```

```
      SUBROUTINE PLOT1 ( XPOW,XRL,XTL,XT,XTLMX,XRLR,XTLR,XCPGP,XCPU,
     *                   XBU,POW,RFRPM,TFRPM,RCB,REB,RUB,RCI,CPU,BU,
     *                   RFR,TFR,IT,N )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX
      COMMON /XFRIC/ FST,FDY,GAP
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      DIMENSION  XPOW(NBUF) , XRL(NBUF,NP1) , XTL(NBUF,NP1) , XT(NBUF) ,
     *           XTLMX(NBUF) , XRLR(NBUF,2) , XTLR(NBUF,2) ,
     *           XCPGP(NBUF) , XCPU(NBUF,NP) , XBU(NBUF) , POW(NN) ,
     *           RFRPM(NN,NP1) , TFRPM(NN,NP1) , RCB(NN) , REB(NN) ,
     *           RUB(NN) , RCI(NN) , CPU(NN,NP) , BU(NN) ,
     *           RFR(NN,NFR1) , TFR(NN,NFR1)
C
C     FUEL REGION PLOTTER PREPARATION
C
C
      XPOW(IT) = POW(N)
      XBU(IT) = BU(N)
      XRLR(IT,1) = RCB(N)*10.
      XRLR(IT,2) = REB(N)*10.
      XCPGP(IT) = GAP * 10.0
      XTLR(IT,1) = TFR(N,NCR+1)
      XTLR(IT,2) = TFR(N,NCR+NER+1)
C
      XTLMAX = 1.E-10
      DO 100 I=1,NP1
      XRL(IT,I) = RFRPM(N,I)*10.0
      XTL(IT,I) = TFRPM(N,I)
      IF ( XTL(IT,I).GT.XTLMAX ) XTLMAX=XTL(IT,I)
      IF ( I.EQ.NP1 ) GO TO 100
      XCPU(IT,I) = CPU(N,I)
  100 CONTINUE
      XTLMX(IT) = XTLMAX
      XT(IT) = XTL(IT,NP1)
      RETURN
      END
```

```
      SUBROUTINE PLOT2 ( XSIGC,SIGC,VOL,IT,N )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      DIMENSION  XSIGC(NBUF) , SIGC(NN,NTR) , VOL(NN,NTR)
C
C
C     CLAD STRESS PLOTTER PREPARATION
C
C     CLAD AVERAGE HOOP STRESS
C
      X = 0.
      Y = 0.
      DO 100 I=1,NCL
      Y = Y + VOL(N,NFR+I)
      X = X + SIGC(N,NFR+I)*VOL(N,NFR+I)
  100 CONTINUE
      XSIGC(IT) = X*0.01 / Y
      RETURN
      END
```

```
      SUBROUTINE PLOT3 ( XCSTN,XTSC,XTPC,RCO,RCOO,TSC,TPC,VOL,RDIS,IT,N)
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      DIMENSION  XCSTN(NBUF) , XTSC(NBUF) , XTPC(NBUF) , RCO(NN) ,
     *           RCOO(NN) , TSC(NN,NTR) , TPC(NN,NTR) , VOL(NN,NTR) ,
     *           RDIS(NN,NTR2)
C
C
C     CLAD STRAIN PLOTTER PREPARATION
C
C
      XCSTN(IT) = ( RCO(N)+RDIS(N,NTR2)-RCOO(N) ) / RCOO(N) * 100.
      X = 0.
      Y = 0.
      Z = 0.
      DO 100 I=1,NCL
      Z = Z + VOL(N,NFR+I)
      X = X + TSC(N,NFR+I)*VOL(N,NFR+I)
      Y = Y + TPC(N,NFR+I)*VOL(N,NFR+I)
  100 CONTINUE
      XTSC(IT) = X*100. / Z
      XTPC(IT) = Y*100. / Z
      RETURN
      END
```

```
      SUBROUTINE PLOT4 ( RFRPM,RCB,REB,LMR,LMC,LMZ,IT,N )
      REAL*8 DAYDAY
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      COMMON /DAY/ DAYDAY
      DIMENSION  RFRPM(NN,NP1) , RCB(NN) , REB(NN) ,
     *           LMR(NN,NTR) , LMC(NN,NTR) , LMZ(NN,NTR)
C
C**** IPCRK = AXIAL SECTION FOR FUEL CRACK
C
C     FUEL CRACK PLOTTER PREPARATION
C
C
C     CRACK PLOT
C
C
C
      IF ( IPCNT.GT.0 ) GO TO 100
      CALL SYMBOL(0.,200.,10.,'** RUN DATE **',0.,14)
      CALL SYMBOL(30.,185.,10.,DAYDAY,0.,8)
      CALL SYMBOL(40.,100.,10.,'ACTIVE-II',0.,9)
      CALL PLOT(200.,0.,-3)
  100 CONTINUE
C
C
      DX = 100. / (RFRPM(N,NP1)*10. )
      RADI = RFRPM(N,1)*10.*DX
      DELR1 = ( RCB(N)-RFRPM(N,1) ) / NCR * 10.0
      DELR2 = ( REB(N)-RCB(N) ) / NER * 10.0
      DELR3 = ( RFRPM(N,NP1)-REB(N) ) / NUR * 10.0
      CALL PLOT ( 150.,0.,-3 )
      CALL NEWPEN(1)
      RAD = RADI
      CALL CIRCLE ( RAD,130.,0.,360.,RAD,RAD,0.0 )
C
C
C     RADIAL
C
C
      DO 10040 K=1,3
      IF ( K-2 ) 10041,10042,10043
10041 KK = 1
      KKK = NCR
      DELR = DELR1
      GO TO 10044
10042 KK = NCR+1
      KKK = NCR+NER
      DELR = DELR2
      GO TO 10044
10043 KK = NCR+NER+1
      KKK = NFR
      DELR = DELR3
10044 CONTINUE
C
C
```

```
      DO 10050 KL=KK,KKK
      CALL NEWPEN(2)
      LR = LMR (N,KL)
      DLR = DELR / (LR+1)
      IF ( LR.EQ.0 ) GO TO 10051
C
      DO 10060 L=1,LR
      RAD = RAD + DLR*DX
      CALL CIRCLE ( RAD,130.,0.,360.,RAD,RAD,0.5 )
10060 CONTINUE
C
10051 CONTINUE
C
      RAD = RAD + DLR*DX
      CALL NEWPEN(1)
      CALL CIRCLE ( RAD,130.,0.,360.,RAD,RAD,0.0 )
10050 CONTINUE
C
      IF ( K.EQ.3 ) GO TO 10040
      CALL SYMBOL ( RAD,130.,3.,K,0.,-1 )
      CALL SYMBOL ( 0.,130.+RAD,3.,K,90.,-1 )
      CALL SYMBOL ( -RAD,130.,3.,K,180.,-1 )
      CALL SYMBOL ( 0.,130.-RAD,3.,K,270.,-1 )
C
10040 CONTINUE
C
      CALL SYMBOL ( -35.,15.,5.,'R-C-DIRECTION CRACK',0.,19 )
      CALL AXIAL ( -50.,0.,IPCRK )
      CALL SYMBOL ( 30.,0.,4.,'STEP =',0.,6 )
      CALL NUMBER ( 58.,0.,4.,FLOAT(IT),0.,-1 )
C
C
C     CIRCUMFERENTIAL
C
      CALL NEWPEN(3)
C
      LCMX = 0
      DO 10070 K=1,NER
      IF ( LMC(N,K).GT.LCMX ) LCMX=LMC(N,K)
10070 CONTINUE
C
      IF ( LCMX.NE.0 ) ANG=2.*PI / LCMX
      RAD = RADI
C
      DO 10080 K=1,3
      IF ( K-2 ) 10081,10082,10083
10081 KK = 1
      KKK = NCR
      DELR = DELR1
      GO TO 10084
10082 KK = NCR+1
      KKK = NCR+NER
      DELR = DELR2
      GO TO 10084
10083 KK = NCR+NER+1
      KKK = NER
      DELR = DELR3
10084 CONTINUE
```

```
C
      DO 10090 L=KK,KKK
      RADO = RAD
      RAD = RAD + DELR*DX
      IF ( LMC (N,L).EQ.0 ) GO TO 10090
      LC = LMC (N,L)
C
      DO 10100 LL=1,LC
      ANGL = ANG*(LL-1)
      X0 = RAD*COS( ANGL )
      Y0 = RAD*SIN( ANGL ) + 130.
      X1 = RADO*COS( ANGL )
      Y1 = RADO*SIN( ANGL ) + 130.
      CALL AROHD ( X0,Y0,X1,Y1,RAD-RADO,3.0,6 )
10100 CONTINUE
C
10090 CONTINUE
C
10080 CONTINUE
C
      CALL NEWPEN(2)
      CALL SYMBOL ( 150.,180.,4.,'---',0.,3 )
      CALL SYMBOL ( 166.,180.,4.,'CIRCUMF. CRACK',0.,14 )
      CALL NEWPEN(3)
      CALL SYMBOL ( 150.,160.,4.,'---',0.,3 )
      CALL SYMBOL ( 166.,160.,4.,'RADIAL CRACK',0.,12 )
      CALL NEWPEN(1)
      CALL SYMBOL ( 150.,140.,4.,1,0.,-1 )
      CALL SYMBOL ( 166.,140.,4.,'COLUMNAR BNDRY',0.,14 )
      CALL SYMBOL ( 150.,120.,4.,2,0.,-1 )
      CALL SYMBOL ( 166.,120.,4.,'EQUIAXED BNDRY',0.,14 )
      CALL PLOT ( 300.,0.,-3 )
C
C
C     AXIAL
C
C
      LZMX = 0
      DO 10110 K=1,NFR
      IF ( LMZ(N,K).GT.LZMX ) LZMX=LMZ(N,K)
10110 CONTINUE
C
      DY = 100. / (LZMX+1)
      CALL NEWPEN(1)
C
      Y0 = 190.
      DO 10119 K=1,4
      CALL PLOT ( 0.,Y0,3 )
      Y0 = Y0 - 20.4
      CALL PLOT ( 0.,Y0,2 )
      Y0 = Y0 - 1.5
      CALL PLOT ( 0.,Y0,3 )
      Y0 = Y0 - 1.5
      CALL PLOT ( 0.,Y0,2 )
      Y0 = Y0 - 1.5
10119 CONTINUE
      CALL PLOT ( 0.,Y0,3 )
      CALL PLOT ( 0.,70.,2 )
```

```
C
      RAD = RADI
      CALL PLOT ( RAD,180.,3 )
      CALL PLOT ( RAD,80.,2 )
C
      DO 10120 K=1,3
      IF ( K-2 ) 10121,10122,10123
10121 KK = 1
      KKK = NCR
      DELR = DELR1
      GO TO 10124
10122 KK = NCR+1
      KKK = NCR+NER
      DELR = DELR2
      GO TO 10124
10123 KK = NCR+NER+1
      KKK = NFR
      DELR = DELR3
10124 CONTINUE
C
      DO 10130 L=KK,KKK
      CALL NEWPEN(1)
      RADO = RAD
      RAD = RAD + DELR*DX
      CALL PLOT ( RAD,180.,3 )
      IF ( L.LT.KKK ) GO TO 10125
      CALL PLOT ( RAD,80.,2 )
      GO TO 10126
10125 CALL DASHPT ( RAD,80.,2.0*FLOAT(K) )
10126 CONTINUE
      IF ( LMZ (N,L).EQ.0 ) GO TO 10130
      LZ = LMZ (N,L)
      CALL NEWPEN(2)
C
      RADC = (RAD+RADO)*0.5
      DO 10140 LL=1,LZ
      Y = 180.-DY*LL
      CALL AROHD ( RADC,Y,RADO,Y,RADC-RADO,2.,6 )
      CALL AROHD ( RADC,Y,RAD,Y,RAD-RADC,2.,6 )
10140 CONTINUE
C
10130 CONTINUE
C
10120 CONTINUE
C
      CALL NEWPEN(1)
      CALL PLOT ( RADI,180.,3 )
      CALL PLOT ( RAD,180.,2 )
      CALL PLOT ( RADI,80.,3 )
      CALL PLOT ( RAD,80.,2 )
      CALL SYMBOL (   0.,15.,5.,'Z-DIRECTION CRACK',0.,17 )
      CALL AXIAL (   0.,0.,IPCRK )
      CALL SYMBOL ( 80.,0.,4.,'STEP =',0.,6 )
      CALL NUMBER ( 108.,0.,4.,FLOAT(IT),0.,-1 )
C
C
C
C
```

```
      CALL NEWPEN(2)
      CALL SYMBOL ( 130.,170.,4.,'---',0.,3 )
      CALL SYMBOL ( 146.,170.,4.,'AXIAL CRACK',0.,11 )
      CALL NEWPEN(1)
      CALL SYMBOL ( 130.,150.,2.,'-------',0.,6 )
      CALL SYMBOL ( 146.,150.,4.,'COLUMNAR GR. REGION',0.,19 )
      CALL SYMBOL ( 130.,130.,4.,'---',0.,3 )
      CALL SYMBOL ( 146.,130.,4.,'EQUIAXED GR. REGION',0.,19 )
      CALL SYMBOL ( 130.,110.,6.,'--',0.,2 )
      CALL SYMBOL ( 146.,110.,4.,'UNDISTURBED GR. REGION',0.,22 )
      CALL PLOT ( 240.,0.,-3 )
C
      RETURN
      END
```

```
      SUBROUTINE PLOT5 ( XTIME,XPLPR,XYP,ITP,IT,ITN,TIM,PLPR )          00033500
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,     00033600
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1         00033700
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2), 00033800
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX     00033900
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT                00034000
      COMMON /COMPLT/ YP,DOOD,TITLE(20)                                 00034100
      DIMENSION  XTIME(NBUF) , XPLPR(NBUF) , XYP(NBUF) , ITP(NBUF)      00034200
      ITP(IT) = ITN                                                     00034300
      XTIME(IT) = TIM                                                   00034400
      XPLPR(IT) = PLPR                                                  00034500
      XYP(IT) = YP                                                      00034600
      RETURN                                                            00034700
      END                                                               00034800
```

```
      SUBROUTINE PLTOR ( XPOW,XRL,XTL,XT,XTLMX,XRLR,XTLR,XCPGAP,XCPU,
     *                   XBU,XSIGC,XCSTN,XTSC,XTPC,XTIME,XPLPR,XYP,
     *                   ITP,IT,ITN )
      REAL*8  DAYDAY
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX
      COMMON /DAY/ DAYDAY
      COMMON /COMPLT/ YP,DDOD,TITLE(20)
      COMMON /PLTOPT/ IPFUL,IPSTS,IPSTN,IPCRK,NBUF,IPCNT
      DIMENSION  XPOW(NBUF) , XRL(NBUF,NP1) , XTL(NBUF,NP1) ,
     *           XT(NBUF) , XTLMX(NBUF) , XRLR(NBUF,2) , XTLR(NBUF,2) ,
     *           XCPGAP(NBUF) , XCPU(NBUF,NP) , XBU(NBUF) ,
     *           XSIGC(NBUF) , XCSTN(NBUF) , XTSC(NBUF) ,
     *           XTPC(NBUF) , XTIME(NBUF) , XPLPR(NBUF) ,
     *           XYP(NBUF) , ITP(NBUF)
C****
C**** ITP = TIME STEP NUMBER SELECTED
C**** XPOW  = POWER   (W/CM)
C**** XBU   = BURN-UP (ATOMIC-PERCENT)
C**** XTL   = FUEL TEMP   (DEGC)
C**** XRL   = FUEL RADIAL COODINATE    (MM)
C**** XTLMX = FUEL PIN MAXIMUM TEMP    (DEGC)
C**** XPLPR = PLENUM PRESSURE    (KG/CM**2)
C**** XYP   = FISSION GAS RELEASED PERCENT     ( ( )
C**** XT    = PELLET SURFACE TEMP    (DEGC)
C**** XTIME = TIME    (HRS)
C**** XSIGC = CLAD HOOP STRESS   (KG/MM**2)
C**** XRLR(1) = COLUMNAR GRAIN REGION BOUNDARY   (MM)
C**** XRLR(2) = EQUIAXED REGION BOUNDARY         (MM)
C**** XCPGAP  = CLAD-PELLET GAP  (MM)
C**** XCSTN = CLAD STRAIN - DELTAD/D - PERCENT   (()
C**** XCPU  = PLUTONIUM CONCENTRATION
C**** XTSC  = CIRCUMFERENTIAL SWELLING STRAIN COMPONENT ACCUMULATED
C**** XTPC  = HOOP CREEP STRAIN ACCUMULATED
C**** IPFUL = AXIAL SECTION FOR FUEL AND CLAD-PELLET GAP
C**** IPSTS = AXIAL SECTION FOR CLAD HOOP STRESS
C**** IPSTN = AXIAL SECTION FOR CLAD STRAIN - PERCENT CHANGE
C****         IN ORIGINAL O.D.
C****
C**** PLOT PEN SELECTION    N = 1  --- BLACK
C****                           2  --- RED
C****                           3  --- GREEN
      DATA  IPLT /0/
C**** IPLT = 1 ---SKIP PLOT AXIS
      ITN = ITN - 1
      XRL(1,1) = 0.
C****
      IF ( IPCNT.GT.0 ) GO TO 100
      CALL SYMBOL(0.,200.,10.,'** RUN DATE **',0.,14)
      CALL SYMBOL(30.,185.,10.,DAYDAY,0.,8)
      CALL SYMBOL(40.,100.,10.,'ACTIVE-II',0.,9)
      CALL PLOT(100.,0.,-3)
  100 CONTINUE
      CALL PLOT (150.,0.,-3 )
C****
      FMPMAX = 1.E-10
```

```
      DO 1000 I=1,IT
      IF(XTLMX(I).GT.FMPMAX) FMPMAX=XTLMX(I)
 1000 CONTINUE
      DO 1100 I=1,10
      MM =1000*I
      IF(FLOAT(MM).GT.FMPMAX) GO TO 1200
 1100 CONTINUE
 1200 CONTINUE
      MY = MM / 100
C****
      MN = 50000
      IF(XTIME(IT).LT.100000.) MN = 5000
      IF(XTIME(IT).LT.10000.)  MN = 500
      IF(XTIME(IT).LT.1000.)   MN = 50
      IF(XTIME(IT).LT.100.)    MN = 5
      DO 1300 I=1,10000
      NMN = MN*I
      IF(FLOAT(NMN).GT.XTIME(IT))  GO TO 1400
 1300 CONTINUE
 1400 CONTINUE
      MX = NMN / MN
C**** WRITE AXIS
      DY = MY / 220.
      DX = MX / 300.
      CALL NEWPEN (3)
      CALL PLOT(-34.,0.,3)
      CALL PLOT(-34.,220.,2)
      CALL NEWPEN (2)
      CALL PLOT (0.,220.,3)
      CALL PLOT (0.,0.,2)
      CALL NEWPEN (1)
      CALL PLOT (300.,0.,2)
      CALL PLOT (300.,220.,2)
      CALL PLOT (0.,220.,2)
      DO  1500 I=1,MY
      J=MY-I+1
      IF(MOD(I,5).NE.1) GO TO 1600
      CALL PLOT (298.,J/DY,3)
      CALL PLOT (302.,J/DY,2)
      F = FLOAT(J*100)
      CALL NUMBER (303.,J/DY-2.0,4.,F,0.,-1)
      GO TO 1500
 1600 CONTINUE
      CALL PLOT (299.0,J/DY,3)
      CALL PLOT (301.0,J/DY,2)
 1500 CONTINUE
      CALL NUMBER (303.,-2.0,4.,0.,0.,-1)
      CALL SYMBOL (328.,80.0,5.,'TEMPERATURE ( C )',90.,17)
      CALL NUMBER (324.,148.,2.,0.,90.,-1)
      DO 1700 I=1,MX
      CALL NEWPEN(3)
      J=MX-I+1
      IF(I.EQ.1) GO TO 1650
      CALL PLOT(J/DX,220.,3)
      CALL PLOT(J/DX,0.,2)
 1650 CONTINUE
      F=FLOAT(MN*J)
      CALL NEWPEN(1)
```

```
      CALL NUMBER(J/DX-4.0,-7.0,4.0,F,0.,-1)
 1700 CONTINUE
      CALL NUMBER(-2.0,-7.,4.,0.,0.,-1)
      CALL SYMBOL(100.0,-20.0,5.,'OPERATION TIME (HOUR)',0.,21)
      CALL NEWPEN(2)
      CALL NUMBER(-14.,-2.5,5.,0.,0.,-1)
      I=0
 2000 I=I+1
      CALL PLOT(-1.0,I*11.0,3)
      CALL PLOT(1.0,I*11.0,2)
      I=I+1
      F=FLOAT(I*5)
      CALL PLOT(-2.0,FLOAT(I)*11.0,3)
      CALL PLOT(2.0,FLOAT(I)*11.0,2)
      CALL NUMBER (-15.0,I*11.0-2.0,4.,F,0.,-1)
      IF(I.LT.20) GO TO 2000
      CALL SYMBOL(-20.,80.,5.,'FP GAS RELEASED (     )',90.,21)
      CALL SYMBOL (-20.,170.,5.,108,90.,-1)
      CALL NEWPEN(3)
      CALL PLOT(-36.0,0.,3)
      CALL PLOT(-32.0,0.,2)
      CALL NUMBER(-42.,-2.0,4.,0.,0.,-1)
      XPLMX=1.E-20
      DO 2050 I=1,IT
      IF(XPLMX.LT.XPLPR(I)) XPLMX=XPLPR(I)
 2050 CONTINUE
      DO 2060 I=1,100
      IF(XPLMX.GT.FLOAT(I*100)) GO TO 2060
      IPLN=I
      GO TO 2070
 2060 CONTINUE
 2070 CONTINUE
      I=0
 2100 I=I+1
      CALL PLOT(-35.0,FLOAT(I)*11.0,3)
      CALL PLOT(-33.0,FLOAT(I)*11.0,2)
      I=I+1
      F = FLOAT(I*5*IPLN)
      CALL PLOT(-36.,FLOAT(I)*11.0,3)
      CALL PLOT(-32.,FLOAT(I)*11.0,2)
      CALL NUMBER(-46.,FLOAT(I)*11.0-2.0,4.,F,0.,-1)
      IF(I.LT.20)  GO TO 2100
      CALL SYMBOL(-50.,80.,5.,'PLENUM PRESSURE (KG/CM**2)',90.,26)
      CALL NEWPEN(1)
      DX = NMN / 300.
      DY = MM / 220.
      ITM = IT-1
      DO 4000 I=1,ITM
      CALL SYMBOL (XTIME(I)/DX,XTLMX(I)/DY,2.0,0,0.,-1)
      CALL PLOT (XTIME(I)/DX,XTLMX(I)/DY,3)
      CALL PLOT (XTIME(I+1)/DX,XTLMX(I+1)/DY,2)
 4000 CONTINUE
      CALL SYMBOL (XTIME(IT)/DX,XTLMX(IT)/DY,2.0,0,0.,-1)
      DO 4100 I=1,ITM
      CALL SYMBOL (XTIME(I)/DX,XT(I)/DY,2.0,1,0.,-1)
      CALL PLOT (XTIME(I)/DX,XT(I)/DY,3)
      CALL PLOT (XTIME(I+1)/DX,XT(I+1)/DY,2)
 4100 CONTINUE
```

```
      CALL SYMBOL (XTIME(IT)/DX,XTLMX(IT)/DY,2.0,1,0.,-1)                00052300
      CALL NEWPEN(3)                                                     00052400
      DY=50.0*IPLN / 110.0                                               00052500
      DO 4200 I=1,ITM                                                    00052600
      CALL SYMBOL (XTIME(I)/DX,XPLPR(I)/DY,2.0,2,0.,-1)                  00052700
      CALL PLOT (XTIME(I)/DX,XPLPR(I)/DY,3)                              00052800
      CALL PLOT (XTIME(I+1)/DX,XPLPR(I+1)/DY,2)                          00052900
 4200 CONTINUE                                                           00053000
      CALL SYMBOL (XTIME(IT)/DX,XPLPR(IT)/DY,2.0,2,0.,-1)                00053100
      CALL NEWPEN(2)                                                     00053200
      DY = 50./110.                                                      00053300
      DO 4300 I=1,ITM                                                    00053400
      CALL SYMBOL (XTIME(I)/DX,XYP(I)/DY,2.0,4,0.,-1)                    00053500
      CALL PLOT (XTIME(I)/DX,XYP(I)/DY,3)                                00053600
      CALL PLOT (XTIME(I+1)/DX,XYP(I+1)/DY,2)                            00053700
 4300 CONTINUE                                                           00053800
      CALL SYMBOL (XTIME(IT)/DX,XYP(IT)/DY,2.0,4,0.,-1)                  00053900
      CALL NEWPEN(1)                                                     00054000
      CALL SUBTIT(TITLE)                                                 00054100
      CALL AXIAL (340.,210.,IPFUL)                                       00054200
      CALL SYMBOL (340.,160.,4.,0,0.,-1)                                 00054300
      CALL SYMBOL (342.,158.,4.,' -- MAXIMUM TEMP. IN FUEL',0.,25)       00054400
      CALL SYMBOL (340.,154.,4.,1,0.,-1)                                 00054500
      CALL SYMBOL (342.,152.,4.,' -- PELLET SURFACE TEMP.',0.,24)        00054600
      CALL NEWPEN(2)                                                     00054700
      CALL SYMBOL (340.,148.,4.,4,0.,-1)                                 00054800
      CALL SYMBOL (342.,146.,4.,' -- FP GAS RELEASED',0.,19)             00054900
      CALL NEWPEN(3)                                                     00055000
      CALL SYMBOL (340.,142.,4.,2,0.,-1)                                 00055100
      CALL SYMBOL (342.,140.,4.,' -- PLENUM PRESSURE',0.,19)             00055200
C**** WRITE AXIS FOR FUEL TEMP                                           00055300
      CALL NEWPEN(1)                                                     00055400
      CALL PLOT (520.,0.,-3)                                             00055500
      CALL PLOT (300.,0.,2)                                              00055600
      CALL PLOT(300.,220.,2)                                             00055700
      CALL PLOT(0.,220.,2)                                               00055800
      XRLMAX=1.E-10                                                      00055900
      DO 4400 I=1,IT                                                     00056000
      IF(XRL(I,NP1).GT.XRLMAX) XRLMAX=XRL(I,NP1)                         00056100
 4400 CONTINUE                                                           00056200
      DO 3000 I=1,20                                                     00056300
      X=0.5*(I-1)                                                        00056400
      IF(XRLMAX.LT.X) GO TO 3100                                         00056500
 3000 CONTINUE                                                           00056600
 3100 CONTINUE                                                           00056700
      MXD = I                                                            00056800
      DXD = X/300.                                                       00056900
      CALL AXISD (MXD,0.5,DXD)                                           00057000
      CALL SYMBOL (65.,-17.,5.,'DISTANCE FROM CENTER OF PELLET (MM)',    00057100
     *                0.,35)                                             00057200
      CALL PLOT(0.,0.,3)                                                 00057300
      CALL PLOT(0.,220.,2)                                               00057400
      DY=FLOAT(MM)/220.                                                  00057500
      DO 3200 I=1,MY                                                     00057600
      CALL NEWPEN(1)                                                     00057700
      J=MY-I+1                                                           00057800
      Y= 100.*J                                                          00057900
      CALL NUMBER(-17.,Y/DY-2.,4.,Y,0.,-1)                               00058000
```

```
      IF(I.EQ.1) GO TO 3200
      CALL NEWPEN(3)
      CALL PLOT(0.,Y/DY,3)
      CALL PLOT(300.,Y/DY,2)
 3200 CONTINUE
      CALL NEWPEN(1)
      CALL NUMBER(-17.,-2.,4.,0.,0.,-1)
      CALL SYMBOL(-23.,70.0,5.,'FUEL TEMPERATURE ( C )',90.,22)
      CALL NUMBER(-27.,163.0,2.0,0.,90.,-1)
      DO 3300 I=1,IT
      ICODE = I-1
      ISW1 = 0
      ISW2 = 0
      IF(ICODE.GE.13) ICODE=13
      CALL PLOT (XRL(I,1)/DXD,XTL(I,1)/DY,3)
      DO 3400 J=2,NP1
      CALL SYMBOL(XRL(I,J-1)/DXD,XTL(I,J-1)/DY,2.0,ICODE,0.,-1)
      IF(ISW1.EQ.1) GO TO 3410
      IF(XRL(I,J).LT.XRLR(I,1)) GO TO 3410
      CALL PLOT(XRLR(I,1)/DXD,XTLR(I,1)/DY,2)
      CALL NEWPEN(2)
      CALL SYMBOL(XRLR(I,1)/DXD,XTLR(I,1)/DY,2.,ICODE,0.,-1)
      CALL NEWPEN(1)
      ISW1 = 1
 3410 CONTINUE
      IF(ISW2.EQ.1) GO TO 3420
      IF(XRL(I,J).LT.XRLR(I,2)) GO TO 3420
      CALL PLOT(XRLR(I,2)/DXD,XTLR(I,2)/DY,2)
      CALL NEWPEN(3)
      CALL SYMBOL(XRLR(I,2)/DXD,XTLR(I,2)/DY,2.,ICODE,0.,-1)
      CALL NEWPEN(1)
      ISW2 = 1
 3420 CONTINUE
      CALL PLOT(XRL(I,J)/DXD,XTL(I,J)/DY,2)
 3400 CONTINUE
      CALL SYMBOL(XRL(I,NP1)/DXD,XTL(I,NP1)/DY,2.0,ICODE,0.,-1)
 3300 CONTINUE
      CALL NEWPEN(2)
      CALL SYMBOL(320.,190.,4.,'COLUMNAR BNDRY',0.,14)
      CALL NEWPEN(3)
      CALL SYMBOL(320.,180.,4.,'EQUIAXED BNDRY',0.,14 )
      CALL NEWPEN(1)
      CALL AXIAL (320.,210.,IPFUL)
      DO 5000 I=1,IT
      IF(I.GE.15) GO TO 5010
      ITX = ITP( I )
      ICODE = I-1
      YY = 160.0-5.0*(I-1)
      CALL SYMBOL(320.,YY,4.0,ICODE,0.,-1)
      CALL SYMBOL(323.,YY-2.0,4.0,' --- STEP',0.,9)
      CALL NUMBER(360.,YY-2.0,4.,FLOAT(ITX),0.,-1)
 5000 CONTINUE
      GO TO 5020
 5010 CONTINUE
      CALL SYMBOL (369.,93.,4.,30,0.,-1)
C     CALL NUMBER (374.,93.,4.,FLOAT(IT),0.,-1)
 5020 CONTINUE
      CALL SUBTIT(TITLE)
```

```
C**** WRITE AXIS FOR VOID RADIUS
      CALL NEWPEN(1)
      CALL PLOT(480.,0.,-3)
      CALL PLOT(300.,0.,2)
      CALL PLOT (300.,220.,2)
      CALL PLOT (0.,220.,2)
      CALL AXIST(MX,FLOAT(MN),MX/300.)
      CALL SYMBOL (100.,-20.,5.,'OPERATION TIME (HOUR)',0.,21)
      CALL  PLOT(0.,0.,3)
      CALL  PLOT(0.,220.,2)
      VMAX = 1.E-20
      DO 5100 I=1,IT
      IF(VMAX.LT.XRL(I,1)) VMAX=XRL(I,1)
 5100 CONTINUE
      DO 5150 I=1,100
      XX = 0.1*(I-1)
      IF(VMAX.LT.XX) GO TO 5200
 5150 CONTINUE
 5200 CONTINUE
      MMM=(I-1)*10+1
      DY = X/220.
      DXT = NMN / 300.
      DO 6500 I=1,MXD
      F=0.5*(MXD-I)
      CALL NEWPEN(1)
      CALL NUMBER(-20.,F/DY-2.0,4.0,F,0.,2)
      IF(I.EQ.1.OR.I.EQ.MXD) GO TO 6500
      CALL NEWPEN(3)
      CALL PLOT(0. ,F/DY,3)
      CALL PLOT(300.,F/DY,2)
 6500 CONTINUE
      CALL NEWPEN(1)
      CALL SYMBOL(-28.,80.,5.,'VOID RADIUS (MM)',90.,16)
      DO 6600 I=1,ITM
      CALL SYMBOL (XTIME(I)/DXT,XRL(I,1)/DY,2.,  0 ,0.,-1)
      CALL PLOT (XTIME(I+1)/DXT,XRL(I+1,1)/DY,2)
 6600 CONTINUE
      CALL SYMBOL (XTIME(IT)/DXT,XRL(IT,1)/DY,2.,  0 ,0.,-1)
      DO 6610 I=1,ITM
      CALL SYMBOL(XTIME(I)/DXT,XRLR(I,1)/DY,2.,3,0.,-1)
      CALL PLOT(XTIME(I+1)/DXT,XRLR(I+1,1)/DY,2)
 6610 CONTINUE
      CALL SYMBOL(XTIME(IT)/DXT,XRLR(IT,1)/DY,2.,3,0.,-1)
      DO 6620 I=1,ITM
      CALL SYMBOL(XTIME(I)/DXT,XRLR(I,2)/DY,2.,4,0.,-1)
      CALL PLOT(XTIME(I+1)/DXT,XRLR(I+1,2)/DY,2)
 6620 CONTINUE
      CALL SYMBOL(XTIME(IT)/DXT,XRLR(IT,2)/DY,2.,4,0.,-1)
      CALL AXIAL(320.,210.,IPFUL)
      CALL SYMBOL (320.,180.,4.,0,0.,-1)
      CALL SYMBOL (323.,178.,4.,' --- VOID RADIUS',0.,16)
      CALL SYMBOL (320.,172.,4.,3,0.,-1)
      CALL SYMBOL (323.,170.,4.,' --- COLUMNAR BNDRY',0.,19)
      CALL SYMBOL (320.,164.,4.,4,0.,-1)
      CALL SYMBOL (323.,162.,4.,' --- EQUIAXED BNDRY',0.,19)
      CALL SUBTIT(TITLE)
C**** WRITE AXIS FOR MAXIMUM FUEL TEMP.
      CALL NEWPEN(1)
```

```
      CALL PLOT(480.,0.,-3)
      CALL PLOT(300.,0.,2)
      CALL PLOT(300.,220.,2)
      CALL PLOT(0.,220.,2)
      CALL AXIST(MX,FLOAT(MN),MX/300.)
      CALL SYMBOL(100.,-20.0,5.,'OPERATION TIME (HOUR)',0.,21)
      CALL PLOT(0.,0.,3)
      CALL PLOT(0.,220.,2)
      DY = MY/220.
      DO 6000 I=1,MY
      J=MY-I+1
      CALL NEWPEN(1)
      IF(MOD(I,5).NE.1) GO TO 6100
      F=FLOAT(J*100)
      CALL NUMBER(-19.,J/DY-2.,4.,F,0.,-1)
      IF(I.EQ.1) GO TO 6000
      CALL NEWPEN(3)
      CALL PLOT(0.0,J/DY,3)
      CALL PLOT(300.,J/DY,2)
      GO TO 6000
 6100 CONTINUE
      CALL PLOT (0.0,J/DY,3)
      CALL PLOT (2.,J/DY,2)
 6000 CONTINUE
      CALL NEWPEN(1)
      CALL NUMBER (-19.,-2.0,4.,0.,0.,-1)
      CALL SYMBOL (-22.,40.0,5.,'MAXIMUM TEMP. IN FUEL PIN ( C )',90.,
     *             31)
      CALL NUMBER(-26.,178.,2.,0.,90.,-1)
      DY=MM/220.
      DO 6200 I=1,ITM
      CALL SYMBOL (XTIME(I)/DXT,XTLMX(I)/DY,2.0,IPLT,0.,-1)
      CALL PLOT (XTIME(I+1)/DXT,XTLMX(I+1)/DY,2)
 6200 CONTINUE
      CALL SYMBOL (XTIME(IT)/DXT,XTLMX(IT)/DY,2.0,IPLT,0.,-1)
      CALL AXIAL(320.,210.,IPFUL)
      CALL SUBTIT(TITLE)
C**** WRITE AXIS FOR POWER
      PMAX=1.E-20
      DO 7000 I=1,IT
      IF(PMAX.LT.XPOW(I)) PMAX=XPOW(I)
 7000 CONTINUE
      DO 7100 I=1,10000
      MP=100*I
      IF(FLOAT(MP).GT.PMAX) GO TO 7200
 7100 CONTINUE
 7200 CONTINUE
      DYP = MP/220.
      DY = DYP
      MPY=MP/50
      CALL NEWPEN(1)
      CALL PLOT (480.,0.,-3)
      CALL PLOT (300.,0.,2)
      CALL PLOT(300.,220.,2)
      CALL PLOT(0.,220.,2)
      CALL AXIST (MX,FLOAT(MN),MX/300.)
      CALL SYMBOL (100.,-20.,5.,'OPERATION TIME (HOUR)',0.,21)
      CALL PLOT (0.,0.,3)
```

```
      CALL PLOT (0.,220.,2)
      DO 7300 I=1,MPY
      CALL NEWPEN(1)
      J=MPY-I+1
      F=J*50.
      IF(MOD(I,2).NE.1) GO TO 7400
      CALL NUMBER(-17.,F/DY-2.,4.,F,0.,-1)
      IF(I.EQ.1) GO TO 7300
      CALL NEWPEN(3)
      CALL PLOT (0.,F/DY,3)
      CALL PLOT (300.,F/DY,2)
      GO TO 7300
 7400 CONTINUE
      CALL PLOT (2.,F/DY,3)
      CALL PLOT (0.,F/DY,2)
 7300 CONTINUE
      CALL NEWPEN(1)
      CALL NUMBER (-17.,-2.,4.,0.,0.,-1)
      CALL SYMBOL(-23.,83.0,5.,'POWER (W/CM)',90.,12)
      CALL PLOT (0.,0.,3)
      DO 7500 I=1,IT
      CALL PLOT (XTIME(I)/DXT,XPOW(I)/DY,2)
      CALL SYMBOL (XTIME(I)/DXT,XPOW(I)/DY,2.0,IPLT,0.,-1)
 7500 CONTINUE
      YBUMAX=20.0
      IF(XBU(IT).LT.10.) YBUMAX=10.
      IF(XBU(IT).LT.1.0) YBUMAX=1.0
      DY= YBUMAX/220.
      CALL PLOT (330.,0.,3)
      CALL PLOT (330.,220.,2)
      DO 7510 I=1,10
      F = YBUMAX-YBUMAX*(I-1)/10
      CALL PLOT (330.,F/DY,3)
      CALL PLOT (332.,F/DY,2)
      CALL NUMBER (334.,F/DY-2.,4.,F,0.,1)
 7510 CONTINUE
      CALL PLOT(330.,0.,3)
      CALL PLOT(332.,0.,2)
      CALL NUMBER (334.,-2.,4.,0.0,0.,1)
      CALL SYMBOL (360.,60.,5.,'BURN-UP (ATOMIC-PERCENT)',90., 24)
      CALL PLOT (0.,0.,3)
      DO 7520 I=1,IT
      CALL PLOT (XTIME(I)/DXT,XBU(I)/DY,2)
      CALL SYMBOL (XTIME(I)/DXT,XBU(I)/DY,2.,4,0.,-1)
 7520 CONTINUE
      CALL AXIAL ( 380.,210.,IPFUL )
      CALL SYMBOL (380.,160.,4.,0,0.,-1)
      CALL SYMBOL (382.,158.,4.,' -- POWER',0.,9)
      CALL SYMBOL (380.,140.,4.,4,0.,-1)
      CALL SYMBOL (382.,138.,4.,' -- BURN UP',0.,11)
      CALL SUBTIT(TITLE)
C**** CLAD-PELLET GAP  AND  CLAD HOOP STRESS
      GAPMAX=1.E-10
      SIGMAX=1.E-10
      SIGMIN = 0.
      DO 8000 I=1,IT
      IF(GAPMAX.LT.XCPGAP(I)) GAPMAX=XCPGAP(I)
      IF(SIGMAX.LT.XSIGC(I))  SIGMAX=XSIGC(I)
```

```
      IF ( SIGMIN.GT.XSIGC(I) ) SIGMIN=XSIGC(I)
 8000 CONTINUE
      DO 8100 I=1,10000
      XG=0.05*I
      IF(XG.GT.GAPMAX) GO TO 8200
 8100 CONTINUE
 8200 CONTINUE
      DYG=XG/220.
      MGY=5*I
      DO 8300 I=1,10000
      XS=5.0*I
      IF(XS.GT.SIGMAX) GO TO 8400
 8300 CONTINUE
 8400 CONTINUE
      ISIGX = I
      DO 8410 I=1,10000
      XSMIN = -5.0*I
      IF ( XSMIN.LT.SIGMIN ) GO TO 8420
 8410 CONTINUE
 8420 CONTINUE
      ISIGN = I
      IF ( SIGMIN.GE.0. ) ISIGN = 0
      IF( SIGMIN.GE.0. ) XSMIN=0.
      XS = XS-XSMIN
      I = ISIGX+ISIGN
      DYS=XS/220.
      MSY= I*5
      CALL NEWPEN(1)
      CALL PLOT (510.,0.,-3)
      CALL PLOT (300.,0.,2)
      CALL PLOT (300.,220.,2)
      CALL PLOT (0.,220.,2)
      CALL PLOT (0.,0.,2)
      DO 8500 I=1,MSY
      J=MSY-I+1
      JNUM = J-5*ISIGN
      IF(MOD(J,5).NE.0) GO TO 8450
      CALL PLOT (298.,1.0*J/DYS,3)
      CALL PLOT (300.,1.0*J/DYS,2)
      IF(JNUM.LT.0) CALL PLOT(302.,1.*J/DYS,3)
      IF(JNUM.LT.0) CALL PLOT(304.,1.*J/DYS,2)
      CALLNUMBER(305.,1.0*J/DYS-2.,4.,1.0*IABS(JNUM),0.,1)
      GO TO 8500
 8450 CONTINUE
      CALL PLOT(299.,1.*J/DYS,3)
      CALL PLOT(300.,1.*J/DYS,2)
 8500 CONTINUE
      IF(ISIGN.GT.0) CALL PLOT(302.,0.,3)
      IF(ISIGN.GT.0) CALL PLOT(304.,0.,2)
      CALL NUMBER(305.,-2.,4.,5.*ISIGN,0.,1)
      CALL SYMBOL(329.,80.,5.,'CLAD HOOP STRESS (KG/MM**2)',90.,27)
      CALL AXIST (MX,FLOAT(MN),MX/300.)
      CALL SYMBOL(100.,-20.,5.,'OPERATION TIME (HOUR)',0.,21)
      CALL NUMBER (-17.,-2.,4.,0.,0.,2)
      DO 8600 I=1,MGY
      IF(MOD(I,5).NE.0) GO TO 8700
      CALL NUMBER (-17.,0.01*I/DYG-2.,4.,0.01*I,0.,2)
      CALL PLOT (0.,0.01*I/DYG,3)
```

```
      CALL PLOT (2.,0.01*I/DYG,2)
      GO TO 8600
 8700 CONTINUE
      CALL PLOT (0.,0.01*I/DYG,3)
      CALL PLOT (1.,0.01*I/DYG,2)
 8600 CONTINUE
      CALL SYMBOL (-22.,80.,5.,'RADIAL GAP SIZE (MM)',90.,20)
      XO = XTIME(1)/DXT
      YO = XCPGAP(1)/DYG
      CALL SYMBOL (XO,YO,2.0,0,0.,-1)
      DO 8800 I=2,IT
      XO = XTIME(I)/DXT
      YO = XCPGAP(I)/DYG
      CALL PLOT (XO,YO,2)
      CALL SYMBOL (XO,YO,2.,0,0.,-1)
 8800 CONTINUE
      XO = XTIME(1)/DXT
      YO = (XSIGC(1)-XSMIN)/DYS
      CALL SYMBOL (XO,YO,2.,4,0.,-1)
      DO 8900 I=2,IT
      XO = XTIME(I)/DXT
      YO = (XSIGC(I)-XSMIN)/DYS
      CALL PLOT (XO,YO,2)
      CALL SYMBOL (XO,YO,2.,4,0.,-1)
 8900 CONTINUE
      CALL SUBTIT(TITLE)
      CALL SYMBOL(343.,160.,4.,0,0.,-1)
      CALL SYMBOL(345.,158.,4.,' -- CLAD PELLET GAP',0.,19)
      CALL AXIAL (361.,152.,IPFUL)
      CALL SYMBOL(343.,130.,4.,4,0.,-1)
      CALL SYMBOL(345.,128.,4.,' -- CLAD HOOP STRESS',0.,20)
      CALL AXIAL (361.,122.,IPSTS)
C**** CLAD STRAIN AND POWER
      STNMAX = 1.E-10
      STNMIN = 0.
      DO 9000 I=1,IT
      IF(STNMAX.LT.XCSTN(I)) STNMAX=XCSTN(I)
      IF ( STNMIN.GT.XCSTN(I) ) STNMIN=XCSTN(I)
      IF ( STNMIN.GT.XTPC(I) ) STNMIN=XTPC(I)
 9000 CONTINUE
      STNXXX = STNMAX-STNMIN
      HABA = 5.0
      IF(STNXXX.LT.10.) HABA=0.50
      IF(STNXXX.LT.1.) HABA=0.05
      DO 9100 I=1,10000
      XS = HABA*I
      IF(XS.GT.STNMAX) GO TO 9200
 9100 CONTINUE
 9200 CONTINUE
      ISTNX = I
      DO 9210 I=1,10000
      XS = -HABA*I
      IF ( XS.LT.STNMIN ) GO TO 9220
 9210 CONTINUE
 9220 CONTINUE
      ISTNN = I
      IF ( STNMIN.GE.0. ) ISTNN=0
      I = ISTNX+ISTNN
```

```
      DY = I*HABA/220.                                                  00092900
      MSY = I                                                           00093000
      CALL NEWPEN(1)                                                    00093100
      CALL PLOT(500.,0.,-3)                                             00093200
      CALL PLOT(300.,0.,2)                                              00093300
      CALL PLOT(300.,220.,2)                                            00093400
      CALL PLOT(0.,220.,2)                                              00093500
      CALL PLOT(0.,0.,2)                                                00093600
      DO 9300 I=1,MPY                                                   00093700
      J = MPY-I+1                                                       00093800
      F = J*50.                                                         00093900
      CALL PLOT(298.,F/DYP,3)                                           00094000
      CALL PLOT(300.,F/DYP,2)                                           00094100
      CALL NUMBER(301.,F/DYP-2.,4.,F,0.,-1)                             00094200
 9300 CONTINUE                                                          00094300
      CALL NUMBER(301.,-2.,4.,0.,0.,-1)                                 00094400
      CALL SYMBOL(320.,90.,5.,'POWER (W/CM)',90.,12)                    00094500
      CALL AXIST(MX,FLOAT(MN),MX/300.)                                  00094600
      CALL SYMBOL(100.,-20.,5.,'OPERATION TIME (HOUR)',0.,21)           00094700
      YO = 0.                                                           00094800
      IF(YO-HABA*ISTNN.LT.0.) CALL PLOT(-22.,YO/DY,3)                   00094900
      IF(YO-HABA*ISTNN.LT.0.) CALL PLOT(-20.,YO/DY,2)                   00095000
      CALL NUMBER(-19.,-2.,4.,ABS(YO-HABA*ISTNN),0.,2)                  00095100
      DO 9400 I=1,MSY                                                   00095200
      YO = YO+HABA                                                      00095300
      IF(YO-HABA*ISTNN.LT.0.) CALL PLOT(-22.,YO/DY,3)                   00095400
      IF(YO-HABA*ISTNN.LT.0.) CALL PLOT(-20.,YO/DY,2)                   00095500
      CALL NUMBER(-19.,YO/DY-2.,4.,ABS(YO-HABA*ISTNN),0.,2)             00095600
      CALL PLOT(0.,YO/DY,3)                                             00095700
      CALL PLOT(2.,YO/DY,2)                                             00095800
 9400 CONTINUE                                                          00095900
      CALL SYMBOL(-34.,80.,5.,'CLAD STRAIN (()',90.,15)                 00096000
      CALL SYMBOL(XTIME(1)/DXT,XPOW(1)/DYP,2.,0,0.,-1)                  00096100
      DO 9500 I=2,IT                                                    00096200
      XO = XTIME(I)/DXT                                                 00096300
      YO = XPOW(I)/DYP                                                  00096400
      CALL PLOT(XO,YO,2)                                                00096500
      CALL SYMBOL(XO,YO,2.,0,0.,-1)                                     00096600
 9500 CONTINUE                                                          00096700
        XS =-ISTNN*HABA                                                 00096800
      CALL SYMBOL(XTIME(1)/DXT,(XCSTN(1)-XS)/DY,2.,4,0.,-1)             00096900
      DO 9600 I=2,IT                                                    00097000
      XO = XTIME(I)/DXT                                                 00097100
      YO = (XCSTN(I)-XS)/DY                                             00097200
      CALL PLOT(XO,YO,2)                                                00097300
      CALL SYMBOL(XO,YO,2.,4,0.,-1)                                     00097400
 9600 CONTINUE                                                          00097500
      CALL NEWPEN(2)                                                    00097600
      CALL SYMBOL (XTIME(1)/DXT,(XTPC(1)-XS)/DY,2.,2,0.,-1)             00097700
      DO 9610 I=2,IT                                                    00097800
      XO = XTIME(I)/DXT                                                 00097900
      YO = (XTPC(I)-XS)/DY                                              00098000
      CALL PLOT (XO,YO,2)                                               00098100
      CALL SYMBOL (XO,YO,2.,2,0.,-1)                                    00098200
 9610 CONTINUE                                                          00098300
      CALL NEWPEN(3)                                                    00098400
      CALL SYMBOL (XTIME(1)/DXT,(XTSC(1)-XS)/DY,2.,3,0.,-1)             00098500
      DO 9620 I=2,IT                                                    00098600
```

```
      XO = XTIME(I)/DXT
      YO = (XTSC(I)-XS)/DY
      CALL PLOT (XO,YO,2)
      CALL SYMBOL (XO,YO,2.,3,0.,-1)
 9620 CONTINUE
      CALL NEWPEN(1)
      CALL SUBTIT(TITLE)
      CALL AXIAL (340.,180.,IPSTN)
      CALL SYMBOL (340.,165.,4.,4,0.,-1)
      CALL SYMBOL (342.,163.,4.,' -- CLAD STRAIN',0.,15)
      CALL NEWPEN(2)
      CALL SYMBOL (340.,155.,4.,2,0.,-1)
      CALL SYMBOL (342.,153.,4.,' -- HOOP CREEP STRAIN ACCUMULATED',
     *            0.,33)
      CALL NEWPEN(3)
      CALL SYMBOL (340.,143.,4.,3,0.,-1)
      CALL SYMBOL (342.,141.,4.,' -- CIRCUMFERENTIAL SWELLING STRAIN',
     *            0.,35)
      CALL SYMBOL (358.,135.,4.,'COMPONENT ACCUMULATED',0.,21)
      CALL NEWPEN(1)
      CALL SYMBOL (340.,110.,4.,0,0.,-1)
      CALL SYMBOL (342.,108.,4.,' -- POWER',0.,9)
      CALL AXIAL ( 360.,102.,IPFUL )
C**** PLUTONIUM CONCENTRATION
      CALL PLOT (560.,0.,-3)
      CALL PLOT (300.,0.,2)
      CALL PLOT (300.,220.,2)
      CALL PLOT (0.,220.,2)
      CALL SUBTIT(TITLE)
      XCPUMX=1.E-10
      DO 9700 I=1,IT
      DO 9700 J=1,NP
      IF(XCPU(I,J).GT.XCPUMX) XCPUMX=XCPU(I,J)
 9700 CONTINUE
      DO 9710 I=1,100
      X=0.05*I
      IF(X.GT.XCPUMX) GO TO 9720
 9710 CONTINUE
 9720 CONTINUE
      MY = I*5
      DY = X/220.
      CALL AXISD (MXD,0.5,DXD)
      CALL SYMBOL (65.,-17.,5.,'DISTANCE FROM CENTER OF PELLET (MM)',
     *            0.,35)
      CALL PLOT (0.,0.,3)
      CALL PLOT (0.,220.,2)
      DO 9730 I=1,MY
      CALL NEWPEN(1)
      J=MY-I+1
      Y=0.01*J
      IF(MOD(J,5).NE.0) GO TO 9740
      CALL NUMBER (-17.,Y/DY-2.,4.,Y,0.,2)
      IF(I.EQ.1) GO TO 9730
      CALL NEWPEN(3)
      CALL PLOT (0.,Y/DY,3)
      CALL PLOT (300.,Y/DY,2)
      GO TO 9730
 9740 CONTINUE
```

```
      CALL PLOT (0.,Y/DY,3)
      CALL PLOT (2.,Y/DY,2)
 9730 CONTINUE
      CALL NEWPEN(1)
      CALL NUMBER (-17.,-2.,4.,0.,0.,-1)
      CALL SYMBOL (-23.,70.,5.,'PLUTONIUM CONCENTRATION',90.,23)
      DO 9800 I=1,IT
      ICODE=I-1
      ISW1=0
      ISW2=0
      IF(ICODE.GE.13) ICODE=13
      X2 = ( XRL(I,1)+ XRL(I,2))/2.0
      DO 9900 J=2,NP
      X1= X2
      X2=  ( XRL(I,J)+ XRL(I,J+1))/2.0
      CALL SYMBOL (X1/DXD,XCPU(I,J-1)/DY,2.0,ICODE,0.,-1)
      IF(ISW1.EQ.1)  GO TO 9910
      IF(X2.LT.XRLR(I,1)) GO TO 9910
      Y = XCPU(I,J) + (XCPU(I,J-1)-XCPU(I,J))*(X2-XRLR(I,1))/(X2-X1)
      CALL PLOT (XRLR(I,1)/DXD,Y/DY,2)
      CALL NEWPEN(2)
      CALL SYMBOL(XRLR(I,1)/DXD,Y/DY,2.,ICODE,0.,-1)
      CALL NEWPEN(1)
      ISW1=1
 9910 CONTINUE
      IF(ISW2.EQ.1) GO TO 9920
      IF(X2.LT.XRLR(I,2)) GO TO 9920
      Y = XCPU(I,J) + (XCPU(I,J-1)-XCPU(I,J))*(X2-XRLR(I,2))/(X2-X1)
      CALL PLOT (XRLR(I,2)/DXD,Y/DY,2)
      CALL NEWPEN(3)
      CALL SYMBOL(XRLR(I,2)/DXD,Y/DY,2.,ICODE,0.,-1)
      CALL NEWPEN(1)
      ISW2=1
 9920 CONTINUE
      CALL PLOT (X2/DXD,XCPU(I,J)/DY,2)
 9900 CONTINUE
      CALL SYMBOL (X2/DXD,XCPU(I,NP)/DY,2.,ICODE,0.,-1)
 9800 CONTINUE
      CALL NEWPEN(1)
      CALL SYMBOL (320.,200.,4.,'PLUTONIUM CONCENTRATION',0.,23)
      CALL AXIAL (320.,193.,IPFUL)
      CALL NEWPEN(2)
      CALL SYMBOL(320.,180.,4.,'COLUMNAR BNDRY',0.,14)
      CALL NEWPEN(3)
      CALL SYMBOL(320.,170.,4.,'EQUIAXED BNDRY',0.,14)
      CALL NEWPEN(1)
      DO 10000 I=1,IT
      IF(I.GE.15) GO TO 10010
      ITX = ITP( I )
      ICODE = I-1
      YY=160.-5.*(I-1)
      CALL SYMBOL(320.,YY,4.0,ICODE,0.,-1)
      CALL SYMBOL(323.,YY-2.0,4.0,' --- STEP',0.,9)
      CALL NUMBER(360.,YY-2.0,4.,FLOAT(ITX),0.,-1)
10000 CONTINUE
      GO TO 10020
10010 CONTINUE
      CALL SYMBOL (369.,93.,4.,30,0.,-1)
```

```
C       CALL NUMBER (374.,93.,4.,FLOAT(IT),0.,-1)                       00110300
10020 CONTINUE                                                          00110400
C                                                                       00110500
        CALL PLOT(600.,0.,-3)                                           00110600
        RETURN                                                          00110700
        END                                                             00110800
```

```
      SUBROUTINE PLUTO ( CP,RVB,RCB,REB,RP,TP,GDT,XM,RH,EPZ,DELT,D )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *          N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /IOFIL/ I05,I06,I07
C
      DIMENSION  CP(NN,NP) , RVB(NN) , RCB(NN) , REB(NN) , RP(NN,NP1) ,
     *           TP(NN,NP1) , GDT(NN,NP1) , XM(NN,3) , RH(NN,NTR) ,
     *           EPZ(NN,NTR) , D(NP)
C
C
C     PLUTONIUM MIGRATION CALC.
C
C
      DO 1000 N=1,NNF
C
      ISW = 0
      S = DELT*3600.
      CX = XM(N,1)+XM(N,2)+XM(N,3)
  900 CONTINUE
      I = 3
      B = 0.
      X = 0.0
C
      DO 1100 J=1,NP
C
      K = NP-J+1
      R = RP(N,K)
      T = TP(N,K)
C
  600 IF(I-2) 100,200,300
C
  100 RR = RVB(N)
      II = 1
      GO TO 400
C
  200 RR = RCB(N)
      II = NCR+1
      GO TO 400
C
  300 RR = REB(N)
      II = NCR+NER+1
C
  400 CONTINUE
C
      IF(R-RR) 450,500,500
  450 I=I-1
      IF ( I.EQ.0 ) GO TO 500
      GO TO 600
  500 CONTINUE
C
```

```
      XY = -1.E+05/(1.9865*T)
C     U = 1.+EPZ(N,II)
      U = 1.0
      H = CX/U/NP
      IF( XY+60. .LT. 0.  .OR.  K .EQ. 1 ) GO TO 700
C
      V = CP(N,K)-CP(N,K-1)
      W = (1.-(CP(N,K)+CP(N,K-1))/2.)*(CP(N,K)+CP(N,K-1))/2.
      IF( V.GT.W ) ISW=1
      A = PI2*RP(N,K)*U*0.046*EXP( XY )*( W*11.0*GDT(N,K)/T -
     *    2.*V/((RP(N,K+1)-RP(N,K-1)) ) ) * RH(N,II)
      GO TO 800
C
  700 A = 0.
C
  800 D(K) = ( A-B ) / H
C
      Y = ABS( D(K)/CP(N,K) )
      IF(Y.GT.X) X=Y
      B = A
C
 1100 CONTINUE
C
      IF( X ) 950,950,960
  950 DO 1200 J=1,NP
      CP(N,J) = CP(N,J) + D(J)*S
 1200 CONTINUE
      GO TO 1400
  960 P = 0.01/X
      IF(P.GT.S) GO TO 950
      DO 1300 J=1,NP
      CP(N,J) = CP(N,J) + D(J)*P
 1300 CONTINUE
      S = S-P
      GO TO 900
C
 1400 CONTINUE
C
      IF( ISW.EQ.0 ) GO TO 1000
      WRITE(IO6,9000) N,( CP (N,J),J=1,NP )
 9000 FORMAT(1H1,'----- SUBROUTINE PLUTO ----- MASS INCREMENT TOO BIG --
     *---',/1H0,I3,/(1H0,10E12.4))
 1000 CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE POWCUT ( TIM,TIP,TIN,PTIN,TOUT,PTOUT,REPOW,PPOWA,PTOUR,
     *  TOUTER,DELTOU,DELTIM,DELIN,DELOUT,DELPOW,KUT,NSK,KUTCNT )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C
      DIMENSION  TOUTER(NN) , DELTOU(NN) , PTOUR(NN)
C
C
      IF (KUTCNT.GT.1) GO TO 10
      DELTIM = ( TIM-TIP ) / KUT
      DELIN = ( TIN-PTIN ) / KUT
      DELPOW = ( REPOW-PPOWA ) / KUT
      IF(NSK.GT.0) GO TO 100
      DELOUT = ( TOUT-PTOUT ) / KUT
      GO TO 200
  100 DO 150 I=1,NN
  150 DELTOU(I) = ( TOUTER(I)-PTOUR(I) ) / KUT
  200 CONTINUE
C
C
C
   10 CONTINUE
      TIM = TIP + DELTIM
      TIN = PTIN + DELIN
      IF(NSK.GT.0) GO TO 300
      TOUT = PTOUT + DELOUT
      GO TO 400
  300 TOUT = PTOUR(NN) + DELTOU(NN)
  400 CONTINUE
      REPOW = PPOWA + DELPOW
      RETURN
      END
```

```
      SUBROUTINE PROPTY ( CPU,RHOP,RHOT )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /CONST/ FKA1,FKB1,FKC1,FKA2,FKB2,FKC2,FKA3,FKB3,FKC3,
     *               FA11,FA12,FA21,FA22,FA31,FA32,
     *               CK1,CK2,CA1,CA2,XEWTON(7)
      COMMON /XCRAK/ THLC,HEAL1,HEAL2,SIGFO,SIGF1
      COMMON /XCREC/ CCA,QCC,AE,BE,CE,FMC,FNC
      COMMON /XCREF/ CFAO,CFA1,CFR,CFB,QCF,QCF1,FMF,FNF,GDF
      COMMON /XFRIC/ FRICST,FRICDY
      COMMON /XGASO/ AF,AN,CT,QRG
      COMMON /XGOLD/ ASSO,ASS1,ANAO,ANA1,ANA2,ACO,AC1,AC2
      COMMON /XHKNS/ HCS,HBO,HCO,HC1,HC2,HC3
      COMMON /XITER/ CI,QRK
      COMMON /XSWEL/ SVPF
      COMMON /XXLAM/ EO,EP,TLG,ONU,ONUP,EA1,EA2,EA3,EA4
      COMMON /FBRATR/ JFA
C
      DIMENSION  CPU(NN,NP) , RHOP(NN,3) , RHOT(NN)
C
C
C     MATERIAL PROPERTIES , AND CONSTANTS SET UP
C
C
C
C
C     /OTHERS/
C
C
      PI = 3.1415927
      PI2 = 6.2831853
      DP = 1.
      GRAV = 1.E-03
      CFLX = 1.041E+14 * 12.*2.54 / 1000.
C     CFLX NOW IN (N/CM**2) / (W/CM)
      BUCNS=(3.1E+10)*3600./(2.246E+19)
C
C     BURNUP IS 1. AFTER 2.*E+19 FISSIONS PER GRAM OF FUEL.
C
      A91 = 1.116E+14
C
C     UNIT = 3.1*10**10FISSIONS/WATT/SEC*3600SEC/HR = 11.16E+13 FISSIONS
C     /WATT/HR
C
      RK = 8.31436*1.0197*10.
      RKR = 8.31436/1.8/4.186
C
C     R = 8.31436 JOULE/MOLE/DEG K  ,  1 JOULE = 1.0197*10. KG*CM
C     RK IS IDEAL GAS CONSTANT IN KG*CM/MOLE/DEG K
C     RKR IS IDEAL GAS CONSTANT IN CAL/MOLE/DEG R
```

```
C
C
C     CONVERSION TO KG/CM**2 FROM PSI
C
C
      DO 500 I=1,3
      PG(I) = PG(I) * 7.03E-02
  500 CONTINUE
C     PG NOW IN KG/CM**2
C
C     /CONST/
C
C
      IF(JFA-2) 470,480,480
  470 CONTINUE
      FA11 = 8.143E-06
      FA12 = 2.430E-09
      FA21 = 8.143E-06
      FA22 = 2.430E-09
      FA31 = 8.143E-06
      FA32 = 2.430E-09
      GO TO 490
  480 FA11 = 9.61E-06
      FA12 = 1.57E-09
      FA21 = 9.61E-06
      FA22 = 1.57E-09
      FA31 = 9.61E-06
      FA32 = 1.57E-09
  490 CONTINUE
C
      IF(JFA-2) 510,520,520
  510 CK1 = 0.130
      CK2 = 1.36E-04
      CA1 = 17.0E-06
      CA2 = 4.25E-09
      GO TO 530
  520 CK1 = 0.121
      CK2 = 1.24E-04
      CA1 = 5.675E-06
      CA2 = 1.7E-09
  530 CONTINUE
C
C     RHOT FOR UO2 = 10.96 G/CC FOR U-PUO2 = 11.06 (20WT.PC)   ROBERTS
C
      DO 1000 N=1,NNF
      DO 1000 I=2,NP
      CPU(N,I) = CPU(N,1)
 1000 CONTINUE
C
      DO 1500 N=1,NNF
      RHOT(N) = 10.96 + 0.5*CPU(N,1)
      RHOP(N,1) = 0.767*RHOP(N,3) + 0.233*RHOT(N)
      RHOP(N,2) = RHOP(N,3)
 1500 CONTINUE
      RHOT(NN) = 0.
C
C
C
```

```
C
C      /XCRAK/                  . ==  SUBROUTINE CRACK   ==
C
C
       THLC = 1400.
       HEAL1 = -7000.
       HEAL2 = -70.
       SIGF0 = 1000.
       SIGF1 = 0.3
C
C
C
C
C
C      /XCREC/                    ==  SUBROUTINE  CREEC  ==
C
C
       CCA = 2.7E-11
       QCC = 95000.
       AE = 1.0E-08
       BE = 2.2E+20
       CE = 4.3E-30
       FMC = 7.0
       FNC = 1.0
C
C
C
C      /XCREF/                    ==  SUBROUTINE  CREEF  ==
C
C
       CFA0 = 1.83E-03
C      CFA1 = 2.82E+07
       CFA1 = 1.07E+06
       CFR = 3.0E+12
C      CFR IN FISSIONS/SEC/CC
       CFB = 6.0E-23
       QCF = 140000.
       QCF1 = 110000.
       FMF = 4.5
       FNF = 1.0
       GDF = 10.0
C
C
C      /XGASO/                    ==  SUBROUTINE  GASOUT  ==
C
C
       AF = 0.246
C
C      AF IS ATOMS OF GAS PER FISSION
C
       AN = 6.02E+23
C
C      AN IS AVOGADRO'S NUMBER
C
       CT = 4.48
       QRG = 8000.
C
C
```

```
C       /XFRIC/                    ==  SUBROUTINE  FRICHK  ==
C
C
        FRICST = 0.8
C       FRICDY = 0.6
        FRICDY = 0.
C
C
C       /XGOLD/                    ==  SUBROUTINE  GOLDN  ==
C
C
        ASSO = 9.18 * 252. / (12.*2.54)
        ASS1=4.214E-3 * 252. / (12.*2.54)
        ANA0=54.306 * 252. / (12.*2.54)
        ANA1=-0.01878 * 252. / (12.*2.54)
        ANA2=2.0914E-6 * 252. / (12.*2.54)
        AC0 = 0.345696 * 252. / 0.4536
        AC1 = -0.79211E-04 * 252. / 0.4536
        AC2 = 3.41178E-08 * 252. / 0.4536
C
C
C
C
C       /XHKNS/                    ==  SUBROUTINE  HKNS  ==
C
C
        HCS = 10.0
        HB0 = 9.0E-35
        HC0 = 4.028
        HC1 = -3.712E-02
        HC2 = 1.0145E-04
        HC3 = -7.879E-08
C
C
C       /XITER/                    ==  SUBROUTINE  ITERAT  ==
C
C
        CI = 11.13E+06
C       CI NOW IN DEG K *CM**2/HR/KG  ,  1 KG = 9.8E+05 DYNE
        QRK = 49600.
C
C
C       /XXLAM/                    ==  SUBROUTINE  XLAME  ==
C
C
        E0 = 3.59E+07 * 7.03E-02
        EP = -2.35
        TLG = 22200.
        ONU = 0.317
        ONUP = -0.46
C
        IF(JFA-2) 1910,1920,1920
  1910  CONTINUE
        EA1 = 2.0793E+06
        EA2=-8.8592E+02
        EA3=0.3015
        EA4=0.850E-04
        GO TO 1930
```

```
 1920 EA1=9.898E+05                                                      00042700
      EA2 = -9.265E+02                                                   00042800
      EA3 = 0.325                                                        00042900
      EA4 = 2.14E-04                                                     00043000
 1930 CONTINUE                                                           00043100
C                                                                        00043200
C     /XSWEL/                  ==  SUBROUTINE  SWELL  ==                 00043300
C                                                                        00043400
C                                                                        00043500
      SVPF = 0.70E-24                                                    00043600
C                                                                        00043700
C     NET SOLID VOLUME CREATED PER FISSION ...                           00043800
C                                                                        00043900
C                                                                        00044000
C                                                                        00044100
      RETURN                                                             00044200
      END                                                                00044300
```

```
      SUBROUTINE RHOMOD ( RH,RHOP,EPZ,RVB,RCB,REB,RUB,BU,XM,RHTHEO,RDIS,
     *                    DXM,RVBO,RCBO,REBO,RUBO,RCI,RCO,XMINT,EPC,
     *                    RDIST,EPR,XEPR,XEPC,R,WOLD,WNEW,IT,DELT )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALF0(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /IOFIL/ I05,I06,I07
      DIMENSION RH(NN,NTR) , RHOP(NN, 3 ) , EPZ(NN,NTR) ,
     *          RVB(NN) , RCB(NN) , REB(NN) , RUB(NN) , BU(NN) ,
     *          XM(NN,3) , RHTHEO(NN) , RDIS(NN,NTR2) , DXM(NN,3) ,
     *          RVBO(NN) , RCBO(NN) , REBO(NN) , RUBO(NN) ,
     *          RCI(NN) , RCO(NN) , XMINT(NN) , EPC(NN,NTR) ,
     *          RDIST(NN,4) , EPR(NN,NTR)
      DIMENSION  R(NN,NFR1) , XEPR(NFR) , XEPC(NFR) , WNEW(NFR) ,
     *           WOLD(NN,NFR)
C
C
C     DETERMINATION OF EACH ZONE BOUNDARY
C
C
      DO 1000 N=1,NNF
C
      DXM(N,1) = DXM(N,1) * DELT * 3600.
      DXM(N,2) = DXM(N,2) * DELT * 3600.
      DXM(N,3) = DXM(N,3) * DELT * 3600.
      RDIST(N,3) = RDIST(N,3) + DXM(N,3)
      RDIST(N,2) = RDIST(N,2) + DXM(N,2)
      RVB(N) = RVB(N) + RDIS(N,1)
      RCB(N) = RCB (N) + RDIS(N,NCR+1) + DXM(N,2)
      REB(N) = REB (N) + RDIS(N,NCR+NER+1) + DXM(N,3)
      IF(RVB(N).LT.RVBO(N)) RVB(N)=RVBO(N)
      IF(RCB(N).LT.RCBO(N)) RCB(N)=RCBO(N)
      IF(REB(N).LT.REBO(N)) REB(N)=REBO(N)
      RUB(N) = RUB (N) + RDIS(N,NFR+1)
      RCI(N) = RCI (N) + RDIS(N,NFR+2)
      RCO(N) = RCO (N) + RDIS(N,NTR+2)
      IF(REB(N).GE.RUB(N)) REB(N) = RUB(N)-0.001
      IF(RCB(N).GE.REB(N)) RCB(N) = REB(N)-0.001
      RHOT = RHTHEO(N)
      SM = XMINT(N) / ( 1.+EPZ(N,1) )
C
      DO 1100 I=1,3
C
      IF(I-2) 100,200,300
  100 KK=1
      KKK=NCR
      RI=RVB(N)
      RO = RCB(N)
      RIO = RVBO(N)
      ROO = RCBO(N)
      COF = 1.
```

```
      RAT = 0.99
      BUR = BURH1
      GO TO 400
  200 KK=NCR+1
      KKK=NCR+NER
      RI=RCB(N)
      RO = REB(N)
      RIO = RCBO(N)
      ROO = REBO(N)
      COF = 1.
      RAT = 0.965
      BUR = BURH2
      GO TO 400
  300 KK=NCR+NER+1
      KKK=NFR
      RI=REB(N)
      RO = RUB(N)
      RIO = REBO(N)
      ROO = RUBO(N)
      COF = 0.
      RAT = 1.
      BUR = 1.0
  400 CONTINUE
C
      ERAV = 0.
      ECAV = 0.
      IF(IT.LE.1) GO TO 501
      DO 500 K=KK,KKK
      ERAV = ERAV+EPR(N,K)
      ECAV = ECAV+EPC(N,K)
  500 CONTINUE
      ERAV = ERAV/(KKK-KK+1)
      ECAV = ECAV/(KKK-KK+1)
  501 CONTINUE
      CX = (1.+ERAV)*(1.+ECAV)
C     CX = 1.+ERAV+ECAV+EPZ(N,1)
C
C
C     BURN-UP DEPENDENT DENSITY CALC.  AND  AXIAL MODIFICATION
C
C
      DO 1200 K=KK,KKK
      RH(N,K) = RHOP(N,I) + COF*(RAT*RHOT-RHOP(N,I)) * BU(N)/BUR
      RH(N,K) = RH(N,K) / ( (1.+EPZ(N,K)) * CX )
C     RH(N,K) = RH(N,K) / CX
 1200 CONTINUE
C
      IF ( I.EQ.1 ) GO TO 1100
      XM(N,I) = PI*(RO+RI)*(RO-RI)*RH(N,KK)
C
 1100 CONTINUE
      RDIST(N,4) = 1.+EPZ(N,1)
      XM(N,1) = SM-XM(N,2)-XM(N,3)
C
C     REGION BOUNDARY UPDATE
C
C      ****
      IF(XM(N,1).GE.0.0001) GO TO 1300
```

```
      XM(N,1) = 0.0001
 1300 XAR = RCB(N)*RCB(N)-XM(N,1)/(PI*RH(N,1))
      IF(XAR.LE.0.) XAR = 0.0
      RVB(N) = SQRT(XAR)
      IF(RVB(N).GE.RVBO(N)) GO TO 1400
      RVB(N) = RVBO(N)
 1400 XM(N,1) = PI*RH(N,1)*(RCB(N)*RCB(N)-RVB(N)*RVB(N))
      IF(XM(N,1).GT.0.) GO TO 1405
      WRITE(IO6,9000) N
      STOP
 1405 CONTINUE
C
      SMD = XM(N,1)+XM(N,2)+XM(N,3)
C
      DO 1500 I=1,3
      IF(I-2) 1410,1420,1430
 1410 KK = 1
      KKK = NCR
      GO TO 1440
 1420 KK = NCR+1
      KKK = NCR+NER
      GO TO 1440
 1430 KK = NCR+NER+1
      KKK = NFR
 1440 CONTINUE
      DO 1450 K=KK,KKK
      RH(N,K) = RH(N,K)*SMD/SM
 1450 CONTINUE
C
      XM(N,I) = XM(N,I)*SMD/SM
C
 1500 CONTINUE
C      ****
C
      RDIST(N,4) = SM
 1000 CONTINUE
C
C     PLENUM
C
      RCI(NN) = RCI (NN) + RDIS(NN,NFR+2)
      RCO(NN) = RCO (NN) + RDIS(NN,NTR+2)
C
      RETURN
C
 9000 FORMAT(1H1,'----- SUBROUTINE RHOMOD -----' , / ,
     * 1H0,'MASS IN COLUMNAR ZONE OF AXIAL SECTION ',I2,'  IS NEGATIVE')
C
      END
```

```
      SUBROUTINE SUBTIT (TITLE)                                         00060800
      DIMENSION TITLE(1)                                                00060900
      CALL SYMBOL (3.0,222.0,3.0,TITLE,0.,80)                           00061000
      CALL SYMBOL (250.,222.,5.,'ACTIVE-II',0.,9)                       00061100
      RETURN                                                            00061200
      END                                                               00061300
```

```
      SUBROUTINE SWELL ( POW,SWS,FD,SWS3,DELT )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQ0,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XSWEL/ SVPF
      DIMENSION POW(NN) , SWS(NN,NTR) , FD(NN,NFR) , SWS3(NN,NUR)
C
C
C     FUEL SWELLING STRAIN DUE TO SOLID FISSION PRODUCTS
C
C
C     SVPF = NET SOLID VOLUME CREATED PER FISSION
C
      DO 1000 N=1,NNF
      DFN = A91*POW(N)*DELT
C
C     DFN IS THE NUMBER OF FISSIONS IN THE CROSS SECTION OF FUEL ROD
C     OF ONE CM IN HEIGHT
C
      J = 1
      DO 1100 I=1,NFR
      DSWS = SVPF*DFN*FD(N,I)*2.54*2.54*2.54
      IF(I.LE.NCR+NER) GO TO 100
      SWS3(N,J) = SWS3(N,J) + DSWS
      J = J+1
      GO TO 1100
  100 CONTINUE
      SWS(N,I) = SWS(N,I) + DSWS
 1100 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TIMCUT(CX,DELTO,EPSPOW)                                00120600
      CX = 2.                                                           00120700
      DELTO = 240.0                                                     00120800
      EPSPOW = 50.0                                                     00120900
      RETURN                                                            00121000
      END                                                               00121100
```

```
      SUBROUTINE UPDATE ( SIGR,SIGC,SIGZ,SIGRO,SIGCO,SIGZO,EPR,EPC,
     *                    EPZF,EPZC,EPRO,EPCO,EPZO,TPR,TPC,TPZ,TPRO,
     *                    TPCO,TPZO,DSR,DSC,DSZ,TSRO,TSCO,TSZO,DISP,
     *                    RDIS,N,EPSEQ )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C
      DIMENSION  SIGR(NTR) , SIGC(NTR) , SIGZ(NTR) , SIGRO(NN,NTR) ,
     *           SIGCO(NN,NTR) , SIGZO(NN,NTR) , EPR(NTR) , EPC(NTR) ,
     *           EPRO(NN,NTR) , EPCO(NN,NTR) , EPZO(NN,NTR) , TPR(NTR) ,
     *           TPC(NTR) , TPZ(NTR) , TPRO(NN,NTR) , TPCO(NN,NTR) ,
     *           TPZO(NN,NTR) , DSR(NTR) , DSC(NTR) , DSZ(NTR) ,
     *           TSRO(NN,NTR) , TSCO(NN,NTR) , TSZO(NN,NTR) ,
     *           DISP(NTR2) , RDIS(NN,NTR2) , EPSEQ(NN,NTR)
C
C
C
C     UPDATE SOME VALUES FOR NEXT TIME STEP
C
C
C
      DO 1000 I=1,NTR
      SIGRO(N,I) = SIGR(I)
      SIGCO(N,I) = SIGC(I)
      SIGZO(N,I) = SIGZ(I)
      EPRO(N,I) = EPR(I)
      EPCO(N,I) = EPC(I)
C
      XXX = (EPR(I)+EPC(I)+EPZF)/3.
      X1  = TPR(I) - TPRO(N,I)
      XX1 = EPR(I)-TPRO(N,I)-XXX
      TPRO(N,I) = TPR(I)
      TPR(I) = X1
      X2  = TPC(I) - TPCO(N,I)
      XX2 = EPC(I)-TPCO(N,I)-XXX
      TPCO(N,I) = TPC(I)
      TPC(I) = X2
      X3  = TPZ(I) - TPZO(N,I)
      XX3 = EPZF-TPZO(N,I)-XXX
      TPZO(N,I) = TPZ(I)
      TPZ(I) = X3
      XXX = 2.*((XX1-XX2)**2+(XX2-XX3)**2+(XX3-XX1)**2)
      EPSEQ(N,I) = SQRT( XXX )/3.0
C
      IF ( I-NFR ) 100,100,200
  100 EPZO(N,I) = EPZF
      GO TO 1000
  200 EPZO(N,I) = EPZC
C
 1000 CONTINUE
C
C
      DO 2000 I=1,NTR2
```

```
      RDIS(N,I) = DISP( I )
 2000 CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE UPDATP ( SIGR,SIGC,SIGZ,SIGRO,SIGCO,SIGZO,EPR,EPC,
     *                    EPZC,EPRO,EPCO,EPZO,TPR,TPC,TPZ,TPRO,TPCO,
     *                    TPZO,TSR,TSC,TSZ,TSRO,TSCO,TSZO,DISP,RDIS,N )
C
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C
      DIMENSION  SIGR(NCL) , SIGC(NCL) , SIGZ(NCL) , SIGRO(NN,NTR) ,
     *           SIGCO(NN,NTR) , SIGZO(NN,NTR) , EPR(NCL) , EPC(NCL) ,
     *           EPRO(NN,NTR) , EPCO(NN,NTR) , EPZO(NN,NTR) , TPR(NCL) ,
     *           TPC(NCL) , TPZ(NCL) , TPRO(NN,NTR) , TPCO(NN,NTR) ,
     *           TPZO(NN,NTR) , TSR(NCL) , TSC(NCL) , TSZ(NCL) ,
     *           TSRO(NN,NTR) , TSCO(NN,NTR) , TSZO(NN,NTR) ,
     *           DISP(NCL1), RDIS(NN,NTR2)
C
C
C
C     UPDATE SOME VALUES FOR NEXT TIME STEP  ---  PLENUM SECTION ONLY
C
C
C     TPR,TPC,TPZ,TSR,TSC,TSZ ARE USED TEMPORARY FOR INCREMENT OF
C     STRAIN RESPECTIVELY
C
C
      DO 1000 I=1,NCL
      K = NFR+I
      SIGRO(N,K) = SIGR(I)
      SIGCO(N,K) = SIGC(I)
      SIGZO(N,K) = SIGZ(I)
      EPRO(N,K) = EPR(I)
      EPCO(N,K) = EPC(I)
      XXX = TPR(I) - TPRO(N,K)
      TPRO(N,K) = TPR(I)
      TPR(I) = XXX
      XXX = TPC(I) - TPCO(N,K)
      TPCO(N,K) = TPC(I)
      TPC(I) = XXX
      XXX = TPZ(I) - TPZO(N,K)
      TPZO(N,K) = TPZ(I)
      TPZ(I) = XXX
      EPZO(N,K) = EPZC
      RDIS(N,K+1) = DISP( I )
C
 1000 CONTINUE
C
      RDIS(N,NTR2) = DISP( NCL1 )
C
      RETURN
      END
```

```
      SUBROUTINE UPDTE ( RVB,RCB,REB,RUB,RCI,RCO,BU,RDIST,DXM,
     *                   FLNC,WOLD,YRVB,YRCB,YREB,YRUB,YRCI,YRCO,
     *                   YBU,YRDIST,YDXM,YFLNC,YWOLD,ISW )
      DIMENSION  RVB(NN),RCB(NN),REB(NN),RUB(NN),RCI(NN),RCO(NN),
     *           BU(NN),RDIST(NN,4),DXM(NN,3),FLNC(NN),WOLD(NN,NFR),
     *           YRVB(NN),YRCB(NN),YREB(NN),YRUB(NN),YRCI(NN),
     *           YRCO(NN),YBU(NN),YRDIST(NN,4),YDXM(NN,3),YFLNC(NN),
     *           YWOLD(NN,NFR)
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
C
C
      GO TO ( 1,2 ) , ISW
C
    1 CONTINUE
C
      DO 100 N=1,NN
      YRVB(N) = RVB(N)
      YRCB(N) = RCB(N)
      YREB(N) = REB(N)
      YRUB(N) = RUB(N)
      YRCI(N) = RCI(N)
      YRCO(N) = RCO(N)
      YBU(N) = BU(N)
      YFLNC(N) = FLNC(N)
      DO 110 J=1,4
      YRDIST(N,J) = RDIST(N,J)
  110 CONTINUE
      DO 120 J=1,3
      YDXM(N,J) = DXM(N,J)
  120 CONTINUE
      DO 130 J=1,NFR
      YWOLD(N,J) = WOLD(N,J)
  130 CONTINUE
C
  100 CONTINUE
      RETURN
C
    2 CONTINUE
C
      DO 200 N=1,NN
      RVB(N) = YRVB(N)
      RCB(N) = YRCB(N)
      REB(N) = YREB(N)
      RUB(N) = YRUB(N)
      RCI(N) = YRCI(N)
      RCO(N) = YRCO(N)
      BU(N) = YBU(N)
      FLNC(N) = YFLNC(N)
      DO 210 J=1,4
      RDIST(N,J) = YRDIST(N,J)
  210 CONTINUE
      DO 220 J=1,3
      DXM(N,J) = YDXM(N,J)
```

```
  220 CONTINUE
      DO 230 J=1,NFR
      WOLD(N,J) = YWOLD(N,J)
  230 CONTINUE
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE VOLUME ( VOL,RFR,RCL,FLZ,CLZ )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      DIMENSION  VOL(NN,NTR) , RFR(NN,NFR1) , RCL(NN,NCL1) ,
     *           FLZ(NN) , CLZ(NN)
C
C
C     VOLUME CALC.
C
C
      DO 1000 N=1,NNF
C
C     FUEL
C
      DO 1100 I=1,NFR
      VOL(N,I) = PI*(RFR(N,I+1)+RFR(N,I))*(RFR(N,I+1)-RFR(N,I))*FLZ(N)
 1100 CONTINUE
C
C     CLAD
C
      DO 1200 I=1,NCL
      VOL(N,NFR+I) = PI*(RCL(N,I+1)+RCL(N,I))*(RCL(N,I+1)-RCL(N,I)) *
     *                   * CLZ(N)
 1200 CONTINUE
C
C
 1000 CONTINUE
C
C
C     PLENUM
C
      DO 2000 I=1,NCL
      VOL(NN,NFR+I) = PI*(RCL(NN,I+1)+RCL(NN,I))*(RCL(NN,I+1)-RCL(NN,I))
     *                 *CLZ(NN)
 2000 CONTINUE
C
C
      RETURN
      END
```

```
      SUBROUTINE XLAME( TBC,TBF,E,P,G,RH,LM,BM,PB,GB,VOL,RHTHEO,N )
      COMMON /ALWAYS/ NN,NCR,NER,NUR,NCL,NP,NOK,NOS,NOC,NPLOT,NCRK,
     *                NTR,NFR,NTR1,NTR2,NNTR2,NNF,NCL1,NFR1,NP1,
     *                IGEN,IGEN1,NONZR,IGENP,IGENP1,NONZRP,
     *                N1C,N2C,N3C,N4C,N5C,N6C,N7C,N8C,N9C,N10C,N11C,
     *                N12C,N13C,N14C,N15C,
     *           N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /OTHERS/ PG(3),CM1(2),DHR(2),ALFO(3),ALF1(3),XML(3),CM(2),
     *                PI,PI2,DP,GRAV,SMQ,SMQO,RHOC,BURH1,BURH2,CFLX,
     *                CPOW,BUCNS,RK,RKR,A91,GASI,TCON,PDBU,BUB
      COMMON /XXLAM/ EO,EP,TLG,CNU,ONUP,A1,A2,A3,A4
      COMMON /FBRATR/ JFA
      COMMON /IOFIL/ IO5,IO6,IO7
      DIMENSION TBC(NN,NCL) , TBF(NN,NFR) , E(NN,NTR) , P(NN,NTR) ,
     *           G(NN,NTR) , RH(NN,NTR) , LM(NN,NTR) , BM(NN,NTR) ,
     *           PB(NN,2) , GB(NN,2) , VOL(NN,NTR) , RHTHEO(NN)
C
C
C     YOUNG'S MODULUS,POISSON'S RATIO,SHEAR MODULUS , AND BULK MODULUS
C
C
C
C     GIVEN VALUES OF EP,ONU,ONUP PERTAIN TO UO2 . GIVEN EO AND TLG
C     REFER TO (U.8 PU.2)O2 .O. FOR UO2 EO=3.16 AND TLG=16300.
C
C
C
C     FUEL
C
C
      IF ( N.EQ.NN ) GO TO 500
C
      RHOT = RHTHEO( N )
      J = 0
      K = 1
      KK = NCR
C
      SP = 0.
      SG = 0.
      SV = 0.
C
C
  400 CONTINUE
C
      DO 1100 I=K,KK
      LML = LM(N,I)
      T = TBF(N,I)*1.8 + 32.0
      IF(T-TLG+100. .GT. 0.) T=TLG-100.
      PS = 1.-RH(N,I)/RHOT
      EX = EO*(1.+EP*PS)*(1.-T/TLG)
      PX = ONU*(1.+ONUP*PS)
      P(N,I) = (PX*(TLG-T)+0.5*(T-77.)) / (TLG-77.)
      P(N,I) = P(N,I) / (2.)**LML
      E(N,I) = EX / (1.5)**LML
      G(N,I) = E(N,I) / (2.+2.*P(N,I))
      IF(G(N,I).LT.10.0) G(N,I)=10.0
      BM(N,I) = E(N,I) / (3.*(1.-2.*P(N,I)))
C
```

```
      SV = SV + VOL(N,I)
      SP = SP + P(N,I)*VOL(N,I)
      SG = SG + G(N,I)*VOL(N,I)
C
 1100 CONTINUE
      J = J+1
C
      IF(J-2) 100,200,300
  100 K=KK+1
      KK = NCR+NER
      GO TO 400
  200 K=KK+1
      KK = NFR
      GO TO 400
  300 CONTINUE
C                      ,
      PB(N,1) = SP / SV
      GB(N,1) = SG / SV
C
  500 CONTINUE
C
C     CLAD
C
      K=NFR
C
      SP = 0.
      SG = 0.
      SV = 0.
C
      DO 1200 I=1,NCL
      K = K+1
      TC = TBC(N,I)
      P(N,K) = A3+A4*TC
      IF(TC+A1/A2.GT.0) GO TO 1111
      E(N,K) = A1+A2*TC
      G(N,K) = E(N,K)/(2.+2.*P(N,K))
      BM(N,K) = E(N,K) / (3.*(1.-2.*P(N,K)))
C
      SV = SV + VOL(N,K)
      SP = SP + P(N,K)*VOL(N,K)
      SG = SG + G(N,K)*VOL(N,K)
C
 1200 CONTINUE
      PB(N,2) = SP / SV
      GB(N,2) = SG / SV
C
C
C     E . G . BM    NOW IN KG/CM**2
C
C
 1000 CONTINUE
C
      RETURN
 1111 WRITE(IO6,1112) N
 1112 FORMAT(1H1,'----- SUBROUTINE XLAME -----' , / ,
     * 1H0,'CLADDING TEMPERATURE OF AXIAL SECTION ',I2,'  IS TOO HIGH')
      STOP
      END
```