Gapcon-Thermal-3 Code Description

January 1978

**Prepared for the
U.S. Nuclear Regulatory Commission**

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Nuclear Regulatory Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
Under Contract EY-76-C-06-1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

Price: Printed Copy \$____*; Microfiche \$3.00

*Pages	NTIS Selling Price
001-025	\$4.50
026-050	\$5.00
051-075	\$5.50
076-100	\$6.00
101-125	\$6.50
126-150	\$7.00
151-175	\$7.75
176-200	\$8.50
201-225	\$8.75
226-250	\$9.00
251-275	\$10.00
276-300	\$10.25

GAPCON-THERMAL-3 CODE DESCRIPTION

by
D. D. Lanning
C. L. Mohr
F. E. Panisko
K. B. Stewart

January , 1978

BATTELLE
Pacific Northwest Laboratories
Richland, Washington 99352

ACKNOWLEDGEMENT

The authors wish to thank the Core Performance Branch of the Nuclear Regulatory Commission, Division of Systems Safety, for their support and guidance of GAPCON-THERMAL-3 development. We would also like to acknowledge the review comments by W. J. Bailey, K. Granzow and C. R. Hann.

ABSTRACT

- NAME: GAPCON-THERMAL-3
- COMPUTER: CDC CYBER-74 (6600) or CYBER-76 (7600)
- DESCRIPTION OF PROBLEM: GAPCON-3 calculates the thermal and mechanical state of the fuel and cladding in a nuclear fuel rod during normal operation both steady state and operational transients.
- METHOD OF SOLUTION: Radial temperature profile obtained by orthogonal collocation. Stresses and strains obtained by chained iterative radial and axial finite element models. Fission gas release and gap conductance obtained from empirical models. The solution iterates to convergence on the gap conductance for each axial node and subsequently on gas release rate for the fuel rod. Additional iterative loops handle plasticity, creep, and nonlinear structural effects due to gap closure.
- RESTRICTIONS: Modeling is axisymmetric for the thermal and mechanical calculations. As many as 17 variable length axial regions can be modeled with no code limitation on the number of fuel and cladding radial nodes. No code limitations on time steps except no more than 49 time steps greater than or equal to 1 day duration. See section on code limitations in this report.
- TYPICAL RUNNING TIME: 3 to 30 seconds per time step.
- UNUSUAL FEATURES: The finite element chained radial-axial-radial mechanics modeling yields good approaches to total two dimensional finite element calculations without commensurate large computer storage and running time. Program includes variable dimensions to limit core storage to fit each individual problem. On CYBER-74, memory size requirements vary dynamically with each problem. Restart tape and pilot plot tape are included in main code operation as options.
- RELATED PROGRAMS: PLOT3 - Plotting program.

- STATUS: New Program.
- REFERENCES: PNL-2434 (this report) and (PNL-2435).
- MACHINE REQUIREMENTS: CALCOMP hardware and software for PLOT3.
- PROGRAMMING LANGUAGE: CDC Extended FORTRAN 4.
- OPERATING SYSTEM: SCOPE 3.4, NOS/BE, and SCOPE 2.1.
- MATERIAL AVAILABLE: Source deck, PLOT3 deck, control cards for CDC 6600.

CONTENTS

ABSTRACT	iii
I. INTRODUCTION	1
II. SUMMARY	3
III. OVERVIEW OF GAPCON-3 CAPABILITIES	5
III.A HISTORY OF GAPCON SERIES	5
III.B GAPCON-3 CAPABILITIES AND OUTPUT	6
III.C LIMITATIONS ON GAPCON-3 CAPABILITY AND FUNCTION	8
IV. FUNCTIONAL DESCRIPTION OF THE CODE	13
IV.A FLOW LOGIC AND SUBROUTINE DESCRIPTION	13
IV.B GAPCON-3 (MAIN PROGRAM)	19
IV.C THERMAL PERFORMANCE CALCULATIONS (TEMPERATURE AND GAS RELEASE)	20
IV.D THE MECHANICS CALCULATIONS	21
V. DETAILED DESCRIPTION OF MODELS AND CALCULATIONS	25
V.A TEMPERATURE CALCULATIONS	25
V.B MECHANICAL ANALYSIS	35
V.C MODELS FROM GAPCON-2 WHICH HAVE BEEN ALTERED	68
VI. VERIFICATION RUNS	71
VI.A TEMPERATURE CALCULATIONS	71
VI.B STRESS CALCULATIONS IN THE CLADDING	73
VI.C COMPARISON WITH IN-REACTOR DATA	77
APPENDIX A INPUT INSTRUCTIONS	A-1
APPENDIX B INPUT FOR COMPARISON CASES	B-1
APPENDIX C RESTART AND PLOT OPTIONS	C-1
APPENDIX D CODE LISTING AND SAMPLE OUTPUT	D-1
REFERENCES	Ref-1

FIGURES

1	Calling Sequence for GAPCON-3 Subroutines	16
2	Iteration Scheme for Steady-State Cases	17
3	Iteration Scheme for Transient Cases	18
4	Comparison Between Subroutine DDLR and the Finite-Difference Code TRUMP (UCRL-14754, 1972) for a Step Power Decrease	34
5	Basic Axisymmetric Ring Element Used in the GAPCON-3 Stress Analysis Models	37
6	Radial Calculation Model for GAPCON-3 Showing the Minimum Number of Axial Locations Where Radial Calculations Are Made	38
7	The Axial Mechanical Analysis Model Using a Quadrilateral Element Composed of Four Triangular Elements	39
8	The Chained Radial-Axial-Radial Solution Algorithm Used to Couple the Radial and Axial Finite Element Models Through Compatibility of Forces	41
9a	Node and D.O.F. Numbering System for Fuel or Cladding Separate Radial Model	54
9b	Node and D.O.F. Numbering System for Fuel-Cladding Combined Radial Model	54
10	Global Stiffness Matrix for the Four Elements in Figure 9a	55
11	Node and D.O.F. Numbering System for Fuel and Cladding Combined Axial Model	57
12	Comparison of Strain Hardening and Time Hardening Options Available in Subrouting RATE	63
13a	Elements 1, 2, 3 and 4 are Plastic and are Brought Back to Prior Subplastic State	66
13b	Showing the Plastic Strain Taken by Alternating Plastic and Elastic Steps	66
14	Yield Stress Versus Strain used in GAPCON-3 Calculations for Temperature Range 550°F to 700°F	67
15	Normalized Fuel Radial Temperature Profiles	72
16	Fuel Transient Radial Temperature Profiles	74
17	Cladding Response to Pressure Load (Radial Stress)	74
18	Cladding Response to Pressure Load (Hoop Stress)	75
19	Cladding Response to Temperature Gradient (Radial Stress)	76
20	Cladding Response to Temperature Gradient (Hoop Stress)	76
21	Comparison to HPR-80 Thermocouple Data	80

22	Comparison to HPR-80 Thermocouple Data	80
23	Comparison to HPR-107 Thermocouple Data	81
24	Comparison to HPR-107 Elongation Data	82
25	Comparison to Estimated Life History of Maine Yankee Rod	83
26	Comparison to IFA-431 Thermocouple Data	84
27	Comparison to IFA-431 Elongation Data	85
28	Comparison to IFA-431 Thermocouple Data	86
29	Comparison to IFA-431 Rod 3 Transient Data	87
30	Comparison to IFA-432 Rod 1 Transient Data	88

TABLES

1	Description of GAPCON-3 Subroutines	14
2	The PELET Finite Element Mechanics Package	15
3	Simultaneous Set of Steady-State Equations	29
4	Simultaneous Equations for Transient	32
5	Nomenclature	33
6	Effect of High Burnup Gas Release Factor	70
7	End-of-Life Data Comparison	78

I. INTRODUCTION

This report contains a description of the GAPCON-THERMAL-3* computer code which simulates the thermal and mechanical behavior of an operating fuel rod during its normal lifetime. It is a status report and defines the current state of development in predicting nuclear fuel rod behavior.

The ability to predict fuel rod behavior was initially limited by the ability to accurately predict the temperature increase from the cladding inner surface to the fuel periphery. The initial series of GAPCON codes was written to remove this limitation. The earlier versions of GAPCON emphasized the thermal aspects of fuel performance and concentrated on predicting "gap conductance." Gap conductance in a fuel rod, however, is dependent on many other fuel pin parameters, including the mechanical deformation. As a result, a detailed iterative system of calculations was required to compute gap conductance for the range of conditions normally encountered.

GAPCON-3 described herein contains most of the thermal features of the earlier GAPCON versions with the addition of an integral finite element mechanics calculation of fuel and cladding deflection, stresses, and strains.

This report contains a functional description of the subroutines contained in GAPCON-3. A separate description of both the thermal and mechanical packages is presented. Finally, the results of the code predictions are given for some analytical problems and irradiation experiments. Input instructions, sample input, PLOT3 program description and listing, and GAPCON-3 code listing are given in appendices. The user familiar with GAPCON-THERMAL-2 should have little difficulty using GAPCON-3.

* For convenience, references to the program are written as simply "GAPCON-3" throughout this report.

II. SUMMARY

GAPCON-3 is a computer program that predicts the thermal and mechanical behavior of an operating fuel rod during its normal lifetime. The code calculates temperatures, dimensions, stresses, and strains for the fuel and the cladding in both the radial and axial directions for each step of the user specified power history. The method of weighted residuals is for the steady state temperature calculation, and is combined with a finite difference approximation of the time derivative for transient conditions. The stress strain analysis employs an iterative axisymmetric finite element procedure that includes plasticity and creep for normal and pellet-clad mechanical interaction loads.

The code uses two main convergence criteria which are satisfied for each time-power step: first, for every axial region a converged value of the ΔT (temperature difference) across the fuel-to-cladding gap is achieved. Secondly, the quantity of gas contained within the pin is obtained via convergence on the fuel pin gas release rate. Using these convergence criteria, combined with the thermal and mechanical analysis packages, GAPCON-3 can solve steady state and operational transient problems. Comparisons of GAPCON-3 predictions to both closed form analytical solutions and actual in-pile instrumented fuel rod data have demonstrated the ability of the code to calculate fuel rod behavior.

GAPCON-3 features a restart capability and an associated plot package unavailable in previous GAPCON series codes.

III. OVERVIEW OF GAPCON-3 CAPABILITIES

This section states in general terms what calculations and output GAPCON-3 can produce. The limitations of the code are also discussed.

III.A HISTORY OF GAPCON SERIES

The development of the GAPCON codes has evolved in response to the problems facing the Nuclear Regulatory Commission. The goal of GAPCON THERMAL-1⁽¹⁾ and 2⁽²⁾ as well as GAPCON-3⁽³⁾ has been to provide a fast running fuel performance code that can be used to evaluate various assumptions about the fuel. The code itself has both best-estimate (i.e., mean value) and lower bound models for several fuel properties. In addition, new models can easily be substituted to evaluate their effect on the over-all fuel rod performance.

GAPCON THERMAL-1 was the first version specifically set up for the light water reactor use. It was designed for use in parametric studies in which the effect of some key design variable could be evaluated. It provided the capability of looking at thermal axial profiles but was limited to a single power level. The mechanical analysis for interfacial pressure was based upon a strength-of-materials elastic formulation using radial loads. This was sufficient since the initial NRC and licensing evaluations concentrated on thermal performance associated with conditions prior to a postulated loss-of-coolant accident.

GAPCON-2 incorporated many significant changes designed to reduce the error in the previous models used in the thermal calculation of GAPCON THERMAL-1. A very detailed fission gas generation model was added, the power history capability was added, and the code output was changed to include SI units. A significant effort was spent in evaluation of the models used in the code and determining the statistical inferences of the experimental error on the predicted results.

The evolution and understanding of the fuel modeling process showed the dependence of the fuel temperature on the cladding deformation and resulting gap dimensions. The observation of cladding deformation related to cladding failures identified the need for a better mechanical analysis package for the GAPCON code.

Neither GAPCON THERMAL-1 nor 2 were designed to perform a detailed incremental mechanical analysis. GAPCON-3 was developed to incorporate the thermal packages of GAPCON THERMAL-2 with a more detailed non-linear mechanical analysis to evaluate the interactive effects.

The design goal of GAPCON-3 was to provide a fast-running code that would handle creep, plasticity, and axial fuel cladding interaction as well as the kinds of fuel performance considerations handled previously. The analysis procedures were selected to handle normal operating transients involving pellet clad interaction (PCI).

To meet the basic requirements of fast computation time and reduced core storage, the entire set of computational approaches previously used in the GAPCON codes was reviewed. The previously used finite difference procedure for temperature calculation was replaced by a much faster Method of Weighted Residuals procedure (MWR). This procedure was extended to operational transient calculations as well, so that GAPCON-3 will calculate accurate fuel temperatures even for rapid power changes.

GAPCON-3 incorporates a finite element mechanics package that provides an estimate of the general stress state in the fuel and cladding, calculates cladding elastic deflections, and accounts for fuel and cladding creep plasticity. The model currently does not compute the strain at pellet-pellet interfaces or allow fuel-cladding axial slip under friction after contact has occurred. This results in persistently high predictions of axial elongation relative to available data. A later version would remove these deficiencies.

III.B GAPCON-3 CAPABILITIES AND OUTPUT

GAPCON-3 calculates and outputs the same thermal variables that GAPCON THERMAL-2 produced (e.g., fuel and cladding temperatures, gap conductance, fission gas release, and stored energy). In addition, GAPCON-3 calculates and outputs fuel and cladding stresses and strains in radial, hoop, and axial directions. Fuel and cladding axial and radial displacements are output as well.

The code has the capability to write information to a tape at user-specified intervals and to restart from this tape. It can also use the same tape to store additional information for use by an associated plot package. The power history may be input in NAMELIST as with GAPCON-2. But this power history may be modified (e.g., on restart) by cards following the NAMELIST cards. These same cards may also permit the following options:

- time dependence of external system pressure
- time dependence of axial power profile
- axial and time dependence of the cladding to coolant heat transfer coefficient (film coefficient).

Up to 17 axial regions in the fuel are permitted, for which detailed radial information may be output. The axial regions may be of variable length.

The user has the option of inputting fuel and cladding mechanical and thermal properties as in GAPCON-2. Similarly, the option is available for user-supplied flux depression. The code has even better modular structure than GAPCON-2 which facilitates substitution of alternate models.

Run time per time step varies from approximately 3 to 30 seconds on the CDC 6600, depending on the number of axial and radial regions, the occurrence of fuel-cladding contact, and the occurrence of fuel restructuring.

Restart Option

GAPCON-3 is able to predict fuel pin behavior over a detailed power history. Rather than starting such a problem from the beginning each time a run is made, the user has an option to restart the run without redoing previously calculated time steps. This mode of operation is very economical for running all but the shortest problems. By using the start-restart option, the user can study the effects of various transients or power cycles using the same pre-transient initial conditions. The user can elect to have the code put requisite data on tape and then can make a restart run beginning at any particular time step that had been recorded on the tape. Restart runs can begin at any of the various time steps on the tape. The start-restart runs use magnetic

tape for saving values of the various calculated fuel rod parameters. On tape these values can be saved for an indefinite period of time. See Appendix C for a description of the operational details of actually making start and restart runs.

Plotting Option

A separate program, PLOT3, was coded to produce high quality off-line plots of GAPCON-3 output. The output from GAPCON-3 is large in scope and the uses of the code may be diverse. Therefore, PLOT3 attempts to satisfy the user demand in visually displaying pertinent code predictions. PLOT3 is a simple, versatile code to use.

PLOT3 uses data stored on a start tape created with both of the NAMELIST input parameters KSAVE and IPLLOT greater than one.

Two different types of plots can be made. One is axial node or axial location dependent. The second is time or time step dependent. A simple example of the former type would be a plot which shows fuel centerline temperature as a function of axial position. This plot could be produced for one or more particular time steps. An example of the latter type would be a plot which shows, for one particular axial location, the dimension of the cladding outer radius as a function of time. One PLOT3 run can produce as many as seven figures per time step with as many as six curves per figure as well as seven time dependent figures also with as many as six curves per figure. Appendix C describes PLOT3 with sample input and output.

III.C LIMITATIONS ON GAPCON-3 CAPABILITY AND FUNCTION

This section emphasizes limitations on GAPCON-3 capability and use which the user should remember when preparing input and interpreting output.

Limitations Due to Approximations

1. The incremental finite element mechanics solution is not appropriate for strains above 10%. If larger strains, or nonuniform strains, are anticipated, another code should be used.

2. The flux depression calculation internal to the code is not reliable for cases other than solid low-enriched pellets (up to 4% U-235) at beginning-of-life. The diffusion-theory-solution (modified Bessel-function of zero order) used to calculate the flux depression has no neutron spectrum dependence (2200 m/sec cross sections are used to calculate the inverse diffusion length). The function also has no burnup dependence and account is not taken for plutonium isotopes. The user has the option to input a flux depression function, and it is recommended that he do so for problems involving significant flux depression.
3. The thermal conductivity calculation does not include any low-temperature burnup effects. If the problem involves fuel temperatures below $\sim 500^{\circ}\text{C}$, the user is advised to supplement the thermal conductivity calculation via the input options.
4. The code does not account for the following effects:
 - fuel axial gap formation
 - fuel-cladding axial sliding with frictional loading
 - circumferential crack formation
 - aximuthal or axial heat flow due to flux tilt or pellet eccentricity^(a)
 - cladding failure
 - transient gas release
 - localized pellet cladding interaction (clad ridging)
 - molten fuel effects
 - chemical effects on cladding
 - swelling into fuel porosity and dish volumes.

If any of these items are deemed significant, further calculations with another code may be needed.

Limitations in Program Input and Operation

1. The number of time steps is not directly limited; however, the total number of time steps of duration greater than 1 day may not exceed 49. Furthermore, the total of time steps greater than 1 day plus groups of consecutive steps chronologically totaling 1 day may not exceed 49. This limit is a practical one related to core storage requirements.

(a) To assess the possible magnitude of these effects, the user should see Reference 4 and related papers.

2. The size of a power change when fuel and cladding are in contact is limited to 1.5 kW/ft. Larger power steps may be input, but if the code calculates that contact has occurred, it will automatically subdivide the step in the following manner: The input power change will be accounted for by a series of <1.0 kW/ft increments of 15-minute duration, as many as are needed to reach the power level input by the user. These increments are treated as regular time steps; the results of each will be output. The code then accounts for the remaining time in the input step by taking a step of that duration at the user-input power. The total number of these "extra" time steps that are allowed in a case is set by the user (ITMCUT). If ITMCUT is exceeded, the program stops. This limitation is necessary to hold the incremental stresses and strains within bounds where reasonable accuracy can be guaranteed.
3. Time steps less than 5-minutes duration are interpreted as transient. In transient operation, the code automatically subdivides all input power-time steps down to 0.1 seconds, with the power changes calculated by linear interpolation. These subdivided steps are treated much like gap conductance iterations; results from them are not output. These extra steps are limited by ITMCUT. The user should carefully calculate ITMCUT before attempting a transient case. The reason for the 0.1 second interval limit is for guaranteed accuracy of transient temperature calculation. The user can obtain acceptable accuracy with larger time steps for power changes less than $\sim 0.5\%/sec$, but considerable user experience is required in this area.

Limitation in Gap Closure Predictions

The combined effect of 1) stress-free thermal expansion, 2) unrestrained fuel swelling to contact, and 3) the empirical fuel relocation model in the code serves to close the gap in a way that is commensurate with in-reactor temperature data (see Reference 5). However, all three of these models are supported by data that is either not well characterized or highly scattered. Therefore, we feel that the gap closure effects as separate effects are poorly understood. This limits our confidence in the code predictions for medium-to-high burnup problems that vary in any significant way from the experiments

against which we have compared the code. The user is encouraged to study Reference 5 and critically interpret his output if his problems fall outside the range of cases discussed in that reference.

IV. FUNCTIONAL DESCRIPTION OF THE CODE

This section reviews the operation of the code. The subroutines are described briefly and the flow logic is presented in detail. Then the relationship of the subroutines in both the thermal and mechanical calculations is described. Detailed descriptions of the temperature and mechanical calculations are presented in Section V. Detailed descriptions of specific models such as fission gas release and fuel swelling are also to be found in Section V.

IV.A FLOW LOGIC AND SUBROUTINE DESCRIPTION

The GAPCON-3 subroutines are listed and briefly described in Tables 1 and 2. The calling sequence is shown in Figure 1, and the flow charts for both steady-state and transient calculation are shown in Figures 2 and 3, respectively.

The outer convergence loop in steady-state is on gas release rate. This loop does not exist in the transient case because no gas release is permitted during a transient (we do not have a transient gas release model in the program). The next inner loop, the axial node loop, is not a convergence loop but a loop that marches from one axial boundary to the next up the fuel pin. This loop is the same in both transient and steady state calculations.

The innermost loop in steady-state cases is the gap ΔT convergence loop which functions as follows: for a given axial boundary an initial ΔT_{gap} is assumed on the first iteration. This is based on current power and the previous h_{gap} for that boundary; $h_{\text{gap}}^{\text{assumed}} = 5.675 \times 10^{+3} \text{ W/m}^2 - \text{C}$ for the first time step. The $\Delta T_{\text{gap}}^{\text{assumed}}$ is used to calculate a radial temperature profile for the fuel, and then the fuel thermal expansion is calculated. The cladding elastic deflection and changes in fuel radius due to relocation densification and swelling are combined with the fuel thermal expansion to calculate a new gap size. Then a new h_{gap} and ΔT_{gap} are calculated. The program then uses this new ΔT_{gap} to get a new fuel temperature profile and continues to iterate to convergence on ΔT_{gap} .

TABLE 1. Description of GAPCON-3 Subroutines

GAPCON-3	- Calls INPUT and calculates pointers for variable dimensions
INPT	- Reads the NAMELIST input and sets cladding and fuel properties.
GAPCON	- Drives the thermal calculation by calculating the gap conductance in the same manner as in GAPCON-THERMAL-2. Subroutine GAPCON also contains the decision-making on the occurrence of fuel-clad contact.
INPOUT	- Writes out the main input variables
FLUXD	- Calculates flux depression (using I_0 Bessel function approximation)
POWDIS	- Calculates the powers at each axial boundary
FISGAS	- Calculates fission gas production, based on input gas fraction
TEMP1	- Assigns coolant temperatures to the upper and lower boundaries of the rod
DENSF	- Calculates fuel densification (Meyer model)
RELOC	- Calculates fuel relocation (GAPCON-2 model)
TEMP2	- Calculates radial boundary temperatures at each axial boundary
DDLRL	- Calculates coefficients for power series expansion of fuel and clad temperatures
SIMQ	- solves simultaneous linear equations
HCAP	- returns fuel heat capacity and
	$\int_{T_{ref}}^T C_d T' \text{ for the fuel}$
	where $T_{ref} = 21^\circ\text{C}$
TCOR, TCW1	- returns thermal conductivity for UO_2 or mixed oxide fuel, respectively
EXPAND	- Calculates fuel expansion (free thermal expansion used in GAPCON-2 model) prior to clad contact
SWELL	- Calculates fission product swelling (unrestrained swelling to contact)
FULEX	- Calculates the linear expansion ($\Delta L/L$) for the fuel at a given temperature, starting from a reference temperature of 21°C
GASREL	- Calculates gas release (Beyer-Hann model)
TEMP3	- Calculates element temperatures for PELET
PELET	- Calculates clad deflections and fuel/clad stresses and strains
OUTPUT	- Output results

TABLE 2. The PELET Finite Element Mechanics Package

INPUT	- Sets up node numbering and degree-of-freedom array for the radial model (called once only)
INPAXL	- Sets up node numbering and d.o.f. array for axial model (called once only)
STIFF	- Sets up the stiffness matrix for radial problem
BOUND	- Puts boundary conditions into stiffness matrix for radial model
SOLVET	- Solves load-displacement equations for radial displacements
LOAD	- Calculates load vector for radial problem
DMCHB	- Solves linear equations with a banded coefficient matrix
STRESS	- Calculates total and incremental stresses and strains for the radial model
XFACE	- Translates stresses to interfacial pressures
STFAXL	- Sets up stiffness matrix for axial problems
ABOUND	- Applies axial boundary conditions to the stiffness matrix
ASOLVE	- Solves axial load displacement problem
ALOAD	- Sets up load vector for the axial problem
DMCHB	- Solves linear equations with a banded coefficient matrix
ASTRESS	- Calculates incremental stresses and strains for the axial model
CREEP	- Controller for getting creep loads to feed back into the above chain to calculate creep displacements
EFFECT	- Calculates effective stresses
RATE	- Calculates incremental creep strain or plastic strain increment for a given time step

FIGURE 1. Calling Sequence for GAPCON-3 Subroutines

FIGURE 2. Iteration Scheme for Steady-State Cases

FIGURE 3. Iteration Scheme for Transient Cases

When converged temperatures for all axial boundaries have been obtained GAPCON calls TEMP3 (see Figure 1). TEMP3 returns element-averaged temperatures to be assigned to finite elements in PELET (the mechanics package). Subroutine PELET sets up a chained radial-axial-radial finite element calculation^(a) of cladding displacements, stresses, and strains and returns elastic deflections of the cladding. In transient calculations, these incremental elastic corrections are factored into one last series of iterations on boundary temperature profiles to refine the ΔT_{gap} 's. Then the program moves to the next time step.

For steady-state calculations, the cladding deflections are also used to correct the ΔT_{gap} 's but in addition gas release is recalculated. The program may iterate several times through the entire process of converging on boundary temperatures, recalculating fuel temperatures, and recalculating gas release, until the gas release convergence criterion of $\pm 1.0\%$ is achieved. The steady-state calculation then recalls PELET to get creep deflections before going to the next time step.

IV.B GAPCON-3 (MAIN PROGRAM)

GAPCON-3 is actually the main program although it contains no loops. GAPCON-3 first calls HEADER, which writes a banner page, then calls INPT, which reads in the NAMELIST variables and the fuel and cladding material property cards, and sets up the temperature-dependent properties arrays for the fuel and cladding (arrays CF and AA, Respectively). Then the array D is set up which contains the variably dimensioned parameters in the program. The use of the D array with its pointers plus subroutine EXFL permits automatically varied core requirements on the CDC 6600, depending on the input problem. The way this is accomplished is best described by an example.

Suppose the program contained only three variably dimensioned arrays, X, Y, and Z, which should have the dimensions X(NX), Y(NY,2), Z(4,NZ,LONG). Since all three arrays are actually stored in a single dimension, they can

(a) This calculation is described in Section V.B.

be part of a single-dimension super array. The starting position ("pointers") for each variable within the super array D can be calculated as follows:

```
IX = 1
IY = IX + NX
IZ = IY + 2*NY
ILAST = IZ + 4*LONG*NZ
```

Clearly the value of ILAST is the required dimension of the super array D. GAPCON-3 calculates the pointers for D, which are stored in blank common, and then uses EXFL to extend the core allocation of blank common from 1 to ILAST. Returning to our three-variable example, the subroutine call statements, subroutine argument lists, and dimension statements are handled as follows:

```
CALL SUB (D(IX),D(IY),D(IZ),NY,NZ)
SUBROUTINE SUB(X,Y,Z,NY,NZ)
DIMENSION X(1), Y(NY,1), Z(4,NZ,1)
```

Because of the way most compilers store data, the right-hand index need not be specified in this case. The call statement fixes the variables in array D, and the values of the variables can be assigned, stored, and retrieved in the proper sequence via the above statements.

IV.C THERMAL PERFORMANCE CALCULATIONS (TEMPERATURE AND GAS RELEASE)

Subroutine GAPCON contains all the major loops in the program, including all those indicated in Figures 2 and 3.

Two subroutines are called from GAPCON prior to any loops (hence they are called only once per case). These are FLUXD, which calculates the flux depression, and INPOUT, which outputs the input variables, as in GAPCON-2.

The time step loop begins next. There are three subroutines at its beginning called once each time step: POWDIS, which calculates absolute values of linear power at each axial boundary; FISGAS, which calculates the

current increment of fission gas produced, and TEMP1, which fixes the temperatures at the rod top and bottom boundaries to the inlet and outlet temperatures, respectively.

The iteration loop on gas release begins next (index K). This loop contains a large loop (index IPOW) which marches from one axial boundary to the next up the fuel rod. At each axial boundary, fuel relocation and densification are first evaluated (subroutines RELOC and DENSF, respectively). Then the fuel temperature profile is calculated (subroutines TEMP2-DDLR), and the fuel thermal expansion and swelling are estimated (subroutines EXPAND and SWELL). Then a gap size and a gap conductance is calculated, from which a new estimate of fuel temperatures is made. The program iterates to convergence on gap conductance, then local gas release is calculated (subroutine GASREL). The process is repeated for each axial boundary. The program then corrects the cladding deflections (subroutine PELET) and returns to repeat the entire gas release sequence until the release rates match within 1%.

GAPCON then readjusts the free gas inventory and calls OUTPUT to print out the results for the time step.

IV.D THE MECHANICS CALCULATIONS

PELET is a subroutine which connects axial finite element model of the fuel and cladding with a series of radial models of the fuel and cladding at various axial elevations. Prior to contact the radial slices are examined in the plenum region, the top fuel region, a mid-fuel region, and the bottom region. Axial stresses from the axial model are imposed on the radial model. After contact has occurred anywhere along the rod, the radial slices are examined at the standard regions and the contact regions. Interfacial pressures from the contact regions are translated to radial loads for the axial model; then its axial stresses are applied to the radial models in the form of loads along the top of each slice.

The interrelation of the PELET subroutines is as follows: PELET first calls INPAXL and INPUT to set up the element numbering, node numbering, and degree-of-freedom numbering for the axial and radial models, respectively. This need be done only once per case.

Next follows a loop (the NKN loop) which is only executed once until temperature convergence is achieved; then it is executed twice. For $NKN = 1$, only elastic calculations are performed; for $NKN = 2$, only creep and plastic calculations are performed. Subroutine CREEP is called for $NKN = 2$. It provides estimates of generalized creep and plastic strain for each element which are translated into loads in the same way as thermal strains.

Within the NKN loop is the KKA loop, executed twice per NKN. For $KKA = 1$, radial and then axial model calculations are performed; for $KKA = 2$, only a final radial model calculation is done. The first radial calculation obtains interfacial pressures at contact regions and passes them to the axial calculation where they are translated into radial loads. The appropriate axial stresses from the axial calculations are passed to the second radial calculation and translated to axial loads. Those axial stresses also form the axial component of the stresses used for creep calculations.

Within the KKA loop is the KA loop, which moves up the axial regions and calls for the radial model calculation at preselected regions. These regions are the top of the cladding, the top of the fuel boundary of highest power, and fuel bottom, plus any additional regions at which contact has occurred. The radial calculation involves STIFF, which sets up the global stiffness matrix for the cladding or fuel cladding model, depending on whether contact has occurred; BOUND, which adjusts the stiffness matrix to account for axial conditions; and SOLVET, which calls LOAD to get the appropriate load vector (thermal plus external); calls DMCIB to solve the load-displacement equations, calls STRES to calculate incremental stresses and strains, and calls XFACE to calculate incremental interfacial pressure.

Similarly for the axial model, subroutine STFAXL sets up the stiffness matrix for fuel and cladding and ABOUND applies the axial boundary conditions; ASOLVE calls ALOAD to get the load vector, calls DMCHB to solve the load-displacement equations, and calls ASTRES to calculate incremental axial stresses and strains.

It should be emphasized that the calculations are all incremental after the first time step. That is, the loads and the resulting displacements, stresses, and strains represent only the changes from the previous to current time step, and the total changes are cumulative totals of these incremental results. It is imperative that the first time step be positioned at zero time and extremely low power (less than 1 kW/ft).

V. DETAILED DESCRIPTION OF MODELS AND CALCULATIONS

This section presents detailed descriptions of the temperature calculation and the stress/strain calculation. Models which have been altered from GAPCON-2, including fission gas release and fuel expansion and swelling, are also reviewed in detail.

V.A TEMPERATURE CALCULATIONS

The radial temperature profiles are solved by an orthogonal collocation method of weighted residual (MWR) technique.⁽⁶⁾ The difference between the solution formulations for the steady-state and transient cases follows directly from differences in the governing equations and differences in boundary conditions. The derivation of the MWR solution will be done in some detail (and follow a parallel path for the two cases) so that differences will be clear.

The Steady-State Solution

The general statement of the steady-state problem is the coupled set of equations (see Table 5 for Nomenclature).

$$\text{fuel: } \vec{\nabla} \cdot [K_f(T_f) \vec{\nabla} T_f(\vec{r})] + Q(\vec{r}) = 0 \quad (1)$$

$$\text{cladding: } \vec{\nabla} \cdot [K_c(T_c) \vec{\nabla} T_c(\vec{r})] = 0 \quad (2)$$

with 12 boundary conditions.

GAPCON-3 (together with most fuel performance programs) only solves the radial version of the above equations:

$$\nabla_r^2 T_f(r) + \beta_f(T) \left(\frac{dT_f}{dr} \right)^2 + \frac{Q(r)}{K} = 0 \quad (3)$$

$$\nabla_r^2 T_c(r) + \beta_c(T) \left(\frac{dT_c}{dr} \right)^2 = 0 \quad (4)$$

where $\nabla_r^2 \equiv \left[\frac{d^2}{dr^2} + \frac{1}{r} \frac{d}{dr} \right]$

and $\beta \equiv \frac{1}{K} \frac{dK}{dT}$.

Equations 3 and 4 are still a coupled set, with the coupling occurring in the boundary condition. The boundary conditions are:

$$\left. \frac{dT_f}{dr} \right|_{r=a} = 0 \quad (\text{Inner Boundary}) \quad (5)$$

$$\frac{q}{A_f h_{\text{gap}}} = T_f(b) - T_f(c) \quad (\text{Gap Condition}) \quad (6)$$

$$-K_c \left. \frac{dT_c}{dr} \right|_{r=c} = \frac{q}{A_{ci}} \quad (\text{Continuity of Heat Flow}) \quad (7)$$

$$T_c(d) = T_{\text{cool}} + \frac{q}{A_{cs} h_{\text{film}}} \quad (\text{Outer Boundary}) \quad (8)$$

Again, see Table 5 for Nomenclature.

The MWR technique for solving this set of equations is as follows. Trial solutions for the fuel and the cladding are assumed, of the form

$$T_f = \sum_{j=1}^{N_F+2} d_j r^{j-1} \quad a \leq r \leq b \quad (9)$$

$$T_c = \sum_{j=1}^{N_C+2} d_j r^{j-1} \quad c \leq r \leq d \quad (10)$$

The trial solutions are inserted in equations for the fuel and cladding, and the respective equations are evaluated at N_F points in the fuel and N_C points in the cladding. Setting the resulting expressions = 0 at the "collocation points" generates $N_F + N_C$ equations, which together with the four boundary

conditions is, sufficient to solve for the $N_F + N_C + 4$ unknown coefficients d_j . We have found that four properly spaced collocation points in the fuel and two in the cladding are sufficient for a system of $4 + 2 + 4 = 10$ equations.

The iterative solution of the collocation equations is performed on the basis that the thermal conductivity and its temperature derivative will be evaluated from the temperatures of the previous iteration, but all other terms will be evaluated at the current iteration. At first glance this seems impossible but it can be achieved by applying Fourier's law to rewrite the term $\beta \left(\frac{dT}{dr} \right)^2$.

We know that at each radial boundary R_i in the fuel, Fourier's law must apply, i.e.,

$$-K_f \frac{dT_f}{dr} \Big|_{r=R_i} = \text{surface heat flux at } R_i \quad (11)$$

The heat flux is just the heat produced within the volume bounded by the radius R_i divided by the cylindrical surface area at R_i , such that

$$-K_f \frac{dT_f}{dr} \Big|_{r=R_i} = \frac{1}{2\pi R_i} \int_{V \text{ inside } R_i} Q(r) dV \quad (12)$$

$$= \frac{1}{R_i} \int_{r=a}^{r=R_i} Q(r) r dr \equiv I(r) \quad (13)$$

and

$$\beta_f (dT_f/dr)^2 = \frac{1}{K^3} \frac{dk}{dT} I^2(r) \equiv \delta_f(T) I^2(r) \quad (14)$$

Since there is no heat generation in the cladding, the corresponding expression for the cladding is:

$$\beta_c (dT_c/dr)^2 = \frac{\delta_c(T) q^2}{(2\pi r)^2} \quad (15)$$

Now the required assignment of unknown and known terms can be made. Equations 3 and 4 become

$$\nabla_r^2 T_f^{n+1} = \frac{-Q}{K^n} - \delta_f^n I^2(r) \quad (16)$$

$$\nabla_r^2 T_c^{n+1} = - \frac{\delta_c^n q^2}{(2\pi r)^2} \quad (17)$$

where now the unknown temperatures of the current iteration (superscript n+1) can be found in terms of the temperature dependent function $\delta(T)$ evaluated at temperatures from the old iteration (superscript n).

To summarize, the steady-state solution proceeds as follows:

1. Arrange the boundary conditions and collocation equations as in Table 3, and substitute in the trial functions to form a set of 10 linear equations with expansion coefficients as unknowns.
2. Guess an initial temperature distribution. Evaluate the K's and δ 's on that basis and solve for the d_j 's (call subroutine SIMQ).
3. Recalculate the temperature distribution, reevaluate the K's and δ 's and resolve for the d_j 's.
4. Repeat to convergence.

The present convergence criterion in GAPCON-3 is that the sum of the absolute differences between previous and current temperatures at all points of evaluation shall be less than 1.0°C.

TABLE 3. Simultaneous Set of Steady-State Equations

Evaluation Boundary	Equation Form	Comments
$r = a$	$dT_f^{n+1}/dr = 0$	Inner b.c.
Coll.Pts.1-4 ^(a)	$\nabla_r^2 T_{f_i}^{n+1} = \frac{-Q_i}{K_{f_i}^n} - \delta_{f_i}^n I_i^2$	Collocation
Gap	$T_{f(b)}^{n+1} - T_{c(c)}^{n+1} = \frac{q}{2\pi b h_{gap}}$	Gap Condition
$r = c$	$dT_c^{n+1}/dr = \frac{-q}{A_{cI} K_c^n}$	Continuity of Heat Flow
Coll.Pts.1,2 ^(b)	$\nabla_r^2 T_c^{n+1} = -\delta_c^n \frac{q}{2\pi r_i} I_i^2$	Collocation
$r = d$	$T_c^{n+1} = T_{cs}$	Outer b.c.

- (a) The collocation points for the MWR solution in the fuel are selected at the Zeros of the $P_4(t)$ Legendre polynomial with a change of variable $t = (1-x)/2$. The modified polynomial defined as a "shifted" Legendre polynomial has the following form:

$$P_4(t) = 1 - 20t + 90t^2 - 140t^3 - 70t^4$$

where t is defined from $0 < 1$ the zeros become :

0.0694320
0.3300095
0.6699905
0.9305680

which are dimensionless ratios of the fuel radius where $r = (b - a) P_4(t)/d$.

- (b) The collocation points for the cladding use a $P_2(t)$ Legendre polynomial using the similar change of variables $t = (1 - x)/2$ and have the following form:

$$P_2(t) = 1 - 6t + 6t^2$$

the zeros become:

0.2113248
0.78867513

which are defined in terms of dimensionless ratios of the cladding thickness $r = (d - c)P_2(t)/d$.

The Transient Solution

The equivalent expressions like Equations 3 and 4 for the transient situation are:

$$\frac{1}{\alpha_f(T)} \frac{\delta T_f(r,t)}{\delta t} = \nabla_r^2 T_f + \frac{Q(r,t)}{K_f} + \beta_f \frac{\delta T_f}{\delta r}^2 \quad (18)$$

$$\frac{1}{\alpha_c(T)} \frac{\delta T_c(r,t)}{\delta t} = \nabla_r^2 T_c + \beta_c \frac{\delta T_c}{\delta r}^2 \quad (19)$$

where

$$\alpha = \frac{K}{\zeta C}, \text{ here } C = \text{heat capacity}.$$

The approach taken is to approximate the time derivative as

$$\frac{\delta T}{\delta t} = \frac{T^{n+1} - T^n}{\Delta t}$$

where Δt is the time step size, n indicates the beginning of the time step and $n+1$ indicates the end of the time step. We evaluate K , C , dK/dT , and $(\delta T/\delta r)^2$ at the beginning of the time step, using temperature from the previous time step. With all this in mind, we can isolate all terms involving T^{n+1} (the "new" temperatures) on the lefthand sides and solve for them. Note that no iteration is implied! The equivalents of Equations 16 and 17 can be written as

$$\nabla_r^2 T_f^{n+1} - \frac{1}{\alpha_f^n} \frac{T_f^{n+1}}{\Delta t} = - \frac{T_f^n}{\alpha_f^n \Delta t} - \frac{Q^{n+1}}{K^n} - \beta_f^n \frac{\delta T_f^n}{\delta r}^2 \quad (20)$$

and

$$\nabla_r^2 T_c^{n+1} - \frac{T_c^{n+1}}{\alpha_c^n \Delta t} = - \frac{T_c^n}{\alpha_c^n \Delta t} - \beta_c^n \frac{\delta T_c^n}{\delta r}^2 \quad (21)$$

The boundary conditions for the transient case are also different (except for the first one). They are:

$$\left. \frac{\delta T_f}{\delta r} \right|_{r=a} = 0 \quad \text{(Inner Boundary Condition)} \quad (22)$$

$$K_f \frac{\delta T_f}{\delta r} \bigg|_{r=b} + h_{\text{gap}} [T_f(b) - T_c(c)] = 0 \quad \text{(Gap Condition)} \quad (23)$$

$$- 2\pi b K_f^n \frac{\delta T_f}{\delta r} \bigg|_{r=b} = - 2\pi c \frac{\delta T_c}{\delta r} \bigg|_{r=c} \quad \text{(Continuity of Heat Flow)} \quad (24)$$

$$- K_c \frac{\delta T_c}{\delta r} \bigg|_{r=d} = h_f [T_c(d) - T_{\text{cool}}] \quad \text{(Outer Boundary Condition)} \quad (25)$$

Somewhat as with steady-state, one substitutes the trial functions into the boundary conditions and collocation equations, arranged as in Table 4, insists that the boundary conditions be satisfied in current time, uses the previous time step's results to evaluate the temperature-dependent properties and $(\delta T / \delta r)^2$, solves, then moves to the next time step.

The inverse effective time constant for the fuel rod is in the range of 3 to 10 seconds. We have found time steps of 0.1 second sufficiently small to insure accuracy and stability. No iteration on conductance is performed. The code uses the gap conductance from the previous time step, satisfies the gap condition in current time, and moves on.

We have found the collocation technique to be two to three times faster than the previously used 50-node finite difference technique for generating fuel temperatures. Figure 4 displays results of comparative transient calculations.

TABLE 4. Simultaneous Equations for Transient

Evaluation Boundary	Equation Form	Comments
$r = a$	$\frac{\delta T_f^{n+1}}{\delta r} = 0$	Inner b.c.
Coll.Pts.1-4	$\nabla_r^2 T_{f_i}^{n+1} - \frac{T_{f_i}^{n+1}}{\alpha_i^n \Delta t} = \frac{T_{f_i}^n}{\alpha_i^n \Delta t} + \frac{Q_i^{n+1}}{K_i^n} - \beta_{f_i}^n \left(\frac{\delta T_{f_i}^n}{\delta r} \right)^2$	Collocation
Gap	$h_{gap} \left[T_f^{n+1}(b) - T_c^{n+1}(c) \right] + K_f^n \frac{\delta T_f^{n+1}}{\delta r} = 0$	Fuel Heat Loss
$r = c, b$	$-b K_f^n \frac{\delta T_f^n}{\delta r} + c K_c^n \frac{\delta T_c^{n+1}}{\delta r} = 0$	Heat Flow Continuity
Coll.Pts.1,2	$\nabla_r T_{c_i}^{n+1} - \frac{T_{c_i}^{n+1}}{\alpha_c^n \Delta t} = - \frac{T_{c_i}^n}{\alpha_c^n \Delta t} - \beta_c^n \left(\frac{\delta T_{c_i}^n}{\delta r} \right)^2$	Collocation
$r = d$	$K_c \frac{\delta T_c^{n+1}}{\delta r} + h_{film} T_c^{n+1} = h_{film} T_{cool}$	Outer b.c.

TABLE 5. Nomenclature

<u>Symbol</u>	<u>Meaning</u>	<u>SI Units</u>
A_f	Fuel surface area/unit length	m
A_{CI}	Cladding inner surface area/unit length	m
A_{cs}	Cladding outer surface area/unit length	m
a	Inner fuel radius	m
b	Outer fuel radius	m
c	Inner cladding radius	m
d	Outer cladding radius	m
d_j	Expansion coefficient	--
h_{gap}	Gap conductance	W/m^2-C
h_{film}	Film coefficient	W/m^2-C
q	Linear power	W/m
Q	Volumetric heat generation	W/m^3
T_c	Cladding temperature	C
T_{cool}	Coolant temperature	C
T_{cs}	Cladding outer temperature	C
T_f	Fuel	C
K_f	Thermal conductivity - fuel	$W/M-C$
K_c	Thermal conductivity - cladding	$W/M-C$
Greek:		
α	Diffusivity, $K/\xi C$	m^2-sec^{-1}
β	Special conductivity function	--
δ	Special conductivity function	--
ξ	Density	Kg/m^3

FIGURE 4. Comparison Between Subroutine DDLR and the Finite-Difference Code TRUMP (UCRL-14754, 1972) for a Step Power Decrease

$$\text{Note } T_N = \frac{T_C - T_{\text{coolant}}}{T_{\text{init.}} - T_{\text{coolant}}}$$

V.B MECHANICAL ANALYSIS

The analysis of the mechanical behavior of the fuel rod is based on a geometric model which divides the fuel rod into a series of axial regions. Using this model both the cladding and the fuel can either move independently of each other or can be locked to simulate pellet and cladding mechanical interaction.

The mathematical formulation uses a series of axisymmetric finite element models specifically designed to represent the effects of both radial and axial interaction between the fuel and the cladding. The models are designed to account for both plastic creep and elastic anisotropic behavior. The mathematical formulation also provides the capability for modeling structural anisotropic effects such as axial and radial fuel cracking and mechanistic fuel relocation concepts.

The objective of this analysis approach is to provide a mathematical mechanical analysis formulation base which conceptually can handle all major components of fuel behavior analysis without any foreseeable major corrections in either scope or concept. The matrix formulation technique that has been used provides this flexibility and can be modified to include more advanced concepts without major revisions to the code.

The current version of GAPCON-3 specifically treats the following areas:

1. Stress and strain for the fuel and cladding resulting from thermal gradients as well as from external applied loads.
2. Radial and axial mechanical interaction of fuel and cladding (PCI).
3. Thick shell analysis of the cladding and fuel which provides stress and strains as a function of radius.
4. The statically indeterminate solution of displacement and interfacial contact forces for the fuel and cladding when contact has occurred.
5. Axial stack interactions on multiple regions and the cyclic locking and unlocking of axial region as a function of path accounting for incremental changes in system strain energy.
6. A mechanism for treating elastic structural anisotropy of the fuel regions as well as material elastic anisotropy of the cladding.

7. Creep and plasticity as incremental strains and accounting for plastic anisotropy for the fuel and cladding.
8. Provide a fast running solution procedure that will provide the detail without taking excessive amounts of computer time.
9. Provide a technique that can accept transient deformation data using any arbitrary equation describing creep strain or plastic strain conditions.

V.B.1 General Formulation

The requirements outlined in the previous section can be satisfied by mathematically treating the structure using a finite element solution procedure. The finite element formulation can model the statically indeterminate PCI solution by the introduction of gap elements and can model the anisotropy and temperature dependent material properties by changing them on an element by element basis.

The treatment of cracked fuel and the differences in radial and axial elastic compliance of a given fuel region caused by cracking and stacking irregularities can be mathematically modeled using an orthotropic material matrix for the element formulation. By using these techniques on an incremental basis the nonlinear changes in loading and structure can be included on a step-by-step basis following a very general path dependent event. The effects of plasticity, creep, and structural discontinuities associated with gap closure and deformed structural conditions are examples of the problem complexity that can be considered.

The finite element formulation used in GAPCON-3 is based on an axisymmetric ring element shown in Figure 5. This element provides stresses in the radial $\{\sigma_r\}$, hoop $\{\sigma_\theta\}$, axial $\{\sigma_z\}$, and shear $\{\sigma_{rz}\}$ directions which are the principal coordinates of interest. The basic equation defines incremental load vectors $\{f\}_i$ being combined to form a total displacement vector $\{q\}$. This is true in the general sense, however in reality $\{q\}$ is the summation of $\sum_{i=1}^n \{q\}_i$. GAPCON-3 operates in an incremental mode which means that although the individual solution for a given time step may not require (from the plasticity/elasticity viewpoint) an incremental solution, the general nature

FIGURE 5. Basic Axisymmetric Ring Element Used in the GAPCON-3 Stress Analysis Models

of the total fuel system analysis makes such a calculation inherent in the procedure. The transformation matrix $[K]$ is the structural stiffness matrix for a single element and can be summed for the combined stiffness of many elements in a total structure. It is changed at the completion of every step to account for changes in geometry or temperature effects on material properties. For example, see Reference 7.

The matrix $[K]$ is composed of material elastic coefficients that are sensitive to the principal directions of interest and geometric effects which are integrated over the volume of the element. In GAPCON-3 a constant strain triangle is used as the basic element either in series or in grouped quadrilateral combinations.

Figure 6 shows the radial model formulation which uses the constant strain triangle in series to simulate an average condition of a given fuel and cladding axial region. Figure 6 also shows the application of a gap element which is used for simulating gap or the fuel and cladding in mechanical contact.

FIGURE 6. Radial Calculation Model for GAPCON-3 Showing the Minimum Number of Axial Locations Where Radial Calculations Are Made. This number can be increased by the user to every axial region. Regions where contact occurs are regions where radial calculations are automatically made.

Mathematically the gap element couples the displacements or degrees of freedom D.O.F. of the fuel stiffness matrix $[K]$ fuel with the appropriate D.O.F. of the cladding stiffness matrix $[K]$ cladding. The solution, using the combined stiffness matrix, represents the statically indeterminate solution with coupling of the radial and axial displacements at the fuel and cladding interface.

The radial models are located conceptually in the center of a given axial region. The temperatures are an average of nodes placed on the upper and lower region interfaces for a given radius. This type of averaging provides radial temperature gradients which are representative of the region. External loading is averaged over the region and scaled to the element dimension.

Radial models are always constructed, as a basic minimum, at the bottom of the fuel column at the maximum power point, the top of the fuel column and the top of the fuel rod. (See Figure 6). Additional elevations are added

automatically if a region is defined as being in contact. If a region has at one time been in contact and goes out of contact it will continue to be classed as a region where a radial model calculation is performed. The results of the calculations are interpolated between axial elevations where radial results are available.

Before contact, an individual radial slice at a given elevation provides cladding stress and strain information which reflects the effects of external pressure loading in both the axial and radial directions. After contact has occurred the radial model gives axial stresses for the local areas, the interfacial pressures that are developed, and the resultant strains and displacements.

Radial interaction always results in changing the values of the axial stresses and strains in the regions not in contact due to the relative axial motion of the fuel stack and cladding. The radial model can compute the axial stress for the region in contact but not in the regions where no contact has occurred. In general, these regions will have high axial stress with relatively low hoop and radial stresses. The interaction of stresses in regions below the axial zone that has mechanical contact with the axial thermal expansion can be computed using the axial interaction model (see Figure 7).

FIGURE 7. The Axial Mechanical Analysis Model Using a Quadrilateral Element Composed of Four Triangular Elements. Axial interaction between the fuel and cladding stacks is accomplished by closure of the gap element.

This model, which employs quadrilateral elements composed of axisymmetric triangles, is used to evaluate the axial stresses and strains in the fuel and cladding. It is responsive to the pressure loads applied by the fuel and the thermal and external loading for each region. The axial results of this model are then in turn used in an additional radial model calculation where the axial loads applied to the radial model are those values obtained by the axial solution for that particular axial zone. The computed results of the second time through the radial model provide an iterative technique at obtaining a fully interactive radial-axial stress model. This technique has the benefits of a detailed stress calculation without the penalties of actually performing all of the calculations normally required for this type of refinement.

The procedure is called the "Chained Radial-Axial-Radial Solution" and is shown schematically in Figure 8. The axial model is called once for a given solution; the radial models are called twice. The procedure allows the radial calculation of the interfacial pressure between the fuel and the cladding or the gap dimension to be corrected by the Poisson effects from the axial loads. Since the computation of thermal effects in the fuel is the primary objective, whatever liberties that are taken in this approach can be justified on the basis of the effect on thermal predictions. As an example of the correction to the state of stress and strain that will be obtained at each step in the chained algorithm, a set of idealized conditions can be assumed where:

$$\sigma_2 = 0.5\sigma_1 \quad \text{and} \quad \mu = 0.3.$$

For this case then: $\epsilon_1 = \frac{1}{E} (\sigma_1 - \mu\sigma_2)$

or $\epsilon_1 = \frac{1}{E} (\sigma_1 - \mu \cdot 0.5\sigma_1)$

where: $\sigma_1 = \text{hoop stress}$

$\sigma_2 = \text{axial stress}$

If σ_2 is not included or not known, as is the case for the first time through the radial model calculation where the correct axial stresses are not known, the error in ϵ_1 becomes:

$$\epsilon_1 \text{ error} = \frac{\mu\sigma_2}{E} \sim \frac{0.15\sigma_1}{E}$$

FIGURE 8. The Chained Radial-Axial-Radial Solution Algorithm Used to Couple the Radial and Axial Finite Element Models Through Compatibility of Forces. The axial model is called once for a given calculation.

For the next step in the calculation where the axial model uses the results from the radial models:

$$\epsilon_1 = \frac{1}{E} (\sigma_1 - \mu \sigma_2)$$

the error in axial direction becomes:

$$\epsilon_2 \text{ error} = \frac{0.3 \times 0.15 \sigma_1}{E} = \frac{0.045 \sigma_1}{E}$$

For the second time through the radial model using the results from the previous calculations the error in the radial direction becomes:

$$\epsilon_1 \text{ error} = \frac{\mu \times 0.045 \sigma_1}{E} = \frac{0.0135 \sigma_1}{E}$$

As a result of one complete iteration through the chained procedure the error in the elastic solution will be approximated by:

$$\text{error} = \mu^n$$

Where n is the number of calculational steps. For one complete cycle n = 3, and $0.3 < \mu < 0.42$, so that

$$\text{error} = 2.7\% \text{ to } 7.4\%.$$

The advantages of the chained solution procedure for a finite element approach are significant when viewed from the viewpoint of computer core

storage and computation time. The standard finite element approach using an axial/radial element geometry aspect ratio of approximately 10 to 1 (ideal is from 1:1 to 2:1) and using six fuel and four cladding nodes would require 15,000 unknowns or D.O.F. Such an approach for a fuel rod is of limited usefulness. The solution time required is proportional to the number of terms in the stiffness matrix [K] which for 15,000 unknowns with bandwidths of 50 becomes $15,000 \times 50 = 750 \times 10^3$ calculations.

For the chained solution approach assuming every axial zone is used

Radial	20 x 40 x 6	= 4,800	operations
Axial	232 x 24	= 5,568	operations
Radial	20 x 40 x 6	= <u>4,800</u>	operations
		14,240	operations or a relative

savings in time of:

$$\frac{750}{14.2} \sim 50 \times \text{faster}$$

If a lesser number than 20 axial zones were used such as 10 with five zones where radial slices are computed, the conditions become:

Radial	5 x 40 x 6	= 1200
Axial	1 x 116 x 24	= 2784
Radial	5 x 40 x 6	= <u>1200</u>
		5184
relative savings		$\frac{750}{5.2} = 144 \times \text{faster}$

For a simple elastic calculation the savings are significant; for an iterative plasticity or creep calculation the savings obtained by using this procedure are even more significant in terms of reducing the cost of an individual run.

The justification for using this procedure in terms of compatibility of strains and displacements is based on the physical structure of a fuel rod.

A fuel rod, due to its very high length to diameter ratio is more compliant in the axial direction than in the radial direction. The coupling in the axial mode tends to be one of force equilibrium or compatibility as opposed to strain compatibility alone. This can be restated by saying that

the solution in the axial direction obtained by compatibility of strain/displacement approach would be very similar to that obtained by the compatibility of forces or flexibility approach. This comparison is a rather loose application of the terms of displacement and flexibility; however, the concept of a system that is much more insensitive to errors in the axial displacements on stress than it is to radial displacements on stress is the key point in understanding the justification for the procedure.

In the radial direction the radial model, which is based on compatibility of strain and displacements, gives a good estimate of stress conditions. During contact the local axial stresses caused by interaction between the fuel cladding are relatively high and the interaction zone is generally limited to a relatively small area. The net result is that regardless of the approach (either full finite element or chained iterative finite element), the computed results will be similar for the conditions under consideration. A specific example is given in PNL-2435, the verification document for the code, where a comparison calculation is shown for a complete finite element solution.

V.B.2 Finite Element Formulation

The governing equation in a finite element matrix mechanical analysis procedure relates the applied loads to the resulting displacement through a linear transformation matrix. This can be described by Equation 26.

$$[K]\{q\}_i = \{f\}_{\text{external}} + \{f\}_{\text{creep}} + \{f\}_{\text{thermal}} + \{f\}_{\text{plastic}} = \{f\}_i \quad (26)$$

where: $\{f\}_i$ = incremental load vector due to external, creep, thermal, and plasticity loads.

$\{q\}_i$ = nodal displacement vector for a given incremental load vector

$[K]$ = stiffness matrix or transformation matrix relating applied loads and displacements.

The fuel-cladding system experiences strains resulting from displacements due to thermal expansion, internal and external pressure, and pellet/cladding interaction. Further displacements/strains are incurred through creep and plasticity. In GAPCON-3, these sources of strain are all expressed as line loads $\{f\}_i$ at the element boundaries.

Changes in the element stresses and strains resulting from changes in the incremental load vectors $\{f\}_i$ are found by:

1. Translating thermal expansion and external load changes to incremental nodal load vectors $\{f\}_i$.
2. Solving Equation 26 for the entire model to get all the nodal displacements.
3. Translating the nodal displacements of each element to element strain.
4. Subtracting the initial strain. (See following sections.)
5. Calculating stress from strain.
6. Correcting for creep and plastic strains by translating these to nodal loads and repeating Steps 1 through 5.

The stiffness matrix for the entire model is assembled from the matrices for the individual elements, subject to compatibility of nodal displacements and to boundary conditions. Each element matrix is a 6 x 6 symmetric matrix, reflecting 6 degrees of freedom (radial and axial movement of the three-corner node points). The matrices for all elements will be identical algebraically, since all elements are identical structurally. But the numerical values of the K_{ij} from the different matrices will of course be different because

1. The elements differ geometrically. (Nodes and internal degrees of freedom are always numbered counter-clockwise beginning at the lower left. For some elements $r_1 = r_3$ whereas for others $r_2 = r_3$, etc.)
2. The material properties will be evaluated at the element temperature, which differs from element to element.
3. The dimensions of the elements will vary as deformation proceeds.

The single element shown in Figure 5 identified the three triangular nodal points IJK. The figure indicated that for each node there are two displacements u and v corresponding to the radial and axial directions for a

total of six displacements^(a) for a given element. The element stiffness matrix $[K]$ is then a 6 x 6 matrix relating the six unknown displacements with six loads applied at the element nodes.

The derivation of Equation 26 is based on the minimization of the potential or strain energy in a continuum which can be specialized to a separate element. The strain energy can be written as shown in Equation 27 using the concept of virtual strains and displacements and relating the strain energy of the body to the work done by body loads and surface loads.

$$\Pi = \int_{vol} dU - \int_{vol} X dV - \int_{surface} T ds \quad (27a)$$

where: $dU = \frac{1}{2} \{\epsilon\}^T \{\sigma\} dV = \frac{1}{2} \{\epsilon\}^T [C] [(\{\epsilon\} - \{\epsilon_o\})] dV$

hence $\Pi = \frac{1}{2} \int_{volume} \{\epsilon^*\}^T [C] \{\epsilon\} dV - \frac{1}{2} \int_{volume} \{\epsilon^*\}^T [C] \{\epsilon_o\} dV - \int_{volume} X q^* dV - \int_{surface} T q^* ds \quad (27b)$

where: $\{\epsilon\}$ = strains (unknown and virtual)

$\{\epsilon_o\}$ = initial strains (known) thermal, creep, etc.

$[C]$ = material matrix

X = body forces $\equiv 0$ for this application

T = surface tractions

q^* = virtual displacements

dU = internal energy

(a) Throughout, these vectors are arranged as follows for a single element.

u_I Where I, J, K refer to the nodal indices read counterclockwise
 as in Figure 5.
 v_I
 u_J
 v_J
 u_K
 u_k

The component parts of Equation 27 can be rewritten in terms of matrix notation for an element where internal stresses and strains are described in terms of element nodal values. Equation 28 defines a transformation matrix $[N]$ for an interpolation function which relates the displacement of the element nodal points $\{q\}$ with the displacements internal to the element $\{u\}$.

$$\{u\} = [N]\{q\} \quad (28)$$

An additional transformation matrix $[B]$ can be developed which relates the element nodal point displacements $\{q\}$ with the element strains $\{\epsilon\}$ as defined in Equation 29.

$$\{\epsilon\} = [B]\{q\} \quad (29)$$

By using Equation 29 and the material coefficient matrix $[C]$ the element strain $\{\epsilon\}$ and element displacement $\{q\}$ can be related to the element stresses $\{\sigma\}$ as shown by Equation 30.

$$\begin{aligned} \{\sigma\} &= [C]\{\epsilon\} \\ \text{or } \{\sigma\} &= [C][B]\{q\} \end{aligned} \quad (30)$$

Using Equations 28, 29 and 30, the expression for describing the potential energy (Equation 27) can be written by direct substitution as shown in Equation 31.

$$\begin{aligned} \Pi = & \frac{1}{2} \int_{\text{volume}} \{q^*\}^T [B]^T [C] [B] \{q\} dV - \frac{1}{2} \int_{\text{volume}} \{q^*\}^T [B]^T [C] \{\epsilon_0\} dV - \\ & \int_{\text{volume}} \{q^*\}^T [N]^T \{X\} dV - \int_{\text{surface}} \{q^*\}^T [N]^T \{d\} ds \end{aligned} \quad (31)$$

Using the variational principle for finding the minimum potential energy shown in Equation 27 and differentiating with respect to the virtual displacements we can rewrite the expression in terms of the functional Π , the strain energy U and the external work W as Equation 32.

$$\Delta \Pi = \Delta U + \Delta W = 0 \quad (32)$$

Or, extracting the virtual displacement vector $\{q^*\}$ from Equation 31 and performing the first variation indicated in Equation 32 with respect to the virtual displacements

$$0 = \int_{\text{volume}} [B]^T [C] [B] \{q\} dV - \int_{\text{volume}} [B]^T [C] \{\epsilon_0\} dV - \int_{\text{volume}} [N]^T \{X\} dV - \int_{\text{surface}} [N]^T \{T\} ds \quad (33)$$

Equation 33 can be rewritten (Equation 34) for the specific application by neglecting the body forces.

$$\int_{\text{volume}} [B]^T [C] [B] dV \{q\} - \int_{\text{volume}} [B]^T [C] \{\epsilon_0\} dV - \int_{\text{surface}} [N]^T \{T\} ds = 0$$

or $[K]\{q\} = \{f_b\} + \{f_T\} \quad (34)$

where:

$$[K] = \int_{\text{volume}} [B]^T [C] [B] dV = (\text{stiffness matrix for a single replacement})$$

$$\{f_b\} = \int_{\text{volume}} [B]^T [C] \{\epsilon_0\} dV = (\text{initial strain equivalent load vector})$$

$$\{f_T\} = \int_{\text{surface}} [N]^T \{T\} ds = (\text{external applied surface forces translated into nodal loads})$$

Note: $[B]$, $[C]$ and $[N]$ have yet to be defined for an axisymmetric constant strain triangular element

For the case of interest the initial strain term $\{\epsilon_0\}$ or grouped as $\{f_b\}$ takes the form of the thermal, plasticity, and creep strains. The Equation 34 can then be written as Equation 35 which is identical to Equation 26.

$$[K]\{q\} = \underbrace{\{f\}}_{\text{thermal}} + \underbrace{\{f\}}_{\text{plastic creep}} + \underbrace{\{f\}}_{\text{external}} \quad (35)$$

The actual application of this procedure to a given problem requires two major things: (a) the selection of an element configuration, i.e., triangle, square, rectangle and (b) the shape function $[N]$ for describing the relationship between nodal point displacements $\{q\}$ and displacements internal to the element $\{U\}$.

In the case of GAPCON, the element shape is a triangular ring. The triangular element is the least complicated of any of the special elements. The operation defined in Equation 34 indicates that the element formulation requires a volume integral to be performed. The axisymmetric triangular ring element allows integration as shown in Equation 36.

$$[K] = \int_{\text{volume}} [B]^T [C] [B] dV$$

$$\text{or } [K] = 2\pi \iint [B]^T [C] [B] r dr dz \quad (36)$$

which can be approximated by using centroidal values denoted by Equation 37.

$$[K] = 2\pi [\bar{B}]^T [C] [\bar{B}] \bar{r} \Delta \quad (37)$$

where:

Δ = cross sectional area of the triangle and the bar indicates evaluation at the element centroid

Equation 37 has a further restriction in that the shape function $[N]$ which is used for this must be linear and does not require any additional nodal points on the element other than at the corners.

The shape function $[N]$ can best be described by the geometry shown by Figure 5 using the nodes IJK. The purpose of $[N]$ is to provide a means of interpolating displacements within the boundary of the element defined by the nodal displacements of the element corners.

In general, the displacement of any node (I) can be broken into two components u and v shown in Figure 5. For the element a total of displacements of six components is needed to complete the nodal displacement vector $\{q\}$ for a given element shown in Equation 38.

$$\{q\} = \begin{Bmatrix} u_I \\ v_I \\ u_J \\ v_J \\ u_K \\ v_K \end{Bmatrix} \quad (38)$$

Using a linear expression for u_I and v_I we can write Equation 39 for all the nodes:

$$\begin{aligned} u_I &= \alpha_1 + \alpha_2 r_I + \alpha_3 z_I \\ v_I &= \alpha_4 + \alpha_5 r_I + \alpha_6 z_I \\ u_J &= \alpha_1 + \alpha_2 r_J + \alpha_3 z_J \\ v_J &= \alpha_4 + \alpha_5 r_J + \alpha_6 z_J \\ u_K &= \alpha_1 + \alpha_2 r_K + \alpha_3 z_K \\ v_K &= \alpha_4 + \alpha_5 r_K + \alpha_6 z_K \end{aligned} \quad (39)$$

where:

r_I = radial dimension of the nodal point I
 z_I = axial dimension of the nodal point I
 α = coefficient to be determined
 $\begin{matrix} u_I \\ v_I \end{matrix}$ = nodal displacement vectors

Equation 39 can be solved for the α and an expression can then be written for the displacements u and v of any point within the element at some location r and z (Equation 40). This transformation between nodal point values and internal conditions is the interpolation function $[N]$.

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} (a_I + b_I r + c_I z) & 0 & (a_J + b_J r + c_J z) & 0 & (a_K + b_K r + c_K z) & 0 \\ 0 & (a_I + b_I r + c_I z) & 0 & (a_J + b_J r + c_J z) & 0 & (a_K + b_K r + c_K z) \end{bmatrix} \begin{pmatrix} u_I \\ v_I \\ u_J \\ v_J \\ u_K \\ v_K \end{pmatrix} \quad (40)$$

where:

Δ = area of the triangular cross section IJK

$$a_I = r_J z_K - r_K z_J$$

$$a_J = r_K z_I - r_I z_K$$

$$a_K = r_I z_J - r_J z_I$$

$$b_I = z_J - z_K$$

$$b_J = z_K - z_I$$

$$b_K = z_I - z_J$$

$$c_I = r_K - r_J$$

$$c_J = r_I - r_K$$

$$c_K = r_J - r_I$$

which can be rewritten in the form used in Equation 28 as shown in Equation 41.

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = [N] \{q\} \quad (41)$$

The $[B]$ matrix, defined by Equation 29, relates displacements to strains and can be developed by looking at the strain terms which are defined for the axisymmetric condition (Equation 42).

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_r \\ \epsilon_\theta \\ \epsilon_z \\ \gamma_{rz} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{u}{r} \\ \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \end{Bmatrix} \quad (42)$$

or noting that u and v are $\{q\}$:

$$\{\epsilon\} = [B]\{q\}$$

Where $[\bar{B}]$ is evaluated at the element centroid (Equation 43) and the bar denotes centroidal values for r denoted by $\bar{r}$.

$$[\bar{B}] = \begin{bmatrix} \frac{\partial N_I}{\partial r} & 0 & \frac{\partial N_J}{\partial r} & 0 & \frac{\partial N_K}{\partial r} & 0 \\ \frac{N_I}{\bar{r}} & 0 & \frac{N_J}{\bar{r}} & 0 & \frac{N_K}{\bar{r}} & 0 \\ 0 & \frac{\partial N_I}{\partial z} & 0 & \frac{\partial N_J}{\partial z} & 0 & \frac{\partial N_K}{\partial z} \\ \frac{\partial N_I}{\partial z} & \frac{\partial N_I}{\partial r} & \frac{\partial N_J}{\partial z} & \frac{\partial N_J}{\partial r} & \frac{\partial N_K}{\partial z} & \frac{\partial N_K}{\partial r} \end{bmatrix} \quad (43)$$

where:

$$\frac{\partial N_I}{\partial r} = b_I$$

$$\frac{\partial N_I}{\partial z} = c_I$$

$$\frac{\partial N_J}{\partial r} = b_J$$

$$\frac{\partial N_J}{\partial z} = c_J$$

$$\frac{\partial N_K}{\partial r} = b_K$$

$$\frac{\partial N_K}{\partial z} = c_K$$

$$\frac{N_I}{\bar{r}} = \frac{a_I}{\bar{r}} + b_I + \frac{c_I \bar{z}}{\bar{r}}$$

$$\frac{N_J}{\bar{r}} = \frac{a_J}{\bar{r}} + b_J + \frac{c_J \bar{z}}{\bar{r}}$$

$$\frac{N_K}{\bar{r}} = \frac{a_K}{\bar{r}} + b_K + \frac{c_K \bar{z}}{\bar{r}}$$

The material matrix [C] can either be written for orthotropic anisotropic material properties or for isotropic conditions. The present version of GAPCON-3 uses the latter condition.

$$[C] = \frac{E}{(1+\mu)(1-2\mu)} \begin{bmatrix} 1-\mu & \mu & \mu & 0 \\ \mu & 1-\mu & \mu & 0 \\ \mu & \mu & 1-\mu & 0 \\ 0 & 0 & 0 & \frac{1-2\mu}{2} \end{bmatrix} \quad (44)$$

where:

E = elastic modulus

μ = Poisson's ratio

By using Equations 40, 41, 42, 43 and 44, all of the matrix formulation terms previously defined can be found needed to develop the element stiffness matrix and load vectors for a single element. The implementation of this procedure for the radial and axial models is carried out in subroutines STIFF for the radial model and STFAXL for the axial model. In these subroutines, stiffness matrices for each individual element (XKBAR) are formulated and merged into a total stiffness matrix for the appropriate radial or axial model.

V.B.3 Assembly of the Global Stiffness Matrix-Radial Model

The assembly of individual finite elements into a total model involves a careful assessment of the relative displacements or degrees of freedom of each element. The radial model shown in Figure 6 has pairs of two triangular elements connected in series with additional pairs of elements. The "merge" operation of forming the global stiffness matrix from the element stiffness matrices consists of matching components in the individual stiffness matrices that involve common D.O.F. from other elements and summing them.

Each individual element has 6 D.O.F. or two displacements at each of three corner nodes. A combined or global stiffness matrix may contain several hundred or thousand D.O.F. The individual matrices as well as the total global matrix are symmetric, banded and positive definite. The bandwidth is defined by the maximum number of D.O.F. that interact with other D.O.F.

If the fuel and cladding are not in contact, the last part of subroutine STIFF assembles stiffness matrices for the fuel and cladding separately. If the fuel and cladding are in contact, a matrix for the total system is formed, with the gap degrees of freedom double-numbered (only for the radial model). Figure 9 illustrates the node and degree of freedom numbering for the two different cases.

To see how the element stiffness components are assigned to the global stiffness matrix, consider Figure 9a in detail. It is clear that node 1, for example, is shared by elements 1 and 2. It is therefore to be expected that some components from elements 1 and 2 will be present in the global matrix in (global) degrees of freedom 1 and 2. Similarly, node 5 is shared by elements 1, 2, and 3 so we will expect to see components from these elements influencing (global) degrees of freedom 3 and 4. The global stiffness matrix for the four elements shown in Figure 9a is presented schematically in Figure 10. An entry I_{jk} is meant to indicate the jk th component of the element stiffness matrix for element I . We see that the diagonal components from contributing elements appear on the diagonal of the global matrix at the degrees of freedom those elements share. The assignment of the off-diagonal element components is a bit more subtle, but in every case we see the shared degrees of freedom influenced properly by the contributing elements; e.g., degree of freedom 4 has off-diagonal influence from elements 1, 2, and 3 only. We can also see that the global stiffness matrix is banded and symmetric. Because of the symmetry only the main diagonal and either the upper or lower codiagonals need be stored, which greatly reduces the required computer core space as compared with a square matrix of similar size.

The bandwidth for the radial model is eight. In the program, the numbering sequences for elements, nodes, and degrees of freedom are set by a single call to subroutine INPUT. In actual practice the global stiffness matrices are stored as a single dimension array using only the main diagonal and upper codiagonals. The matrix structure is reconstituted whenever it is needed from this single column vector.

FIGURE 9a. Node and D.O.F. Numbering System for Fuel or Cladding Separate Radial Model

FIGURE 9b. Node and D.O.F. Numbering System for Fuel-Cladding Combined Radial Model

	1	2	3	4	5	6	7	8	9	10	11	12
1	111 211											
2	121 221	122 222										
3	131 251	132 252	133 255 355									
4	141 261	142 262	143 265 365	144 266 366								
5	151	152	153	154	155							
6	161	162	163	164	165	166						
7	231	232	235 315	236 316			237 377 411					
8	241	242	245 325	246 326			243 321 421	244 322 422				
9			335	336			331 451	332 452	333 455			
10							341 461	342 462	343 465	344 466		
11							431	432	435	436	437	
12							441	442	445	446	443	444

ELEMENT NO.
 COMPONENT OF
 ELEMENT STIFFNESS
 MATRIX

FIGURE 10. Global Stiffness Matrix for the Four Elements in Figure 9a. Global D.O.F. numbers are along the outside.

V.B.4 Assembly of the Global Matrix-Axial Model

The basic steps for calculating the element stiffness matrices used in the radial model are identical in the axial model and are performed in the first half of subroutine STFAXL. The node and degree of freedom numbering system for the axial model are shown in Figure 11. There is a basic difference from the radial model in that there is no renumbering of degrees of freedom in case of contact. If it is decided that an axial region is in contact, the axial degrees of freedom for the boundaries of that region are forced to move together through the following procedure:

1. Let i and ii be the two axial degrees of freedom to be locked. In the global stiffness matrix, add row ii to row i . To preserve symmetry, add column ii to column i .
2. Zero out row and column ii . Place a 1.0 on the diagonal at ii , ii .
3. Add load ii to load i and zero out load ii .
4. In the solution displacement it will be zero and displacement i will represent the axial displacement in degrees of freedom i and ii . Arbitrarily set displacement $ii - \text{displacement } i$.

This procedure has been shown to collect all the proper influence from contributing elements and in reality consists of a series of algebraic simple row and column operations.

V.B.5 Application of Boundary Conditions

The only boundary conditions in the models are condition of zero axial displacement along the bottom.* Accordingly, the rows and columns corresponding to those degrees of freedom are set to zero in the global matrix, the diagonal term is set to 1.0, and the corresponding load is set to zero. This forces the proper displacements to be zero. The manipulation of the stiffness matrix is accomplished in subroutine BOUND for the radial model and ABOUND for the axial.

* The fuel is always assumed to have a small hole in the center (1 mil diameter) unless a larger void is input or forms from fuel restructuring. Thus, there is no inner radial boundary condition except that of symmetry.

FIGURE 11. Node and D.O.F. Numbering System for Fuel and Cladding Combined Axial Model

V.B.6 Formulation of the Load Vector

The last step in the formulation of the problem is to return to the right-hand side of Equation 34 and transform it into a series of nodal loads (vector $\{f\}$). This operation is performed in subroutine LOAD for the radial model and ALOAD for the axial model.

The first load vector term in Equation 34 makes use of the $[B]$ matrix to transform internal element initial strains into nodal loads (Equation 45).

$$2\pi \int_{\text{volume}} [B]^T [C] \{\epsilon_0\} r dr d\theta = \{f\}_{\substack{\text{thermal} \\ \text{plastic} \\ \text{creep}}} \quad (45)$$

or for centroidal values:

$$\{f\}_{\substack{\text{thermal} \\ \text{plastic} \\ \text{creep}}} = 2\pi \bar{r} \Delta [\bar{B}]^T [C] \{\epsilon_0\} \quad (46)$$

The external load vectors $\{f\}_{\text{external}}$ are developed by integrating the surface tractions or pressures and developing equivalent nodal loads (Equation 47).

$$\int_{\text{surface}} [N]^T T ds = \{f\}_{\text{external}} \quad (47)$$

Due to the geometry that is used the development of the external loads become equivalent line loads and are pressure driven (Equation 48).

$$\{f\}_i^{\text{external}} = \frac{\text{Pressure Acting on Element Face} * \text{Face Area}}{\text{Total Length of Nodal Line on Boundaries of Element Face (2 Nodes)}} * \text{Length of Node Line (i)} \quad (48)$$

The same result is obtained as if the total force on the exterior surface of the element had been lumped at its exterior nodes. This process of lumping is carried on throughout the program.

V.B.7 Solution Procedure

Having formulated the global stiffness matrix and the loads, the remaining step is to solve for displacements and translate these back into element stresses and strains. The solution routine used is DMCHB, which takes advantage of the banded symmetric nature of the stiffness matrix. If contact has not occurred, the factored form of $[K]$ from the first iteration is retained, which means that the solution only involves one matrix multiplication for each D.O.F. Solutions for the radial model are called from subroutines SOLVET and ASOLVE for the axial model.

V.B.8 Incrementation of Strains and Stresses

Incremental elastic strains and stresses are added to the total strain array after temperature convergence. Then creep/plastic strains are calculated and fed through the entire chain of radial/axial calculations to produce corrections to the stress and strain values. This process may have to be repeated several times. The solution is based on the concept that total strain (a) ϵ_T can be described as (Equation 49a).

$$\epsilon_T = \epsilon_{\text{elastic}} + \epsilon_{\text{creep}} + \epsilon_{\text{thermal}} + \epsilon_{\text{plastic}} \quad (49a)$$

and that for any one incremental step the strains and associated load vectors will be separated as follows (Equation 49b):

$$\left. \begin{aligned} \epsilon_{T_i} &= \epsilon_{\text{elastic}_i} + \epsilon_{\text{thermal}_i} \\ \epsilon_{T_{i+1}} &= \epsilon_{\text{elastic}_{i+1}} + \epsilon_{\text{plastic}_{i+1}} \\ \epsilon_{T_{i+2}} &= \epsilon_{\text{elastic}_{i+2}} + \epsilon_{\text{creep}_{i+2}} \end{aligned} \right\} (b) \quad (49b)$$

-
- (a) Stresses for the radial model are computed in subroutine STRES and in ASTRES for the axial model.
- (b) The elastic stresses/strains for the $i + 1$ and $i + 2$ steps change the elastic stress and strain distributions obtained as a result of the imposed plastic or creep strains. The total strain at the end of step T will be a sum of $\epsilon_{T_i} + \epsilon_{T_{i+1}} + \epsilon_{T_{i+2}}$ with external loads and thermal loads from the ϵ_{T_i} increment and the plastic and creep effects from the $i+1$ and $i+2$ increments.

The state of stress can be computed at the conclusion of a given time step or load step and the corrections to the stress distribution is as follows (see Equation 50):

Thermal + Elastic

$$\{\sigma\}_i = [C]\{\epsilon_T\} - [C]\{\epsilon\}_{\text{thermal}}$$

Creep

$$\{\sigma\}_i = [C]\{\epsilon_T\} - [C]\{\epsilon\}_{\text{creep}}$$

Plasticity

$$\{\sigma\}_i = [C]\{\epsilon_T\} - [C]\{\epsilon\}_{\text{plastic}} \quad (50)$$

Where the total strain and stress vectors for the current point in time is a summation of the individual components (see Equation 51).

$$\begin{aligned} \{\epsilon_{\text{total}}\} &= \sum_{i=1}^n \{\epsilon_T\}_i \\ \{\sigma_{\text{total}}\} &= \sum_{i=1}^n \{\sigma\}_i \end{aligned} \quad (51)$$

V.B.9 Anisotropic Behavior

Anisotropy in plastic and creep deformation has a significant effect on the resulting deformation. The anisotropy model that is used is based on Hills⁽⁶⁾ concept of changing the definition of the effective stress and strain terms, and in addition to change the coefficients in the Prandtl-Reuss equations. The modified Prandtl-Reuss equations allocate the inelastic strain increment normal to the yield surface as a function of the stress state.

The definition of the anisotropy parameters change as a function of direction of testing. The equations for this application are referenced to the axial direction. This means that anisotropic strain characterizations should be based on uniaxial-axial material test results. A similar set of equations can be developed for any of the other three directions.

For the isotropic case, the effective stress is given by Equation 52.

$$\sigma_e = \frac{1}{\sqrt{2}} [(\sigma_{11}-\sigma_{22})^2 + (\sigma_{22}-\sigma_{33})^2 + (\sigma_{33}-\sigma_{11})^2]^{1/2} \quad (52)$$

where:

1 = axial

2 = hoop

3 = radial

The effective strain is given by Equation 53.

$$d\epsilon_e^P = \frac{\sqrt{2}}{3} [(d\epsilon_{11}-d\epsilon_{22})^2 + (d\epsilon_{22}-d\epsilon_{33})^2 + (d\epsilon_{33}-d\epsilon_{11})^2]^{1/2} \quad (53)$$

The Prandtl-Reuss equations are as follows:

$$\begin{aligned} d\epsilon_{11}^P &= \frac{d\epsilon_P}{\sigma_e} [\sigma_{11} - 1/2(\sigma_{22} + \sigma_{33})] \\ d\epsilon_{22}^P &= \frac{d\epsilon_P}{\sigma_e} [\sigma_{22} - 1/2(\sigma_{11} + \sigma_{33})] \\ d\epsilon_{33}^P &= \frac{d\epsilon_P}{\sigma_e} [\sigma_{33} - 1/2(\sigma_{22} + \sigma_{11})] \end{aligned} \quad (54)$$

For the case of anisotropy referenced to the axial direction, the effective stress is given by Equation 55. (Compare to Equation 52.)

$$\sigma_e = \left[\frac{R}{R+1} (\sigma_{11}-\sigma_{22})^2 + \frac{R}{P(R+1)} (\sigma_{22}-\sigma_{33})^2 + \frac{R}{R+1} (\sigma_{33}-\sigma_{11})^2 \right]^{1/2} \quad (55)$$

where:

$$R = \frac{\delta\epsilon_{hoop}}{\delta\epsilon_{radial}} \quad (\text{obtained from a uniaxial axial test})$$

$$P = \frac{\delta\epsilon_{axial}}{\delta\epsilon_{radial}} \quad (\text{obtained from a uniaxial hoop test})$$

$\delta\epsilon$ = an increment of creep or plastic strain

The effective strain is given by Equation 56. (Compare to Equation 53.)

$$d\epsilon_e^P = \frac{1}{G} [a_1(a_2 d\epsilon_{11} - a_3 d\epsilon_{22})^2 + a_2(a_3 d\epsilon_{22} - a_1 d\epsilon_{33})^2 + a_3(a_1 d\epsilon_{33} - a_2 d\epsilon_{11})^2]^{1/2} \quad (56)$$

where:

$$G = \frac{R^2 + R + P}{P(R+1)^2}$$

$$a_1 = \frac{R}{R+1}$$

$$a_2 = \frac{R}{P(R+1)}$$

$$a_3 = \frac{1}{R+1}$$

The modified Prandtl-Reuss flow equations are as follows (compare to Equation 54):

$$\begin{aligned} d\epsilon_{33}^P &= \frac{d\epsilon_P}{\sigma_e} \left[\frac{P+R}{P(1+R)} \sigma_{33} - \frac{R}{P(1+R)} \sigma_{22} - \frac{1}{1+R} \sigma_{11} \right] \\ d\epsilon_{22}^P &= \frac{d\epsilon_P}{\sigma_e} \left[\frac{R(P+1)}{P(R+1)} \sigma_{22} - \frac{R}{P(R+1)} \sigma_{33} - \frac{R}{R+1} \sigma_{11} \right] \\ d\epsilon_{11}^P &= \frac{d\epsilon_P}{\sigma_e} \left[\sigma_{11} - \frac{R}{R+1} \sigma_{22} - \frac{1}{R+1} \sigma_{33} \right] \end{aligned} \quad (57)$$

For the case where the material is isotropic, then $R = 1$, $P = 1$, and Equations 55, 56, and 57 become equal to Equations 52, 53, and 54.

These equations are found subroutine EFFECT, CREEP, and RATE.

V.B.10 Creep

The solution routine as indicated in Equation 49 breaks the procedure into elastic and non-elastic segments. The elastic computations are completed first and then followed by creep and plastic corrections. The creep loop is designed to handle primary and secondary creep conditions and is based on computing increments of inelastic strain.

The subroutine RATE called by CREEP computes an array of incremental strains $d\epsilon_e^P$ using either a strain hardening or time hardening option. (See Figure 12.) It is very general in nature and does not require a closed solution explicit in strain. The solution for a strain hardening calculation uses a modified Newton Rapheson iteration technique.

The modified Prandtl-Reuss equations are used to develop initial strain vectors for each element $\{\epsilon_0\}$. Using Equation 46, a set of creep load vectors are developed that, when applied to the finite element structure, will cause it to deform. The computed strains are the total strains caused by the increments of creep strain and the corrections to the total state of stress are the differences between the initial strains $\{\epsilon_T\}$ as indicated by Equation 50.

The creep equation for Zircaloy used in RATE is due to P. J. Pankaskie and was developed in reference (9):

$$\dot{\epsilon} = A[1 + \alpha \cdot k \exp(-k \cdot t)] \cdot \exp(EAE) \cdot [(1 - \tanh(B \cdot \Delta T)) \cdot \sinh(S_1 \cdot \sigma) + (1 + \tanh(B \cdot \Delta T)) \cdot \sinh(S_2 \sigma)]/2 \quad (58)$$

FIGURE 12. Comparison of Strain Hardening and Time Hardening Options Available in Subroutine RATE

where EAE is the effective activation energy consisting of the sum of a high temperature thermal activation energy term (TAE), and a fast flux or low temperature activation energy term (FAE) as follows:

$$\text{TAE} = -[C(1-\tanh(B\Delta T)) + (D-0.038\sigma)(1+\tanh(B\Delta T))]/2RT$$

$$\text{FAE} = (E-F)(1+\tanh(B\Delta T))/2$$

If $\phi > 0$, $E = n \cdot \ln \phi$

$$\phi = 0, E = (C-0.061\sigma)(3-T_0/T)/4RT_0$$

$$\phi = \phi_{th} \cdot F_R$$

$$T_0 = 638.5(1+0.021 \cdot \ln(1+\phi/6.0E12)) \cdot \text{Exp}(-7.18E-7 \cdot \sigma)$$

$$\Delta T = T_0 - T$$

where: ϕ = fast flux, ($E \geq 1$ Mev)

ϕ_{th} = thermal flux

F_R = user-supplied ratio

σ = effective stress, psi

T_0 = transition temperature from the low temperature regime where creep is activated by both temperature and fast flux and the high temperature regime where creep is activated, for the same applied stress, by temperature alone

$\alpha = 3310$, hr

$k = 0.0044$, hr⁻¹

$A = 5.96 \text{ E}14$, hr⁻¹

$B = 100.0$, °K⁻¹

$C = 63600.0$, cal/mole °K

$D = 9500.0$, cal/mole °K

$F = 65.77$

$n = 0.85$

$S_1 = 1.05 \text{ E-}4$ psi⁻¹

$S_2 = 1.0 \text{ E-}5$ psi⁻¹

The creep rate for stainless steel was developed from reference (10), and is as follows:

$$\dot{\epsilon} = A e^{-B/T} (\sinh \alpha \sigma)^n \quad (59)$$

where: $B = 17576$

$n = 24.608 - 0.01923T$

$$\alpha = \frac{(-5.225 \times 10^{-4} - 8.716 \times 10^{-7} T)}{(23.55 - 0.01796T)}$$

$A = 1.609 \times 10^5$

The code presently has limitations on time step size and that restrict the amount of creep strain in any one time step to a stable value. If instability should result even with these safeguards, a reduced time step might be needed. As a rule of thumb, creep strains resulting in strains in a given time step of more than 5% or 10% of the elastic strain can result in unstable behavior.

V.B.11 Plasticity

The calculation of plastic strains is based on using the same initial strain method as is used for the creep calculation. The solution procedure however is implemented in a different manner and the criteria controlling the calculation are significantly altered.

The plasticity loop calculation is performed at the start of a creep loop. It is exercised when the stresses obtained from the just completed elastic load step have driven the stresses in one or more elements above the yield stress. A test is made for each element based on the temperature and generalized strain in each element for that time.

As an example, consider the four-element cladding model shown in Figure 9. For some loading conditions, the stress for this four-element model are shown in Figure 13a. At the conclusion of the just completed elastic step, the stresses are shown above the yield stress curve. The stresses at this point are then reduced back to the starting point shown by the stress conditions below the line. The total load step is then divided into substeps such that the first substep just brings the most plastic element to the yield surface. At this point, a second load substep is taken that will bring the second element to the yield surface. At the completion of this substep, a corrective plastic strain increment is computed using the initial strain load vector. The strain that is incurred by element 4 as it leaves the yield surface is considered as plastic for both the pseudo elastic and plastic portions. This procedure is repeated until all four elements are plastic as is shown when element 1 is brought to the yield surface. A corrective plastic load step is taken and then the final portion of the original load step can be taken driving all elements above the yield surface. This step is concluded with a final plastic load step to correct the stress back to yield condition (Figure 13b).

FIGURE 13a. Elements 1, 2, 3 and 4 are Plastic and are Brought Back to Prior Subplastic State

FIGURE 13b. Showing the Plastic Strain Taken by Alternating Plastic and Elastic Steps

In actual practice, the oscillation of the elements below the yield surface after a plastic corrective substep is less than 500 psi out of 40,000 to 60,000 psi. The actual spread between stresses decreases as more plastic substeps are taken and in the limit; elements 1 through 4 would be at the same stress at the conclusion of the load step if the material were perfectly plastic. The plastic strain is accumulated for each of the substeps and provides the basis for computing the plastic strain distributions through the cladding wall.

At the conclusion of the load step, all elements will have been returned to a stress condition on the yield curve or slightly below. At this time a creep step is then taken in a normal manner completing the load and time step. For each substep, both plastic and creep, the solution procedure uses the chained radial axial radial sequence to compute the displacements.

The yield stress (Figure 14) for Zircaloy is calculated from an equation due to P. J. Pankaskie⁽¹¹⁾ and related to data in his report. It is as follows:

$$\text{yield stress} = C(T) \sigma^n$$

$$\text{where: } C = 128260 - 146.5T + 0.1378T^2 - 7.28 \times 10^{-5}T^3$$

$$n = 1.75 \times 10^{-2} + 3.58 \times 10^{-4}T - 7.638 \times 10^{-7}T^2 - 4.95 \times 10^{-10}T^3$$

σ = effective stress

T = temperature in degrees Kelvin

FIGURE 14. Yield Stress Versus Strain used in GAPCON-3 Calculations for Temperature Range 550°F to 700°F

V.B.12 Fuel Thermal Expansion

The thermal expansion of the fuel is treated in a manner commensurate with the current level of understanding of its behavior. Prior to contact the dimensions of the fuel are basically unknown, the pellet structure is cracked, and the pellet fragments are subject to relocation. The result of this is that only an equivalent fuel dimension can be calculated based upon temperature and thermal expansion data. The relationship between the computed equivalent dimensions and the actual location of the fuel due to thermal expansion can only be inferred.

Prior to contact the fuel radial thermal expansion model used in GAPCON-3 is the "stress free" thermal expansion in which each of 50 concentric rings of fuel is allowed to expand without restraint, and the total is summed to produce the total movement of the outer radius. The basis for this model is the assumption that the radial cracks which are known to form in fuel upon the first rise to power will relieve the hoop and radial stresses arising from the thermal gradient, resulting in nearly stress-free expansion. (See reference 12.)

After contact the integral finite element solution for the fuel is used to compute the radial motion. This model again uses a thermal stress-free expansion where a cracked geometry is assumed to relieve the thermal stresses within the pellet and only interfacial restraints are assumed.

The axial thermal expansion of the relative fuel/cladding stack length comes from the axial model for both regimes.

V.C. MODELS FROM GAPCON-2 WHICH HAVE BEEN ALTERED

Fission Gas Release

Both the low and high temperature fission gas release models that were developed for GAPCON-THERMAL-2 are used in GAPCON-3. The low temperature model proposed by Bellamy and Rich⁽¹³⁾ includes a fuel surface to volume ratio (S/V) term which is exponentially dependent on burnup, releasing additional gas from all fuel which is below 1200°C. Because the S/V relationship used in GAPCON-THERMAL-2 can give excessive gas release value, whereas the original

model was based on data up to 8% gas release at 5 at. % burnup, the low temperature gas release model in GAPCON-3 sets an upper limit of 0.08 on the low temperature gas release fraction. Otherwise the two codes use identical gas release models.

There is, however, increasing evidence that high burnups enhance fission gas release. There is not a well verified model to quantitatively describe such release in light water reactor fuel. GAPCON-3 has an option: an NRC-developed fission gas release enhancement factor for burnups greater than 20,000 MWD/MTM.

A high burnup addition factor is used. The factor is the product of two terms, FRETN and BRNEF, where

$$\text{FRETN} = 1 - \text{RDOT}$$

$$\text{BRNEF} = \frac{1 - e^{-0.0000436 (\text{Burnup}-20,000)}}{1 + \frac{0.665}{\text{RDOT}} e^{-0.0001107 (\text{Burnup}-20,000)}}$$

and RDOT, gas release fraction, is just the sum of the low and high temperature release models at 20,000 MWD/t.

For a discussion of the low and high temperature gas release models, the reader is referred to Reference 2.

Table 6 gives GAPCON-3 calculated local gas release fractions with and without the high burnup gas release factor for the peak power region of a PWR sized, prepressurized fuel rod operating at a local power of 11.7 kW/ft.

Fuel Thermal Expansion (Subroutine EXPAND)

The thermal expansion model used in GAPCON-3 is the "stress free" thermal expansion in which each of 50 concentric rings of fuel is allowed to expand without restraint, and the total is summed to produce the total movement of the outer radius. The basis for this model is the assumption that the radial cracks which are known to form in fuel upon first rise to power will relieve the hoop and radial stresses arising from the thermal gradient, resulting in nearly stress-free expansion.

TABLE 6. Effect of High Burnup Gas Release Factor

Local Burnup, MWD/MTM	High Burnup Local Gas Release Fraction	
	Without Factor	With Factor
20,000	0.0116	0.0116
20,460	0.0118	0.0122
21,450	0.0112	0.0135
22,440	0.0125	0.0150
23,430	0.0129	0.0168
24,420	0.0132	0.0188
25,410	0.0136	0.0213
26,400	0.0139	0.0241
27,390	0.0143	0.0275
28,380	0.0146	0.0314
29,370	0.0149	0.0360

Fuel Swelling (Subroutine FLUXD)

The fuel swelling model used in GAPCON-3 is the same as for GAPCON-2 (Geithoff model)⁽¹⁴⁾ except that no porosity is assumed to be consumed by the swelling prior to fuel-cladding contact. Thus

$$\frac{\Delta V(\%)}{V} = [H_f V_f(f) + h_c V_c] \frac{BU}{10,000}$$

where $h_f = 1.6\%$

V_f = volume of fuel/unit length

V_c = volume of fuel between 1300 and 1700°C

$h_c = 0.4\%$

BU = burnup, MWD/MTM.

Flux Depression (Subroutine FLUXD)

As in GAPCON-THERMAL-2, the thermal neutron flux depression is calculated as proportional to a modified Bessel function of zero order, as derived by Robertson.⁽¹⁵⁾ The argument of this function is the radius times the effective inverse diffusion length (Kappa). As a result of comparison to more exact BRT-1 code predictions, we have modified our Kappa values (which are estimated from 2200 m/sec cross sections) by the factor 1.6.

VI. VERIFICATION RUNS

The first parts of this section present thermal and mechanical results obtained with GAPCON-3 for specific, simplified problems for which analytical solutions are well known. The last part presents code predictions versus in-reactor experimental data for centerline temperature, fission gas release, cladding creep-down, and axial elongation.

VI.A TEMPERATURE CALCULATIONS

The temperature distribution in a solid cylindrical pellet (radius r_{FS}) with constant volumetric heat generation rate Q throughout and a fixed thermal conductivity, K , is known to be

$$T(r) = T_{FS} + \frac{Q}{4K} (r_{FS}^2 - r^2) \quad (1)$$

where T_{FS} is the fuel surface temperature.

In normalized coordinates,

$$T_N(y) = 1 - y^2 \quad (2)$$

where

$$T_N = \frac{T - T_{FS}}{T_O - T_{FS}},$$
$$T_O = T_{FS} + \frac{Qr_{FS}^2}{4K},$$

and $y = \frac{r}{r_{FS}}.$

Figure 15 shows the steady-state results of GAPCON-3 for a solid pellet with constant heat generation and conductivity, plotted in normalized coordinates. Also shown on the same graph are the results at the same power level and surface temperature but with 1) flux depression corresponding to 10%

FIGURE 15. Normalized Fuel Radial Temperature Profiles

enrichment, 2) a temperature-dependent thermal conductivity (Lyons equation), and 3) a hole in the center of the fuel. The result has been normalized to the $(T_0 - T_{FS})$ from the previous example to emphasize the flattening effect of the hole and flux depression. The solution, although not exact, can be brought to any desired accuracy by iteration. The line plotted is accurate within 0.5°C , and as the figure shows the agreement with GAPCON-3 is as good for the previous problem--within 1°C in both cases.

Now let us consider a transient problem for which an analytical solution exists. Consider a solid pellet with radius r_{FS} , conductivity K , diffusivity Y , going from zero volumetric heat generation to Q watts/cm³ at time $t = 0$. Prior to $t = 0$, the entire pellet temperature $T = T_{FS}$; after $t = 0$, the surface temperature remains fixed at T_{FS} . The solution for temperature as a function of time and radius can be shown to be⁽¹⁶⁾

$$T = T_{FS} + \frac{Q}{4K} (r_{FS}^2 - r^2) - \frac{2Q}{r_{FS}K} \sum_{n=1}^{\infty} \frac{J_0(\alpha_n r) e^{-Y \alpha_n^2 t}}{J_1(\alpha_n r_{FS}) \alpha_n^3} \quad (3)$$

here α_n are roots of $J_0(\alpha_n r_{FS}) = 0$. J_1 and J_0 are ordinary Bessel functions.

In normalized coordinates,

$$T_N = 1 - y^2 - 8 \sum_{n=1}^{\infty} \frac{J_0(\alpha_n y) e^{-\alpha_n^2 \tau}}{\alpha_n^2 J_1(\alpha_n)} \quad (4)$$

where α_n are roots of $J_0(\alpha_n) = 0$

and

$$\begin{aligned} T_N &= \frac{T - T_{FS}}{T_o^f - T_{FS}} \\ (T_o^f - T_{FS}) &= \frac{Q r_{FS}^2}{4K} \\ y &= r/r_{FS} \\ \tau &\equiv Yt/r_{FS}^2 \end{aligned} \quad (5)$$

Figure 16 shows the results of a jump from 0 to 10 kW/ft at approximately 3, 9, and 18 seconds. The sum of exponentials was carried out eight terms which brought the accuracy to within 0.1°C at the times considered. The agreement with GAPCON-3 is again within 1°C.

VI.B STRESS CALCULATIONS IN THE CLADDING

Figures 17 and 18 show the stress response of the cladding to a pure external pressure load (no thermal gradient). For this situation the stress distributions are known to be⁽¹⁷⁾

FIGURE 16. Fuel Transient Radial Temperature Profiles

FIGURE 17. Cladding Response to Pressure Load (Radial Stress)

FIGURE 18. Cladding Response to Pressure Load (Hoop Stress)

$$\begin{aligned}\sigma_{rad} &= - \frac{P_o}{1 - R^2} \left(1 - \frac{R^2}{Y^2} \right) \\ \sigma_{hoop} &= - \frac{P_o}{1 - R^2} \left(1 + \frac{R^2}{Y^2} \right)\end{aligned}\tag{6}$$

where P_o = external pressure

$Y = r/r_{CS}$, r_{CS} = cladding outer radius

$R = r_{CI}/r_{CS}$, r_{CI} = cladding inner radius.

The stresses match the expected value within 1%.

Figures 19 and 20 show the results of a temperature gradient (logarithmic) of 115°C across the cladding. For fixed mechanical and thermal properties, the solution is known to be⁽¹⁷⁾

FIGURE 19. Cladding Response to Temperature Gradient (Radial Stress)

FIGURE 20. Cladding Response to Temperature Gradient (Hoop Stress)

$$\sigma_{\text{rad}} = - \frac{E\alpha\Delta t}{2(1-\mu) \log(1/R)} \left[\ln \frac{1}{Y} - \left(\frac{R^2}{1-R^2} \right) \left(1 - \frac{1}{Y^2} \right) \ln \frac{1}{R} \right] \quad (7)$$

$$\sigma_{\text{hoop}} = \frac{E\alpha\Delta t}{2(1-\mu) \ln(1/R)} \left[\ln \frac{1}{Y} - \left(\frac{R^2}{1-R^2} \right) (1 + Y^2) \ln \frac{1}{R} \right]$$

where Y and R have the same meaning as before, and

E = Young's modulus

α = coefficient of thermal expansion

$\Delta t = T_{\text{CI}} - T_{\text{CS}}$

μ = Poisson's ratio.

The hoop stresses agree within 1%, as before. The radial stresses are more scattered, but their small absolute value must be taken into account.

VI.C COMPARISON WITH IN-REACTOR DATA

A few examples will be given which compare GAPCON-3 to in-reactor data to demonstrate both the capabilities and limitations of the program. Detailed input lists for each run are compiled in Appendix B.

HPR-80 Fuel Centerline Temperature Data

The beginning-of-life fuel centerline temperature data for Halden irradiation tests IFA-11 and IFA-21 have provided a classic bench mark for many fuel performance codes, including previous versions of GAPCON. The data are presented in Halden Project Report HPR-80.⁽¹⁸⁾ In these well characterized tests, the full centerline temperature was measured by means of W-Rh thermocouples, and local linear heat rating was deduced by gamma thermometers calibrated in-reactor. Figures 21 and 22 present data from two of the rods and show not only the data and GAPCON-3 predictions but also the GAPCON-THERMAL-2 predictions.

HPR-107 Temperature and Elongation Data

The IFA-116 and IFA-117 assemblies in Halden also carried centerline thermocouples, but in addition had elongation detectors on the cladding. The data are reported in HPR-107.⁽¹⁹⁾ Figure 23 shows typical temperature

results from the beginning of life for two different assumptions of initial (fabricated) gap size: 1) the nominal gap of 8.8 mils and 2) the minimum gap size (accounting for tolerances in fuel and cladding) of 5.0 mils. The two inputs appear to bracket the data as shown. Figure 24 shows the results for cladding elongation, assuming the minimum gap size. It should be noted that no pellet-cladding contact was predicted by the code with the nominal gap size. With the minimum gap size, we see from Figure 23 that the power at which contact occurs is predicted well in this case (as indicated by the slope change in the elongation curve). However, the slope of axial elongation as a function of increasing power after contact is too large. We attribute this to a lack of pellet-cladding slip in the current GAPCON-3 model.

Maine Yankee Rod (EPRI Case D)

EPRI has presented fission gas release data, cladding deformation data, and microstructure data from a group of Maine Yankee Rods.^(20,21) In a fuel performance code comparison, they presented data from one of the rods as Case D. Figure 25 shows the GAPCON-3 temperature history of this rod, as predicted from EPRI standard input for this case.⁽²¹⁾ Also shown in EPRI's estimate is the value and burnup location of the peak temperature achieved at the 36-inch elevation along the rod. The error bars represent our own estimate of the possible uncertainty (2σ limits) on this temperature, which was inferred from equiaxed grain growth.

Other data from this source include measurements of end-of-life fission gas release and cladding creep down. Comparison to GAPCON-3 is indicated in Table 7.

TABLE 7. End-of-Life Data Comparison

	<u>Data</u>	<u>GAPCON-3</u>
End-of-life gas release, %	13.4	5.6
Diametrical change in cladding dimension at 36-inch elevation, mills	2.4	1.9

One reason the gas release figure is low is that the input power profile and peak power may be low. There is indication in the grain growth data that the actual power experienced by the rod may have been higher than the nominal value used in the EPRI report.

IFA-431 Rod 3 Ramps

Data from the NRC-sponsored Halden test IFA-431 have been released⁽²²⁾ and can be compared to GAPCON-3. One of the rods showing pellet-cladding interaction (PCI) from this test was rod 3 which had a 2-mil initial diametrical gap. Figures 26 and 27 show the comparison of centerline temperature and cladding elongation, respectively, from the first rise to power. On the temperature plot we see GAPCON-3 overpredicting the data slightly. The prediction comes closer to the data in subsequent power ramps (c.f. Figure 28), presumably because of decreased thermal conductivity as a consequence of pellet cracking. The elongation plot (Figure 27) shows up the same shortcoming as previously noted; whereas, the elongation at full power is predicted reasonably well but the slope of the elongation versus power is too steep.

Rod 3 Transient

As a final check on the transient capability of the code, Figure 29 is presented. The figure shows centerline temperature data and predictions for a 20% power drop in 30 seconds. Because there is a slight mismatch in the initial conditions between the data and the computer run, both are presented in normalized coordinates. The normalized temperature is defined here as

$$T_N = \frac{T_o - 240}{T_i - 240}$$

where T_o is the fuel center temperature and T_i is its initial value. The data initial conditions were a centerline temperature of 890°C at a power of 10.5 kW/ft, with a coolant temperature of ~240°C.

The slight overestimation of the speed of the temperature fall is probably due to an underestimation of the fraction of the total conductance due to solid contact. For rod 1, where there is no solid-solid contact, the code matches the data more closely, as shown in Figure 30.

FIGURE 21. Comparison to HPR-80 Thermocouple Data

FIGURE 22. Comparison to HPR-80 Thermocouple Data

FIGURE 23. Comparison to HPR-107 Thermocouple Data

FIGURE 24. Comparison to HPR-107 Elongation Data
(Solid symbols indicate increasing power,
open symbols indicate decreasing power)

FIGURE 25. Comparison to Estimated Life History of Maine Yankee Rod

FIGURE 26. Comparison to IFA-431 Thermocouple Data (Initial Rise to Power)

FIGURE 27. Comparison to IFA-431 Elongation Data (Initial Rise to Power)

FIGURE 28. Comparison to IFA-431 Thermocouple Data

FIGURE 29. Comparison to IFA-431 Rod 3 Transient Data

FIGURE 30. Comparison to IFA-432 Rod 1 Transient Data

APPENDIX A

This appendix contains the input instructions for GAPCON-3.

INPUT INSTRUCTIONS

The GAPCON-3 code requires approximately 220K of addressable core storage. Run time will depend on the particular computer, the number of steps, and the number of axial and radial nodes. A typical case will take 2 to 5 minutes on the CDC 6600.

GAPCON-3 uses a combination of format and namelist input. This combination minimizes errors in input and simplifies restarting the problem from some predetermined point. The mandatory and optional inputs have been divided into data blocks which are described in the following pages. Preceding the data blocks is a mandatory TITLE card which contains the title of the case (in columns 2 through 40). A non-zero integer entry in columns 46 through 50 will activate standard problem input for system checkout, and subsequent cards are ignored. Certain variables have default values, and these are noted in Table A.2.

DATA BLOCK 1 - NAMELIST INPUT

<u>Card Number</u>	<u>Column Numbers</u>	<u>Variable</u>	<u>Definitions and Comments</u>
1	2	--	A dollar sign (\$) is required in this position.
	3-7	--	The word "INPUT" is required in this position.
2 (and following)	2-72	--	The NAMELIST variables are listed in Table 1 and defined in Table 2. They are entered as simple algebraic statements separated by commas; e.g., FRDEN = .92, FRISIN = .98, etc. Certain variables, indicated by (1) in Table 1, are arrays of numbers, with the array entries separated by commas; e.g., TIME (1) = 0., 60., 90., 100., etc. If the array cannot be contained on one card, initiate following cards with the variable name; e.g., TIME (15) = 303., where 303. is the 15th entry in the TIME array, and the 14th entry was the last value on the preceding card.
last card	--	--	A (\$) must appear somewhere in columns 2-72 after the last NAMELIST variable.

DATA BLOCK 2 - Flux Depression Card (Omit if NFLX is not greater than zero)

<u>Card Number</u>	<u>Column Numbers</u>	<u>Variable</u>	<u>Definitions and Comments</u>
1	1-10	ZZ	If NFLX is greater than 1, the program reads in flux depression parameters to be used in the relation: Heat generation = $ZZ + YY \cdot R^{**2} + WW \cdot R^{**4}$, where R is radial measure in centimeters.
	11-20	YY	
	21-30	WW	

DATA BLOCK 3 - Cladding Properties (Omit if NCLAD is not greater than zero)

<u>Card Number</u>	<u>Column Numbers</u>	<u>Variable</u>	<u>Definitions and Comments</u>
1	1-10	AA(1,1)	A temperature, in °F, which cladding properties are to be evaluated.
	11-20	AA(1,2)	Cladding conductivity, Btu/hr-ft-°F.
	31-40	AA(1,4)	Young's modulus, psi.
	41-50	AA(1,5)	Poisson's ratio.
	51-60	AA(1,6)	Integrated thermal expansion relative to 70°F reference, per °F.
	61-70	AA(1,7)	Meyer hardness, in kg/cm ² .

Card 2 and all following cards are identical in format and entries to card 1. The entries must be E 10.4 format, and a total of NCLAD cards must be included in this block, in ascending or descending temperature sequence. Code will then use these input cladding properties.

DATA BLOCK 4 - Fuel Properties (Omit if NFUEL is not greater than zero)

<u>Card Number</u>	<u>Column Numbers</u>	<u>Variable</u>	<u>Definition and Comments</u>
1	1-10	CF(1,1)	A temperature in °F at which fuel properties will be evaluated.
	31-40	CF(1,4)	Young's modulus, psi.
	41-50	CF(1,5)	Poisson's ratio.
	51-60	CF(1,6)	Integrated thermal expansion relative to a reference temperature of 70°F, per °F.
2	1-10	CF(1,2)	A second temperature, °F.

Card 2 and all following cards are identical in format and entries to card 1. The entries must be in E 10.4 format. A total of NFUEL cards must be included in this block, in ascending or descending temperature sequence. Code will then use these input fuel properties.

NFUEL must not exceed 7.

DATA BLOCK 5 - Time Step Option (Omit if IOPT is zero)

The purpose of this block is to permit the variation of certain NAMELIST variables with each time step. The items that may be varied are the external (system) pressure, (EXTP), the film coefficient (HF) (the code otherwise assumes 15,000 Btu/hr-sq.ft.-F), the rod power, the axial power profile, and the time step size. Thus one has the opportunity to make the constant EXTP time dependent, to make the constant HF both axially and time dependent, to make the array PROFIL(1) time dependent, and to redefine the power-time history of the rod. If IOPT is greater than zero, at least one card is required for each time step.

<u>Card Number</u>	<u>Column Numbers</u>	<u>Variable</u>	<u>Definitions and Comments</u>	<u>Format</u>
1	1-10	EXTP	External pressure, in PSI.	F
	11-15	IPRFL	Non-zero entry indicates that (NPOW + 1) values of PROFIL will be given on the next card.	I
	16-20	IHF	Non-zero entry indicates that (NPOW + 1) values of HF, one each for the fueled boundaries, will appear on the next card after the last PROFIL card.	I
	21-25	ITM	A non-zero entry signals a change in the power-time history.	I
	31-40	TDL	The new time step size (days).	F
	41-50	POWR	The new rod power (kw/ft)	F
2 (omit if IPRFL = 0)	--	PROFIL	The values of PROFIL are entered under an 8F10.4 format.	F
3 (omit if IHF = 0)	--	HF	Values of HF are entered under an 8F10.4 format in units of Btu/ft ² -F-hr.	F

At least one card must be included for each time step, containing as a minimum the external pressure (EXTP) entry.

TABLE A.1. NAMELIST Variables

ATMOS	ITAPE	XH
CP	ITMCUT	XN
CR	JSTART	Z
DCI	KSAVE	ZCLAD
DCO	LFUEL	
DELZA (1)	LVOIDZ	
DFS	MINI (1)	
DSINZ	MXPOWR	
DTEMP	NC	
DVOIDZ	NCLAD	
EXTP	NF	
FLXRAT	NFLX	
FRACAR	NFUEL	
FRACH	NOH	
FRACHE	NPOW	
FRACKR	NSTEPS	
FRACN	NSH	
FRACXE	NTIME	
FRDEN	PRCDH	
FRDEN2	PROFIL (1)	
FRPUO ₂	PSEUDO (1)	
FRSIN	RADS	
FR35	RNEWK (1)	
FR40	ROUC	
FR41	ROUF	
IDENSF	S	
IGAS	TIME (1)	
IOPT	TINLET	
IPEAK	TM	
IPLLOT	TPLAS	
IRELOC	VPLENZ	
IRL	X	
ISTOR	XCO	

TABLE A.2. NAMELIST Variables for GAPCON-3

<u>Variable</u>	<u>Description and Comments</u>
ATMOS	Fill gas pressure (atmospheres)
CP	Ratio of plastic strain increments $\epsilon_{\text{axial}}/\epsilon_{\text{RAD}}$, for uniaxial hoop load. Default value is 1.0.
CR	Ratio of plastic $\epsilon_{\text{Hoop}}/\epsilon_{\text{RAD}}$, for uniaxial load. Default value is 1.0.
DCI	Cladding inside diameter (inches).
DCO	Cladding outside diameter (inches).
DELZA (1)	An array of axial segment lengths, including the insulator region at the bottom and the non-fueled plenum region at the top. NPOW + 2 values should be input. If no values are input, the code assigns 0.5 inch to the bottom region, LFUEL/NPOW inches to each fuel region, and VPLENZ/PI*RCI**2 inches to the plenum.
DFS	Fuel pellet diameter (inches).
DSINZ	Initial diameter of restructured fuel (normally equals 0.) (inches).
TEMP	The axial ΔT across the core, (i.e., T outlet - T inlet). (See MXPOWR).
DVOIDZ	Diameter of initial central void in the fuel pellets (inches).
EXTP	Coolant pressure (psi).
FLXRAT	Ratio of fast (>1 Mev) to thermal flux used in creep equation. Default value is 0.01.
FRACAR	Fraction of initial fill gas which is argon.
FRACH	Fraction of initial fill gas which is hydrogen.
FRACHE	Fraction of initial fill gas which is helium.
FRACKR	Fraction of initial fill gas which is krypton.
FRACN	Fraction of initial fill gas which is nitrogen.
FRACXE	Fraction of initial fill gas which is xenon.
FRDEN	Fractional density of fuel pellet. (Fraction of TD)
FRDEN2	Maximum density of fuel after densification. (Fraction of TD)
FRPUO ₂	The weight fraction of the fuel which is PuO ₂ (the remainder is assumed to be UO ₂).
FRSIN	Density of restructured fuel. (Fraction of TD)
FR35	The weight fraction of the U which is ²³⁵ U (the remainder is assumed to be ²³⁸ U).
FR40	The weight fraction of the Pu which is ²⁴⁰ Pu.

<u>Variable</u>	<u>Description and Comments</u>
FR41	The weight fraction of the Pu which is ^{241}Pu . The remaining fraction of Pu is assumed to be ^{239}Pu .
IDENSF	A signal that allows the user to include the effects of radial fuel shrinkage on the fuel-to-cladding gap size due to isotropic densification. The code assumes the final density of the densified fuel to be FRDEN2 of theoretical. If IDENSF = 0, densification is not taken into account. If IDENSF $\neq$ 0, the change in fuel diameter due to densification is included.
IGAS	An integer signal to designate whether conservative or best estimate calculations are desired for gas release Set IGAS = 0 for best estimate $\neq$ 0 for conservative estimate (i.e., estimate is the 95% confidence boundary and yields more gas release).
IOPT	An integer which allows input of time dependent external pressure, film coefficient, and axial power profile, and allows the user to alter the input power history (see data block 4).
IPEAK	An integer to specify whether an average or peak power is to be input via PSEUDO. If IPEAK $\neq$ 0, an average power needs to be input; if IPEAK = 0, peak power needs to be input via PSEUDO.
IPLOT	An integer to initiate writing to a plot tape IPLOT = 0 yields no plot (KSAVE must be non-zero in order to save data for plotting). (See Appendix C)
IRELOC	A signal that allows the user to use the fuel relocation model developed for this code. If IRELOC = 0, change in fuel diameter due to relocation is 0 (i.e., relocation is not taken into account). If IRELOC > 0, the change in fuel radius due to relocation is taken into account by a best estimate value. If IRELOC < 0, the change in fuel radius due to relocation is taken into account by a conservative estimate; i.e., gives less gap closure.
IRL	An integer to specify the output of flux depression value (from subroutine FLUXD). If IRL = 0, 11 flux depression values and their respective pellet diameters will be printed out, with the first value given at the pellet centerline and the last at the pellet surface. If IRL > 0, the subroutine FLUXD will divide the fuel pellet into IRL equal nodes and print out the flux depression values and their appropriate diameters at the midplane of each node.
ITAPE	An integer signal allowing one to restart from and continue writing on the same tape.

<u>Variable</u>	<u>Description and Comments</u>
ITMCUT	An integer which specifies the total number of extra time steps the code may take in order to hold power changes after fuel-cladding contact has occurred to 1.5 KW/ft maximum. If contact is anticipated, it is recommended that ITMCUT be 10 or greater.
JSTART	An integer to be used only on restart cases which specifies the time step number where the restart is to begin.*
KSAVE	An integer which specifies the increment of time steps for which restart information will be written to tape. If KSAVE = 1, every time step is saved; if KSAVE = 3, every third time step is saved; etc. Note that the control cards for the run should include instructions for tape mounting if KSAVE is greater than 0.
LFUEL	Length of fuel column (inches).
LVOIDZ	Length of initial central void in the fuel pellets (inches).
MINI (1)	An integer array specifying the output. If MINI(1) = 0 or negative, only the peak power boundary is output. If MINI(1) > 0, all boundaries "J" are output for which MINI(J) > 0.
MXPOWR	The value of PSEUDO for which DTEMP is to be associated. The program will calculate a value for total coolant temperature rise along the rod as $DTEMP * PSEUDO / MXPOWR$
NC	The number of radial boundaries in the cladding in the finite element model. Default value is 4.
NCLAD	An integer signal to specify type of cladding. If NCLAD = 0, cladding is Zircaloy. If NCLAD < 0, cladding is 304SS. If NCLAD > 0, cladding properties are input as described previously with the number of points (temperatures) input equal to NCLAD.
NF	The number of radial boundaries in the fuel in the finite element model. Note that this is also the number of boundaries at which temperatures will be printed in the output. Default value is 6. NF should never be less than 4; and if the sum of NF and NC exceeds 10, the output per boundary will require more than one page.

* Note that the tape designation for a "start" run is TAPE 1, whereas it is TAPE 2 for a "restart" run.

<u>Variable</u>	<u>Description and Comments</u>
NFLX	An integer specifying the flux depression. If NFLX < 0, no flux depression occurs. If NFLX = 0, flux depression is calculated on the Robertson method, based on the U-235 content. Not valid for FR35 greater than 4. If NFLX > 0, flux is assumed to be of the form $ZZ + YY \cdot R^2 + WW \cdot R^4$ where ZZ, YY, and WW are input variables and R is the radius measured in cm. If NFLX > 0, NFLX must not exceed 20.
NFUEL	An integer specifying the fuel thermal conductivity and mechanical properties (elastic modulus, Poisson's ratio, and thermal expansion). If NFUEL < 0, the code automatically assigns UO ₂ property to the fuel, except that the thermal conductivity is lowered by 1% for each percent PuO ₂ , up to 5%. Note that if the PuO ₂ content exceeds 5%, the code automatically uses the CRBR formula unless another formula is input (with NFUEL > 0). If NFUEL = 0, the code uses UO ₂ mechanical properties and Lyons thermal conductivity. If NFUEL > 0, the code uses input mechanical properties and thermal conductivity (see input instructions). Limit is 7.
NOH	An integer signal to specify disposition of the hydrogen present in the sorbed gas. If NOH = 0, the hydrogen is assumed to be released from the rod. If NOH ≠ 0, the hydrogen is assumed to remain in the fuel rod as a gas.
NPOW	The number of axial fueled regions. Limit is 17.
NSTEPS	An integer to be used only on restarts which will redefine the total number of time steps in the problem including those previously taken.
NSH	Option for strain hardening. NSH = 1 for strain hardening, NSH = 0 for time hardening.
NTIME	Number of time increments. Total number of time steps > 1 day is limited to 49.
PRCDH	Percent of fuel column volume that is dish volume (i.e., 100 x total dish volume/total fuel column volume).
PROFIL (1)	A table that is used to input a normalized axial power profile for the rod. NPOW + 1 values for each profile needs to be input with the first and last values corresponding to the bottom and top of the fuel, respectively.
PSEUDO (1)	Power for each time step (KW/ft); allows the user to follow a power history. NTIME values need to be input. PSEUDO (1) and PSEUDO (2) may not = 0. (See IPEAK) LIMIT (100).
RADS	Radius of fuel pellet dish, inches.

<u>Variable</u>	<u>Description and Comments</u>
RNEWK (1)	An array of values for inputting fuel thermal conductivity, according to the form, $K_{\text{Fuel}}(\text{W/cm-c}) = R2 \frac{1. - P}{1. + R1*P} * \frac{R3}{R4 + TK} + R5*TK**3$
ROUC	Arithmetic mean cladding ID surface roughness (inches).
ROUF	Arithmetic mean fuel surface roughness (inches).
S	Fuel sorbed gas content (cc/g of fuel).
TIME(1)	A set of accumulative time increments (days) that allows the user to follow a power history. NTIME values need to be input. Limited to 100 values. <u>TIME(1) must = 0.</u> TIME(n) must be larger than previous time, TIME (n-1).
TINLET	Inlet coolant temperature, °F, associated with MXPOWR.
TM	Melting temperature of fuel (°C). If no value is input, the code uses 2790°C.
TPLAS	The temperature at which the fuel becomes plastic. If no value is input, the code uses 1200°C.
VPLENZ	Volume of gas plenum included in the fuel rod (cubic inches).
X	Fraction of fission products that are gaseous.
XCO	Fraction of sorbed gas that is carbon monoxide and carbon dioxide.
XH	Fraction of sorbed gas that is hydrogen and moisture.
XN	Fraction of sorbed gas that is nitrogen. Note: XCO + XH + XN should = 1.0 when S > 0.
Z	Fraction of fission product gas that is krypton.
ZCLAD	A signal to specify Zr-2 or Zr-4 cladding material properties. If ZCLAD > 0, cladding is Zr-4. If ZCLAD ≤ 0, cladding is Zr-2. Note: NCLAD must have a value of 0 to use ZCLAD.

* Note that during a transient (i.e., for any time step less than 5 minutes) the code automatically subdivides the power history into 0.1-second intervals using linear interpolation between the input values.

APPENDIX B

The following tables present the GAPCON-3 input for the comparisons to in-reactor experiments in the same order in which they occur in the text.

TABLE B.1. Rod HBC, HPR-80

ATMOS 1.	ITMCUT 20	XN 0
CP.	JSTART 0	WW 0
CR 1.	KSAVE 0	YY 0
DCI .4959	LFUEL 67.48	ZZ 0.
DCO .5394	LVOIDZ 67.48	Z .125
DELZA (1) .5,11.2,11.2,11.2, 11.2,11.2,11.2,.497	MINI (1) 1,0,3	ZCLAD -1
DFS .489	MXPOWR 15.24	
DSINZ 0.	NC 4	
DTEMP 20.	NCLAD 0	
DVOIDZ .0472	NF 6	
EXTP 500.	NFUEL 0	
FLXRAT 0.01	NFLX 0	
FRACAR -	NOH 0	
FRACH -	NPOW 6	
FRACHE 1.0	NSTEPS 0	
FRACKR -	NSH 0	
FRACN -	NTIME 7	
FRACXE -	PRCDH 0	
FRDEN .96	PROFIL (1) .719,.932,.1,.916,.737,.461,.186	
FRDEN2 .965	PSEUDO (1) .05,1.52,3.05,6.10,9.1,12.2,15.2	
FRPUO ₂	RADS 0.	
FRSIN .98	RNEWK (1) -	
FR35 .05	ROUC 0.5 E-4	
FR40 -	ROUF .1.6 E-4	
FR41 -	S 0	
IDENSF 0	TIME (1) 0.,.017,.035,.069,.099,.139,.174	
IGAS 0	TINLET 446.	
IOPT 0	TM (2790.)	
IPEAK 1.	TPLAS (1200)	
IPLOT 0	VPLENZ .096	
IRELOC 1	X .27	
IRL 0	XCO 0	
ISTOR	XH 0	
ITAPE		

TABLE B.2. Rod HBA, HPR-80

ATMOS 1.	ITAPE	XH 0.
CP 1.	ITMCUT 20	XN 0.
CR 1.	JSTART 0	WW 0.
DCI .4956	KSAVE 0	YY 0.
DCO .5394	LFUEL 67.48	ZZ 0.
DELZA (1) .5,11.2,11.2,11.2,	LVOIDZ 67.48	Z 0.125
DFS .4937 11.2,11.2,11.2,.497	MINI (1) 1,0,3	ZCLAD -1
DSINZ 0.	MXPOWR	
DTEMP 20.	NC 4	
DVOIDZ 0.0472	NCLAD 0	
EXTP 500.	NF 6	
FLXRAT	NFLX 0	
FRACAR -	NFUEL 0	
FRACH -	NOH 0	
FRACHE 1.0	NPOW 6	
FRACKR -	NSTEPS 0	
FRACN -	NSH 0	
FRACXE -	NTIME 6	
FRDEN 0.98	PRCDH 0.	
FRDENZ .965	PROFIL (1) .719,.932,1.,.916,.737,.461,.186	
FRPUO ₂ -	PSEUDO (1) .05,3.,5.,7.,10.,15.	
FRSIN 0.98	RADS	
FR35 .05	RNEWK (1)	
FR40	ROUC 0.5 E-4	
FR41	ROUF 1.6 E-4	
IDENSF 0	S	
IGAS 0	TIME (1) 0.,0.017,.035,.099,.137,	
IOPT 0	TINLET 446	
IPEAK 1	TM 2790	
IPLOT 0	TPLAS 1200	
IRELOC 1	VPLENZ .096	
IRL 0	X .27	
ISTOR	XCO 0.	

TABLE B.3. IFA 117 (HPR-107) Rod 5

ATMOS 1.	ITAPE	XH 0
CP 1.	ITMCUT 20	XN 0
CR 1.	JSTART 0	WW 0
DCI .5543	KSAVE 0	YY 0.
DCO .6140	LFUEL 19.2	ZZ 0.
DELZA (1) .5,.48,.48,.23,	LVOIDZ 2.5	Z 0.125
DFS .5455 .25,.12	MINI (1) 1,2,3,4,5,6	ZCLAD 0.
DSINZ 0.	MXPOWR 15.24	
DTEMP 20.	NC 4	
DVOIDZ .126	NCLAD 0	
EXTP 485.	NF 6	
FLXRAT	NFLX 0	
FRACAR -	NFUEL 0	
FRACH -	NOH 0	
FRACHE 1.	NPOW 5	
FRACKR -	NSTEPS 0	
FRACN -	NSH 0	
FRACXE -	NTIME 12	
FRDEN .913	PRCDH 2.	
FRDEN2 .965	PROFIL (1) .85,.9,.99,1.,10.99	
FRPUO ₂ -	PSEUDO (1) .05,2,3.9,6.2,8.5,10.5,12.9,13.5,11.7,9.6,7.1,4.5	
FRSIN .98	RADS .23	
FR35 6.	RNEWK (1)	
FR40 -	ROUC 4.E-4	
FR41 -	ROUF 1.E-3	
IDENSF 0	S	
IGAS 0	TIME (1) 0.,0.042,0.94,.125,.167,208,.25,.292,.33,.375,	
IOPT 0	.417,.458	
IPEAK 0	TINLET 464	
IPLOT 0	TM (2790.)	
IRELOC 1	TPLAS (1200)	
IRL 0	VPLENZ .293	
ISTOR	X .3	
	XCO 0	

TABLE B.4. EPRI Case D (Maine Yankee Rod)

ATMOS 1.	ITAPE	XH 0
CP	ITMCUT 20	XN 0
CR	JSTART 0	WW 0
DCI .3885	KSAVE 0	YY 0
DCO .4145	LFUEL 136.7	ZZ 0
DELZA (1) See Table B.4a	LVOIDZ 0.-	Z .125
DFS .3795	MINI (1) 1,0,3,0,5,0,0,0,0,	ZCLAD 0.
DSINZ 0.	MXPOWR 9. 0,11	
DTEMP 54.	NC 2	
DVOIDZ 0.	NCLAD 0	
EXTP 1800.	NF 3	
FLXRAT .63	NFLX 0	
FRACAR -	NFUEL 0	
FRACH -	NOH 0	
FRACHE 1.	NPOW 0	
FRACKR -	NSTEPS 0	
FRACN -	NSH 0	
FRACXE -	NTIME 39	
FRDEN .928	PRCDH 2.2	
FRDEN2 .965	PROFIL (1) See Table B.4a	
FRPUO ₂	PSEUDO (1) See Table B.4a	
FRSIN .965	RADS .146	
FR35 .02	RNEWK (1)	
FR40 -	ROUC 1.E-5	
FR41 -	ROUF .98E-4	
IDENSF 1	S 0	
IGAS 0	TIME (1) See Table B.4a	
IOPT 1	TINLET 530.	
IPEAK 0	TM (2790.)	
IPL0T 0	TPLAS (1200)	
IRELOC 1	VPLENZ 1.126	
IRL	X .27	
ISTOR	XCO 0	

TABLE B.4a. Supplemental Input for EPRI Case D

DELZA	PROFIL	PSEUDO	TIME
0.5	0.29 ^(a)	0.05	0.
6.	0.41	4.	0.24
14.5	0.69	6.2	0.36
15.0	0.86	6.2	0.62
16.0	0.98	6.2	1.8
16.0	1.	6.2	8.0
20.0	0.95	6.2	18.5
21.4	0.8	6.6	37
9.6	0.67	6.6	62
9.6	0.47	6.6	86
9.5	0.28	6.6	111
8.7	(b)	6.6	134
9.5		6.0	134.2
		4.0	134.3
		6.0	134.4
		6.0	135.3
		6.0	162
		5.1	194
		2.6	240
		7.2	240.1
		7.2	264
		6.9	275
		7.1	301
		7.1	341
		7.5	363
		7.1	387
		7.1	412
		7.0	412.1
		4.0	412.3
		7.8	412.5
		7.8	418
		8.4	433
		7.1	440
		7.1	443
		7.1	467
		7.1	498
		7.1	526
		0.05	526.1

(a) First step.

(b) See Appendix D for further steps.

TABLE B.5. IFA-431 Rod 3, Ramp A

ATMOS 1.	ITAPE	XH 0.
CP 1.	ITMCUT 20	XN 0.
CR 1.	JSTART 0	WW 0.
DCI .4293	KSAVE 0	YY 0.
DCO .5035	LFUEL 22.89	ZZ 0.
DELZA (1) .5,10*2.289,.6469	LVOIDZ 6.5	Z 0.125
DFS .4275	MINI (1) 2,0,0,0,0,0,0,8,9,	ZCLAD 0.
DSINZ 0.	MXPOWR 12.6 10,11,12	
DTEMP 27.	NC 4	
DVOIDZ .069	NCLAD 0	
EXTP 485.	NF 6	
FLXRAT	NFLX 0	
FRACAR -	NFUEL 0	
FRACH -	NOH 0	
FRACHE 1.	NPOW 10	
FRACKR -	NSTEPS 0	
FRACN -	NSH 0	
FRACXE -	NTIME 15	
FRDEN .95	PRCDH 0.0	
FRDEN2 .965	PROFIL (1) 0.59,.66,.72,.78,.84,.89,.93,.96,.99,1.01,1.03	
FRPUO ₂ 0.	PSEUDO (1) See Table B.5a	
FRSIN .95	RADS 0.	
FR35 0.1	RNEWK (1)	
FR40 -	ROUC 2.1E-5	
FR41 -	ROUF 8.7E-5	
IDENSF 0	S 0.	
IGAS 0	TIME (1) See Table B.5a	
IOPT 0	TINLET 463.	
IPEAK 0	TM (2790.)	
IPLOT 0	TPLAS (1200.)	
IRELOC 1	VPLENZ .94	
IRL 0	X .27	
ISTOR	XCO 0.	

TABLE B.5a. Supplemental Information for
IFA-431 Rod 3, Ramp A

<u>PSEUDO</u>	<u>TIME</u>
0.01	0.
0.43	0.05
0.9	0.11
0.56	0.40
0.7	0.50
1.24	0.55
2.1	0.60
2.85	0.65
3.6	0.71
4.5	0.75
5.55	0.80
6.0	0.85
6.6	0.90
7.28	0.95
7.42	

TABLE B.5b. Supplemental Information for
IFA-431, Rod 3, Ramp C

<u>PSEUDO</u>	<u>Time</u>
0.8	0.
1.94	0.7
3.1	0.73
3.89	0.75
4.72	0.77
5.7	0.79
6.64	0.82
7.96	0.87
8.82	0.89
9.74	0.91
10.71	0.93
11.6	0.98
11.15	1.40
10.56	2.15
12.2	2.70
12.1	3.40

TABLE B.5c. Supplemental Information for
IFA-431, Rod 3 Transient

<u>PSEUDO</u>		<u>TIME</u>
0.05	9.32	0.
3.0	9.26	0.4
5.0	9.17	0.5
7.0	9.08	0.55
9.5	9.00	0.6
9.5	8.91	1 second steps
9.5	8.82	from here on
9.51	8.74	
9.52	8.64	
9.53	8.54	
9.54	8.44	
9.55	8.37	
9.57	8.29	
9.60	8.22	
9.59	8.12	
9.58	8.02	
9.57	7.93	
	7.84	
9.54	7.74	
9.52	7.65	
9.49	7.61	
9.48	7.56	
9.47	7.53	
9.46	7.54	
9.46	7.55	
9.45	7.51	
9.45	7.47	
9.44	7.43	
9.38		

APPENDIX C

Restart and Plot Options

APPENDIX C

GAPCON-3 is able, as a user option, to produce either a restart or a restart-plot tape. Part I of this appendix discusses how start-restart runs are made and includes sample card deck images which are located at the end of Part I. Part II discusses PLOT3, a separate program which utilizes the restart-plot tape and plots GAPCON-3 results. A sample run is provided along with the PLOT3 listing.

PART I

The restart option in GAPCON-3 is a user-oriented feature to save computation time and money. Figure C1 shows that the use of the restart option provides the user with the ability to run multiple variations of a problem without continually starting each case in the beginning. A restart run must be preceded by a start run which saves predetermined pertinent GAPCON-3 variable values. These values must be stored on some permanent device, the magnetic tape being the most convenient. The start run differs from a regular GAPCON-3 run in two areas: (1) control cards and (2) one NAMELIST variable value.

The control cards must indicate to the computer operator and the system that a magnetic tape is needed and that data will be written, rather than read, to the tape. An MT1 on the job card informs the operator and the system that one tape is needed for the job. For an example, see Deck 1 card images which are located at the end of Part I. The COMMENT card is used by the operator and tape librarian as a check on the tape user. The REQUEST card supplies needed information to the operator and the operating system. TAPE1 is the normal GAPCON-3 write file and 99999 is the tape number which applies only to this example. The user will have to be assigned a tape by the librarian at the computer facility. For additional details the user should refer to a System Users Guide section on tape processing.

The NAMELIST variable KSAVE must have a value greater than zero (Deck 1 set KSAVE = 1). As indicated in the input instructions the KSAVE value is that repeat cycle for writing on the tape file. For example KSAVE = 1 causes

writing every time step; KSAVE = 3 writes the results from every third time step. So Deck 1 will produce a start tape containing data from each of the (NTIME) time steps.

After the start tape is created the GAPCON-3 user has three restart options.

- A. He may restart at any precalculated time step without writing any new results on the restart tape. See Decks 2, 3, and 4.
- B. He may restart at any precalculated time step writing all new results on restart tape. Note: This will destroy the corresponding values created by the start run. See Deck 5.
- C. He may restart at any precalculated time step writing all the start values up to the restart time step and subsequently writing all new results all on a second magnetic tape. See Deck 6.

These options require (1) modified control cards and (2) modified NAMELIST variable values which are discussed below.

Option A

MT1 is needed on job card--mount one tape. REQUEST card is needed with file name TAPE2 instead of TAPE1 which was used on the start run. File TAPE2 is the name for the GAPCON-3 normal read file. No ring is needed on the magnetic tape reel because GAPCON-3 will only read and not write from the tape.

Set KSAVE equal to 0 so the code will not write values from any time step. JSTART is set equal to the time step number just before new calculations are desired. To restart new calculations at time step 10 set JSTART = 9. Thus, the value of JSTART is the last time step read from the tape prior to making new calculations.

Option B

MT1 is needed.

REQUEST file TAPE2 with the ring is needed to permit writing on the tape.

KSAVE must be greater than zero.

ITAPE must be an integer set equal to zero. This is the signal for the code to first read from TAPE2 up to time step number JSTART, then write to TAPE2 all subsequent KSAVE time step increments.

JSTART as with option A must be greater than 1.

Option C

MT2 is needed--mount two magnetic tapes.

REQUEST start tape, file TAPE2, no ring.

REQUEST second tape, file TAPE1 with a ring.

KSAVE must be greater than zero.

JSTART must be greater than 1.

Table C-1 briefly summarizes the start-restart control and NAMELIST requirements.

Using any of these three options may require updated rod power history, axial power profiles, coolant external pressure and/or coolant to cladding heat transfer film coefficients. These updates are implemented using any nonzero value for the NAMELIST variable IOPT when restarting a problem. Whenever IOPT is greater than zero, even on a start run, values for the following parameters are read for each subsequent time step with a F10, 3I5, 5X, 2F10 format.

- a external coolant pressure, psi,
- b integer flag for new axial power profile
- c integer flag for new coolant film coefficients
- d integer flag for new power history
- e time increment for subsequent time step, days,
- f rod power for current time step, kW/ft.

When IOPT is greater than zero one data card for each subsequent time step containing at least the external coolant pressure must be included after the last NAMELIST dollar sign. Whenever the axial power profile flag is

nonzero the immediately following card(s) must contain NPOW+1 values for the new rod axial power profile, 8F10 format. If the coolant film coefficient flag is nonzero the next immediately following card(s) must contain NPOW new values of the film coefficients, 8F10 format. If the power history flag is nonzero, the value for the subsequent time step increment and the current time step rod power, items e and f above, are used. For an example of the last case see Decks 4, 5 and 6.

The user can just write out the GAPCON-3 predictions which are stored on a tape (created using both KSAVE and IPLOT greater than zero). All that is needed is a restart run with MT1, TAPE2 and no ring and both JSTART and ITAPE each set equal to the value of NTIME.

FIGURE C1. GAPCON-3 Restart Option

TABLE C-1. Outline of GAPCON-3 Start-Restart
Requirements

START

- MT1 on Job card
- REQUEST file TAPE1 with ring
- KSAVE > 1

RESTART

Option A (no writing to tape)

- MT1 on Job card
- REQUEST file TAPE2 no ring
- JSTART > 1

Option B (rewrite on tape starting with first restart time step)

- MT1 on Job card
- REQUEST file TAPE2 with ring
- JSTART > 1
- KSAVE > 0
- ITAPE = 1

Option C (write on a second tape)

- MT2 on Job card
- REQUEST file TAPE2 - start tape - no ring
- REQUEST file TAPE1 - new tape - ring
- JSTART > 1
- KSAVE > 0

THE FOLLOWING SAMPLE DECK 1 IS FOR A START RUN WITH IOPT=0

FEPUV3,P2,T250,MT1.

ACCOUNT,PRP000,PNL.

COMMENT. PRP000/PANISKO GAPCON3 SAMPLE /10

REQUEST(TAPE1,MT,SV,VSU=99999,PING.

ATTACH(OLDLGO,GAPLGO,ID=MOHR.

OLDLGO(PL=20000)

*

GAPCON3 START SAMPLE IOPT=0

\$INPUT

ATMOS=27.21,CP=1.0,CR=1.0,DCI=0.3734,DCO=0.4220,DFS=.3659,DTMP=100.,

EXTP=2250.,FR35=0.03,FRACHE=1.0,FRDEN=.965,FRSIN=.965,ITMCUT=10,

IPEAK=1,IREFLC=1, RQUF=.00008,RQUC=.00004,TINLET=545.,VPLENZ=.525,

NPOW=9, DELZA(1)=.5,.25,.25,28.6,28.6,28.6,28.6,28.6,.25,.25,4.79,

PROFIL(1)=.08,.8,.8,1.,1.,1.,1.,1.,1.,.8,.8,.08,MAXPOWR=13.4,

MINI(1)=1,2,3,4,5,6,7,8,9,10,11,

FRDEN2=.965, LFUEL=144,TDENSEF=1,ZCLAD=1,ISTOR=1,IOPT=0,NTIME=15,NC=3,

PSEUDO(1)=.05,5.3,10.6,0,5.3,10.6,7.95,5.3,7.95,10.6,13.4,0,5.3,10.6,0,

TIME(1)=0,.5,125.5,126,126.125,251.125,251.25,251.5,251.625,376.625,

TIME(11)=376.667,378.917,379.167,416.667,416.7,

IPLT=1,

KSAVE=1,

\$

*

THE FOLLOWING SAMPLE DECK 2 IS FOR A RESTART RUN WITH IOPT=0

FEPUV3,P2,T250,MT1.

ACCOUNT,PRP000,PNL.

COMMENT. PRP000/PANISKO GAPCON3 SAMPLE /10

REQUEST(TAPE2,MT,SV,VSU=99999)

ATTACH(OLDLGO,GAPLGO,ID=MOHR.

OLDLGO(PL=20000)

*

GAPCON3 RESTART SAMPLE IOPT=0

\$INPUT

ATMOS=27.21,CP=1.0,CR=1.0,DCI=0.3734,DCO=0.4220,DFS=.3659,DTMP=100.,

EXTP=2250.,FR35=0.03,FRACHE=1.0,FRDEN=.965,FRSIN=.965,ITMCUT=10,

IPEAK=1,IREFLC=1, RQUF=.00008,RQUC=.00004,TINLET=545.,VPLENZ=.525,

NPOW=9, DELZA(1)=.5,.25,.25,28.6,28.6,28.6,28.6,28.6,.25,.25,4.79,

PROFIL(1)=.08,.8,.8,1.,1.,1.,1.,1.,1.,.8,.8,.08,MAXPOWR=13.4,

FRDEN2=.965, LFUEL=144,TDENSEF=1,ZCLAD=1,ISTOR=1,IOPT=0,NTIME=79,NC=3,

MINI(1)=1,2,3,4,5,6,7,8,9,10,11,

PSEUDO(1)=.05,5.3,10.6,0,5.3,10.6,7.95,5.3,7.95,10.6,13.4,0,5.3,10.6,0,

TIME(1)=0,.5,125.5,126,126.125,251.125,251.25,251.5,251.625,376.625,

TIME(11)=376.667,378.917,379.167,416.667,416.7,

IPLT=1,

KSAVE=0,

JSTART=9,

\$

*

THE FOLLOWING SAMPLE DECK 3 WILL GET IDENTICAL RESULTS AS THE PRECEDING ONE
 FERJVS,P2,T250,MT1.
 ACCOUNT,PRP000,PNL.
 COMMENT, PRP000/PANISKO GAPCON3 SAMPLE /10
 REQUEST(TAPE2,MT,SV,VSX=99999)
 ATTACH(OLDLGO,GAPLGO,ID=MOHR.
 OLDLGO(PL=20000)

*
 GAPCON3 RESTART SAMPLE IORT=0
 \$INPUT
 MINI(1)=1,2,3,4,5,6,7,8,9,10,11,
 IPIOT=1,
 JSTART=9,
 \$

*

THE FOLLOWING SAMPLE DECK 4 IS FOR A RESTART WITH IORT=1
 NOTE THAT THE FIRST CARD AFTER THE * CONTAINS VALUES FOR THE JSTART+1 TIME STEP
 THE SHORT INPUT FORM WILL BE USED AS FOR THE PRECEDING SAMPLE

FERJVS,P2,T250,MT1.
 ACCOUNT,PRP000,PNL.
 COMMENT, PRP000/PANISKO GAPCON3 SAMPLE /10
 REQUEST(TAPE2,MT,SV,VSX=99999)
 ATTACH(OLDLGO,GAPLGO,ID=MOHR.
 OLDLGO(PL=20000)

*
 GAPCON3 RESTART SAMPLE IORT=1
 \$INPUT
 IORT=1,
 MINI(1)=1,2,3,4,5,6,7,8,9,10,11,
 IPIOT=1,
 JSTART=9,
 NSTEPS=17,
 \$

2250.	1	5.	10.6
2250.	1	5.	10.6
2250.	1	10.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6

*

THE FOLLOWING SAMPLE DECK 5 WILL RESTART AND WRITE ON THE SAME TAPE
WRITING WILL COMMENCE AFTER THE JSTART TIME STEP AND CONTINUE THRU NSTEP

FERJUV3,P2,T250,MT1.
ACCOUNT,PRP000,PNL.
COMMENT, PRP000/PANISKO GAPCON3 SAMPLE /10
REQUEST(TAPE2,MT,SV,VSN=99999,RING.
ATTACH(OLDLGO,GAPLGO,ID=MOHR.
OLDLGO(PL=20000)

*

GAPCON3 RESTART SAMPLE IOPT=1 WRITE ON TAPE

\$INPUT

IOPT=1,

MINI(1)=1,2,3,4,5,6,7,8,9,10,11,

IPL0T=1,

JSTART=0,

NSTEPS=17,

ITAPE=1,

KSAVE=1,

\$

2250.	1	5.	10.6
2250.	1	5.	10.6
2250.	1	10.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6

*

THE FOLLOWING SAMPLE DECK 6 WILL RESTART AND WRITE ON A SECOND TAPE
THIS SECOND TAPE WILL CONTAIN VALUES FROM THE START RUN THRU TIME STEP
JSTART THEN WILL HAVE VALUES FROM THE RESTART RUN

FERJUV3,P2,T250,MT2.
ACCOUNT,PRP000,PNL.
COMMENT, PRP000/LANNING GAPCON3 SAMPLE /10
REQUEST(TAPE1,MT,SV,VSN=88888,RING.
COMMENT, PRP000/PANISKO GAPCON3 SAMPLE /10
REQUEST(TAPE2,MT,SV,VSN=99999)
ATTACH(OLDLGO,GAPLGO,ID=MOHR.
OLDLGO(PL=20000)

*

GAPCON3 RESTART SAMPLE IOPT=1 WRITE ON TAPES

\$INPUT

IOPT=1,

MINI(1)=1,2,3,4,5,6,7,8,9,10,11,

IPL0T=1,

JSTART=0,

NSTEPS=17,

KSAVE=1,

\$

2250.	1	5.	10.6
2250.	1	5.	10.6
2250.	1	10.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6
2250.	1	15.	10.6

PART II

PLOT3 is a separate computer program used to plot GAPCON-3 results. The plotting program utilizes CALCOMP* hardware and software in conjunction with a GAPCON-3 start-plot magnetic tape to produce high quality graphs. This part of Appendix C contains a description of PLOT3 plus an example and a listing of the code.

PLOT3 was designed to minimize input yet maintain flexibility to accommodate diverse user demands. Graphic summaries of the GAPCON-3 predictions are useful for both the user and his readers. Some of the numerous data saved for restart runs are used along with a few additional data for making plots. These needed additional data are written to tape when $KSAVE > 1$ and $IPLLOT > 1$ in the GAPCON-3 input. Refer to Part I of the appendix for a description of a "start" tape. Whenever plots are desired a GAPCON-3 start run must be made with both KSAVE and IPLLOT greater than zero. Note that the previously mentioned Decks 1 through 6 (see Part I) all have $IPLLOT = 1$. This causes sufficient additional data to be stored on start and restart tapes to permit future plotting.

Figure C2 is a flow diagram for PLOT3 and its subroutine BOX. Initially the first six values are read in PLOT3 from the first data block of the start-plot tape. Using these values, pointers for 38 fuel rod parameters are calculated. A listing of these 38 parameters which can be plotted are given in Table C2. A pointer is the starting address in the C array for one of the parameters. After the pointers are calculated the user input is read. Refer again to the flow diagram. Based on the user commands, input flags are set for parameters to be plotted.

Next the time step loop begins:

- A block of data is read from the tape generated by GAPCON-3
- Appropriate data based on user commands are stored consecutively for curves 1 through 7 of graphs 1 through 7 as a function of axial nodes at a given time step.

* California Computer Products Inc.

- Appropriate values are stored consecutively for curves 1 through 7 of graphs 1 through 7 for one axial node as a function of time step.
- Subroutine BOX is called to generate commands to plot all the isochronous graphs.
- The procedure starts again by reading another block of data from next time step.

After the last time step is read, subroutine BOX is called again to generate graphs as a function of time.

As shown in Figure C2, BOX generates the commands to line the graph. For each curve BOX does the following:

- Scales the data in D array to fit user restrictions.
- Loads data into plot arrays, YARRAY and XARRAY.
- Labels curve.
- Plots curve.

After the last curve is plotted the title is written and the axes are labelled. After the last graph is generated for the current time step the program begins again in PLOT3 by reading the next block of data from the next time step off the GAPCON-3 tape. Finally, after the last graph as a function of axial node is plotted, then PLOT3 calls BOX one final time to plot time dependent data, on the E array.

PLOT3 is easy to use. The user is required to input (1) a title card which also contains eight values to specify the size of the graphs and (2) at least one card containing integer signals for NAMELIST input parameters to be plotted. See the sample input cards which follow the PLOT3 listing at the end of Part II. The title card contains the following nine consecutive parameters, (values in parenthesis are for the example).

- maximum ordinate value (100)
 - ordinate length, inches (7)
 - maximum abscissa value (10)
 - abscissa length, inches (8)
- } for axial node graphs

- maximum ordinate value (100 implied)
 - ordinate length, inches (7 implied)
 - maximum abscissa value (60)
 - maximum abscissa length, inches (8 implied)
 - graphs title
- } for time dependent graphs

The format for these parameters is 8F5.0, 10X, 3A10.

If plots are desired only at specific time steps, NAMELIST variable INTERV is used. For example, to plot only the 1st and 4th time step, INTERV(1) = 1,4 is needed. Finally, using the signals defined in Table C2 the user simply inputs integer values for each desired parameter to be plotted. Like integer values appear on the same plot; positive integers refer to axial dependent plots, and negative integers refer to time dependent plots. Integers must be between -7 and +7.

As shown in the sample case, Integer Temperature Fuel Inside (ITFI) is set equal to 1 and -1. This causes one curve of fuel centerline temperature to be plotted on the 1st isochronous graph for every time step and causes one curve of fuel centerline temperature for axial region 4 (KPOW=4) to be plotted as a function of time on the 1st time dependent graph. The same applies for both Integer Temperature Fuel Outside (ITFO) and Integer Temperature Cladding Inside (ITCI). IBURN set equal to 1 causes burnup to be plotted on the 1st isochronous graph. IPOWER equal to 1, 2, -1, and -2 causes rod axial power profiles to be plotted on the 1st and 2nd isochronous graphs as well as on the 1st and 2nd time dependent graph. Since KPOW=4 in this sample, all the time dependent graphs will be for fuel rod axial region 4.

To indicate that the abscissa variable refers to axial node, IAXNOD is set equal to 11, 12 for isochronous graphs 1 and 2. Likewise, to indicate that the abscissa variable for the history graphs is days, ITDAYS is set equal to -11, -12 for graphs 1 and 2.

With the simple user input of the sample case the resulting plots shown after the PLOT3 listing were generated.

a CURVE LOOP
b FIGURE LOOP
c TIME STEP LOOP

FIGURE C2. Plot 3 Flow Sheet


```

*****
C
C      PLOT3 FUEL ROD PARAMETER KEY WORDS
C
C*****
C      PARAMETER                                KEY WORD
C      1 = TIME (DAYS)                          ITDAYS
C      2 = TIME (SECONDS)                       ITSEC
C      3 = INTERNAL GAS PRESSURE (PSI)          IPREST
C      4 = POWER (KW/FT)                       IPOWER
C      5 = BURNUP (MWD/KG)                     IBURNU
C      6 = FILM COEFF (BTU/HR-FT2-DEG F)        IHF
C      7 = FUEL  $\Delta$ R SWELLING (IN)              IDRB
C      8 = FUEL  $\Delta$ R THERMAL EXPANSION (IN)      IDRCI
C      9 = FUEL  $\Delta$ R RELOCATION (IN)             IDRGDM
C     10 = FUEL  $\Delta$ R DENSIFICATION (IN)         IDRPI
C     11 = PIN AXIAL NODE                      IAXNOD
C     12 = CLAD  $\Delta$ R (IN)                      IDRCT
C     13 = TOTAL FUEL  $\Delta$ R (IN)                IDRFT
C     14 = RESTRUCTURED RADIUS (IN)            IRSIN
C     15 = RADIAL HOT GAP (IN)                 IRGAP
C     16 = COOLANT TEMP (DEG F)                ITCOOL
C     17 = FUEL-CLAD PRESSURE (PSI)             IPRESS
C     18 = GAP CONDUCTANCE (BTU/HR-FT-DEG F)   IHG
C     19 = CLAD  $\Delta$ R ELASTIC (IN)              ICRELS
C     20 = CLAD  $\Delta$ R CREEP (IN)               ICRCPP
C     21 = FUEL VOID RADIUS (IN)              IRFI
C     22 = RADIUS FUEL SURFACE (IN)            IRFD
C     23 = RADIUS CLAD INSIDE (IN)            IRCI
C     24 = RADIUS CLAD OUTSIDE (IN)           IRCO
C     25 = TEMP FUEL CENTER (DEG F)           ITFI
C     26 = TEMP FUEL SURFACE (DEG F)          ITFO
C     27 = TEMP CLAD INSIDE (DEG F)           ITCI
C     28 = TEMP CLAD OUTSIDE (DEG F)          ITCO
C     29 = FUEL RADIAL STRESS (PSI)            -
C     30 = FUEL HOOP STRESS (PSI)              -
C     31 = FUEL AXIAL STRESS (PSI)             -
C     32 = FUEL SHEAR STRESS (PSI)            CURRENTLY
C     33 = CLAD RADIAL STRESS (PSI)            -
C     34 = CLAD HOOP STRESS (PSI)            NOT
C     35 = CLAD AXIAL STRESS (PSI)            -
C     36 = CLAD SHEAR STRESS (PSI)           AVAILABLE
C     37 = GAS RELEASE FRACTION               IRDOT
C     38 = CLAD TOTAL LENGTH CHANGE           SUMM
C
C*****
C*****

```

TABLE C2.

FEPPLO3,P2,T200,MT2.

ACCOUNT CARD

COMMENT CARD USED AT UCC

REQUEST(TAPE7,HY,S,RING,IB,VSN=64814)

REQUEST(TAPE2,MT,SV,VSN=65666.

ATTACH(PLTLIB,CALCMP,ID=UCCLIB)

FTN(R=2.

MAP(PART)

LDSET(LIB=P(TLIB)

LGO.

*

```
PROGRAM PLOT3(INPUT,OUTPUT,TAPE2,TAPES=INPUT,TAPE6=OUTPUT)
C   PLOT3 IS A PLOT PROGRAM THAT USES DATA FROM GAPCON=3 RESTART=PLOT
C   TAPES. THE MAIN PROGRAM READS THE RESTART=PLOT TAPE AND, BASED ON
C   USER INPUT LOADS THE APPROPRIATE DATA INTO EITHER OF TWO ARRAYS.
C   THE D ARRAY HAS THE DATA FOR A PARTICULAR TIME FOR ALL AXIAL
C   NODES, THE E ARRAY HAS THE TIME (TIME STEP) DEPENDENT DATA FOR
C   NODE KPOW (USER SPECIFIED).
C   D CONTAINS AS MANY AS 20 DATA PTS=7/CURVE, AS MANY AS 6 CURVES/PLOT,
C   AND AS MANY AS 7 PLOTS AT ANY TIME
C   E CONTAINS AS MANY AS 100 DATA PTS., 1/TIME STEP, FOR AS MANY AS 6 CURVES
C   AND AS MANY AS 7 PLOTS
C   PLOT D ARRAY EACH TIME STEP
C   SAVE VALUES ON E AT EACH TIME STEP TO BE PLOTTED AFTER LAST TIME STEP
C   EXTERNALS... BOX PLOT PLOTS
C   FUNCTIONS... FLOAT IAMS
C   DIMENSION PTITL(3),IBUF(1000),C(4000),KSUB(7,40),IFL(7,7,2)
+ ,ISUB(40),KONTR(7),KONTER(7),INTERV(50),D(100,7,7),E(100,7,7)
COMMON /A/ TIMEN,TSEC,PRESTO,POWER(20),BURNUP(20),HF(20),
IDELRB(20),          DELRCI(20),          DELGDM(20),DELPI(20),
IDELRCO(20),DELRCO(20),DELRCO(20),PSIN(20),RGAPX(20),TCOOL(20),
+PRESS(20),HGC(20),CRELS(20),CRCRP(20),KDC(15,20),TFB(10,19),TCB(5,
+20),TSTSR(4,26,10),ROOT(50,20),SUMM
COMMON /B/ ITDAYS(7),ITSEC(7),IPREST(7),IPOWER(7),IBURNU(7),
+IHF(7),IDRB(7),IDRCI(7),IDRGDM(7),IDRPI(7),IAXNOD(7),IDRCT(7),
+IDRFT(7),IRFIN(7),IRGAP(7),ITCOOL(7),IPRESS(7),IHG(7),ICRELS(7),
+ICRCRP(7),IRFI(7),IRFO(7),IRCI(7),IRCO(7),ITFI(7),ITFU(7),ITCI(7)
+ ,ITCU(7),IRFSTS(7),ITFSTS(7),IZFSTS(7),ISFSTS(7),IRCSTS(7),ITCSTS
+ (7),IZCSTS(7),ISCSTS(7),IROOT(7),ICLADL(7),INTIME(7)
EQUIVALENCE(TIMEN,C(1))
EQUIVALENCE(ITDAYS(1),KSUB(1,1))
NAMELIST /IN/ ITDAYS,ITSEC,IPREST,IPOWER,IBURNU,IHF,IDRB,
+IDRCI,IDRGDM,IDRPI,IAXNOD,IDRCT,IRFIN,IRGAP,ITCOOL,
+IPRESS,IHG,ICRELS,ICRCRP,IRFI,IRFO,IRCI,IRCO,ITFI,ITFU,ITCI,ITCO
+ ,IRFSTS,ITFSTS,IZFSTS,ISFSTS,IRCSTS,ITCSTS,IZCSTS,ISCSTS,IROOT
+ ,ICLADL,INTERV,KPOW,NSTEPS
CALL PLOTS(IBUF,1000,7)
IXAX=1 FOR TIME INDEPENDENT PLOTS
IXAX=1
REWIND 2
READ(2) ILEN,NPOW,NTIME,NF,NC,IPLDT
WRITE(6,10) NPOW,NTIME,NF,NC,IPLDT
IF(NC.GT.5) GO TO 1002
IF(NF.GT.10) GO TO 1001
GO TO 1005
1001 WRITE(6,1003)
STOP
1002 WRITE(6,1004)
STOP
1005 CONTINUE
```

```

1005 CONTINUE
    IF(NPDW.GT.10) GO TO 1006
    GO TO 1007
1006 WRITE(6,1008)
    STOP
1007 CONTINUE
    NARF=NPDW+2
    NART=NPDW+3
    NFPNC=NF+NF
    NART=NART+1
    NCELD=2+NFPNC+4
    NATED=NART+4+(NARF-1)*4
C   SET UP THE POINTERS TO USE WITH THE CMA-RAY
    DO 20 I=1,4
        ISUB(I)=1
20 CONTINUE
    DO 21 I=5,21
        ISUB(I)=ISUB(I-1)+20
21 CONTINUE
    ISUB(22)=ISUB(21)+NF-1
    ISUB(23)=ISUB(21)+NF
    ISUB(24)=ISUB(21)+NF+NC-1
    ISUB(25)=ISUB(21)+300
    ISUB(26)=ISUB(25)+NF-1
    ISUB(27)=ISUB(25)+100
    ISUB(28)=ISUB(27)+NC-1
    ISUB(29)=ISUB(27)+100
    ISUB(30)=ISUB(29)+1
    ISUB(31)=ISUB(30)+1
    ISUB(32)=ISUB(31)+1
    ISUB(33)=ISUB(29)+71
    ISUB(34)=ISUB(33)+1
    ISUB(35)=ISUB(34)+1
    ISUB(36)=ISUB(35)+1
    ISUB(37)=ISUB(29)+1040
    ISUB(38)=ISUB(37)+1000
C   A = INCREMENTS ON THE Y AXIS          COL. 1-5
C   YL = LENGTH OF THE Y AXIS IN INCHES    COL. 6-10
C   R = INCREMENTS ON THE X AXIS          COL. 11-15
C   XL = LENGTH OF THE X AXIS IN INCHES    COL. 16-20
C   AT= INCREMENTS ON THE Y AXIS(TIME PLOTS) COL. 21-25
C   YLT= LENGTH OF THE Y AXIS IN INCHES    COL. 26-30
C   RT= INCREMENTS ON THE X AXIS(TIME PLOTS) COL. 31-35
C   XLT= LENGTH OF THE X AXIS IN INCHES    COL. 36-40
C   PTITL = TITLE FOR THE TOP OF THE PLOTS COL. 51-80
    READ(5,22) A,YL,R,XL,AT,YLT,RT,XLT,(PTITL(K),K=1,3)
    READ(5,IN)
    WRITE(6,IN)
    KPDW=KPDW+1
    IF(NSTEPS.NE.0) NTIME=NSTEPS
    IF(KPDW.GT.NART) STOP*KPDW GREATER THAN NART*
    WRITE(6,700)
    WRITE(6,9) (KSUB(IQ,JQ),IQ=1,7),ISUB(JQ),JQ=1,40)
C*****
C
C   NOW DETERMINE WHAT DATA ARE TO BE PLOTTED FOR EACH TIME STEP
C   AND WHAT DATA ARE TO BE SAVED TO PLOT AT THE END AS A FUNCTION
C   OF TIME.
C   ARRAY IF(I,J,1) CONTAINS THE ISUB NO. FOR DATA TO BE PLOTTED
C   EVERY TIME STEP.

```

```

C      ARRAY IFL(I,J,2) CONTAINS THE ISUR NO. FOR DATA TO BE PLOTTED AS
C      A FUNCTION OF TIME.
C
C*****
      DO 50 I1=1,7
      KONTER(I1)=0
      KONTR(I1)=0
50  CONTINUE
      DO 150 I2=1,40
      DO 200 I3=1,7
      IF(KSUB(I3,I2).EQ.0) GO TO 200
      IZ=KSUB(I3,I2)
      IF(IZ.LT.0) GO TO 60
      IF(KSUB(I3,I2).GT.7) GO TO 201
      KONTR(IZ)=KONTR(IZ)+1
      IF(KONTR(IZ).GT.6) STOP#YOU HAVE ATTEMPTED TO PLOT TOO MANY CURVES
      + FOR EVERY TIME STEP#
      IFL(KONTR(IZ),IZ,1)=I2
      GO TO 200
201  CONTINUE
      IFL(7,IZ-10,1)=I2
      GO TO 200
60   IP=IABS(IZ)
      IF(KSUB(I3,I2).LT.-10) GO TO 202
      KONTER(IP)=KONTER(IP)+1
      IF(KONTER(IP).GT.6) STOP#YOU HAVE ATTEMPTED TO PLOT TOO MANY CURVE
      +S AS A FUNCTION OF TIME#
      IFL(KONTER(IP),IP,2)=I2
      GO TO 200
202  CONTINUE
      IFL(7,IP-10,2)=I2
200  CONTINUE
150  CONTINUE
      DO 155 I1=1,2
      DO 160 I2=1,7
      IF(IFL(1,I2,I1).NE.0.AND.IFL(7,I2,I1).EQ.0) GO TO 162
      GO TO 160
162  IFL(1,I2,I1)=0
      WRITE(6,161)I2
160  CONTINUE
155  CONTINUE
      WRITE(6,401)IFL,NT,NTIME
C      BEGIN THE TIME LOOP
      IC=1
      DO 1000 J1=1,NTIME
      READ(2) TIMEN,PRESTO,DELGM,DELPT,RSIN,PRESS,HGC,CRELST,CRCRP
      +,((RD(L1,L2),L1=1,NFPNC),L2=1,NART)
      +,(((TSTR(L1,L2,L3),L1=1,4),L2=1,NDEL),L3=1,NART),RDOT,SUMH,NT
      IF(JPLOT.EQ.0) STOP#PLOT DATA NOT ON TAPE JPLOT IS EQ. TO 0#
      READ(2) POWER,BURNUP,HF,DELRF,DELRCI,DELRCO,DELRCF,DELRCR,DELRCY,
      +,TCOOL
      +,((TR(L1,L2),L1=1,NF),L2=1,NARF),((TCR(L1,L2),L1=1,NC),L2=1,NART)
      +,NX
      TSEC=TIMEN*86400
      IF(INTERV(1).GT.0.AND.INTERV(1).LE.NT) GO TO 1000
      IC=IC+1
C      IF INTERV(1).EQ.0 PLOT EACH TIME STEP ON THE RESTART TAPE
C      IF INTERV(1).NE.0 PLOT TIME STEP INTERV(1)
C
C      BEGIN THE PLOT LOOP

```

```

DO 400 I1=1,7
D(100,1,I1)=0.
BEGIN THE CURVE LOOP
DO 500 I2=1,7
IF(IFL(I2,I1,1).EQ.0) GO TO 500
ISTART=ISUB(IFL(I2,I1,1))
IF(NXT.NE.1.AND.IFL(I2,I1,1).EQ.37) ISTART=ISUB(IFL(I2,I1,1))+(NXT+1
+)*1
I=IFL(I2,I1,1)
J=1
IF(I.GT. 3.AND.I.LE.20) J=1
IF(I.LE.20) ILAST=ISTART+NAHT-1
IF(I.GT.20.AND.I.LE.24) ILAST=ISUB(21)+15*NAHT-1
IF(I.GT.24.AND.I.LE.26) ILAST=ISUB(25)+10*NAHT-1
IF(I.GT.26.AND.I.LE.28) ILAST=ISUB(27)+5*NAHT-1
IF(I.GT.28.AND.I.LE.36) ILAST=ISUB(29)+4+26*NAHT-1
IF(I.EQ.37) ILAST=ISUB(37)+50+20-1
IF(I.GT.20.AND.I.LE.24) J=15
IF(I.GT.24.AND.I.LE.26) J=10
IF(I.GT.26.AND.I.LE.28) J=5
IF(I.GT.28.AND.I.LE.36) J=26
IF(I.EQ.37) J=50
IF(I.EQ.38) ILAST=ISUB(38)
I4=0
DO 600 I3=ISTART,ILAST,J
I4=I4+1
IF(I.EQ.11.OR.I.EQ.38) C(I3)=FLOAT(I4)
D(I4,I2,I1)=C(I3)
IF(I.LE.3) D(I4,I2,I1)=D(I,I2,I1)
600 CONTINUE
D(100,I2,I1)=I4
500 CONTINUE
IF(D(100,1,I1).EQ.0.) GO TO 499
WRITE(6,730) XL,YL
WRITE(6,735) B,A
WRITE(6,710)
WRITE(6,402) (((D(K,L,I1),L=1,7),K,I1),K=1,NAHT)
499 CONTINUE
400 CONTINUE
DO 300 I1=1,7
DO 350 I2=1,7
IF(IFL(I2,I1,2).EQ.0) GO TO 350
ISTART=ISUB(IFL(I2,I1,2))
J=0
I=IFL(I2,I1,2)
IF(I.GT. 3.AND.I.LE.20) J=1
IF(I.GT.20.AND.I.LE.24) J=15
IF(I.GT.24.AND.I.LE.26) J=10
IF(I.GT.26.AND.I.LE.28) J=5
IF(I.GT.28.AND.I.LE.36) J=26
IF(I.EQ.37) J=50
I3=ISTART+J*KPOW
E(J,I2,I1)=C(I3)
E(100,I2,I1)=J1
350 CONTINUE
300 CONTINUE
CALL ROY(NABT,NTIME,0,IXAX,4,IFL,XL,YL,TIMEN,PTITL,KPOW,B)
1000 CONTINUE
C IXAX=2 FOR TIME DEPENDENT PLOTS
IXAX=2

```

```

C      NOW PLOT E
      IF(NTIME.GT,IC-1) NTIME=IC-1
      DO 599 I5=1,7
      IF(E(100,1,I5).EQ.0.) GO TO 598
      WRITE(6,750)
      WRITE(6,720)
      WRITE(6,402) (((E(K,L,I5),L=1,7),K,I5),K=1,NTIME)
598 CONTINUE
599 CONTINUE
C      CHECK TO SEE IF TIME PLOTS ARE DIFFERENT SIZE
      IF(AT.GT.0.) A=AT
      IF(YLT.GT.0.) YL=YLT
      IF(RT.GT.0.) R=RT
      IF(XLT.GT.0.) XL=XLT
      CALL BOX(NABT,NTIME,E,IXAX,A,IFL,XL,YL,TIMEN,PTITL,KPOW,R)
C      WHEN NT=NTIME PLOT E RIGHT AFTER D
C      I IS THE COUNTER, IF NEEDED, FOR INTERV(I)
C      IF ISYCHRONOUS PLOTS ARE DESIRED FOR NT = 3,23 AND 46 THEN
C      INTERV(1)= 3,23,46
C      PLOT EACH TIME STEP D(20,7,7) ISYCHRONOUS
C      SAVE EACH TIME STEP E(100,7,7) TIME DEPENDENT
C      SAVE VALUES IN E TO BE PLOTTED AFTER LAST TIME STEP
C      PLOT D ARRAY D(I1,L2,L3) I1=1, UP TO NABT POINTS PER CURVE L2=1 UP
C      TO 6 CURVES PER PLOT L3=1 UP TO 7 PLOTS
      CALL PLOT(12,0,0,0,999)
C      CLOSE THE PLOT TAPE AFTER PROCESSING THE LAST CASE (END-OF-FILE REACHED)
      STOP
      9 FORMAT(1H ,8I12)
      10 FORMAT(1H ,5I10)
      22 FORMAT(AF5.1, 10X,3A10)
      161 FORMAT(10X,#YOU FORGOT TO SPECIFY THE ARSCISSA FOR PLOT NO.,I5)
      401 FORMAT(1H ,14(7I10/),2I10)
      402 FORMAT(1H ,2X,7G13.4,2I5)
      700 FORMAT(1H1)
      710 FORMAT(5X,#D ARRAY FOR PLOT #)
      720 FORMAT(5X,#E ARRAY FOR PLOT #)
      730 FORMAT(5X,#X AXIS LENGTH =#,F4.0,# INCHES#,5X,#Y AXIS LENGTH =#,
      +F4.0,# INCHES#)
      735 FORMAT(5X,#THE X AXIS = #,F4.0,# UNITS Y AXIS = #,F4.0,# UNITS
      +#)
      750 FORMAT(///)
      1003 FORMAT(10X,#THE VALUE OF NE IS GREATER THAN 10#)
      1004 FORMAT(10X,#THE VALUE OF NC IS GREATER THAN 5#)
      1008 FORMAT(1H1,#NPOW .GT. 10 SO INCREASE DIMENSION OF TSTR(I,J,10) #
      +AND INCREASE THE 10 ON CARD NO 9999 AND 1040 ON 9998#)
      END
      SUBROUTINE BOX(NABT,NTIME,D,IXAX,A,IFL,XL,YL,TDAYS,PTITL,KPOW,C)
C      SUBROUTINE BOX TAKES THE D OR E ARRAY FROM PLOT3 AND PLOTS THE
C      VALUES. ALL THE PLOTTING INSTRUCTIONS FOR THE CALCOMP PLOTTER
C      (EXCEPT TWO CALLS IN PLOT3 - ONE TO OPEN THE PLOT FILE #PLOTS#
C      AND THE OTHER TO CLOSE THE PLOT FILE #PLOT#).
      INTEGER XTITLE,A1
      DIMENSION TITL(156),PTITL(3),XLABL(4),XARRAY(102),YARRAY(102),
      +KFIX(40),IFL(7,7,2),D(100,7,7),NUM(40),JFIX(40)
C*****
C*****
C      VALUES FOR XTITLE TO PUT TITLES ON PLOTS
C      1 = TIME (DAYS) ITOAYS
C      2 = TIME (SECONDS) ITSEC

```

C	3 = INTERNAL GAS PRESSURE (PSI)	IPREST
C	4 = POWER (KW/FT)	TPOWER
C	5 = BURNUP (MW/KG)	IBURNUP
C	6 = FILM COEFF (BTU/HR-FT ² -DEG F)	TFE
C	7 = FUEL >R SWELLING (IN)	TDRT
C	8 = FUEL >R THERMAL EXPANSION (IN)	TDRT
C	9 = FUEL >R RELOCATION (IN)	TDRT
C	10 = FUEL >R DENSIFICATION (IN)	TDRT
C	11 = PIN AXIAL MODE	IAX100
C	12 = CLAD >R (IN)	ICRT
C	13 = TOTAL FUEL >R (IN)	ITRT
C	14 = RESTRUCTURED RADIUS (IN)	IRST
C	15 = RADIAL HOT GAP (IN)	IRGAP
C	16 = COOLANT TEMP (DEG F)	ITC00L
C	17 = FUEL-CLAD PRESSURE (PSI)	IPRESS
C	18 = GAP CONDUCTANCE (BTU/HR-FT-DEG F)	IGG
C	19 = CLAD >R ELASTIC (IN)	ICREL5
C	20 = CLAD >R CREEP (IN)	ICREL0
C	21 = FUEL VOID RADIUS (IN)	IVRT
C	22 = RADIUS FUEL SURFACE (IN)	IRFL
C	23 = RADIUS CLAD INSIDE (IN)	IRCI
C	24 = RADIUS CLAD OUTSIDE (IN)	IRCO
C	25 = TEMP FUEL CENTER (DEG F)	ITFI
C	26 = TEMP FUEL SURFACE (DEG F)	ITFO
C	27 = TEMP CLAD INSIDE (DEG F)	ITCI
C	28 = TEMP CLAD OUTSIDE (DEG F)	ITCO
C	29 = FUEL RADIAL STRESS (PSI)	IRFSTS
C	30 = FUEL HOOP STRESS (PSI)	ITFSTS
C	31 = FUEL AXIAL STRESS (PSI)	IZFSTS
C	32 = FUEL SHEAR STRESS (PSI)	ISFSTS
C	33 = CLAD RADIAL STRESS (PSI)	IRCSTS
C	34 = CLAD HOOP STRESS (PSI)	ITCSTS
C	35 = CLAD AXIAL STRESS (PSI)	IZCSTS
C	36 = CLAD SHEAR STRESS (PSI)	ISCSTS
C	37 = GAS RELEASE FRACTION	IRDOT
C	38 = CLAD TOTAL LENGTH CHANGE	SUMM
C	39 = TIME STEP	INTERV
C	39 = TIME STEP	RTIME

C*****
C*****

DATA (NOM(I),I=1,39)/11,14,28,10,17,30,27,35,28,32,14,18,24,25,20,
+28,25,30,26,24,22,25,24,25,26,27,26,27,25,23,20,24,25,23,24,24,20,
+20,9/

DATA (TITL(I),I=1,76) /
+10H TIME (DAYS,10H) ,1H ,1H ,
+10H TIME (SECO,10HNDST) ,10H ,10H
+10H INTERNAL G,10HAS PRESSUR,10HP (PSI)+ ,10H ,
+10H POWER (KW,10HFT)+ ,10H ,10H
+10H BURNUP (MW,10H0/MTM)+ ,10H ,10H
+10H FILM COEFF,10H (BTU/HR-F,10H12-DEG F)+ ,10H ,
+10H FUEL DELTA,10H R SWELLIN,10HG (IN)+ ,10H ,
+10H FUEL DELTA,10H R THERMAL,10H EXPANSION,10H (IN)+ ,
+10H FUEL DELTA,10H R DENSIFI,10H CATION (IN,10H)+ ,
+10H FUEL DELTA,10H R RELOCAT,10H (IN) ,10H ,
+10H PIN AXIAL ,10H MODE ,10H ,10H ,
+10H CLAD DELTA,10H R (IN)+ ,10H ,10H ,
+10H TOTAL FUEL,10H DELTA R (,10H IN)+ ,10H ,
+10H RESTRUCTUR,10H ED RADIUS ,10H (IN)+ ,10H ,
+10H RADIAL HOT,10H GAP (IN)+ ,10H ,10H ,

```

+10HCONCLANT TE,10HMPERATURE ,10H(DEG F)* ,10H
+10HFUEL-CLAD ,10HMPRESSURE (,10HPSI)* ,10H
+10HGAP CONDC,10HTANCE (RTU,10H/HR-FT-DEG,10H F)*
+10HCLAD DELTA,10H R ELASTIC,10H (IN)* ,10H
DATA (TITL(I),I=77,152) /
+10HCLAD DELTA,10H R CREEP (,10HIN)* ,10H
+10HFUEL VOID ,10HRADIUS (IN,10H)* ,10H
+10HRADIUS FUE,10HL SURFACE ,10H(IN)* ,10H
+10HRADIUS CLA,10HD INSIDE (,10HIN)* ,10H
+10HRADIUS CLA,10HD OUTSIDE ,10H(IN)* ,10H
+10HTEMP. FUEL,10H CENTER (D,10HDEG F)* ,10H
+10HTEMP. FUEL,10H SURFACE (,10HDEG F)* ,10H
+10HTEMP. CLAD,10H INSIDE (D,10HDEG F)* ,10H
+10HTEMP. CLAD,10H OUTSIDE (,10HDEG F)* ,10H
+10HFUEL RADIA,10HL STRESS (,10HPSI)* ,10H
+10HFUEL HOOP ,10HSTRESS (PS,10HI)* ,10H
+10HFUEL AXIAL,10H STRESS (P,10HSI)* ,10H
+10HFUEL SHEAR,10H STRESS (P,10HSI)* ,10H
+10HCLAD RADIA,10HL STRESS (,10HPSI)* ,10H
+10HCLAD HOOP ,10HSTRESS (PS,10HI)* ,10H
+10HCLAD AXIAL,10H STRESS (P,10HSI)* ,10H
+10HCLAD SHEAR,10H STRESS (P,10HSI)* ,10H
+10HGAS RELEASE,10HE FRACTION,10H ,10H
+10HCLAD TOTAL,10H LENGTH CH,10HANGE (IN)* ,10H /
DATA (TITL(I),I=153,156) /
+10HTIME STEP ,1H ,1H ,1H /
DATA YNP,NPLT /0.,0/
Y1=YL
DO 200 I1=1,7
IF(IFL(I1,I1,IXAX),LT,1) GO TO 200
NCUR=0
DO 251 I3=1,6
IF(IFL(I3,I1,IXAX),LT,1) GO TO 251
NCUR=NCUR+1
251 CONTINUE
YL=Y1-.3+FLOAT(NCUR)*.5
C PLOT THE AXIS
CALL NEWPEN(1)
JSEG=C/10
DO 10 I=1,JSEG
X=XL/FLOAT(JSEG)*FLOAT(I)
CALL PLOT(X,0,0,2)
CALL PLOT(X,0,1,2)
CALL PLOT(X,0,0,2)
10 CONTINUE
IC=IFIX(C)
DO 180 I=1,IC
XT=(XL/C)*FLOAT(I)
CALL PLOT(XT,0,0,3)
CALL PLOT(XT,.05,2)
180 CONTINUE
IA=IFIX(A)
ISEG=IA/5
DO 20 I=1,ISEG
Y=YL/FLOAT(ISEG)*FLOAT(I)
CALL PLOT(X,Y,2)
CALL PLOT(X-.1,Y,2)
CALL PLOT(X,Y,2)
20 CONTINUE
DO 170 IT=1,IA

```



```

      YT=(YL/A)*FLD(ATT)
      CALL PLOT(X,YT,3)
      CALL PLOT(X+.05,YT,2)
170 CONTINUE
      DO 30 I=1,JSEG
      XX=XI-(X1/FLD(JSEG))*FLD(ATT)
      CALL PLOT(XX,Y,2)
      CALL PLOT(XX,Y+.1,2)
      CALL PLOT(XX,Y,2)
30 CONTINUE
      DO 181 IT=1,ITC
      XYT=XI-(X1/IC)*FLD(ATT)
      CALL PLOT(XYT,YL,3)
      CALL PLOT(XYT,YL+.05,2)
181 CONTINUE
      DO 40 NI=1,ISEG
      YY=YL-(YL/FLD(ISEG))*FLD(ATT)
      CALL PLOT(XX,YY,2)
      CALL PLOT(XX+.1,YY,2)
      CALL PLOT(XX,YY,2)
40 CONTINUE
      DO 171 IT=1,ITA
      YT=(YL/A)*FLD(ITT)
      CALL PLOT(XX,YT,3)
      CALL PLOT(XX+.05,YT,2)
171 CONTINUE
      YS=YI+.5
      DO 49 J=1,ACUR
      VALUE=0.
      DO 43 I=1,3P
      JFIX(I)=0
      KFIX(I)=0
43 CONTINUE
      NSEG=D/(100,J,IT)
      DO 100 I=1,NSEG
      XARRAY(I)=D(I,7,IT)
      IF(XARRAY(I).GE.VALUE) VALUE=XARRAY(I)
      YARRAY(I)=D(I,1,IT)
      IF(YARRAY(I).GE.VALUE) VALUE=YARRAY(I)
100 CONTINUE
      IF(VALUE.EQ.0.) GO TO 42
      I=IFL(7,IT,IXAX)
      IF(VALUE.GT.C) GO TO 44
      IF(VALUE.LT.C/10.) GO TO 45
      GO TO 42
44 VALUE=VALUE/10.
      JFIX(I)=IFIX(I)+1
      IF(VALUE.GT.C) GO TO 44
      GO TO 42
45 VALUE=VALUE*10.
      JFIX(I)=JFIX(I)+1
      IF(VALUE.LT.C/10.) GO TO 45
42 CONTINUE
      DO 46 K=1,NSEG
      XARRAY(K)=XARRAY(K)*10.++JFIX(I)
46 CONTINUE
      XJFIX=FLD(JFIX(I))
      IF(VALUE.EQ.0.) GO TO 43
      I=IFL(I,IT,IXAX)

```

```

      IF(VALUE.GT.A) GO TO 15
      IF(VALUE.LT.A/10.) GO TO 25
      GO TO A0
15  VALUE=VALUE/10.
      KFIX(I)=KFIX(I)-1
      IF(VALUE.GT.A) GO TO 15
      GO TO A0
25  VALUE=VALUE*10.
      KFIX(I)=KFIX(I)+1
      IF(VALUE.LT.A/10.) GO TO 25
80  CONTINUE
      DO 120 K=1,NSEG
          YARRAY(K)=YARRAY(K)+(10.**KFIX(I))
120 CONTINUE
C   PUT TITLES ON PLOT
      XS=.5
      XTITLE=TFI(J,I,XYAX)
      WRITE(6,740)
      WRITE(6,1000) NDIR,XTITLE
      WRITE(6,740)
      WRITE(6,150)(XARRAY(I),I=1,NSEG)
      WRITE(6,745)
      WRITE(6,150)(YARRAY(I),I=1,NSEG)
      XSN=YS+.3+(.14*FLOAT(NUM(XTITLE)))
      XSN1=XSN+.28
      DO 50 MI=1,4
          XLABEL(MI)=TITL(XTITLE+4*(4-MI))
50  CONTINUE
      XKFIX=FLOAT(KFIX(I))
      CALL SYMBOL(XS,YS+.07,.14,J,0.,-1)
      CALL SYMBOL(XS+.2,YS+.14,XLABEL,0.,NUM(XTITLE))
      CALL SYMBOL(XSN,YS,.14,2HE,0,0,2)
      CALL NUMBER(XSN1,YS,.14,XKFIX,0,0,-1)
      YS=YS-.3
      IF(YARRAY(NSEG).EQ.0.) NSEG=NSEG-1
      XARRAY(NSEG+1)=0.
      XARRAY(NSEG+2)=C/XL
      YARRAY(NSEG+1)=0.
      YARRAY(NSEG+2)=A/YL
      CALL LINE(XARRAY,YARRAY,NSEG,1.,J)
49  CONTINUE
C   LABEL THE X-Y AXIS
      XDIS=0.
      IC=C+1.
      DO 70 I=1,IC,10
          CS=FLOAT(I)-1.
          CALL NUMBER(XDIS,-.4,.14,CS,0,0,-1)
          XDIS=XDIS+(XL/FLOAT(JSEG))
70  CONTINUE
      YDIS=0.
      A1=A+1
      F=1.
      L=5
      IF(A.GT.10.) L=10
      IF(L.EQ.10) F=2.
      DO 60 I=1,A1,L
          AY=FLOAT(I)-1.
          CALL NUMBER(-.5,YDIS,.14,AY,0,0,-1)
          YDIS=YDIS+(YL/FLOAT(ISEG))+F
60  CONTINUE

```

```

CALL SYMBOL(XX+.5,Y1+.2,.14,FTITLE,0.0,30)
IF(IXAX.EQ.2) GO TO 65
CALL SYMBOL(XX+.5,Y1+.2,.14,6RTIME =,0.0,5)
CALL NUMBER(XX+.5,Y1+.2,.14,TRAYS,0.0,1)
CALL SYMBOL(XX+.5,Y1+.2,.14,4HTRAYS,0.0,4)
GO TO 64
65 CONTINUE
CALL SYMBOL(XX+.5,Y1+.2,.14,0-TEMPER =,0.0,9)
ZPOW=FLOAT(KPOW)
CALL NUMBER(XX+.5,Y1+.2,.14,ZPOW,0.0,-1)
66 CONTINUE
PUT TITLES ON THE APOISSA
XS=.5
XTITLE=IFL(7,T1,IXAX)
XSN=XS+.3+.14*FLOAT(MIN(XTITLE,1))
XSN1=XSN+.28
DO 51 MI=1,4
XLAB(MI)=TITLE(XTITLE+1-(4-MI))
51 CONTINUE
CALL SYMBOL(XS+.2,-.6,.14,XLAB,0.0,MIN(XTITLE,1))
CALL SYMBOL(XSN,-.6,.14,2HE,0.0,2)
CALL NUMBER(XSN1,-.6,.14,4JFL,0.0,-1)
YNP=YNP+Y1+3.
NPLT=NPLT+1
YAX=YNP+Y1+3.
IF(YAX.GT.30.) GO TO 2000
CALL PLOT(0.5,Y1+3.,-5)
GO TO 2001
2000 CONTINUE
YSP=(Y1+3.1*(FLOAT(NPLT)-1.))
CALL PLOT(X1+3.,-YSP,+3)
YNP=1.
NPLT=0
2001 CONTINUE
200 CONTINUE
YL=Y1
RETURN
150 FORMAT(5X,10F10.2)
740 FORMAT(3X,2YAPRAY VALUES)
745 FORMAT(3X,2YAPRAY VALUES)
760 FORMAT(//)
1000 FORMAT(10X,ZNDOR =Z,13.5X,XTITLE =Z,3X,TR)
END

```

```

*
100. 7. 10. 8. 60. PLOTS SAMPLE
$IN
ITFI=1,-1.
ITFO=1,-1.
ITCI=1,-1.
ITCQOL=1.
IBURNU=1.
IPOWER=1,2,-1,-2.
IRFI=2,-2.
IRFO=2,-2.
IRCI=2,-2.
IRCQ=2,-2.
IAXNOD=11,12.
ITDAYS=-11,-12.
KPOW=4,
$

```

PLOT3 SAMPLE

TIME =0.0 DAYS

○ POWER (KW/FT)* E 2
 △ BURNUP (MWD/MTM)* E 0
 + COOLANT TEMPERATURE (DEG F)* E -1
 X TEMP. FUEL CENTER (DEG F)* E -1
 ◇ TEMP. FUEL SURFACE (DEG F)* E -1
 + TEMP. CLAD INSIDE (DEG F)* E -1

DATE= 18/10/77 TIME 16.4

C-25

○ POWER (KW/FT)* E 2

△ FUEL VOID RADIUS (IN)* E 4

+ RADIUS FUEL SURFACE (IN)* E 2

X RADIUS CLAD INSIDE (IN)* E 2

◇ RADIUS CLAD OUTSIDE (IN)* E 2

PLOT3 SAMPLE

TIME =15.0 DAYS

○ POWER (KW/FT)* E 1
△ BURNUP (MWD/MTM)* E -1
+ COOLANT TEMPERATURE (DEG F)* E -1
X TEMP. FUEL CENTER (DEG F)* E -2
◇ TEMP. FUEL SURFACE (DEG F)* E -1
⋈ TEMP. CLAD INSIDE (DEG F)* E -1

C-27

PLOT3 SAMPLE

TIME =15.0 DAYS

○ POWER (KW/FT)* E 1
△ FUEL VOID RADIUS (IN)* E 4
+ RADIUS FUEL SURFACE (IN)* E 2
X RADIUS CLAD INSIDE (IN)* E 2
◇ RADIUS CLAD OUTSIDE (IN)* E 2

C-28

PLOT3 SAMPLE

TIME =30.0 DAYS

○ POWER (KW/FT)* E 3
 △ BURNUP (MWD/MTM)* E -1
 + COOLANT TEMPERATURE (DEC F)* E -1
 X TEMP. FUEL CENTER (DEC F)* E -2
 ◊ TEMP. FUEL SURFACE (DEC F)* E -1
 + TEMP. CLAD INSIDE (DEC F)* E -1

PLOT3 SAMPLE

TIME =30.0 DAYS

○ POWER (KW/FT)* E 1
△ FUEL VOID RADIUS (IN)* E 4
+ RADIUS FUEL SURFACE (IN)* E 2
X RADIUS CLAD INSIDE (IN)* E 2
◇ RADIUS CLAD OUTSIDE (IN)* E 2

C-30

PLOT3 SAMPLE

TIME =60.0 DAYS

○ POWER (KW/FT)* E 1
 △ BURNUP (MWD/MTM)* E -2
 + COOLANT TEMPERATURE (DEG F)* E -1
 X TEMP. FUEL CENTER (DEG F)* E -2
 ◁ TEMP. FUEL SURFACE (DEG F)* E -2
 + TEMP. CLAD INSIDE (DEG F)* E -1

PLOT3 SAMPLE

TIME =60.0 DAYS

○ POWER (KW/FT)* E 1
△ FUEL VOID RADIUS (IN)* E 4
+ RADIUS FUEL SURFACE (IN)* E 2
× RADIUS CLAD INSIDE (IN)* E 2
◇ RADIUS CLAD OUTSIDE (IN)* E 2

PLOT3 SAMPLE

NODE NO. 3

○ POWER (KW/FT)* E 1
△ TEMP. FUEL CENTER (DEG F)* E -2
+ TEMP. FUEL SURFACE (DEG F)* E -2
X TEMP. CLAD INSIDE (DEG F)* E -1

PLOT3 SAMPLE

NODE NO. 3

○ POWER (KW/FT)* E 1
△ FUEL VOID RADIUS (IN)* E 4
+ RADIUS FUEL SURFACE (IN)* E 2
X RADIUS CLAD INSIDE (IN)* E 2
◇ RADIUS CLAD OUTSIDE (IN)* E 2

APPENDIX D

This appendix contains a listing of the program and its subroutines arranged in alphabetical order. Output for the sample problem is included at the end.

D-1

C	SIZE INCRIMENTS.	GAPCON3	35
C		GAPCON3	36
C		GAPCON3	37
C	RADIAL CALCULATION PARAMETERS	GAPCON3	38
C		GAPCON3	39
C	NFF=TOTAL NODES IN RADIAL MECHANICAL MODEL	GAPCON3	40
C	NF4=TOTAL D.O.F. IN RADIAL MODEL OF FUEL	GAPCON3	41
C	NC4=TOTAL D.O.F. IN RADIAL MODEL OF CLAD	GAPCON3	42
C	NFF2=NUMBER OF D.O.F. IN COMBINED FUEL-CLAD MODEL	GAPCON3	43
C	PLUS 4.	GAPCON3	44
C	NOELC=NUMBER OF RADIAL CLAD ELEMENTS	GAPCON3	45
C	NOELF=NUMBER OF RADIAL FUEL ELEMENTS	GAPCON3	46
C	NOELC=TOTAL NUMBER OF RADIAL ELEMENTS	GAPCON3	47
C	NFF1=TOTAL NUMBER OF RADIAL BOUNDARIES	GAPCON3	48
C		GAPCON3	49
	NFF = (NF+NC)*2	GAPCON3	50
	NF4 = 4*NF	GAPCON3	51
	NC4 = 4*NC	GAPCON3	52
	NFF1 = NF+NC	GAPCON3	53
	NFF2 = 2*NFF	GAPCON3	54
	NOELC = (NC+1)*2	GAPCON3	55
	NOELF = (NF+1)*2	GAPCON3	56
	NOEL = NOELC+NOELF	GAPCON3	57
C		GAPCON3	58
C	AXIAL CALCULATION PARAMETERS	GAPCON3	59
C		GAPCON3	60
C	NARF=NUMBER OF AXIAL REGIONS IN THE FUEL	GAPCON3	61
C	NART=NARC=TOTAL NUMBER OF AXIAL REGIONS	GAPCON3	62
C	NART,NARF=NUMBER OF AXIAL BOUNDARIES IN CLAD AND FUEL, RESPECTIVELY	GAPCON3	63
C	NAXB1=TOTAL NUMBER OF AXIAL BOUNDARIES	GAPCON3	64
C	NAXB2=TOTAL NUMBER OF NODES IN AXIAL MODEL	GAPCON3	65
C	NAXR4=TOTAL NUMBER OF D.O.F. IN AXIAL MODEL	GAPCON3	66
C	NAEF=NUMBER OF ELEMENTS IN AXIAL FUEL MODEL	GAPCON3	67
C	NAEC=NUMBER OF ELEMENTS IN AXIAL CLAD MODEL	GAPCON3	68
C	NTAE=TOTAL NUMBER OF ELEMENTS IN THE AXIAL MODEL	GAPCON3	69
C		GAPCON3	70
	NABF = NARF+1	GAPCON3	71
	NABT = NART+1	GAPCON3	72
	NAXB1 = NABT+NABF	GAPCON3	73
	NABF2 = NABF*2	GAPCON3	74
	NABT2 = 2*NABT	GAPCON3	75
	NAXB2 = NAXB1*2+NABF+NABT	GAPCON3	76
	NAXR4 = 4*NAXB1+(NABF+NABT)*2	GAPCON3	77
	NAEF = NARF*4	GAPCON3	78
	NAEC = NART*4	GAPCON3	79
	NTAE = NAEF+NAEC	GAPCON3	80
	NABC = NABT	GAPCON3	81
	NARC = NART	GAPCON3	82
C	PARAMETERS FOR CHOLESKY (MINIMUM-SPACE) STORAGE OF STIFFNESS	GAPCON3	83
C	MATRICES	GAPCON3	84
C		GAPCON3	85
C	MAXF=TOTAL NUMBER OF STORAGE LOCATIONS REQUIRED FOR THE FUEL	GAPCON3	86
C	RADIAL STIFFNESS MATRIX	GAPCON3	87
C	MAXC=STORAGE REQUIRED FOR RADIAL CLAD MATRIX	GAPCON3	88
C	MAXT=STORAGE FOR COMBINED FUEL-CLAD RADIAL MATRIX	GAPCON3	89
C	MAXA=STORAGE FOR THE AXIAL MODEL MATRIX	GAPCON3	90
C		GAPCON3	91

MAXF = NF4+7*(NF4-4)
 MAXC = NC4+7*(NC4-4)
 NMX = NFF2-4
 MAXT = NMX+7*(NMX-4)
 MAXA = (MAX64+25*(MAX64-13))*2

C					GAPCON3	92
C					GAPCON3	93
C					GAPCON3	94
C					GAPCON3	95
C					GAPCON3	96
C					GAPCON3	97
C					GAPCON3	98
C					GAPCON3	99
C					GAPCON3	100
C	VARIABLE	POINTER	MEANING	UNITS	GAPCON3	101
C					GAPCON3	102
C	AELM	IAELM	AXIAL MODEL EL. NODE NO'S	---	GAPCON3	103
C	AK	IAK	AXIAL STIFFNESS MATRIX	POUNDS/IN	GAPCON3	104
C	ARAA	IAARA	AXIAL EL. AREA	IN**2	GAPCON3	105
C	AREA	IARE	RADIAL EL. AREA	IN**2	GAPCON3	106
C	B	ISB	COMB. RADIAL MODEL DISP.	IN	GAPCON3	107
C	BA	IHS	IBAAI MODEL DISP.	IN	GAPCON3	108
C	BASAVE	IBASAV	SAVE AXIAL MOD. LOAD	POUNDS	GAPCON3	109
C	BASE	IHASE	SAVE INCR. EL. EFFECT. STRESS FOR PLASTIC CALC.	PSI	GAPCON3	110
C					GAPCON3	111
C	BB	ISBB	COMB. RAD. MODEL LOADS	POUNDS	GAPCON3	112
C	BBA	IBBA	AXIAL MODEL LOADS	POUNDS	GAPCON3	113
C	BBC	ISBBC	CLAD RAD. MODEL LOADS	POUNDS	GAPCON3	114
C	BBF	ISBBF	FUEL RAD. MODEL LOADS	POUNDS	GAPCON3	115
C	BC	ISBC	CLAD RAD. MODEL DISP.	IN	GAPCON3	116
C	BF	ISBF	FUEL RAD MODEL DISP.	IN	GAPCON3	117
C	BBSAV	IBBSV	SAVE COMB. RAD. MOD. LOADS	POUNDS	GAPCON3	118
C	CA	ICA	AXIAL STRAIN-TO-STRESS CONV.	PSI	GAPCON3	119
C	CC	ICC	RADIAL STRAIN-TO-STRESS CONV.	PSI	GAPCON3	120
C	CRATE	ICRATE	GEN. STRAIN EL. INCR.	IN/IN	GAPCON3	121
C	CRPG	ICRPG	GEN. EL. STRAIN	IN/IN	GAPCON3	122
C	DISPA	IDISP	AXIAL MOD. DISPLACEMENTS	IN	GAPCON3	123
C	DISPC	IDISPC	CLAD OR COMB. RAD. MOD. DISP.	IN	GAPCON3	124
C	DISPF	IDISPF	FUEL ONLY RAD MOD. DISP.	IN	GAPCON3	125
C	D6PSV	ID6PSV	SAVE INNER CLAD RAD. DISP.	IN	GAPCON3	126
C	EFSTR	IEFSTR	EFFECTIVE STRESS-RAD. MOD.	PSI	GAPCON3	127
C	ELM	IELM	RADIAL MOD. EL. NODE. NO'S	---	GAPCON3	128
C	ELSTN	IELSTN	INCR. EL. STRAIN-RAD. MOD.	IN/IN	GAPCON3	129
C	ELSTNA	ILSTNA	INCR. EL. STRAIN. AXIAL MOD.	IN/IN	GAPCON3	130
C	ELSTRA	ILSTRA	INCR EL. STRESS-AXIAL MOD.	PSI	GAPCON3	131
C	ELSTRS	IELSTR	INCR. EL. STRESS-RAD. MOD.	PSI	GAPCON3	132
C	EPSLA	IPSLA	DISP.-TO-STRAIN CONV.-AX. MOD.	IN**1	GAPCON3	133
C	EPSLM	IEPSLM	DIS.-TO-STRAIN CONV.-RAD. MOD.	IN**1	GAPCON3	134
C	ESTNV	IESTNV	SAVED VALUES RAD. MOD. INCR. STN.	IN/IN	GAPCON3	135
C	ESTSV	IESTSV	SAVED VALUES RAD. MOD. INCR. STRS.	PSI	GAPCON3	136
C					GAPCON3	137
C	IACON	IACN	SIGNAL OF AX. BOND, CONTACT.	---	GAPCON3	138
C	IDO	IIDO	SIGNAL FOR RAD. CALC.	---	GAPCON3	139
C	IJ	IJJ	POSITION VECTOR, PLASTIC CALC.	---	GAPCON3	140
C	IPDS	IIPDS	POSITION VECTOR, PLASTIC CALC.	---	GAPCON3	141
C	KAN	IKAN	SIGNAL FOR COMB. RAD. CALC.	---	GAPCON3	142
C	NADET	INADET	D.O.F. ARRAY FOR RAD. MOD.	---	GAPCON3	143
C	NDC	INDC	AIDS TO D.O.F. ARRAY-RAD. MOD.	---	GAPCON3	144
C	NDF	INDF	AIDS TO D.O.F. ARRAY-RAD. MOD.	---	GAPCON3	145
C	NDT	INDT	AIDS TO D.O.F. ARRAY-RAD. MOD.	---	GAPCON3	146
C	NODE	INODE	D.O.F. ARRAY-RAD. FUEL MOD.	---	GAPCON3	147
C	NODEC	INODEC	D.O.F. ARRAY-RAD. CLAD MOD.	---	GAPCON3	148

C	NODET	INODET	D.O.F. ARRAY=RAD. COMB. MOD.	---	GAPCON3	149
C	OLDSE	IOLDSE	PREV. EL. EFFECT. STRESS	PSI	GAPCON3	150
C	PRESS	IPRESS	INTERFACIAL PRESSURE	PSI	GAPCON3	151
C	PSSRT	IPSSRT	SORTED ARRAY OF PSSV	PSI	GAPCON3	152
C	PSSV	IPSSV	SAVED VALUES OF PLASTICITY	PSI	GAPCON3	153
C	PSTS	IPSTS	CLAD. EL. YIELD STRESS, RAD. MOD.	PSI	GAPCON3	154
C	R	IR	ARRAY OF RAD. NODE LOCATIONS=	IN	GAPCON3	155
C			RAD. MOD.		GAPCON3	156
C	RA	IRA	ARRAY OF RAD. NODE LOCATIONS=	IN	GAPCON3	157
C			AX. MOD.		GAPCON3	158
C	RBR	IRBR	EL. CENTROID RAD. LOCATION=	IN	GAPCON3	159
C			RAD. MODEL		GAPCON3	160
C	RBRA	IRBRA	EL. CENTROID RAD. LOCATION=	IN	GAPCON3	161
C			AX. MODEL		GAPCON3	162
C	RD	IRD	GEN. ARRAY OF RAD. BOUNDARY	IN	GAPCON3	163
C			LOCATIONS, RAD. MODEL		GAPCON3	164
C	STRESS	ISTRES	DISP.=TO-STRESS CONV.=RAD. MOD.	PSI/IN	GAPCON3	165
C	STRSA	ISTRSA	DISP.=TO-STRESS CONV.=AX. MOD.	PSI/IN	GAPCON3	166
C	TCB	ITCB	CLAD RAD. BOUND. TEMP.=RAD. MOD.	F	GAPCON3	167
C	TCRP	ITCRP	INCR. EL. CREEP-PLAS. STRAIN=	IN/IN	GAPCON3	168
C			RAD. MOD.		GAPCON3	169
C	TDELE	ITDELE	INCR. CHANGE IN TELM	F	GAPCON3	170
C	TELM	ITELM	EL. TEMP.=RAD. MOD.	F	GAPCON3	171
C	TFB	ITFB	FUEL RAD. BOUND. TEM.=RAD. MOD.	F	GAPCON3	172
C	THET	ITHET	INCR. EL. TH. STRAIN=RAD. MOD.	IN/IN	GAPCON3	173
C	THETAX	ITHTAX	INCR. EL. TH. STRAIN=AX. MOD.	IN/IN	GAPCON3	174
C	THSAV	ITHSAV	SAVED VALUES OF THET	IN/IN	GAPCON3	175
C	THXSV	ITHXSV	SAVED VALUES OF THETAX	IN/IN	GAPCON3	176
C	TME	ITME	PAST, PRESENT, FUTURE TIME	DAYS	GAPCON3	177
C	TSTNR	ITSTNR	TOTAL EL. STRAIN=RAD. MOD.	PSI	GAPCON3	178
C	TSTSR	ITSTSR	TOTAL EL. STRESS=RAD. MOD.	PSI	GAPCON3	179
C	TW	ITW	EL. TEMP.=AX. MOD.	F	GAPCON3	180
C	TWDEL	ITWDEL	INCR. CHANGE IN TW	F	GAPCON3	181
C	Y	IY	AXIAL BOUND. LOCATIONS=AX. MOD.	IN	GAPCON3	182
C	YACF	IYACF	Y IN 1-D ARRAY	IN	GAPCON3	183
C	YM	IYM	INTERPOLATION WORK SPACE	---	GAPCON3	184
C	XX	IXK	FUEL ROD. MOD. STIFFNESS	POUNDS/IN	GAPCON3	185
C	XXC	IXKC	CLAD RAD. MOD. STIFFNESS	POUNDS/IN	GAPCON3	186
C	XXCC	IXKCC	SAVE VALUES OF XXC	POUNDS/IN	GAPCON3	187
C	XKT	IXKT	COMB. RAD. MOD. STIFFNESS	POUNDS/IN	GAPCON3	188
C	XKTT	IXKTT	SAVED VALUES OF XKT	POUNDS/IN	GAPCON3	189
C	XLOAD	IXLOAD	STRAIN=TO-LOAD CONV.=RAD. MOD.	POUNDS	GAPCON3	190
C	XLOADA	IXLOADA	STRAIN=TO-LOAD CONV.=AX. MOD.	POUNDS	GAPCON3	191
C	IDUD = 1				GAPCON3	192
	IELM = IDUD+20				GAPCON3	193
	IELSTN = IELM+3*NFF				GAPCON3	194
	IELSTR = IELSTN+4*NOEL				GAPCON3	195
	IR = IELSTR+4*NOEL				GAPCON3	196
	IZ = IR+NFF				GAPCON3	197
	INODE = IZ+NFF				GAPCON3	198
	IXK = INODE+NFF				GAPCON3	199
	IXKC = IXK+MAXF				GAPCON3	200
	IXKCC = IXKC+MAXC				GAPCON3	201
	INODEC = IXKCC+MAXC*NAHT				GAPCON3	202
	INODET = INODEC+NFF				GAPCON3	203
	IXKT = INODET+NFF				GAPCON3	204
	IXKTT = IXKT+MAXT				GAPCON3	205

IEPSTM = IXKTT+MAXT+NARF
 ISTRES = IEPSTM+36*NOEL
 IDISPC = ISTRES+36*NOEL
 IDISPF = IDISPC+NFF1*NABT
 IDISPT = IDISPF+NFF1*NABT
 INDC = IDISPT+NFF2
 INDF = INDC+NC4
 INDT = INDF+NF4
 IKSARE = INDT+NFF2
 IPRESS = IKSARE+36
 ISB = IPRESS+NABT
 ISBB = ISB+NFF2
 ISBF = ISBB+NFF2
 ISBC = ISBF+NF4
 ISBBF = ISBC+NC4
 ISBBC = ISBBF+NF4
 ITDELE = ISBBC+NC4
 ITELM = ITDELE+NOEL*NABF*2
 IARE = ITELM+NOEL*NABT*2
 IRBR = IARE+NOEL
 IXLOAD = IRBR+NOEL
 ICC = IXLOAD+6*4*NOEL
 ITHET = ICC+4*4*NOEL
 ICA = ITHET+4*NOEL
 IHTAX = ICA+4*4*NTAE
 INADET = IHTAX+4*NTAE
 IRA = INADET+NAXB4
 IY = IRA+4*NABT
 IYACF = IY+4*NABT
 IAEIM = IYACF+4*NABT
 IARAA = IAEIM+3*NTAE
 IBA = IARAA+NTAE
 IBBA = IBA+NAXB4
 IRBRA = IBBA+NAXB4
 IXLODA = IRBRA+NTAE
 ITW = IXLODA+6*4*NTAE
 ITWDEL = ITW+NTAE*2
 IDISP = ITWDEL+NTAE
 ILSTRA = IDISP+NAXB4
 ILSTNA = ILSTRA+4*NTAE
 ISTRSA = ILSTNA+4*NTAE
 IPSLA = ISTRSA+36*NTAE
 IAK = IPSLA+36*NTAE
 IACN = IAK+MAXA
 IRD = IACN+40
 IKAN = IRD+NFF1*NABT
 IUD = IKAN+NABT
 IYM = IUD+NABT
 ITSTR = IYM+NARC*(NOEL+1)
 ITSTNR = ITSTR+4*NOEL*NARC
 ITME = ITSTNR+4*NOEL*NARC
 IEFSTR = ITME+20
 ITCRP = IEFSTR+NOEL*NARC
 ICRPG = ITCRP+4*NOEL*NARC
 ICRATE = ICRPG+NOEL*NARC
 ITFB = ICRATE+NOEL*NARC
 ITR = ITFB+NF*NABF

GAPCON3 206
 GAPCON3 207
 GAPCON3 208
 GAPCON3 209
 GAPCON3 210
 GAPCON3 211
 GAPCON3 212
 GAPCON3 213
 GAPCON3 214
 GAPCON3 215
 GAPCON3 216
 GAPCON3 217
 GAPCON3 218
 GAPCON3 219
 GAPCON3 220
 GAPCON3 221
 GAPCON3 222
 GAPCON3 223
 GAPCON3 224
 GAPCON3 225
 GAPCON3 226
 GAPCON3 227
 GAPCON3 228
 GAPCON3 229
 GAPCON3 230
 GAPCON3 231
 GAPCON3 232
 GAPCON3 233
 GAPCON3 234
 GAPCON3 235
 GAPCON3 236
 GAPCON3 237
 GAPCON3 238
 GAPCON3 239
 GAPCON3 240
 GAPCON3 241
 GAPCON3 242
 GAPCON3 243
 GAPCON3 244
 GAPCON3 245
 GAPCON3 246
 GAPCON3 247
 GAPCON3 248
 GAPCON3 249
 GAPCON3 250
 GAPCON3 251
 GAPCON3 252
 GAPCON3 253
 GAPCON3 254
 GAPCON3 255
 GAPCON3 256
 GAPCON3 257
 GAPCON3 258
 GAPCON3 259
 GAPCON3 260
 GAPCON3 261
 GAPCON3 262

	ITCB = ITFM+NF*NARF	GAPCON3	263
	IBBSV = ITCB+NC*NARC	GAPCON3	264
	IESTSV = IBBSV+NFF2*NARC	GAPCON3	265
	IESTNV = IESTSV+NOEL*NARC*4	GAPCON3	266
	ITHSAV = IESTNV+NOEL*NARC*4	GAPCON3	267
	ITHXSV = ITHSAV+NOEL*NARC*4	GAPCON3	268
	IBASAV = ITHXSV+NTAE*4	GAPCON3	269
	IPSTS = IBASAV+NAXB4	GAPCON3	270
	IOLDSE = IPSTS+NOEL*NARF	GAPCON3	271
	IPSSV = IOLDSE+NOEL*NARF	GAPCON3	272
	IPSSRT = IPSSV+NOEL*NARF	GAPCON3	273
	JIIJ = IPSSRT+NOEL*NARF	GAPCON3	274
	IBASE = JIIJ+NOEL*NARF	GAPCON3	275
	IIPDS = IBASE+NOEL*NARF	GAPCON3	276
	IDSPSV = IIPDS+NOEL*NARF	GAPCON3	277
	ILAST = IDSPSV+NOEL*NARF	GAPCON3	278
	WRITE (6,120) ILAST	GAPCON3	279
C		GAPCON3	280
C		GAPCON3	281
	IF (ILAST.LE.6416) GO TO 100	GAPCON3	282
	IF (NABT.LT.4) GO TO 100	GAPCON3	283
	IF (ILAST.LE.ILIMIT) GO TO 110	GAPCON3	284
100	WRITE (6,130)	GAPCON3	285
	STOP	GAPCON3	286
110	CONTINUE	GAPCON3	287
	ILD = LOCF(D(1))	GAPCON3	288
	ILEN = ((ILD+7+ILAST)/64+1)*64	GAPCON3	289
	ILST = ILEN-ILD+1	GAPCON3	290
	CALL DISPLA (#CURE REQUIRED#,ILEN)	GAPCON3	291
	CALL EXFL (D,ILST)	GAPCON3	292
	CALL MOVEKA (0,D,ILST)	GAPCON3	293
C		GAPCON3	294
	IRZ = 0	GAPCON3	295
	CALL GAPCON (D(ITELM),D(ITDELE),D(ITW),D(IWDEL),D(IKAN),D(IPRESS)	GAPCON3	296
	+ ,D(IYM),D(IY),D(IIDO),TREF,RFO,RCI,RCU,HFI,CP,CR,NOELF,D(ITSTSR),D	GAPCON3	297
	+ (ITSTNR),NABT2,D(ITME),D(IDISPC),D(IDISPF),D(IRD),NFF1,D(ITFB),D(I	GAPCON3	298
	+ TFR),D(ITCB),NART,NABF,NABT,NF,NC,NOEL,NARF,NTAE,D(IEFSTR),D(IOLDS	GAPCON3	299
	+ E),D(IDSPSV))	GAPCON3	300
	STOP	GAPCON3	301
C		GAPCON3	302
C		GAPCON3	303
C		GAPCON3	304
C		GAPCON3	305
120	FORMAT (5X,#ILAST IS#,I10)	GAPCON3	306
130	FORMAT (5X,#VALUE OF NF,NC,OR NPOW IS TOO LARGE #)	GAPCON3	307
	END	GAPCON3	308

```

SUBROUTINE BLKDAT
C
C SOME COMPUTERS MAY REQUIRE THIS SUBROUTINE BE RENAMED BLOCK DATA
C IF SO ALSO REMOVE THE RETURN AT THE END OF THIS SUBROUTINE
C
COMMON/FP/BETA,D,E,G,H
COMMON/GA/NPOW,NTIME,NFZ,NCZ,IPLT,ATMOS,DCI,DCO,DFS,DSINZ,DTEMP
+,DVOIDZ,FR35,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRACN,FRACXE,FRD
+,EN,FRPUO2,FRSIN,IGAS,IPEAK,LFUEL,LVOIDZ, NFUEL,NOM,PRCDH,R
+,ADS,ROUC,ROUF,S,TINLET,YM,TPLAS,VPLENZ,XCO,XH,XN,IRELOC,IRL,IDENSF
+,FLXRAT,X,Z,ZZ,YY,WW,EXTP,ISTOR,NCLAD,ZCLAD,NSH,PROFIL(20),PSEUDO(
+,100),TIME(100),NFLX,TITLE(10),MXPWR,IOPT,JSTART,KSAVE,ISAVE,ITMCU
+,T,ISTAND,ITAPE,NSTEPS,MINI(20),SUMARY,FRDENZ
INTEGER SUMARY
REAL LFUEL,LVOIDZ,MXPWR
COMMON/BA/AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN
+,NCNN,ISTRT,ITER
+,FLXR
COMMON/FEP/HMF,NT,KKZZZ,ISTOP,ROOLD(20),RRSIN(20),ICNOLD(20)
+,IDOOOLD(20)
+,PINOLD,POUTLD,POLD,NTOLD(20)
+,ICHNG,CRPAD(20)
+,STEPS
COMMON/DA/ TABLE(80,2),GMWT(7),SIGLJ(7),EKLJ(7),HF(20),RTCD(5)
+,RTC(5),EPSIF,EP6IC
COMMON/EA/ZR(6,7),ZR4(6,7),ST(12,7),RV(20,2),UD2(7,7)
COMMON/FA/PI,CCPIN3,SECDAY,AVOGAD,RR,CONEN
DATA PI,CCPIN3,SECDAY,AVOGAD,RR,CONEN/3.14159,16.387,86400.,6.023E
+,23,22400.,3.2E-14/
DATA EP6IF,EP6IC/,8.,9/
DATA ((ZR(I,J),J=1,7),I=1,6)/75.,7.30,4.48E4,1.42E7,.366,3.24E-6,9
+,470.,212.,7.74,3.48E4,1.35E7,.361,3.47E-6,7360.,392.,8.38,2.38E4,1
+,25E7,.353,3.70E-6,5030.,572.,9.02,1.58E4,1.16E7,.345,3.87E-6,3340
+,752.,9.83,1.21E4,1.07E7,.337,3.99E-6,2560.,932.,10.64,1.00E4,9.8
+,0E6,.330,4.08E-6,2110./
C
C ZIRC=4 DATA BY D.B. SCOTT (WCAP-3629-41) PAGE 8
C
DATA ((ZR4(I,J),J=1,7),I=1,6)/75.,7.45,4.48E4,1.42E7,.366,3.24E-6,
+,9470.,212.,7.86,3.48E4,1.35E7,.361,3.47E-6,7360.,392.,8.27,3.48E4,
+,1.25E7,.353,3.47E-6,7360.,572.,8.79,3.48E4,1.16E7,.345,3.47E-6,736
+,0.,752.,9.48,3.48E4,1.07E7,.337,3.47E-6,7360.,932.,10.40,3.48E4,9.
+,80E6,.330,3.47E-6,7360./
DATA((ST(I,J),J=1,7),I=1,12)/
+ 100, 8.50, 29.0E3, 28.0E6, .266, 9.15E-6, 0.13E3,
+ 200, 8.93, 25.1E3, 27.6E6, .271, 9.33E-6, 5.31E3,
+ 300, 9.38, 22.4E3, 27.2E6, .277, 9.48E-6, 4.73E3,
+ 400, 9.80, 20.5E3, 26.6E6, .281, 9.60E-6, 4.33E3,
+ 500, 10.27, 19.2E3, 26.1E6, .285, 9.71E-6, 4.06E3,
+ 600, 10.70, 18.3E3, 25.4E6, .289, 9.82E-6, 3.87E3,
+ 700, 11.13, 17.5E3, 24.8E6, .293, 9.94E-6, 3.70E3,
+ 800, 11.57, 16.8E3, 24.1E6, .297, 10.07E-6, 3.51E3,
+ 900, 12.00, 16.2E3, 23.3E6, .301, 10.19E-6, 3.42E3,
+ 1000, 12.44, 15.6E3, 22.5E6, .305, 10.30E-6, 3.28E3,
+ 1100, 12.88, 15.0E3, 21.7E6, .309, 10.39E-6, 3.17E3,
+ 1200, 13.31, 14.2E3, 20.9E6, .313, 10.47E-6, 3.00E3/
C
TABLE OF THE OMEGA FUNCTION FROM BIRD, STEWART, AND LIGHTFOOT.
C

```

```

DATA ((TABLE(I,J),J=1,2),I=1,40)/.0,2.785,.3,2.785,.35,2.628,.40,2 BLKDAT 43
+.492,.45,2.368,.50,2.257,.55,2.156,.60,2.065,.65,1.982,.70,1.908,. BLKDAT 44
+75,1.841,.80,1.780,.85,1.725,.90,1.675,.95,1.629,1.0,1.587,1.05,1. BLKDAT 45
+549,1.10,1.514,1.15,1.462,1.20,1.424,1.25,1.424,1.30,1.399,1.35,1. BLKDAT 46
+375,1.40,1.353,1.45,1.333,1.50,1.314,1.55,1.296,1.60,1.279,1.65,1. BLKDAT 47
+264,1.70,1.248,1.75,1.234,1.80,1.221,1.85,1.209,1.90,1.197,1.95,1. BLKDAT 48
+186,2.00,1.175,2.10,1.156,2.20,1.138,2.30,1.122,2.4,1.107/ BLKDAT 49
DATA ((TABLE(I,J),J=1,2),I=41,80)/2.50,1.093,2.60,1.081,2.7,1.069, BLKDAT 50
+2.8,1.058,2.9,1.048,3.0,1.039,3.1,1.030,3.2,1.022,3.3,1.014,3.4,1. BLKDAT 51
+007,3.5,.9999,3.6,.9932,3.7,.9870,3.8,.9811,3.9,.9755,4.0,.9700,4. BLKDAT 52
+1.9649,4.2,.9600,4.3,.9533,4.4,.9507,4.5,.9464,4.6,.9422,4.7,.938 BLKDAT 53
+2.4,8,.9343,4.9,.9305,5.0,.9269,6.0,.8963,7.0,.8727,8.0,.8538,9.0,.83 BLKDAT 54
+79,10.0,.8242,20.0,.7432,30.0,.7005,40.0,.6718,50.0,.6504,60.0,.6335,70. BLKDAT 55
+.6194,80.0,.6076,90.0,.5973,200.0,.5882/ BLKDAT 56
DATA RTC/1.429E-3,1.51E-3,1.622E-3,1.699E-3,1.876E-3/ BLKDAT 57
DATA RTCO/2.05E-15,3.9E-15,5.2E-15,7.82E-15,10.04E-15/ BLKDAT 58
DATA GMMT/4.003,39.944,2.016,28.02,28.01,83.8,131.3/ BLKDAT 59
DATA SIGLJ/2.576,3.418,2.915,3.59,3.681,3.498,4.055/ BLKDAT 60
DATA EKLJ/10.2,124.0,38.0,91.5,110.3,225.0,229.0/ BLKDAT 61
DATA ((UG2(I,J),J=1,7),I=1,7)/77.0,2*0.31,7E6,.316,4.39E-6,0.932,.2 BLKDAT 62
+*0.28,9E6,.310,4.39E-6,0.1832,.2*0.28,3E6,.304,5.289E-6,0.2732,.2* BLKDAT 63
+0.22,5E6,.298,6.126E-6,0.3632,.2*0.27,0E6,.295,6.946E-6,0.4532,.2* BLKDAT 64
+0.13,6E6,.295,7.760E-6,0.5054,.2*0.9,5E6,.295,8.231E-6,0/ BLKDAT 65
DATA MF/20*15000./ BLKDAT 66
DATA DELZA/20*0.0/ BLKDAT 67
DATA JSTART/0/,KSAVE/0/,NSTEP5/0/,ITAPE/0/ BLKDAT 68
C THE FOLLOWING CARDS INITIALIZE SAMPLE PROBLEM VALUES BLKDAT 69
DATA TITLE/1H,4MSAMP,4HLE P,4HROBL,2HEM,5*1H / BLKDAT 70
DATA D,BETA,E,G,H/1.133,2.34,07,26.57,6.6E-13/ BLKDAT 71
DATA ATMOS,DCI,DCU,DFS,DTEMP/1.0,429.5035,42,20./ BLKDAT 72
DATA FR3S,FRACHE,FRDEN,FRDEN2,FRSIN,LFUEL/.03,1.0,95.95,.96,144./ BLKDAT 73
DATA MXPQWR,NPOW,NTIME,ROUC,ROUF,TINLET/10.4,4.0,2E-4,.39E-4,533./ BLKDAT 74
DATA TH,TPLAS,X,Z,VPLENZ/2790.1200.0,27.125,2.0/ BLKDAT 75
DATA TIME/.0,15.30.00,96*0/ BLKDAT 76
DATA PSEUDO/.5,3.6,10.96*0/ BLKDAT 77
DATA EXTP/1050./ BLKDAT 78
DATA PROFIL/.5,9,1.1,1.1,9,15*0/ BLKDAT 79
DATA ITMCUT/10/ BLKDAT 80
DATA FLXRAT/.01/ BLKDAT 81
DATA (MINI(I),I=1,20)/0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/ BLKDAT 82
DATA DSINZ/.0/,DVOIDZ/.0/,FRACAR/.0/,FRACH/.0/,FRACKH/.0/ BLKDAT 83
DATA FRACH/.0/,FRACXE/.0/,FRPUO2/.0/,FR40/.0/,FR41/.0/,IDENSEF/0/ BLKDAT 84
DATA IGAS/0/,IDPT/0/,IPEAK/0/,IPLUT/0/,IRELOC/0/,IRL/0/,ISTOR/0/ BLKDAT 85
DATA LVOIDZ/0.0/,NCLAD/0/,NFLX/0/,NFUEL/0/,NGH/0/,NSH/0/,PRCDH/0/ BLKDAT 86
DATA RAD5/.0/,S/.0/,SUMARY/0/,XCO/.0/,XH/.0/,XN/.0/,ZCLAD/0/ BLKDAT 87
DATA ZZ/.0/,YY/.0/,WW/.0/ BLKDAT 88
RETURN BLKDAT 89
END BLKDAT 90

```

```

C      SUBROUTINE ABOUND (AK,DISPA,NADET)
C      THE PURPOSE OF THIS SUBROUTINE IS TO DEFINE THE INITIAL BOUNDARY
C      CONDITIONS FOR THE AXIAL FINITE ELEMENT MODEL. IT USES THE
C      STIFFNESS MATRIX AK(I,J) WHERE I,J=ARE EQUAL TO THE TOTAL
C      DEGREES OF FREEDOM IN THE PROBLEM. FOR THIS CASE THE MODEL IS IN
C      THE
C      Z DIRECTION FOR NODES 1, NABF+1,NABF+2+1
C      NABF+2+NABF+1
C      COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IETPSLM,IELSTN,IELSTR
C      +,IKSAVE,IPRESS,ITEM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD
C      +ET,IR,IRBR,ISA,ISAC,ISAF,ISH,ISBB,ISBBC,I
C      +SBBF,ISBC,ISHF,ISK,ISKC,ISTRES,ITDELE,IXK,IXKC,IXKT
C      +,IXLOAD,ICC,ITHET,ICA,ITHTAX,INADET,IRA,IY,IAELM,I
C      +ARAA,IBA,IBBA,IRBBA,IXLODA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I
C      +PSLA,IAK,IACN,IKAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA,
C      +IIDU,ITME,IEFSTR,ITCRP,ICRPG,ICHATE
C      +,IXKCC,IXKTT
C      +,ITFB,ITCB,IBBSV,IESTSV,IESTNV
C      +,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDSE,IPSSV,IPSSRT,JIIJ,IIPDS
C      +,IBASE,ITFR,IYM,IZ
C      +,IDSPSV
C      COMMON /GENPAR/ RFI,RFO,RCI,RCO,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUT,P
C      +DEL
C      COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IDF,NFF,MAXDC,NETOT,NE
C      +WF,NEHT,KA,NARF,NARC,NART,NABF,NABC,NABT,NAEC,NAEF,NTAE,JAM,NAXBT,
C      +NAXB2,NAXB4,NFF1,NABF2,NABT2,CP,CR,PRDEN,KZ1,ISSTOP,IPLAS,NKN
C      +,ICRPP,MAXF,MAXC,MAXT,KRNCH,CRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N
C      +MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD
C      +,KPLAS,ICNT,RRAKR,IRAK,RAKSUM,IPLEND,ISP
C      COMMON /LASIC/ E,XNU,ALPHA
C      COMMON /CONTNT/ PI
C      COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2
C      COMMON /CRP/ PSP(25),TPLASK
C      DIMENSION AK(1), DISPA(1), NADET(1)
C      DIMENSION LKNT(26)
C      INTEGER BND(4)
C      LKNT(1) = 1
C      DO 100 I=2,26
C      LKNT(I) = LKNT(I-1)+1
100 CONTINUE
C      THIS IS THE TOTAL DEGREES OF FREEDOM
C      BND(1) = 1
C      BND(2) = NABF+1
C      BND(3) = NABF+2+1
C      BND(4) = NABF+2+NABF+1
C      DO 130 I=1,4
C      IOUT = BND(I)
C      II = NADET(IOUT)+1
C      MM = MCUD
C      IF (II.GE,KRNCH) MM = MCUD-(II-KRNCH)-1

```

SUBROUTINE ABOUND

74/74 OPT=1

FTN 4,6+452

15/02/78 17.25.31

	DO 110 KM=1,MM	ABOUND	35
	IM = (II-1)*MCUD+KM	ABOUND	36
	NM = II-KRNCH	ABOUND	37
	IF (NM.GT.0) IM = IM-LKNT(NM)	ABOUND	38
	AK(IM) = 0.0	ABOUND	39
110	CONTINUE	ABOUND	40
	MA = MCUD-1	ABOUND	41
	IF (II.LE.MA) MA = II-1	ABOUND	42
	DO 120 KM=1,MA	ABOUND	43
	IJ = II-KM	ABOUND	44
	IM = MCUD*(IJ-1)+II-IJ+1	ABOUND	45
	IF (IJ.GT.KRNCH) IM = IM-LKNT(IJ-KRNCH)	ABOUND	46
	AK(IM) = 0.0	ABOUND	47
120	CONTINUE	ABOUND	48
	IM = (II-1)*MCUD+1	ABOUND	49
	NM = II-KRNCH	ABOUND	50
	IF (NM.GT.0) IM = IM-LKNT(NM)	ABOUND	51
	AK(IM) = 1.0	ABOUND	52
	DISPA(II) = 0.0	ABOUND	53
130	CONTINUE	ABOUND	54
C		ABOUND	55
C	HAVE COMPUTED THE CORRECT BOUNDARY CONDITION	ABOUND	56
C		ABOUND	57
	RETURN	ABOUND	58
	END	ABOUND	59


```

SUBROUTINE ALQAD (AELM,ARAA,RA,NADET,RRRA,RA,TW,TWDEL,CA,THETAX,XL ALQAD      2
+QADA,YACF,TCRP,NOELD,NTAED,BRA,IACN,NABTD,ARE,RBR,TSTSR,RO,Y,NFF1 ALQAD      3
+D,NAB2D,THXSV) ALQAD      4
C ALQAD      5
C PURPOSE OF THIS SUBROUTINE IS TO DEFINE THE AXIAL AND RADIAL LOADS ALQAD      6
C FOR THE SOLUTION OF THE AXIAL PROBLEM ALQAD      7
C COMPUTE THE CORRECT CONCENTRATED NODAL LOADS FOR ALQAD      8
C A. APPLIED LOADS ALQAD      9
C B.THERMAL LOADS ALQAD     10
C C.CREEP LOADS ALQAD     11
C D.ELASTIC LOADS ALQAD     12
C ALQAD     13
C RADIAL FORCES FOR THE EXTERNAL NODES ALQAD     14
C ALQAD     15
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IPELTM,IELSTN,IELSTR PELCOM      2
+,IKSAVE,IPRESS,ITEM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD PELCOM      3
+ET, IR,IRRR, ISA,ISAC,ISAF,ISB,ISBB,ISBBC,I PELCOM      4
+SBBF,ISBC,ISBF,ISK,ISKC,ISTRES, ITDELE, IXK,IXKC,IXKT PELCOM      5
+,IXLOAD, ICC,ITHET,ICA,ITHYAX, INADET,IRA,IY,IAELM,I PELCOM      6
+ARAA,IBA,IBBA,IRBRA,IXLODA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I PELCOM      7
+PSLA,IAC,IACN,IACN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM      8
+IIDO,ITHE,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM      9
+, IXKCC,IXKTT PELCOM     10
+,ITFB, ITCB, IBBSV,IESTSV,IESTNV PELCOM     11
+,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDSE,IPSSV,IPSSRT,JIIJ,IIPDS PELCOM     12
+,IBASE,ITFR,IYM,IZ PELCOM     13
+,IDSPSV PELCOM     14
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELH,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM     15
+DEL PELCOM     16
COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IDF,NFF,MAXDC,NETOT,NE PELCOM     17
+WF,NEWT,KA,NARF,NAHC,NART,NABF,NABC,NABT,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM     18
+NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CH,FROEN,KZ1,ISSTOP,IPLAS,NKN PELCOM     19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM     20
+MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD PELCOM     21
+, KPLAS,ICNT,RRAKR,IRAK,RAKSUM,IPLND,ISP PELCOM     22
COMMON /LASTIC/ E,XNU,ALPHA PELCOM     23
COMMON /CONTNT/ PI PELCOM     24
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM     25
COMMON /CRP/ PSP(25),TPLASK PELCOM     26
DIMENSION RA(1), YACF(1), NADET(1), BBA(1), RA(1), TW(NTAED,2), AR ALQAD     17
+AA(1), RBRA(1), TWDEL(1), XLODA(6,4,1), THERM(6), AELM(3,1), IACO ALQAD     18
+N(NABTD,1) ALQAD     19
DIMENSION NNA(3), IBN(4), CA(4,4,1), THETAX(4,1) ALQAD     20
DIMENSION AVCRP(4), TCRP(4,NOELD,1) ALQAD     21
DIMENSION TSTSR(4,NOELD,1), RD(NFF1D,1), Y(NAB2D,1) ALQAD     22
DIMENSION ARE(1), RBR(1) ALQAD     23
DIMENSION THXSV(4,1) ALQAD     24
C ALQAD     25
COMMON /BA/ AA(23,7),CF(7,8),HKS,DELZA(20),NN,NCONN COMC      2
+, NCON,ISTRT,ITER COMC      3
+, FLXR COMC      4
COMMON/FEP/HHF,NT,KKZZZ,ISTOP,ROOLD(20),RRSIN(20),ICNOLD(20) COMC      5
+, IDOOLD(20) COMC      6
+,PINGLD,POUTLD,POLD,NTOLD(20) COMC      7
+, ICHNG,CRPAD(20) COMC      8
+,STEPS COMC      9
C ALQAD     27

```

	CALL MOVEKA (0,AVCRP,4)	ALOAD	28
	CALL MOVEKA (0,THETA,4*NTAE)	ALOAD	29
	DO 100 I=1,6	ALOAD	30
	THERM(I) = 0.0	ALOAD	31
100	CONTINUE	ALOAD	32
	IF (ILK.GT.0) GO TO 380	ALOAD	33
	IF (NKN.EQ.2) GO TO 120	ALOAD	34
	NDUT = 2*NABF+NABT+1	ALOAD	35
C		ALOAD	36
C	NOW COMPUTE THE AXIAL LOAD	ALOAD	37
C	NOW DIVIDE OVER ALL OF THIS AXIAL CLAD NODES	ALOAD	38
C	TOTAL NODAL LINE LENGTH	ALOAD	39
C		ALOAD	40
C	COMPUTE THE LINE LOAD	ALOAD	41
C		ALOAD	42
	SIG = POUT*PI*RCU**2-PIN*PI*RCI**2	ALOAD	43
	XLINE = PI*(2*(RCI+RCU))	ALOAD	44
	ZF = SIG/XLINE	ALOAD	45
	ZF = -ZF	ALOAD	46
	NIN = 2*NABF+NABT	ALOAD	47
	II = NADET(NIN)	ALOAD	48
C	MUST ADD 1 TO II FOR VERTICAL D.O.F.	ALOAD	49
	BBA(II+1) = BBA(II+1)+ZF*2*PI*RCI	ALOAD	50
	BA(II+1) = BBA(II+1)	ALOAD	51
	NDUT = 2*(NABF+NABT)	ALOAD	52
	II = NADET(NDUT)	ALOAD	53
	BBA(II+1) = BBA(II+1)+ZF*2*PI*RCU	ALOAD	54
	BA(II+1) = BBA(II+1)	ALOAD	55
C	HAVE NOW COMPUTED THE LOAD VECTOR COMPONENTS DUE TO EXTERNAL	ALOAD	56
C	PRESSURE NOW COMPUTE THE THERMAL EXPANSION LOADS.	ALOAD	57
C		ALOAD	58
C		ALOAD	59
C	NOW COMPUTE THE RADIAL PRESSURE LOADS	ALOAD	60
C		ALOAD	61
	NSTART = 2*NABF	ALOAD	62
	FRU = PI*2.0*RCU*POUT	ALOAD	63
	FRI = PI*2.0*RCI*PIN	ALOAD	64
	KIN = NSTART	ALOAD	65
	KOUT = NIN	ALOAD	66
	DO 110 I=1,NABT	ALOAD	67
	KIN = KIN+1	ALOAD	68
	KOUT = KOUT+1	ALOAD	69
	II = NADET(KIN)	ALOAD	70
	III = NADET(KIN+1)	ALOAD	71
	IJ = NADET(KOUT)	ALOAD	72
	IJJ = NADET(KOUT+1)	ALOAD	73
	FI = FRI*(YACF(KIN+1)-YACF(KIN))/2.0	ALOAD	74
	FU = -FRU*(YACF(KOUT+1)-YACF(KOUT))/2.0	ALOAD	75
	BBA(II) = BBA(II)+FI	ALOAD	76
	BA(II) = BBA(II)	ALOAD	77
	BBA(III) = BBA(III)+FI	ALOAD	78
	BA(III) = BBA(III)	ALOAD	79
	BBA(IJ) = BBA(IJ)+FU	ALOAD	80
	BA(IJ) = BBA(IJ)	ALOAD	81
	BBA(IJJ) = BBA(IJJ)+FU	ALOAD	82
	BA(IJJ) = BBA(IJJ)	ALOAD	83
110	CONTINUE	ALOAD	84

120	CONTINUE	ALOAD	85
	BIGT = 0.0	ALOAD	86
	DO 130 I=1,4	ALOAD	87
	BIGT = BIGT+RHRA(I)	ALOAD	88
130	CONTINUE	ALOAD	89
	DO 370 K=1,NTAE	ALOAD	90
C		ALOAD	91
	IF (NKN.EQ.2) GO TO 240	ALOAD	92
C	CALL TERP FOR ALPHA USING THE WEIGHTED TEMPERATURES TW(K)	ALOAD	93
C		ALOAD	94
	IF (K.GT.NAEF) GO TO 220	ALOAD	95
	LA = (K+3)/4	ALOAD	96
	IT = 4*LA	ALOAD	97
	BIG = 0.0	ALOAD	98
	DO 210 I=1,4	ALOAD	99
	FAC = RHRA(IT-I+1)/BIGT	ALOAD	100
	TWW = TW(IT-I+1,NEW)	ALOAD	101
	IF (NCONN.GT.0) GO TO 140	ALOAD	102
	TWC = (TWW-32.)/1.8	ALOAD	103
	ALPHA = FULEX(TWC)	ALOAD	104
	GO TO 150	ALOAD	105
140	CONTINUE	ALOAD	106
	ALPHA = TERP(TWW,CF,CF(1,6),NCONN)	ALOAD	107
150	CONTINUE	ALOAD	108
	IF (IRZ.GT.1) GO TO 180	ALOAD	109
	IF (NCONN.GT.0) GO TO 160	ALOAD	110
	THET = ALPHA*FAC	ALOAD	111
	GO TO 170	ALOAD	112
160	CONTINUE	ALOAD	113
	THET = ALPHA*TWDEL(IT-I+1)*FAC	ALOAD	114
170	CONTINUE	ALOAD	115
	BIG = BIG+THET	ALOAD	116
	GO TO 210	ALOAD	117
180	TLD = TWW-TWDEL(IT-I+1)	ALOAD	118
	ALPH1 = ALPHA	ALOAD	119
	IF (NCONN.GT.0) GO TO 190	ALOAD	120
	TLD = TWC-TWDEL(IT-I+1)/1.8	ALOAD	121
	ALPH2 = FULEX(TLD)	ALOAD	122
	THET = (ALPH1-ALPH2)*FAC	ALOAD	123
	GO TO 200	ALOAD	124
190	CONTINUE	ALOAD	125
	ALPH1 = ALPHA	ALOAD	126
	ALPH2 = TERP(TLD,CF,CF(1,6),NCONN)	ALOAD	127
	THET = (ALPH1*(TWW-TREF)-ALPH2*(TLD-TREF))*FAC	ALOAD	128
200	CONTINUE	ALOAD	129
	BIG = BIG+THET	ALOAD	130
210	CONTINUE	ALOAD	131
	THET = BIG	ALOAD	132
	GO TO 240	ALOAD	133
220	CONTINUE	ALOAD	134
	TWW = TW(K,NEW)	ALOAD	135
	ALPHA = TERP(TWW,AA,AA(1,6),NN)	ALOAD	136
	IF (IRZ.GT.1) GO TO 230	ALOAD	137
	THET = TWDEL(K)*ALPHA	ALOAD	138
	GO TO 240	ALOAD	139
230	TLD = TWW-TWDEL(K)	ALOAD	140
	ALPH1 = ALPHA	ALOAD	141

	ALPH2 = TERP(TLD,AA,AA(1,6),NN)	ALOAD	142
	THET = ALPH1*(TWW-TREF)=ALPH2*(TLD-TREF)	ALOAD	143
240	CONTINUE	ALOAD	144
	IF (NKN.EQ.2) THET = 0.0	ALOAD	145
	IF (NKN.EQ.1) GO TO 300	ALOAD	146
C		ALOAD	147
C	COMPUTE BOUNDARY LIMITS FOR AVERAGING OF FINE MESH CREEP RESULTS	ALOAD	148
C		ALOAD	149
C	COMPUTE FUEL FIRST THEN CLAD	ALOAD	150
C		ALOAD	151
	IF (K.GT.NAEF) GO TO 250	ALOAD	152
	LA = (K+3)/4	ALOAD	153
	LL1 = (NF-1)*2-3	ALOAD	154
	LL2 = (NF-1)*2	ALOAD	155
	GO TO 260	ALOAD	156
250	CONTINUE	ALOAD	157
	LA = (K+3)/4-NAEF/4	ALOAD	158
	LL1 = NOELF+1	ALOAD	159
	LL2 = NOEL	ALOAD	160
260	CONTINUE	ALOAD	161
	DO 290 I=1,4	ALOAD	162
	AVCRP(I) = 0.0	ALOAD	163
	TA = 0.0	ALOAD	164
	DO 270 JK=LL1,LL2	ALOAD	165
	TA = ARE(JK)*RBR(JK)+TA	ALOAD	166
270	CONTINUE	ALOAD	167
	DO 280 JK=LL1,LL2	ALOAD	168
	AVCRP(I) = AVCRP(I)+TCRP(I,JK,LA)*ARE(JK)*RBR(JK)/TA	ALOAD	169
280	CONTINUE	ALOAD	170
290	CONTINUE	ALOAD	171
C		ALOAD	172
300	CONTINUE	ALOAD	173
	PR = 2*PI*AAA(K)*WBRA(K)	ALOAD	174
C		ALOAD	175
	DO 330 I=1,6	ALOAD	176
	THERM(I) = 0.0	ALOAD	177
	SUM = 0.0	ALOAD	178
	DO 310 J=1,3	ALOAD	179
	S = PR*XLOADA(I,J,K)*THET	ALOAD	180
	SUM = SUM+S	ALOAD	181
310	CONTINUE	ALOAD	182
C		ALOAD	183
C	ADD CREEP STRAIN TO LOAD VECTOR	ALOAD	184
C		ALOAD	185
	SCUM = 0.0	ALOAD	186
	DO 320 J=1,4	ALOAD	187
	SC = PR*XLOADA(I,J,K)*AVCRP(J)	ALOAD	188
	SCUM = SCUM+SC	ALOAD	189
320	CONTINUE	ALOAD	190
	SUM = SCUM+SUM	ALOAD	191
	THERM(I) = SUM	ALOAD	192
330	CONTINUE	ALOAD	193
C		ALOAD	194
	DO 340 I=1,3	ALOAD	195
	THETAX(I,K) = THET	ALOAD	196
	IF (NKN.EQ.1.AND.ISSTOP.EQ.1) THXSV(I,K) = THETAX(I,K)	ALOAD	197
340	CONTINUE	ALOAD	198

C	THETAX(4,K) = 0.0	ALOAD	199
C	ADD CREEP STRAIN TO STRESS CORRECTION TERM	ALOAD	200
C		ALOAD	201
	DO 350 I=1,4	ALOAD	202
	THETAX(I,K) = THETAX(I,K)+AVCRP(I)	ALOAD	203
350	CONTINUE	ALOAD	204
C	NOW COMPUTE THE CORRECT D.O.F. AND POSITION IN THE LOAD	ALOAD	205
C		ALOAD	206
	NNA(1) = AELM(1,K)	ALOAD	207
	NNA(2) = AELM(2,K)	ALOAD	208
	NNA(3) = AELM(3,K)	ALOAD	209
C		ALOAD	210
	DO 360 I=1,3	ALOAD	211
	J = 2*I-1	ALOAD	212
	NNI = NNA(I)	ALOAD	213
	II = NADET(NNI)	ALOAD	214
	BBA(II) = BBA(II)+THERM(J)	ALOAD	215
	BA(II) = BBA(II)	ALOAD	216
	BBA(II+1) = BBA(II+1)+THERM(J+1)	ALOAD	217
	BA(II+1) = BBA(II+1)	ALOAD	218
360	CONTINUE	ALOAD	219
370	CONTINUE	ALOAD	220
C		ALOAD	221
	GO TO 460	ALOAD	222
380	CONTINUE	ALOAD	223
C	CREATE EQUILIBRIUM LOAD VECTORS BASED UPON TSTSR AND YACF	ALOAD	224
	DO 450 I=1,ILK	ALOAD	225
	KSTART = LUCK(I,1)	ALOAD	226
	KSTOP = LOCK(I,2)	ALOAD	227
	KDEL = KSTOP-KSTART+1	ALOAD	228
	SUMC = 0	ALOAD	229
	SUMF = 0	ALOAD	230
	DO 410 IM=1,KDEL	ALOAD	231
	MM = KSTART+IM-1	ALOAD	232
	IF (KSTOP.GE.NABF) GO TO 420	ALOAD	233
	IF (I.NE.ILK) GO TO 400	ALOAD	234
	MMHR = 0	ALOAD	235
	DO 390 II=KSTOP,NABF	ALOAD	236
	IF (IACON(II,NEW).EQ.1) MMHR = 1	ALOAD	237
390	CONTINUE	ALOAD	238
400	CONTINUE	ALOAD	239
	IF (MMHR.EQ.0.AND.I.EQ.ILK) GO TO 420	ALOAD	240
C	CHECK STRESS ARRAY	ALOAD	241
	SUMF = SUMF+TSTSR(3,NOELF,MM)*(Y(MM+1,1)-Y(MM,2))	ALOAD	242
	SUMC = SUMC+TSTSR(3,NOEL,MM)*(Y(MM+1,2)-Y(MM,2))	ALOAD	243
410	CONTINUE	ALOAD	244
	SIGAVC = SUMC/(Y(KSTOP+1,2)-Y(KSTART,2))	ALOAD	245
	SIGAVF = SUMF/(Y(KSTOP+1,1)-Y(KSTART,1))	ALOAD	246
	GO TO 430	ALOAD	247
420	CONTINUE	ALOAD	248
	SIGAVC = TSTSR(3,NOEL,NARC)	ALOAD	249
	SIGAVF = 0.0	ALOAD	250
430	CONTINUE	ALOAD	251
	DO 440 IM=1,KDEL	ALOAD	252
	MM = KSTART+IM-1	ALOAD	253
	NOUC = 2*NABF+NABC+MM	ALOAD	254
		ALOAD	255

	NINC = 2*NABF+MM	ALQAD	256
	NINF = MM	ALQAD	257
	NOUF = NABF+MM	ALQAD	258
	NN1 = NADET(NINF)	ALQAD	259
	NN2 = NADET(NOUF)	ALQAD	260
	NN3 = NADET(NINC)	ALQAD	261
	NN4 = NADET(NOUC)	ALQAD	262
C	FORCE CALCULATION	ALQAD	263
C	AREA FOR CLADDING	ALQAD	264
	AREAC = (RD(NFF1,MM)**2-RD(NF+1,MM)**2)*PI	ALQAD	265
	AREAF = (RD(NF,MM)**2-RD(1,MM)**2)*PI	ALQAD	266
	XLINC = (RD(NFF1,MM)**2+RD(NF+1,MM)**2)*PI	ALQAD	267
	XLINF = (RD(NF,MM)**2+RD(1,MM)**2)*PI	ALQAD	268
	XLIN1 = 2.0*RD(1,MM)*PI	ALQAD	269
	XLIN2 = 2.0*RD(NF,MM)*PI	ALQAD	270
	XLIN3 = 2.0*RD(NF+1,MM)*PI	ALQAD	271
	XLIN4 = 2.0*RD(NFF1,MM)*PI	ALQAD	272
	XF1 = (SIGAVF-TSTSR(3,NOELF,MM))*AREAF*XLIN1/XLINF	ALQAD	273
	XF2 = (SIGAVF-TSTSR(3,NOELF,MM))*AREAF*XLIN2/XLINF	ALQAD	274
	XC1 = (SIGAVC-TSTSR(3,NOEL,MM))*AREAC*XLIN3/XLINC	ALQAD	275
	XC2 = (SIGAVC-TSTSR(3,NOEL,MM))*AREAC*XLIN4/XLINC	ALQAD	276
	BBA(NN1+1) = BBA(NN1+1)-XF1	ALQAD	277
	BBA(NN2+1) = BBA(NN2+1)-XF2	ALQAD	278
	BBA(NN3+1) = BBA(NN3+1)-XC1	ALQAD	279
	BBA(NN4+1) = BBA(NN4+1)-XC2	ALQAD	280
	BA(NN1+1) = BBA(NN1+1)	ALQAD	281
	BA(NN2+1) = BBA(NN2+1)	ALQAD	282
	BA(NN3+1) = BBA(NN3+1)	ALQAD	283
	BA(NN4+1) = BBA(NN4+1)	ALQAD	284
	NINC = NINC+1	ALQAD	285
	NOUC = NOUC+1	ALQAD	286
	NINF = NINF+1	ALQAD	287
	NOUF = NOUF+1	ALQAD	288
	NN1 = NADET(NINF)	ALQAD	289
	NN2 = NADET(NOUF)	ALQAD	290
	NN3 = NADET(NINC)	ALQAD	291
	NN4 = NADET(NOUC)	ALQAD	292
	BBA(NN1+1) = BBA(NN1+1)+XF1	ALQAD	293
	BA(NN1+1) = BBA(NN1+1)	ALQAD	294
	BBA(NN2+1) = BBA(NN2+1)+XF2	ALQAD	295
	BA(NN2+1) = BBA(NN2+1)	ALQAD	296
	BBA(NN3+1) = BBA(NN3+1)+XC1	ALQAD	297
	BA(NN3+1) = BBA(NN3+1)	ALQAD	298
	BBA(NN4+1) = BBA(NN4+1)+XC2	ALQAD	299
	BA(NN4+1) = BBA(NN4+1)	ALQAD	300
440	CONTINUE	ALQAD	301
450	CONTINUE	ALQAD	302
460	CONTINUE	ALQAD	303
C	ZERO OUT BOUNDARY CONDITIONS	ALQAD	304
	IBN(1) = 1	ALQAD	305
	IBN(2) = NABF+1	ALQAD	306
	IBN(3) = NABF*2+1	ALQAD	307
	IBN(4) = NABF*2+NABT+1	ALQAD	308
	DO 470 I=1,4	ALQAD	309
	IOUT = IBN(I)	ALQAD	310
	II = NADET(IOUT)+1	ALQAD	311
	BA(II) = 0.0	ALQAD	312

SUBROUTINE ALOAD

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

      BBA(11) = 0.0
470  CONTINUE
      DO 480 I=1,NABF
      IF (IACUN(I,NEW).EQ.0) GO TO 480
      N1 = NADET(I+NABF)+1
      J = 2*NABF+I
      N2 = NADET(J)+1
      BBA(N1) = BBA(N1)+BBA(N2)
      BBA(N2) = 0.0
      BA(N1) = BBA(N1)
      BA(N2) = 0.0
480  CONTINUE
      C
      RETURN
      END

```

```

      ALOAD 313
      ALOAD 314
      ALOAD 315
      ALOAD 316
      ALOAD 317
      ALOAD 318
      ALOAD 319
      ALOAD 320
      ALOAD 321
      ALOAD 322
      ALOAD 323
      ALOAD 324
      ALOAD 325
      ALOAD 326
      ALOAD 327

```

```

SUBROUTINE ASOLVE (HA,BHA,NADET,AK,IACON,NABTO,KAN,PRESS,TW,BASAVE ASOLVE 2
+) ASOLVE 3
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IETSLM,IELSTN,IELSTR PELCOM 2
+,IXSAVE,IPRESS,ITELM,ILAST,IIMIT,INDC,INFF,INDT,INDE,INDEC,INDD PELCOM 3
+ET, IR,IRBK, ISA,ISAC,ISAF,ISH,ISHH,ISBHC,I PELCOM 4
+SBHF,ISHC,ISHF,ISK,ISAC,ISTRES, ITDEL, IXX,IXXC,IXKT PELCOM 5
+,IXLOAD, ICC,ITHET,ICA,ITHAX, INADET,IRA,IY,IAELM,I PELCOM 6
+AAA,IBA,IBHA,IBRA,IXLOAD,ITW,ITWDEL,IDISP,ILSTHA,ILSTNA,ISTRSA,I PELCOM 7
+PSLA,IAK,IACN,KAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM 8
+IIDU,ITME,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM 9
+, IXXCC,IXKTT PELCOM 10
+,ITFB, ITCB, IBBSV,TESTSV,TESTNV PELCOM 11
+,ITHSAV,ITHSV,IBASAV,IPSTS,IOLDE,IPSSV,IPSSRT,JIIJ,IIPGS PELCOM 12
+,IBASE,ITFR,IYM,IZ PELCOM 13
+,IDSPSV PELCOM 14
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM 15
+DEL PELCOM 16
COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IDF,NFF,MAXDC,NETOT,NE PELCOM 17
+WF,NEWT,KA,NARF,NARC,NART,NABF,NAHC,NABT,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM 18
+NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CR,FROEN,KZ1,ISSTOP,IPLAS,KKN PELCOM 19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM 20
+MX,KRNCH,MCDJ,IADJ,ICUT,TIMEC,TIMEF,III,STEP,LOCK(20,2),ILK,NOLD PELCOM 21
+, KPLAS,ICNT,RRAKR,IRAK,RAKSUM,IPLND,ISP PELCOM 22
COMMON /LASTIC/ E,XNU,ALPHA PELCOM 23
COMMON /CONTNT/ PI PELCOM 24
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM 25
COMMON /CRP/ PSP(25),TPLASK PELCOM 26
COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN COMC 2
+, NCON,ISTRT,ITER COMC 3
+, FLXX COMC 4
COMMON/FEP/HMF,NT,KKZZZ,ISTOP,NOLD(20),KRSIN(20),ICNOLD(20) COMC 5
+, ICNOLD(20) COMC 6
+,PINOLD,POUTLD,POLD,NTOLD(20) COMC 7
+, ICHNG,CRPAD(20) COMC 8
+,STEPS COMC 9
DIMENSION KAN(NABTO), PRESS(NABTO) ASOLVE 6
DIMENSION AK(1), IACON(20,2) ASOLVE 7
DIMENSION NADET(1), BBA(1), HA(1) ASOLVE 8
DIMENSION BASAVE(1) ASOLVE 9
COMMON D(1) ASOLVE 10
COMMON /PRSSV/ PRSINC(30),PRSS(30) ASOLVE 11
C PSEUDO=ELASTIC LOAD STEP. ASOLVE 12
IF (IRAK.NE.1) GO TO 120 ASOLVE 13
DO 100 KK=1,NTAE ASOLVE 14
IBAS = (KK-1)*4+ITHXSV=1 ASOLVE 15
IBASTH = (KK-1)*4+ITHAX=1 ASOLVE 16
DO 100 JJ=1,4 ASOLVE 17
D(IBASTH+JJ) = D(IBAS+JJ)*(1.-RRAKR) ASOLVE 18
CONTINUE ASOLVE 19
100 DO 110 J=1,NAXB4 ASOLVE 20
BA(J) = BASAVE(J)*(1.-RRAKR) ASOLVE 21
110 CONTINUE ASOLVE 22
GO TO 150 ASOLVE 23
120 CONTINUE ASOLVE 24
CALL MOVEKA (0,BA,NAXB4) ASOLVE 25
CALL MOVEKA (0,BBA,NAXB4) ASOLVE 26
CALL MOVEKA (0,D(ILSTRA),4*NTAE) ASOLVE 27

```



```

      CALL MOVEKA (0,D(ILSTNA),4*NTAE)
      IF (NKN.EQ.2) GO TO 140
C     ADD IN RADIAL LOADS FROM RADIAL MODEL EST, OF INTERFACIAL PRESSURE
      DO 130 I=1,NARF
        IF (KAN(I).EQ.0) GO TO 130
        III = I+NARF+1
        II = III-1
        INN = NADET(III)
        IN = NADET(II)
        JJ = III+NARF
        ICNN = NADET(JJ)
        J = II+NARF
        ICN = NADET(J)
        RLDD = -PRSINC(I)*RFO*PI*DELZA(I)
        BBA(INN) = -RLDD
        BBA(IN) = -RLDD
        BBA(ICNN) = RLDD
        BBA(ICN) = RLDD
        BA(IN) = -RLDD
        BA(INN) = -RLDD
        BA(ICNN) = RLDD
        BA(ICN) = RLDD
130    CONTINUE
140    CONTINUE
      CALL ALDAD (D(IAELM),D(IAAAA),D(IBA),D(INADET),D(IRBRA),D(IRA),TW,
+D(ITWDEL),D(ICA),D(ITHTAX),D(IXLUDA),D(IYACF),D(ITCRP),NOEL,NTAE,D
+D(IBBA),D(IACN),NABT,D(IAHE),D(IRBR),D(ITSTR),D(IRD),D(IY),NFF1,NA
+BT2,D(ITXSV))
150    CONTINUE
C     SAVE CONVERGED LOAD VALUES
      IF (NKN.NE.1.OR.ISSTOP.NE.1) GO TO 170
      DO 160 I=1,NAXB4
        BASAVE(I) = BA(I)
160    CONTINUE
170    CONTINUE
C     CALL SOLUTION ROUTINE (BA=LOAD ARRAY GOING IN,=DISPLACEMENT ARRAY
C     COMING OUT)
      MUD = 25
      IF (NAXB4.LT.37) MUD = NAXB4-1
      IOP = 1
      IF (ICHNG.EQ.0.AND.KZ1.GT.2.AND.IPLAS.EQ.0.AND.ICUT.EQ.0) IOP = -1
      CALL DMCHB (BA,AK,NAXB4,1,MUD,IOP,1,E=14,IER)
C     RETRIEVE DISPLACEMENTS OF FUEL NODES IN CONTACT
      DO 180 I=1,NABF
        IF (IACON(I,NEW).EQ.0) GO TO 180
        N1 = NADET(I+NABF)+1
        JJ = 2*NABF+1
        N2 = NADET(JJ)+1
        BA(N2) = BA(N1)
        BBA(N2) = BBA(N1)
180    CONTINUE
      CALL ASTRES (D(IAELM),D(IBA),D(IDISP),D(ILSTRA),D(ILSTNA),D(IPSLA)
+D(INADET),D(ISTRSA),D(ICA),D(ITHTAX),NAXB4)
      RETURN
      END

```

```

SUBROUTINE ASTRES (AELM,HA,DISPA,ELSTSA,ELSTNA,EPSLA,NADET,STRSA,C
+ A,THETAX,NAXBTD)
C
C PURPOSE OF THIS SUBROUTINE IS TO COMPUTE THE STRESSES AND
C STRAINS IN THE AXIAL FUEL MODEL FOR BOTH THE FUEL AND THE CLAD
C
COMMON /AXL/ SUMH,SUMFF
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IEPSLM,IELSTN,IELSTH
+ ,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD
+ ET,IR,IRBR,ISA,ISAC,ISAF,ISH,ISHB,ISBHL,I
+ SHBF,ISEC,ISBF,ISK,ISKC,ISTRES,ITDELE,IXK,IXKC,IXKT
+ ,IXLOAD,ICC,ITHT,ICA,ITHTAX,INADET,IRA,IY,IAELM,I
+ AAA,IRA,IBBA,IRRA,IXLDA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I
+ PSA,IAK,IACN,IKAN,IRZ,IAAX,IKD,IYACF,ITSTNH,ITSTNA,ITSTSR,ITSTSA,
+ IDU,ITME,IEFSTR,ITCRP,ICRPG,ICNATE
+ ,IXKCC,IXKIT
+ ,ITFB,ITCB,IBBSV,IESTSV,IESTNV
+ ,ITHSAV,ITHXSV,IBASAV,IPSTS,IULDSE,IPSSV,IPSSRT,JIIJ,IIPDS
+ ,IBASE,ITFR,IYM,IZ
+ ,IDSPSV
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELH,DELZ,ZDEL,KKA,PIN,POUT,P
+ DEL
COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IDF,NFF,MAXDC,NETOT,NE
+ WF,NEWT,KA,NARF,NARC,NART,NABF,NABC,NAHT,NAEC,NAEF,NTAE,JAM,NAXBT,
+ NAXB2,NAXB4,NFF1,NABF2,NAHT2,CP,CR,FRDN,KZ1,ISSTOP,IPLAS,NKN
+ ,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N
+ MX,KRNCH,MOUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD
+ ,KPLAS,ICNT,RHAKR,IRAK,RAKSUM,IPLND,ISP
COMMON /LASTIC/ E,XNU,ALPHA
COMMON /CONTNT/ PI
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2
COMMON /CRP/ PSP(25),TPLASK
DIMENSION AELM(3,1), NN(3), NADET(1), DISP(6), BA(1), EPSLA(4,6,1)
+ , STRSA(4,6,1), ELSTNA(4,1), ELSTSA(4,1)
C
C DIMENSION CA(4,4,1), THETAX(4,1), DISPA(NAXBTD)
C
DO 140 K=1,NTAE
NN(1) = AELM(1,K)
NN(2) = AELM(2,K)
NN(3) = AELM(3,K)
DO 100 I=1,3
J = I+2-1
NNI = NN(I)
IN = NADET(NNI)
DISP(J) = BA(IN)
DISP(J+1) = BA(IN+1)
100 CONTINUE
C
DO 130 I=1,4
S = 0.0
SS = 0.0
DO 110 J=1,6
C
C COMPUTE THE STRAINS
C
S = S+EPSLA(I,J,K)*DISP(J)

```

SUBROUTINE ASTRES

74/74

OPT=1

FIN 4.6+452

15/02/78

17.25.31

C		ASTRES	35
C	COMPUTE THE STRESSES	ASTRES	36
C		ASTRES	37
	SS = SS+STRSA(I,J,K)*DISP(J)	ASTRES	38
110	CONTINUE	ASTRES	39
C		ASTRES	40
C	ELEMENT STRAIN	ASTRES	41
C		ASTRES	42
	ELSTNA(I,K) = S	ASTRES	43
C	SUBTRACT OFF THERMAL STRESS	ASTRES	44
	SIGD = 0.0	ASTRES	45
	DO 120 II=1,4	ASTRES	46
	SIGD = SIGD+CA(I,II,K)*THETAX(II,K)	ASTRES	47
120	CONTINUE	ASTRES	48
	ELSTSA(I,K) = SS-SIGD	ASTRES	49
130	CONTINUE	ASTRES	50
140	CONTINUE	ASTRES	51
C		ASTRES	52
C	RETAIN AXIAL DISPLACEMENTS OF THE TOP OF THE CLAD (SUMM) AND THE	ASTRES	53
C	TOP OF THE FUEL (SUMFF)	ASTRES	54
	IF (ISSTOP.NE.1) GO TO 150	ASTRES	55
	SUMM = SUMM+BA(NAXB4)	ASTRES	56
	IWANT = NADET(NABF2)+1	ASTRES	57
	SUMFF = SUMFF+BA(IWANT)	ASTRES	58
150	CONTINUE	ASTRES	59
C	HAVE NOW COMPUTED THE STRESSES AND STRAINS INT THE	ASTRES	60
C	AXIAL MODEL	ASTRES	61
C		ASTRES	62
	RETURN	ASTRES	63
C		ASTRES	64
	END	ASTRES	65

```

SUBROUTINE BOUND (KK,XKC,XKT,NODE,NODEC,NODET,KAN,IDO,MAXFO,MAXCO, BOUND      2
+MAXTD) BOUND      3
C BOUND      4
C DIMENSION KK(MAXFO), XKC(MAXCO), XKT(MAXTD), NODE(1), NODEC(1), LK BOUND      5
+NT(6), NODET(1), KAN(1), IDO(1) BOUND      6
C BOUND      7
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IETSLM,IELSTN,IELSTR PELCOM      2
+,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD PELCOM      3
+ET, IR,IRBR, ISA,ISAC,ISAF,ISH,ISBB,ISB6C,I PELCOM      4
+SBHF,ISBC,ISBF,ISK,ISKC,ISTRES, ITDELE, IXX,IXKC,IXKT PELCOM      5
+,IXLOAD, ICC,ITHET,ICA,ITHAX, INAOET,IRA,IY,IAELM,I PELCOM      6
+ARAA,IBA,IBBA,IRBRA,IXLUDA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I PELCOM      7
+PSLA,IAK,IACN,IKAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM      8
+IIDO,ITME,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM      9
+, IXKCC,IXKTT PELCOM      10
+,ITFB, ITCB, IBBSV, IESTSV, IESTNV PELCOM      11
+,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDSE,IPSSV,IPSSRT,JIIJ,IIPDS PELCOM      12
+,IBASE,ITFH,IYM,IZ PELCOM      13
+,IDUSPSV PELCOM      14
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELW,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM      15
+DEL PELCOM      16
COMMON /LIMITS/ NF,NC,NDEL,NDELF,NDELC,ICON,IDF,NFF,MAXDC,NETOT,NE PELCOM      17
+NF,NEWT,KA,NARF,NARC,NART,NABF,NABC,NABT,NAEC,NAEF,NTAE,JAM,NAXB1, PELCOM      18
+NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CK,FRDEN,KZ1,ISSTOP,IPLAS,NKN PELCOM      19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHP,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM      20
+MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD PELCOM      21
+, KPLAS,ICNT,KRAKR,IRAK,RAKSUM,IPLEND,ISP PELCOM      22
COMMON /LATIC/ E,XNU,ALPHA PELCOM      23
COMMON /CONTNT/ PI PELCOM      24
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM      25
COMMON /CRP/ PSP(25),IPLASK PELCOM      26
C BOUND      9
C C PURPOSE OF THIS SUBROUTINE IS TO DEFINE THE BOUNDARY CONDITIONS BOUND      10
C C AND TO MODIFY THE STIFFNESS MATRIX FOR THE FUEL BOUND      11
C C THE CLAD, OR THE COMBINED SYSTEM. BOUND      12
C C BOUND      13
C C BOUND      14
C C FOR ICON EQUAL TO ZERO WE HAVE NO CONTACT SO COMPUTE SEPARATE BOUND      15
C C VALUES FOR THE FUEL AND FOR THE CLAD. BOUND      16
C C BOUND      17
C C AXIAL DISPLACEMENTS ON THE BOTTOM OF THE MODEL ARE FORCED TO BOUND      18
C C VANISH. BOUND      19
C C BOUND      20
C LKNT(1) = 1 BOUND      21
C LKNT(2) = 3 BOUND      22
C LKNT(3) = 6 BOUND      23
C LKNT(4) = 10 BOUND      24
C LKNT(5) = 15 BOUND      25
C LKNT(6) = 21 BOUND      26
C IF (ICON.EQ.0) GO TO 160 BOUND      27
C N = NF+NC BOUND      28
C DO 120 I=1,N BOUND      29
C II = NODET(I)+1 BOUND      30
C IIM1 = II-1 BOUND      31
C LOUT = 8 BOUND      32
C IF (II.GE.KRNCHT) LOUT = 7-II+KRNCHT BOUND      33
C IIM1 = 8*(II-1)+1 BOUND      34

```

	IF (II.GT.KRNCHT) IIII = IIII-LKNT(II-KRNCHT)	BOUND	35
	II = IIII	BOUND	36
	DO 100 I=1,LOUT	BOUND	37
	XKT(IIII) = 0.0	BOUND	38
	IIII = IIII+1	BOUND	39
100	CONTINUE	BOUND	40
	ISON = II-7	BOUND	41
	IF (II.LT.8) ISON = 1	BOUND	42
	DO 110 IL=ISON,IIM1	BOUND	43
	III = 8*(IL-1)+II-IL+1	BOUND	44
	IF (IL.GT.KRNCHT) III = III-LKNT(IL-KRNCHT)	BOUND	45
	XKT(III) = 0.0	BOUND	46
110	CONTINUE	BOUND	47
	XKT(IIII) = 1.0	BOUND	48
120	CONTINUE	BOUND	49
C		BOUND	50
C	HAVE COMPUTED THE CORRECT STIFFNESS MATRIX FOR FUEL CONTACTING THE	BOUND	51
C	CLAD.	BOUND	52
C		BOUND	53
	ISKIP = 0	BOUND	54
	IF (ISKIP.EQ.0) GO TO 200	BOUND	55
C		BOUND	56
C	COMPUTE SIMILAR CORRECTIONS FOR THE FUEL ARRAY XK(A,B)	BOUND	57
C		BOUND	58
	N = NF	BOUND	59
	DO 150 I=1,N	BOUND	60
	II = NODE(I)+1	BOUND	61
	IIM1 = II-1	BOUND	62
	LOUT = 8	BOUND	63
	IF (II.GE.KRNCHF) LOUT = 7-II+KRNCHF	BOUND	64
	IIII = 8*(II-1)+1	BOUND	65
	IF (II.GT.KRNCHF) IIII = IIII-LKNT(II-KRNCHF)	BOUND	66
	II = IIII	BOUND	67
	DO 130 I=1,LOUT	BOUND	68
	XK(IIII) = 0.0	BOUND	69
	IIII = IIII+1	BOUND	70
130	CONTINUE	BOUND	71
	ISON = II-7	BOUND	72
	IF (II.LT.8) ISON = 1	BOUND	73
	DO 140 IL=ISON,IIM1	BOUND	74
	III = 8*(IL-1)+II-IL+1	BOUND	75
	IF (IL.GT.KRNCHF) III = III-LKNT(IL-KRNCHF)	BOUND	76
	XK(III) = 0.0	BOUND	77
140	CONTINUE	BOUND	78
	XK(IIII) = 1.0	BOUND	79
150	CONTINUE	BOUND	80
160	CONTINUE	BOUND	81
C		BOUND	82
C	COMPUTE SIMILAR ARRAY FOR THE BOUNDARY COND. ON CLAD	BOUND	83
C		BOUND	84
	N = NC	BOUND	85
	DO 190 I=1,N	BOUND	86
	JJ = NF+I	BOUND	87
	II = NODEC(JJ)+1	BOUND	88
	IIM1 = II-1	BOUND	89
	LOUT = 8	BOUND	90
	IF (II.GE.KRNCHC) LOUT = 7-II+KRNCHC	BOUND	91

SUBROUTINE BOUND

70/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	IIII = 8*(II-1)+1	BOUND	92
	IF (II.GT.KRNCMC) IIII = IIII-LKNT(II-KRNCMC)	BOUND	93
	III = IIII	BOUND	94
	DO 170 IA=1,LOUT	BOUND	95
	XKC(IIII) = 0.0	BOUND	96
	IIII = IIII+1	BOUND	97
170	CONTINUE	BOUND	98
	ISDN = II-7	BOUND	99
	IF (II.LT.8) ISDN = 1	BOUND	100
	DO 180 IL=ISDN,II	BOUND	101
	III = 8*(IL-1)+II-IL+1	BOUND	102
	IF (IL.GT.KRNCMC) III = III-LKNT(IL-KRNCMC)	BOUND	103
	XKC(III) = 0.0	BOUND	104
180	CONTINUE	BOUND	105
	XKC(III) = 1.0	BOUND	106
190	CONTINUE	BOUND	107
C		BOUND	108
200	CONTINUE	BOUND	109
	RETURN	BOUND	110
	END	BOUND	111

```

SUBROUTINE CREEP (GSTR,EFSTR,CRP,TELM,TME,ICRP,CRPG,CRATE,NRAD,NAX CREEP      2
+L,NSH) CREEP      3
C CREEP      4
C PURPOSE OF THIS SUBROUTINE IS TO COMPUTE THE CORRECTION TO CREEP      5
C DISPLACEMENTS STRESSES AND STRAINS CAUSED BY CREEP IN THE FUEL AND CREEP      6
C CLADDING. THIS SUBROUTINE ASSUMES CONTROL OF THE SOLUTION CREEP      7
C ALGORITHM FROM PELLET AT THE START OF THE NEW TIME STEP CREEP      8
C BASE UPON THE GENERALIZED STRESSES COMPUTED IN THIS ROUTINE CREEP      9
C SUBROUTINE. RATE IS CALLED THIS SUBROUTINE COMPUTES INCREMENTED CREEP     10
C CREEP STRAINS BASED UPON STRAIN HARDENING OR TIME HARDENING CREEP     11
C OPTIONS. A SET OF EQUIVALENT CRODIAL LOADS ARE COMPUTED. CREEP     12
C DEFINE NEW VARIABLES CREEP     13
C GSTR(1,I,J)=STRESS VALUES FOR EACH ELEMENT. CREEP     14
C GSTR(2,I,J)=STRESS VALUES FOR EACH ELEMENT. CREEP     15
C GSTR(3,I,J)=STRESS VALUES FOR EACH ELEMENT. CREEP     16
C EFSTR(1,J)=EFFECTIVE STRESS I=RADIAL ELEMENT POSITION CREEP     17
C J=AXIAL ELEMENT POSITION CREEP     18
C CRP(1,I,J)=CREEP STRAIN COMPUTED FROM EQUIVALENT CREEP CREEP     19
C DISPLACEMENTS. CREEP     20
C CRP(2,I,J)=CREEP STRAIN COMPUTED FROM EQUIVALENT CREEP CREEP     21
C DISPLACEMENTS. CREEP     22
C CRP(3,I,J)=CREEP STRAIN COMPUTED FROM EQUIVALENT CREEP CREEP     23
C DISPLACEMENTS. CREEP     24
C CREEP     25
C CRATE(1,J)=CREEP STRAIN COMPUTED FROM RATE=USED TO COMPUTE THE CREEP     26
C INPUT LOAD VECTOR CREEP     27
C DEPT(1,I,J)=INCREMENTAL CREEP STRAINS TRANSFORMED TO THE CREEP     28
C R THETA AND Z COORDINATES CREEP     29
C DEPT(2,I,J)=INCREMENTAL CREEP STRAINS TRANSFORMED TO THE CREEP     30
C R THETA AND Z COORDINATES CREEP     31
C DEPT(3,I,J)=INCREMENTAL CREEP STRAINS TRANSFORMED TO THE CREEP     32
C R THETA AND Z COORDINATES CREEP     33
C DEPT(4,I,J)=INCREMENTAL CREEP STRAINS TRANSFORMED TO THE CREEP     34
C R THETA AND Z COORDINATES CREEP     35
C NSH=1 MEANS STRAIN HARDENING IS USED NSH=0 MEANS IT IS NOT CREEP     36
C AND A TIME HARDENING IS USED. CREEP     37
C CREEP     38
C COMMON // D(1) CREEP     39
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IERSLM,IELSTN,IELSTR PELCOM      2
+,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INOC,INOF,INDT,INGDE,INDEEC,INOD PELCOM      3
+ET, IR,IRBN, ISA,ISAC,ISAF,ISB,ISBH,ISBHC,I PELCOM      4
+SBHF,ISHC,ISBF,ISK,ISKC,ISTRES, ITDELE, IXX,IXKC,IXKT PELCOM      5
+,IXLOAD, ICC,ITHET,ICA,ITHTAX, INADET,IRA,IY,IAELM,I PELCOM      6
+ARAA,IBA,IBBA,IBRA,IXLODA,ITW,ITWDEL,IDISH,ILSTRA,ILSTNA,ISTRSA,I PELCOM      7
+PSLA,IAK,IACN,IKAN,INZ,IAAX,INO,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM      8
+IIDD,ITME,IEFSTR, IICRP,ICRPG,ICRATE PELCOM      9
+IIOU,ITME,IEFSTR, IICRP,ICRPG,ICRATE PELCOM      9
+, IXXCC,IXKTT PELCOM     10
+,ITFH, ITCH, IBBSV,IESTSV,IESTNV PELCOM     11
+,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDSE,IPSSV,IPSSRT,JIIJ,IIPDS PELCOM     12
+,IBASE,ITFH,IYM,I2 PELCOM     13
+,IDSPSV PELCOM     14
COMMON /GEOPAR/ RFI,RFU,RCI,RCO,TRFF,DELK,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM     15
+DEL PELCOM     16
COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICDN,IOF,NFF,MAXDC,NETOT,NE PELCOM     17
+NF,NEWT,KA,NARF,NARC,NART,NARF,NABC,NART,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM     18
+NAXB2,NAXB4,NFF1,NAHF2,NAH12, CF,CR,FRDEN,KZ1,ISSTOP,IPLAS,NKN PELCOM     19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHE,KRNCHE,KRNCHE,NFF2,NC4,NF4,MAXA,NEW,N PELCOM     20

```

```

+MX,KRNCH,MCHD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD      PELCOM      21
+, KPLAS,ICNT,KRAKR,IRAK,RAKSUM,IPLEND,ISP                              PELCOM      22
COMMON /LASIC/ E,XNU,ALPHA                                              PELCOM      23
COMMON /CONTNT/ PI                                                       PELCOM      24
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2                           PELCOM      25
COMMON /CRP/ PSP(25),TPLASK                                              PELCOM      26
C                                                                           CREEP       41
  DIMENSION TME(5)                                                        CREEP       42
  DIMENSION GSTR(4,NRAD,1), EFSTR(NRAD,1), CRP(4,NRAD,1), TELM(NRAD, CREEP       43
+NAXL,2)                                                                  CREEP       44
  DIMENSION CRATE(NRAD,1), CRPG(NRAD,1), TCRP(4,NRAD,1)                 CREEP       45
  CALL MOVEKA (0,EFSTR,NOEL*NARC)                                         CREEP       46
  CALL MOVEKA (0,CRATE,NOEL*NARC)                                         CREEP       47
  CALL MOVEKA (0,CRPG,NOEL*NARC)                                         CREEP       48
C                                                                           CREEP       49
C   COMPUTE EFFECTIVE STRESS                                              CREEP       50
C                                                                           CREEP       51
  CALL EFFECT (EFSTR,D(ITSTR),NARC,NOEL)                                  CREEP       52
C                                                                           CREEP       53
C   NAXL=AXIAL DIMENSION OF THE STRESS AND STRAIN ARRAY                 CREEP       54
C   NRAD= RADIAL DIMENSION OF THE STRESS AND STRAIN ARRAY.              CREEP       55
C                                                                           CREEP       56
C   NOW COMPUTE CREEP STRAINS                                             CREEP       57
C   COMPUTE INCREMENTAL VALUE OF GENERALIZED STRAIN INCREMENT           CREEP       58
C                                                                           CREEP       59
  CONE = CR/(CR+1.0)                                                       CREEP       60
  CTWO = CR/(CP*(CR+1.0))                                                  CREEP       61
  CTHR = 1.0/(CR+1.0)                                                      CREEP       62
  GEE = CONE*CTWO+CTWO*CTHR+CTHR*CONE                                       CREEP       63
  DO 110 J=1,NAXL                                                         CREEP       64
    DO 100 I=1,NRAD                                                       CREEP       65
      CRR = CRP(1,I,J)                                                     CREEP       66
      CRT = CRP(2,I,J)                                                     CREEP       67
      CRZ = CRP(3,I,J)                                                     CREEP       68
      ATEST = ABS(CRP(1,I,J))                                              CREEP       69
      IF (ATEST.LT.1.0E-10) GO TO 100                                     CREEP       70
      DEPE = (1.0/GEE)*(CONE*(CTWO*CRZ-CTHR*CRT)**2+CTWO*(CTHR*CRT CREEP       71
+      -CONE*CRR)**2+CTHR*(CONE*CRR-CTWO*CRZ)**2)**0.5                  CREEP       72
      CRPG(I,J) = DEPE                                                    CREEP       73
    CONTINUE                                                                CREEP       74
  CONTINUE                                                                CREEP       75
C                                                                           CREEP       76
C   HAVE COMPUTED THE GENERALIZED STRAIN INCREMENT MODIFIED FOR THE     CREEP       77
C   HILLS ANISOTROPY COEFFICIENT R AND P.                               CREEP       78
C                                                                           CREEP       79
  CALL RATE (NSH,CRATE,TME,TELM,D(IEFSTR),D(ICRPG),NAXL,NRAD,D(IPST CREEP       80
+),D(IPSSV),D(IPSSRT),D(JIIJ),D(IIPDS),D(IOLDSE),D(IBASE))              CREEP       81
C                                                                           CREEP       82
C   L=TIME INCREMENT                                                      CREEP       83
C   TIME=TIME ARRAY                                                       CREEP       84
C   TEMP(1,J)=TEMPERATURE ARRAY IN KALVIN                                CREEP       85
C                                                                           CREEP       86
C   MUST COMPUTE SIMILAR RESULTS FOR THE FUEL                           CREEP       87
C   NOW USE MODIFIED PRANDLE REURS FLOW RULES TO COMPUTE STRAIN         CREEP       88
C   INCREMENTS                                                            CREEP       89
C                                                                           CREEP       90
  DO 130 J=1,NAXL                                                         CREEP       91

```


SUBROUTINE CREEP

74/74 OPT=1

FIN 4.6+452

15/02/78 17.25.31

	DO 120 I=1,NRAD	CREEP	92
	IF (EFSTR(I,J).LT.10.) GO TO 120	CREEP	93
	A = CHATE(I,J)/EFSTR(I,J)	CREEP	94
	TCRP(1,I,J) = A*((CP+CR)/(CP*(1+CR))+GSTR(1,I,J)-CR/(CP*(CR	CREEP	95
+	+1.))*GSTR(2,I,J)=1.0/(CR+1.0)*GSTR(3,I,J))	CREEP	96
	TCRP(2,I,J) = A*(CR*(CP+1.0)/(CP*(CR+1.0))+GSTR(2,I,J)-CR/(C	CREEP	97
+	P*(CR+1.0))*GSTR(1,I,J)=CR/(CR+1.0)*GSTR(3,I,J))	CREEP	98
	TCRP(3,I,J) = A*(GSTR(3,I,J)-CR/(CR+1.0)*GSTR(2,I,J)-1.0/(CR	CREEP	99
+	+1.0)*GSTR(1,I,J))	CREEP	100
120	CONTINUE	CREEP	101
130	CONTINUE	CREEP	102
C		CREEP	103
	RETURN	CREEP	104
	END	CREEP	105

```

SUBROUTINE DDLR (TCSF,DII,DFP,DCI,DCU,PDW,Z,YY,W,FRDEN,B,DT1,NFUEL DDLR      2
+,FRPOD2,IPDW,RSIN,TSINC,FRSIN,B2,BR,C,NCLAD) DDLR      3
THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR A POWER SERIES DDLR      4
EXPANSION OF THE FUEL AND CLAD TEMPERATURES AS A FUNCTION OF DDLR      5
RADIAL POSITION. THE COEFFICIENTS RETURNED TO TEMP2 ARE AS FOLLOWS DDLR      6
C DDLR      7
C RESTRUCTURED FUEL REGION = BR(1-6) DDLR      8
C UNRESTRUCTURED FUEL REGION = B1(1-6) DDLR      9
C CLAD REGION = B2(1-4) DDLR     10
C DDLR     11
C NOTE THAT THE SERIES EXPANSION IS IN POWERS OF NORMALIZED RADIAL DDLR     12
C POSITION (NORMALIZED TO THE CLAD OUTER RADIUS) A TYPICAL TERM DDLR     13
C OF THE EXPANSION IS DDLR     14
C DDLR     15
C B1(N)*Y**(N-1) DDLR     16
C DDLR     17
C THE SUBROUTINE BRANCHES TO DIFFERENT CALCULATIONS DEPENDING DDLR     18
C ON WHETHER THE CASE IS STEADY-STATE OR TRANSIENT AND DDLR     19
C DEPENDING ON WHETHER THERE IS RESTRUCTURING OR NOT. DDLR     20
C THE METHOD OF WEIGHTED RESIDUALS IS USED TO FIND THE DDLR     21
C EXPANSION COEFFICIENTS, WITH 4TH ORDER COLLOCATION IN DDLR     22
C THE FUEL AND 2ND ORDER IN THE CLAD. DDLR     23
C DDLR     24
COMMON /BA/ AA(23,7),CF(7,8),HKS,DELZA(20),NN,NCUNN DDLR     25
+, NCNN,ISTRT,ITER DDLR     26
+, FLXH DDLR     27
COMMON/FEF/HMF,NT,KKZZZ,ISTOP,ROOLD(20),RRSIN(20),ICNOLD(20) DDLR     28
+, IDOOLD(20) DDLR     29
+, PINOLD,POUTLD,POLD,NTOLD(20) DDLR     30
+, ICHNG,CRPAD(20) DDLR     31
+, STEPS DDLR     32
COMMON /FAD/ FF(5),NFLXN,FFT(5) DDLR     33
COMMON /RSTR/DDZZ,KKNT,HMGG DDLR     34
COMMON /FP/ D,BETA,E,G,H DDLR     35
COMMON /RSTR/ RKCV(20),RKSV(20),RKCSV(20),BBB(10,20),BRULD(16,20) DDLR     36
+, BTOLD(10,20),TRULD(16,20),ARSV(6,6,20),VOID(20),KOUNT(20),DENCUS, DDLR     37
+, DENCUS DDLR     38
COMMON/FRANK/ ARR(6,6),AR(6,6),RT(6),RC(10,6),Q(16),T(16),A(10,10) DDLR     39
+, AS(10,10),TTRM(12),RTC(4),ATS(16,16),AT(16,16),BT(16),RCT(16,6) DDLR     40
+, A1(6,6) DDLR     41
EQUIVALENCE (A8(1),AA(1,1)) DDLR     42
EQUIVALENCE (A2(1),HMF) DDLR     43
EQUIVALENCE (A3(1),FF(1)) DDLR     44
EQUIVALENCE (A4(1),DDZZ) DDLR     45
EQUIVALENCE (A5(1),D) DDLR     46
EQUIVALENCE (A6(1),RKCV(1)) DDLR     47
EQUIVALENCE (A7(1),ARR(1,1)) DDLR     48
DIMENSION A8(244),A2(89), A3(11), A4(3), A5(5), A6(1862), A7(1046) DDLR     49
DIMENSION B(10),B2(4),BR(6) DDLR     50
C DDLR     51
C DEFINE CONSTANTS. RT AND RTC DETERMINE THE COLLOCATION PTS. IN THE DDLR
C DATA A7/1046*0/ DDLR
C FUEL AND CLAD RESPECTIVELY DDLR
C DDLR
C DATA (RT(I),I=1,6)/0.0,0.0694318,0.33000948,0.66999052,0.9305682,1 DDLR
+,0/ DDLR
C DATA (RTC(I),I=1,4)/0.0,0.2113248,0.78867,1.0/ DDLR
C RTIC = DII*1.27 DDLR

```

D-29

	R11 = D11/2.	DDLR	52
	KKC = RKLX(IPOW)	DDLR	53
	KKCS = RKCSV(IPOW)	DDLR	54
	RSINC = RSIN*2.54	DDLR	55
	DF = OFF	DDLR	56
	D1 = D11	DDLR	57
	IF (D11.LT.0.001) D11 = 0.001	DDLR	58
	KK = DF*1.27	DDLR	59
	KDD = DCU*1.27	DDLR	60
	RDD2 = RDD**2	DDLR	61
	RDD4 = RDD**4	DDLR	62
	TFS = BTOLD(7,IPOW)+DT1/1.8	DDLR	63
	IF (NT.EQ.1) VOID(IPOW) = R11C	DDLR	64
	TCS = (TCSF-32.)/1.8	DDLR	65
	MF = MMGG/1761.5	DDLR	66
	MW = MMF/1761.5	DDLR	67
	TCUOL = TCS-4.111*POW/(DCU*MW)	DDLR	68
	STEP = STEPS	DDLR	69
	RVOIDC = VOID(IPOW)	DDLR	70
	DENOM = (Z/2.)*(RR**2-RVOIDC**2)+(YY/4.)*(RR**4-RVOIDC**4)+(W/6.)*	DDLR	71
	*(RR**6-RVOIDC**6)	DDLR	72
	RN = POW*5.22155*RDD**2/DENOM	DDLR	73
	TIN = TCS+1248.	DDLR	74
C	THIS LOOP DEFINES THE INITIAL TEMPERATURE AND THE	DDLR	75
C	VOLUMETRIC HEAT GENERATION AT EACH COLLOCATION PT.(KC(I))	DDLR	76
C		DDLR	77
	DO 120 K=1,10	DDLR	78
	IF (K.GT.6) GO TO 100	DDLR	79
	TIN = TIN-200.0	DDLR	80
	T(K) = TIN	DDLR	81
	IF (NT.GT.1) T(K) = BTOLD(K,IPOW)	DDLR	82
	RC(K,1) = OI/DCU+RT(K)*((DF-D1)/DCU)	DDLR	83
	RM = RC(K,1)*RDD	DDLR	84
	Q(K) = RN*(Z+YY*RM**2+W*RM**4)	DDLR	85
	GO TO 110	DDLR	86
100	TIN = TIN-12.	DDLR	87
	T(K) = TIN	DDLR	88
	IF (NT.GT.1) T(K) = BTOLD(K,IPOW)	DDLR	89
	RC(K,1) = (1.0-DCI/DCU)*RTC(K-6)+DCI/DCU	DDLR	90
110	DO 120 J=2,6	DDLR	91
	RC(K,J) = RC(K,1)*RC(K,J-1)	DDLR	92
120	CONTINUE	DDLR	93
C		DDLR	94
C	BEGIN UNRESTRICTED STEADY STATE CALCULATION	DDLR	95
C	THE FOLLOWING 4 LOOPS SET UP THE B.C. AND COLLOCATION	DDLR	96
C	EQUATIONS (LEFT-HAND SIDES) FOR FUEL AND CLAD	DDLR	97
C	FIRST THE BOUNDARY CONDITIONS FOR FUEL	DDLR	98
C		DDLR	99
	AS(1,1) = 0.0	DDLR	100
	AS(6,1) = 1.0	DDLR	101
	AS(1,2) = 1.0	DDLR	102
	AS(6,2) = RC(6,1)	DDLR	103
	DO 130 J=3,6	DDLR	104
	AS(1,J) = RC(1,J-2)*(J-1)	DDLR	105
	AS(6,J) = RC(6,J-1)	DDLR	106
130	CONTINUE	DDLR	107
	DO 140 I=2,5	DDLR	108

```

      AS(I,1) = 0.0
      AS(I,2) = 1.0/RC(I,1)
C
C   COLLOCATION EQUATIONS FOR FUEL
C
      AS(1,3) = 4.0
      AS(1,4) = 9.0*RC(1,1)
      AS(1,5) = 16.0*RC(1,2)
      AS(1,6) = 25.0*RC(1,3)
140  CONTINUE
C
C   BOUNDARY CONDITIONS FOR THE CLAD
C
      AS(6,7) = -1.0
      AS(7,7) = 0.0
      AS(10,7) = 1.0
      AS(6,8) = -RC(7,1)
      AS(7,8) = 1.0
      AS(10,8) = 1.0
      DO 150 J=9,10
        AS(6,J) = -RC(7,J-7)
        AS(7,J) = (J-7)*RC(7,J-8)
        AS(10,J) = 1.0
150  CONTINUE
C
C   COLLOCATION FOR THE CLAD
C
      DO 160 I=8,9
        AS(I,7) = 0.0
        AS(I,8) = 1.0/RC(I,1)
        AS(I,9) = 4.0
        AS(I,10) = 9.0*RC(I,1)
160  CONTINUE
C
C   BRANCH TO TRANSIENT CALCULATIONS OTHERWISE, SET UP TEMP.
C   DERIVATIVES AT THE COLL. PTS. AND BEGIN ITERATION FOR SS
C   TEMP. CONVERGENCE
C   IF (NT.GT.2.AND.C.LT.0.00347) GO TO 620
      DO 170 I=2,6
        TTRM(I) = (Z/2.)*(RC(I,2)-RC(1,2))+(YY/4.)*(RC(I,4)-RC(1,4))*RD
        + D2+(W/6.)*(RC(I,6)-RC(1,6))*RDD4
170  CONTINUE
      DO 180 N=1,20
        DO 190 J=1,10
          DO 180 I=1,10
            A(I,J) = AS(I,J)
180  CONTINUE
            IF (ITER.EQ.1.AND.KKZZZ.EQ.2) GO TO 190
            IF (J.GT.6) GO TO 190
            A(6,J) = 1.0
            IF (J.GT.1) A(6,J) = RC(6,J-1)
190  CONTINUE
C
C   THIS LOOP SETS UP THE RT. HAND SIDES OF THE COLLOCATION AND
C   BOUNDARY EQUATIONS, B(K)
      DO 250 K=1,10
        IF (K.GT.6) GO TO 240

```

```

      DDLR 109
      DDLR 110
      DDLR 111
      DDLR 112
      DDLR 113
      DDLR 114
      DDLR 115
      DDLR 116
      DDLR 117
      DDLR 118
      DDLR 119
      DDLR 120
      DDLR 121
      DDLR 122
      DDLR 123
      DDLR 124
      DDLR 125
      DDLR 126
      DDLR 127
      DDLR 128
      DDLR 129
      DDLR 130
      DDLR 131
      DDLR 132
      DDLR 133
      DDLR 134
      DDLR 135
      DDLR 136
      DDLR 137
      DDLR 138
      DDLR 139
      DDLR 140
      DDLR 141
      DDLR 142
      DDLR 143
      DDLR 144
      DDLR 145
      DDLR 146
      DDLR 147
      DDLR 148
      DDLR 149
      DDLR 150
      DDLR 151
      DDLR 152
      DDLR 153
      DDLR 154
      DDLR 155
      DDLR 156
      DDLR 157
      DDLR 158
      DDLR 159
      DDLR 160
      DDLR 161
      DDLR 162
      DDLR 163
      DDLR 164
      DDLR 165

```

```

      TK = T(K)+273.15
      F = FNDEN
      P = 1.-F
C
C      BRANCH TO PROPER FUEL CONDUCTIVITY CALCULATION
C
      IF (NFUEL) 200,210,220
200      CONTINUE
      CALL TCW1 (F,FRPUO2,TK,RK,DKDT,DENCH)
      DENCUS = DENCH
      GO TO 230
210      CONTINUE
      CALL TCON (F,TK,RK,DKDT,DENCH)
      DENCUS = DENCH
      GO TO 230
220      CONTINUE
      PM = D*((1.-P)/(1.+BETA*P))
      RK = PM*(E/(G+TK)+H*TK**3.)
      DKDT = PM*(-E/(G+TK)**2.+3.*H*TK**2.)
      DENCUS = PM
230      CONTINUE
      IF (K.EQ.6) RKS = RK
      FAC = RN/(RC(K,1)*RK)
      TFACT = FAC*TTRM(K)
      TFACT = TFACT**2
      B(K) = -(Q(K)+DKDT*TFACT)/RK
      IF (K.EQ.1) B(1) = 0.0
      IF (K.EQ.6) B(6) = DT1/1.8
      GO TO 250
240      CONTINUE
      IF (ITER.GT.1.OR.KKZZZ.GT.2) GO TO 250
      TC = T(K)+1.8+32.0
      TCEST = TC+1.0
      RK = TERP(TC,AA,AA(1,2),NN)
      RKEST = TERP(TCEST,AA,AA(1,2),NN)
      DKDT = (RKEST-RK)*1.8/57.803
      RK = RK/57.803
      IF (K.EQ.7) RKC = RK
      IF (K.EQ.10) RKCS = RK
      TFACT = (-RC(7,1)/RC(K,1))*(RDD/RK)*(4.111*PUW/DCI)
      TFACT = TFACT**2
      B(K) = -DKDT*TFACT/RK
      IF (K.EQ.7) B(7) = -RDD*4.111*PUW/(DCI*RK)
      IF (K.EQ.10) B(10) = TCS
250      CONTINUE
      IF (ITER.GT.1.OR.KKZZZ.GT.2) GO TO 260
      CALL SIMQ (A,B,10)
      GO TO 290
260      B(6) = TFS
      DO 280 IJ=1,6
      DO 270 JK=1,6
      A1(JK,IJ) = A(JK,IJ)
270      CONTINUE
280      CONTINUE
      CALL SIMQ (A1,B,6)
290      CONTINUE
      SUM = 0.0

```

```

DDLR 166
DDLR 167
DDLR 168
DDLR 169
DDLR 170
DDLR 171
DDLR 172
DDLR 173
DDLR 174
DDLR 175
DDLR 176
DDLR 177
DDLR 178
DDLR 179
DDLR 180
DDLR 181
DDLR 182
DDLR 183
DDLR 184
DDLR 185
DDLR 186
DDLR 187
DDLR 188
DDLR 189
DDLR 190
DDLR 191
DDLR 192
DDLR 193
DDLR 194
DDLR 195
DDLR 196
DDLR 197
DDLR 198
DDLR 199
DDLR 200
DDLR 201
DDLR 202
DDLR 203
DDLR 204
DDLR 205
DDLR 206
DDLR 207
DDLR 208
DDLR 209
DDLR 210
DDLR 211
DDLR 212
DDLR 213
DDLR 214
DDLR 215
DDLR 216
DDLR 217
DDLR 218
DDLR 219
DDLR 220
DDLR 221
DDLR 222

```

	DO 320 I=1,10	DDLR	223
	IF (I.GT.6) GO TO 300	DDLR	224
	TT = B(1)+B(2)*RC(I,1)+B(3)*RC(I,2)+B(4)*RC(I,3)+B(5)*RC(I,4	DDLR	225
	+B(6)*RC(I,5)	DDLR	226
	GO TO 310	DDLR	227
300	TT = B(7)+B(8)*RC(I,1)+B(9)*RC(I,2)+B(10)*RC(I,3)	DDLR	228
	IF (ITER.GT.1.OR.KKZZZ.GT.2) GO TO 320	DDLR	229
	BZ(I+6) = B(I)	DDLR	230
	TROLD(I+6,IPOW) = TT	DDLR	231
310	ADEL = ABS(T(I)-TT)	DDLR	232
	SUM = SUM+ADEL	DDLR	233
	T(I) = TT	DDLR	234
	BBB(I,IPOW) = B(I)	DDLR	235
	BRULD(I+6,IPOW) = B(I)	DDLR	236
	BTOLD(I,IPOW) = T(I)	DDLR	237
320	CONTINUE	DDLR	238
	HKSV(IPOW) = HK8	DDLR	239
	RKGSV(IPOW) = RKCS	DDLR	240
	RKGV(IPOW) = RKC	DDLR	241
	IF (SUM.LT.1.0) GO TO 340	DDLR	242
330	CONTINUE	DDLR	243
340	CONTINUE	DDLR	244
	KOUNT(IPOW) = 0	DDLR	245
	IF (T(I).LT.TSINC.AND.RBIN.LE.KII) GO TO 1190	DDLR	246
	IF (NT.EQ.1) GO TO 1190	DDLR	247
	IF (KKNT.EQ.0.AND.T(I).GT.TSINC) GO TO 440	DDLR	248
C		DDLR	249
C	BEGIN CALCULATION OF COEFFICIENTS FOR RESTRICTURED	DDLR	250
C	REGION. SAME PATTERN AS FOR UNRESTRICTURED FUEL.	DDLR	251
C	NOTE POWER DENSITY IS INCREASED.	DDLR	252
C		DDLR	253
	REST = RIIC/RDD	DDLR	254
	DO 360 I=1,10	DDLR	255
	TT = 0.0	DDLR	256
	RST = 1.0	DDLR	257
	DO 350 J=1,5	DDLR	258
	RST = RST*REST	DDLR	259
	TT = B(J+1)*RST+TT	DDLR	260
350	CONTINUE	DDLR	261
	TT = TT+B(1)	DDLR	262
	IF (TT.LT.TSINC) GO TO 370	DDLR	263
	REST = REST+0.2	DDLR	264
360	CONTINUE	DDLR	265
370	REST = REST-0.2	DDLR	266
	DO 390 I=1,10	DDLR	267
	TT = 0.0	DDLR	268
	RST = 1.0	DDLR	269
	DO 380 J=1,5	DDLR	270
	RST = RST*REST	DDLR	271
	TT = B(J+1)*RST+TT	DDLR	272
380	CONTINUE	DDLR	273
	TT = TT+B(1)	DDLR	274
	IF (TT.LT.TSINC) GO TO 400	DDLR	275
	REST = REST+0.02	DDLR	276
390	CONTINUE	DDLR	277
400	REST = REST-0.02	DDLR	278
	DO 420 I=1,10	DDLR	279

	TT = 0.0	DDL	280
	RST = 1.0	DDL	281
	DO 410 J=1,5	DDL	282
	RST = RST*RST	DDL	283
	TT = B(J+1)*RST+TT	DDL	284
410	CONTINUE	DDL	285
	TT = TT+B(1)	DDL	286
	IF (TT.LT.TSINC) GO TO 430	DDL	287
	REST = REST+0.002	DDL	288
420	CONTINUE	DDL	289
430	RSINC = (REST-0.001)*RDD	DDL	290
	RSIN = RSINC/2.54	DDL	291
	IF (RSIN.LT.RRSIN(IPOW)) RSIN = RRSIN(IPOW)	DDL	292
440	CONTINUE	DDL	293
	IF (RSIN.LE.RII) GO TO 1190	DDL	294
	NTOLD(IPOW) = NT	DDL	295
	RRSIN(IPOW) = RSIN	DDL	296
	YR = RSIN*2.54/RDD	DDL	297
	THSINC = B(1)+B(2)+YR+B(3)+YR**2+B(4)+YR**3+B(5)+YR**4+B(6)+YR**5	DDL	298
	DF = RSIN*2.	DDL	299
	DI = DII	DDL	300
	DO 470 K=1,6	DDL	301
	RC(K,1) = DI/DCB+RT(K)*((DF-DI)/DCB)	DDL	302
	RM = RC(K,1)*RDD	DDL	303
	Q(K) = RM*(Z+YY*RM**2+M*RM**4)	DDL	304
	Q(K) = Q(K)*(RSINC**2-RVOIDC**2)/(RSINC**2-RIIC**2)	DDL	305
	DO 450 JJ=2,6	DDL	306
	RC(K,JJ) = RC(K,1)**JJ	DDL	307
450	CONTINUE	DDL	308
	SUM = 0.0	DDL	309
	DO 460 IJ=2,6	DDL	310
	SUM = SUM+BBB(IJ,IPOW)*RC(K,IJ-1)	DDL	311
460	CONTINUE	DDL	312
	T(K) = SUM+BBB(1,IPOW)	DDL	313
470	CONTINUE	DDL	314
	DO 480 J=1,6	DDL	315
	AR(1,J) = RC(1,1)**(J-2)*(J-1)	DDL	316
	AR(6,J) = RC(6,1)**(J-1)	DDL	317
	ARSV(1,J,IPOW) = AR(1,J)	DDL	318
	ARSV(6,J,IPOW) = AR(6,J)	DDL	319
480	CONTINUE	DDL	320
	DO 490 I=2,5	DDL	321
	AR(I,1) = 0.0	DDL	322
	AR(I,2) = 1.0/RC(1,1)	DDL	323
	AR(I,3) = 4.0	DDL	324
	AR(I,4) = 9.0*RC(1,1)	DDL	325
	AR(I,5) = 16.0*RC(1,2)	DDL	326
	AR(I,6) = 25.0*RC(1,3)	DDL	327
490	CONTINUE	DDL	328
	DO 510 JJ=1,6	DDL	329
	DO 500 II=2,5	DDL	330
	ARSV(II,JJ,IPOW) = AR(II,JJ)	DDL	331
500	CONTINUE	DDL	332
510	CONTINUE	DDL	333
	DO 520 I=2,6	DDL	334
	TTRM(I) = (Z/2.)*(RC(1,2)-RC(1,2))+(YY/4.)*(RC(1,4)-RC(1,4))*RD	DDL	335
	+ D2*(W/6.)*(RC(1,6)-RC(1,6))*RDD4	DDL	336

520	CONTINUE	DDL	357
	DO 610 N=1,20	DDL	358
	DO 540 JJ=1,6	DDL	359
	DO 530 KK=1,6	DDL	360
	ARR(KK,JJ) = AR(KK,JJ)	DDL	361
530	CONTINUE	DDL	362
540	CONTINUE	DDL	363
	DO 590 K=2,5	DDL	364
	TK = T(K)+273.15	DDL	365
	F = FRSIN	DDL	366
	P = 1.-F	DDL	367
	IF (NFUEL) 550,560,570	DDL	368
550	CONTINUE	DDL	369
	CALL TCW1 (F,FRPUD2,TK,KK,DKDT,DENCR)	DDL	370
	DENCRS = DENCR	DDL	371
	GO TO 580	DDL	372
560	CONTINUE	DDL	373
	CALL TCUR (F,TK,KK,DKDT,DENCR)	DDL	374
	DENCRS = DENCR	DDL	375
	GO TO 580	DDL	376
570	CONTINUE	DDL	377
	PM = D*((1.-P)/(1.+BETA*P))	DDL	378
	RK = PM*(E/(G+TK)+H*TK**3.)	DDL	379
	DKDT = PM*(-E/(G+TK)**2.+3.*H*TK**2.)	DDL	380
	DENCRS = PM	DDL	381
580	CONTINUE	DDL	382
	FAC = HN/(RC(K,1)*RK)	DDL	383
	TFAC = FAC*TTRM(K)	DDL	384
	TFAC2 = TFAC**2	DDL	385
	BR(K) = -(Q(K)+DKDT*TFAC2)/RK	DDL	386
590	CONTINUE	DDL	387
	BR(1) = 0.0	DDL	388
	BR(6) = TRSINC	DDL	389
	CALL SIMQ (ARR,BR,6)	DDL	390
	SUM = 0.0	DDL	391
	DO 600 I=1,6	DDL	392
	TT = BR(1)+BR(2)*RC(I,1)+BR(3)*RC(I,2)+BR(4)*RC(I,3)+BR(5)*RC(I,4)+BR(6)*RC(I,5)	DDL	393
	ADEL = ABS(T(I)-TT)	DDL	394
	SUM = SUM+ADEL	DDL	395
	T(I) = TT	DDL	396
	BTOLD(I,IPOW) = BR(I)	DDL	397
	TOLD(I,IPOW) = TT	DDL	398
600	CONTINUE	DDL	399
	IF (SUM.LT.10.0) GO TO 620	DDL	400
610	CONTINUE	DDL	401
C		DDL	402
C	END OF STEADY STATE CALCULATIONS.	DDL	403
C	BRANCH TO TRANSIENT WITH RESTRUCTURE OTHERWISE DO	DDL	404
C	TRANSIENT WITHOUT RESTRUCTURE	DDL	405
C		DDL	406
620	CONTINUE	DDL	407
	IF (C.GT.0.00347,OR,NT.LT.3) GO TO 1190	DDL	408
	IF (RSIN.GT,RII) GO TO 830	DDL	409
	IF (BTOLD(1,IPOW),GT,TSINC,AND,FRSIN.GT,FRDEN) GO TO 830	DDL	410
C		DDL	411
C	REDEFINE BOUNDARY CONDITIONS FOR TRANSIENT CASE	DDL	412

C		DDLR	394
	DO 630 I=1,6	DDLR	395
	AS(6,I) = HF*RC(6,I)**(I-1)+(RKSV(IPUW)/RDO)*(I-1)*RC(6,I)**(I-	DDLR	396
	2)	DDLR	397
	AS(7,I) = -(RKSV(IPUW)/RKCV(IPUW))*RC(6,I)/RC(7,I)**(I-1)*RC(6	DDLR	398
	,I)**(I-2)	DDLR	399
630	CONTINUE	DDLR	400
	DO 640 I=7,10	DDLR	401
	AS(6,I) = -HF*RC(7,I)**(I-7)	DDLR	402
	AS(7,I) = (I-7)*RC(7,I)**(I-8)	DDLR	403
	AS(10,I) = HW*(RKCSV(IPUW)/RDO)*(I-7)	DDLR	404
640	CONTINUE	DDLR	405
	DO 790 K=1,10	DDLR	406
	A(1,K) = AS(1,K)	DDLR	407
	A(6,K) = AS(6,K)	DDLR	408
	A(7,K) = AS(7,K)	DDLR	409
	A(10,K) = AS(10,K)	DDLR	410
	IF (K.EQ.1) GO TO 770	DDLR	411
	IF (K.GT.6) GO TO 700	DDLR	412
	TK = BTOLD(K,IPUW)+273.	DDLR	413
	F = FRDEN	DDLR	414
	IF (NFUEL) 650,660,670	DDLR	415
650	CONTINUE	DDLR	416
	CALL TCW1 (F,FRPU02,TK,RK,DKDT,DENCR)	DDLR	417
	GO TO 660	DDLR	418
660	CONTINUE	DDLR	419
	CALL TCR (F,TK,RK,DKDT,DENCR)	DDLR	420
	GO TO 680	DDLR	421
670	CONTINUE	DDLR	422
	P = 1.-F	DDLR	423
	PM = D*((1.-P)/(1.+BETA*P))	DDLR	424
	RK = PM*(E/(G+TK)+H*TK**3.)	DDLR	425
	DKDT = PM*(-E/(G+TK)**2.+3.*H*TK**2.)	DDLR	426
680	CONTINUE	DDLR	427
	IF (K.EQ.6) RKSV(IPUW) = RK	DDLR	428
	IF (K.EQ.6) GO TO 770	DDLR	429
	SUM = 0.0	DDLR	430
	DO 690 M=2,6	DDLR	431
	TRM = (M-1)*RC(K,I)**(M-2)*BBB(M,IPUW)	DDLR	432
	SUM = SUM+TRM	DDLR	433
690	CONTINUE	DDLR	434
	DENC = FRDEN	DDLR	435
	CALL HCAP (BTOLD(K,IPUW),CDT,CT)	DDLR	436
	ALF = STEP*RK/(DENC*CT*DCO**2*17.68)	DDLR	437
	GO TO 720	DDLR	438
700	DENC = 6.57	DDLR	439
	IF (NCLAD.LT.0) DENC = 7.85	DDLR	440
	TC = BTOLD(K,IPUW)*1.6+32.	DDLR	441
	TCEST = TC+1.0	DDLR	442
	RK = TERP(TC,AA,AA(1,2),NN)	DDLR	443
	RKEST = TERP(TCEST,AA,AA(1,2),NN)	DDLR	444
	DKDT = (RKEST-RK)*1.8/57.803	DDLR	445
	RK = RK/57.803	DDLR	446
	IF (K.EQ.7) RKCV(IPUW) = RK	DDLR	447
	IF (K.EQ.10) RKCSV(IPUW) = RK	DDLR	448
	IF (K.EQ.7) GO TO 770	DDLR	449
	IF (K.EQ.10) GO TO 780	DDLR	450

```

      SUM = 0.0
      DO 710 M=8,10
      TRM = BBB(M,IPDW)*(M-7)*RC(K,1)*(M-8)
      SUM = SUM+TRM
710   CONTINUE
      CT = 5.93E-5*BTULD(K,IPDW)+.2916
      IF (NCLAD.LT.0) CT = 0.5341
      ALF = STEP*KK/(UENC*CT*DCO**2*1.613)
720   CONTINUE
      MG = 1
      IF (K.GT.6) MG = 7
      DO 740 M=1,10
      IF (K.LT.6.AND.M.GT.6) GO TO 730
      IF (K.GT.6.AND.M.LE.6) GO TO 730
      A(K,M) = AS(K,M)-RC(K,1)*(M-MG)/ALF
      GO TO 740
730   A(K,M) = 0.0
740   CONTINUE
      TFACT = SUM**2
      IF (K.GT.6) GO TO 750
      BT(K) = -(Q(K)+DKDT*TFACT)/RK-BTULD(K,IPDW)/ALF
      GO TO 760
750   BT(K) = -DKDT*TFACT/RK-BTULD(K,IPDW)/ALF
760   CONTINUE
      GO TO 790
770   BT(K) = 0.0
      GO TO 790
780   BT(K) = HW*TCUOL
790   CONTINUE
      CALL SIMO (A,BT,10)
      DO 820 I=1,10
      IF (I.GT.6) GO TO 800
      BTULD(I,IPDW) = BT(1)+BT(2)*RC(I,1)+BT(3)*RC(I,2)+BT(4)*RC(I,3)
      + BT(5)*RC(I,4)+BT(6)*RC(I,5)
      B(I) = BT(I)
      GO TO 810
800   CONTINUE
      BTULD(I,IPDW) = BT(7)+BT(8)*RC(I,1)+BT(9)*RC(I,2)+BT(10)*RC(I,3)
      + BT(11)*RC(I,4)+BT(12)*RC(I,5)
      B2(I-6) = BT(1)
810   CONTINUE
      BBB(I,IPDW) = BT(1)
820   CONTINUE
      GO TO 1190
C
C   BEGIN TRANSIENT CALCULATION WITH RESTRICTION (A 16X16 PROBLEM)
C
830   CONTINUE
      RSINC = 2.54*RSIN
      DO 860 I=1,16
      IF (I.GT.12) GO TO 850
      IF (I.GT.6) GO TO 840
      RCT(I,1) = DI/DCO+(RSINC-RIIC)*RT(I)/RDO
      RM = RCT(I,1)*RDO
      Q(I) = RN*(Z+YY*RM**2+**RM**4)
      Q(I) = Q(I)*(RSINC**2-RVOIDC**2)/(RSINC**2-RIIC**2)
      GO TO 860

```

```

DDLR 451
DDLR 452
DDLR 453
DDLR 454
DDLR 455
DDLR 456
DDLR 457
DDLR 458
DDLR 459
DDLR 460
DDLR 461
DDLR 462
DDLR 463
DDLR 464
DDLR 465
DDLR 466
DDLR 467
DDLR 468
DDLR 469
DDLR 470
DDLR 471
DDLR 472
DDLR 473
DDLR 474
DDLR 475
DDLR 476
DDLR 477
DDLR 478
DDLR 479
DDLR 480
DDLR 481
DDLR 482
DDLR 483
DDLR 484
DDLR 485
DDLR 486
DDLR 487
DDLR 488
DDLR 489
DDLR 490
DDLR 491
DDLR 492
DDLR 493
DDLR 494
DDLR 495
DDLR 496
DDLR 497
DDLR 498
DDLR 499
DDLR 500
DDLR 501
DDLR 502
DDLR 503
DDLR 504
DDLR 505
DDLR 506
DDLR 507

```

```

840      RCT(I,1) = HSINC/RDD+(RR-HSINC)*RT(I=0)/RDD      DDLR      508
      RM = RCT(I,1)*RDD      DDLR      509
      Q(I) = KN*(Z+YY*RM**2+X*RM**4)      DDLR      510
C      DDLR      511
      GO TO 860      DDLR      512
850      RCT(I,1) = RC(I=0,1)      DDLR      513
860      DO 870 JJ=1,6      DDLR      514
          RCT(I,JJ) = RCT(I,1)**JJ      DDLR      515
870      CONTINUE      DDLR      516
880      CONTINUE      DDLR      517
C      DDLR      518
C      CALCULATE TEMPERATURES AT THE COLL. PTS. OF THE UNRESTRICTED FUE      DDLR      519
C      DDLR      520
      KOUNT(IPOW) = KOUNT(IPOW)+1      DDLR      521
      IF (KOUNT(IPOW).GT.1) GO TO 900      DDLR      522
      DO 890 I=7,12      DDLR      523
          TROLD(I,IPOW) = BBB(1,IPOW)+BBB(2,IPOW)*RCT(I,1)+BBB(3,IPOW)*RC      DDLR      524
      + T(I,2)+BBB(4,IPOW)*RCT(I,3)+BBB(5,IPOW)*RCT(I,4)+BBB(6,IPOW)*RC      DDLR      525
      + T(I,5)      DDLR      526
890      CONTINUE      DDLR      527
900      CONTINUE      DDLR      528
C      DDLR      529
C      CALCULATE ALL THE BOUNDARY CONDITIONS      DDLR      530
      DO 930 K=1,6      DDLR      531
          ATS(1,K) = ARSV(1,K,IPOW)      DDLR      532
          ATS(6,K) = ARSV(6,K,IPOW)      DDLR      533
          ATS(6,K+6) = -RCT(7,1)**(K-1)      DDLR      534
          ATS(7,K) = -(DENCGRS/DENCUS)*(K-1)*RCT(6,1)**(K-2)      DDLR      535
          ATS(7,K+6) = (K-1)*RCT(7,1)**(K-2)      DDLR      536
          ATS(12,K) = 0.0      DDLR      537
      + ATS(12,K+6) = HF*RCT(12,1)**(K-1)+(RKSV(IPOW)/RDD)*(K-1)*RCT(12      DDLR      538
      + ,1)**(K-2)      DDLR      539
          ATS(13,K) = 0.0      DDLR      540
          ATS(13,K+6) = -(RKSV(IPOW)/RKCV(IPOW))*(RCT(12,1)/RCT(13,1))*(K      DDLR      541
      + -1)*RCT(12,1)**(K-2)      DDLR      542
          DO 910 I=1,3      DDLR      543
              ATS(13+I,K) = 0.0      DDLR      544
              ATS(13+I,K+6) = 0.0      DDLR      545
910      CONTINUE      DDLR      546
          DO 920 I=1,4      DDLR      547
              ATS(I+7,K) = 0.0      DDLR      548
920      CONTINUE      DDLR      549
930      CONTINUE      DDLR      550
          DO 940 K=13,16      DDLR      551
              ATS(16,K) = HF+(RKCSV(IPOW)/RDD)*(K-13)      DDLR      552
              ATS(12,K) = -HF*RCT(13,1)**(K-13)      DDLR      553
              ATS(13,K) = AS(7,K-6)      DDLR      554
940      CONTINUE      DDLR      555
C      DDLR      556
C      COLLOCATION FOR THE RESTRICTED FUEL      DDLR      557
C      DDLR      558
          DO 960 JJ=1,6      DDLR      559
              DO 950 II=2,5      DDLR      560
                  ATS(II,JJ) = ARSV(II,JJ,IPOW)      DDLR      561
950      CONTINUE      DDLR      562
960      CONTINUE      DDLR      563
C      DDLR      564

```

C	COLLOCATION FOR THE UNRESTRICTED FUEL	DDL	565
C		DDL	566
	DO 970 JJ=8,11	DDL	567
	ATS(JJ,7) = 0.0	DDL	568
	ATS(JJ,8) = 1.0/RCT(JJ,1)	DDL	569
	ATS(JJ,9) = 4.0	DDL	570
	ATS(JJ,10) = 9.0*RCT(JJ,1)	DDL	571
	ATS(JJ,11) = 16.0*RCT(JJ,2)	DDL	572
	ATS(JJ,12) = 25.0*RCT(JJ,3)	DDL	573
970	CONTINUE	DDL	574
C		DDL	575
C	COLLOCATION FOR THE CLAD	DDL	576
C		DDL	577
	DO 990 I=13,16	DDL	578
	DO 980 J=14,15	DDL	579
	ATS(J,I) = AS(J-6,I-6)	DDL	580
980	CONTINUE	DDL	581
990	CONTINUE	DDL	582
C		DDL	583
C	CALCULATION OF RIGHT-HAND-SIDES	DDL	584
C		DDL	585
	DO 1140 K=1,16	DDL	586
	AT(1,K) = ATS(1,K)	DDL	587
	AT(6,K) = ATS(6,K)	DDL	588
	AT(7,K) = ATS(7,K)	DDL	589
	AT(12,K) = ATS(12,K)	DDL	590
	AT(13,K) = ATS(13,K)	DDL	591
	AT(16,K) = ATS(16,K)	DDL	592
	IF (K,EQ,1) GO TO 1120	DDL	593
	IF (K,GT,12) GO TO 1050	DDL	594
	TK = TROLD(K,IPDH)+273.	DDL	595
	F = FR6IN	DDL	596
	IF (K,GT,6) F = FRDEN	DDL	597
	IF (NFUEL) 1000,1010,1020	DDL	598
1000	CONTINUE	DDL	599
	CALL TCW1 (F,FRPU02,TK,RK,DKDT,DENCR)	DDL	600
	GO TO 1030	DDL	601
1010	CONTINUE	DDL	602
	CALL TCOR (F,TK,RK,DKDT,DENCR)	DDL	603
	GO TO 1030	DDL	604
1020	CONTINUE	DDL	605
	P = 1.-F	DDL	606
	PM = D*((1.-P)/(1.+BETA*P))	DDL	607
	RK = PM*(E/(G+TK)+M*TK**3.)	DDL	608
	DKDT = PM*(-E/(G+TK)**2.+3.+M*TK**2.)	DDL	609
1030	CONTINUE	DDL	610
	IF (K,EQ,12) RKS = RK	DDL	611
	RKSV(IPDH) = RKS	DDL	612
	IF (K,EQ,1.OR,K,EQ,6.OR,K,EQ,7) GO TO 1120	DDL	613
	IF (K,EQ,12) GO TO 1120	DDL	614
	SUM = 0.0	DDL	615
	MG = 0	DDL	616
	IF (K,GT,6) MG = 6	DDL	617
	DO 1040 M=2,6	DDL	618
	TRM = (M-1)*RCT(K,1)**(M-2)*BROLD(M+MG,IPDH)	DDL	619
	SUM = TRM+SUM	DDL	620
1040	CONTINUE	DDL	621

	DENC = FK SIN	DDLR	622
	IF (K.GT.6) DENC = FK DEN	DDLR	623
	CALL HCAP (TRULD(K,IPDW),CDT,CT)	DDLR	624
	ALF = (STEP)*RK/(DENC*CT*DCU**2*17.68)	DDLR	625
	GO TO 1070	DDLR	626
1050	DENC = 6.57	DDLR	627
	IF (NCLAD.LT.0) DENC = 7.85	DDLR	628
	TC = TRULD(K,IPDW)*1.8+32.	DDLR	629
	TCEST = TC+1.0	DDLR	630
	RK = TEMP(TC,AA,AA(1,2),NN)	DDLR	631
	RKEST = TEMP(TCEST,AA,AA(1,2),NN)	DDLR	632
	DKDT = (RKEST-RK)*1.8/57.803	DDLR	633
	RK = RK/57.803	DDLR	634
	IF (K.EQ.13) RKCV(IPDW) = RK	DDLR	635
	IF (K.EQ.16) RKCSV(IPDW) = RK	DDLR	636
	IF (K.EQ.13) GO TO 1120	DDLR	637
	IF (K.EQ.16) GO TO 1130	DDLR	638
	SUM = 0.0	DDLR	639
	DO 1060 M=14,16	DDLR	640
	TRM = BRULD(M,IPDW)*(M-13)*RCT(K,1)**(M-14)	DDLR	641
	SUM = SUM+TRM	DDLR	642
1060	CONTINUE	DDLR	643
	CT = 5.93E-5*TRULD(K,IPDW)+0.2916	DDLR	644
	IF (NCLAD.LT.0) CT = 0.5341	DDLR	645
	ALF = STEP*RK/(DENC*DCU**2*CT*1.613)	DDLR	646
1070	CONTINUE	DDLR	647
C		DDLR	648
C	CORRECTION OF S.S. LEFT SIDES	DDLR	649
	MG = 1	DDLR	650
	IF (K.GT.6) MG = 7	DDLR	651
	IF (K.GT.12) MG = 13	DDLR	652
	DO 1090 M=1,16	DDLR	653
	IF (K.GT.13.AND.M.LT.13) GO TO 1080	DDLR	654
	IF (K.GT.7.AND.K.LT.12.AND.M.LT.7) GO TO 1080	DDLR	655
	IF (K.LT.6.AND.M.GT.6) GO TO 1080	DDLR	656
	IF (K.LT.13.AND.K.GT.7.AND.M.GT.12) GO TO 1080	DDLR	657
	AT(K,M) = ATS(K,M)-(RCT(K,1)**(M-MG))/ALF	DDLR	658
	GO TO 1090	DDLR	659
1080	AT(K,M) = 0.0	DDLR	660
1090	CONTINUE	DDLR	661
	TFACT = SUM**2	DDLR	662
	IF (K.GT.12) GO TO 1100	DDLR	663
	BT(K) = -(Q(K)+DKDT*TFACT)/RK-TRULD(K,IPDW)/ALF	DDLR	664
	GO TO 1110	DDLR	665
1100	BT(K) = -DKDT*TFACT/RK-TRULD(K,IPDW)/ALF	DDLR	666
1110	CONTINUE	DDLR	667
	GO TO 1140	DDLR	668
1120	BT(K) = 0.0	DDLR	669
	GO TO 1140	DDLR	670
1130	BT(K) = M*TCOOL	DDLR	671
1140	CONTINUE	DDLR	672
	CALL SIMQ (AT,BT,16)	DDLR	673
	DO 1180 I=1,16	DDLR	674
	IF (I.GT.12) GO TO 1160	DDLR	675
	IF (I.GT.6) GO TO 1150	DDLR	676
	TRULD(I,IPDW) = BT(1)+BT(2)*RCT(I,1)+BT(3)*RCT(I,2)+BT(4)*RCT(I	DDLR	677
+	,3)+BT(5)*RCT(I,4)+BT(6)*RCT(I,5)	DDLR	678

SUBROUTINE DDLR

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	BR(I) = BT(I)	DDL	679
	GO TO 1170	DDL	680
1150	CONTINUE	DDL	681
	TRULD(I,IPOW) = BT(7)+BT(8)*RCT(I,1)+BT(9)*RCT(I,2)+BT(10)*RCT(I,3)+BT(11)*RCT(I,4)+BT(12)*RCT(I,5)	DDL	682
	B(I-6) = ST(I)	DDL	683
	GO TO 1170	DDL	684
1160	CONTINUE	DDL	685
	B2(I-12) = BT(I)	DDL	686
	TRULD(I,IPOW) = BT(13)+BT(14)*RCT(I,1)+BT(15)*RCT(I,2)+BT(16)*RCT(I,3)+BT(17)*RCT(I,4)+BT(18)*RCT(I,5)	DDL	687
	CT(I,3)	DDL	688
1170	CONTINUE	DDL	689
	BRULD(I,IPOW) = BT(I)	DDL	690
1180	CONTINUE	DDL	691
1190	CONTINUE	DDL	692
	RETURN	DDL	693
C		DDL	694
C		DDL	695
	END	DDL	696
		DDL	697

SUBROUTINE DENSF

74/74

OPT=1

FTN 4.6+452

15/02/78 17.25.31

	SUBROUTINE DENSF (DELPI,DFS,ROW,ROWS,BU)	DESN1	2
C	SUBROUTINE DENSF CALCULATES THE DENSIFICATION OF THE FUEL USING	DESN1	3
C	R.O. MEYER'S DENSIFICATION MODEL DESCRIBED IN NUREG 0085	DESN1	4
	BN = BU	DESN1	5
	DELPI = 0.	DESN1	6
	DR = ROWS-ROW	DESN1	7
	IF (ROWS.LE.ROW) GO TO 120	DESN1	8
	IF (DR.LE..4384) GO TO 100	DESN1	9
	IF (BN.LE.5.) GO TO 120	DESN1	10
	IF (BN.GT.500.) BN = 500.	DESN1	11
	DELH = .5*DR*ALOG10(BN/5.)	DESN1	12
	GO TO 110	DESN1	13
100	IF (BN.LE.20.) GO TO 120	DESN1	14
	IF (BN.GT.2000.) BN = 2000.	DESN1	15
	DELH = .5*DR*ALOG10(BN/20.)	DESN1	16
110	DELPI = -DELH*DFS/6./ROW	DESN1	17
	ROW = ROW+DELH	DESN1	18
120	RETURN	DESN1	19
	END	DESN1	20

SUBROUTINE DMCHB (R,A,M,N,MUD,IOP,EPS,IER)

.....

SUBROUTINE DMCHB

PURPOSE

FOR A GIVEN POSITIVE-DEFINITE M BY M MATRIX A WITH SYMMETRIC
BAND STRUCTURE AND - IF NECESSARY - A GIVEN GENERAL M BY N
MATRIX R, THE FOLLOWING CALCULATIONS (DEPENDENT ON THE
VALUE OF THE DECISION PARAMETER IOP) ARE PERFORMED
(1) MATRIX A IS FACTORIZED (IF IOP IS NOT NEGATIVE), THAT
MEANS BAND MATRIX TU WITH UPPER CODIAGONALS ONLY IS
GENERATED ON THE LOCATIONS OF A SUCH THAT
TRANSPOSE(TU)*TU=A.
(2) MATRIX R IS MULTIPLIED ON THE LEFT BY INVERSE(TU)
AND/OR INVERSE(TRANSPOSE(TU)) AND THE RESULT IS STORED
IN THE LOCATIONS OF R.
THIS SUBROUTINE ESPECIALLY CAN BE USED TO SOLVE THE SYSTEM
OF SIMULTANEOUS LINEAR EQUATIONS A*X=R WITH POSITIVE-
DEFINITE COEFFICIENT MATRIX A OF SYMMETRIC BAND STRUCTURE.

USAGE

CALL DMCHB (R,A,M,N,MUD,IOP,EPS,IER)

DESCRIPTION OF PARAMETERS

R - INPUT IN CASES IOP=-3,-2,-1,1,2,3 DOUBLE PRECISION
M BY N RIGHT HAND SIDE MATRIX,
IN CASE IOP=0 IRRELEVANT,
OUTPUT IN CASES IOP=1,-1 INVERSE(A)*R,
IN CASES IOP=2,-2 INVERSE(TU)*R,
IN CASES IOP=3,-3 INVERSE(TRANSPOSE(TU))*R,
IN CASE IOP=0 UNCHANGED.
A - INPUT IN CASES IOP=0,1,2,3 DOUBLE PRECISION M BY M
POSITIVE-DEFINITE COEFFICIENT MATRIX OF
SYMMETRIC BAND STRUCTURE STORED IN
COMPRESSED FORM (SEE REMARKS),
IN CASES IOP=-1,-2,-3 DOUBLE PRECISION M BY M
BAND MATRIX TU WITH UPPER CODIAGONALS ONLY,
STORED IN COMPRESSED FORM (SEE REMARKS).
OUTPUT IN ALL CASES BAND MATRIX TU WITH UPPER
CODIAGONALS ONLY, STORED IN COMPRESSED FORM
(THAT MEANS UNCHANGED IF IOP=-1,-2,-3).
M - INPUT VALUE SPECIFYING THE NUMBER OF ROWS AND
COLUMNS OF A AND THE NUMBER OF ROWS OF R.
N - INPUT VALUE SPECIFYING THE NUMBER OF COLUMNS OF R
(IRRELEVANT IN CASE IOP=0).
MUD - INPUT VALUE SPECIFYING THE NUMBER OF UPPER
CODIAGONALS OF A.
IOP - ONE OF THE VALUES -3,-2,-1,0,1,2,3 GIVEN AS INPUT
AND USED AS DECISION PARAMETER.
EPS - SINGLE PRECISION INPUT VALUE USED AS RELATIVE
TOLERANCE FOR TEST ON LOSS OF SIGNIFICANT DIGITS.
IER - RESULTING ERROR PARAMETER CODED AS FOLLOWS
IER=0 - NO ERROR,
IER=-1 - NO RESULT BECAUSE OF WRONG INPUT
PARAMETERS M,MUD,IOP (SEE REMARKS),

DMCHB 2
DMCHB 3
DMCHB 4
DMCHB 5
DMCHB 6
DMCHB 7
DMCHB 8
DMCHB 9
DMCHB 10
DMCHB 11
DMCHB 12
DMCHB 13
DMCHB 14
DMCHB 15
DMCHB 16
DMCHB 17
DMCHB 18
DMCHB 19
DMCHB 20
DMCHB 21
DMCHB 22
DMCHB 23
DMCHB 24
DMCHB 25
DMCHB 26
DMCHB 27
DMCHB 28
DMCHB 29
DMCHB 30
DMCHB 31
DMCHB 32
DMCHB 33
DMCHB 34
DMCHB 35
DMCHB 36
DMCHB 37
DMCHB 38
DMCHB 39
DMCHB 40
DMCHB 41
DMCHB 42
DMCHB 43
DMCHB 44
DMCHB 45
DMCHB 46
DMCHB 47
DMCHB 48
DMCHB 49
DMCHB 50
DMCHB 51
DMCHB 52
DMCHB 53
DMCHB 54
DMCHB 55
DMCHB 56
DMCHB 57
DMCHB 58

OR BECAUSE OF A NONPOSITIVE RADICAND AT
SOME FACTORIZATION STEP.
OR BECAUSE OF A ZERO DIAGONAL ELEMENT
AT SOME DIVISION STEP.
IER*K - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-
CANCE INDICATED AT FACTORIZATION STEP K+1
WHERE RADICAND WAS NO LONGER GREATER
THAN EPS*A(K+1,K+1).

REMARKS

UPPER PART OF SYMMETRIC BAND MATRIX A CONSISTING OF MAIN
DIAGONAL AND MUD UPPER CODIAGONALS (RESP. BAND MATRIX TU
CONSISTING OF MAIN DIAGONAL AND MUD UPPER CODIAGONALS)
IS ASSUMED TO BE STORED IN COMPRESSED FORM, I.E. ROWWISE
IN TOTALLY NEEDED M*MUD*(2M-MUD-1)/2 SUCCESSIVE STORAGE
LOCATIONS. ON RETURN UPPER BAND FACTOR TU (ON THE LOCATIONS
OF A) IS STORED IN THE SAME WAY,
RIGHT HAND SIDE MATRIX R IS ASSUMED TO BE STORED COLUMNWISE
IN N*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN RESULT MATRIX
INVERSE(A)*R OR INVERSE(TU)*R OR INVERSE(TRANSPOSE(TU))*R
IS STORED COLUMNWISE TOO ON THE LOCATIONS OF R.
INPUT PARAMETERS M, MUD, IOP SHOULD SATISFY THE FOLLOWING
RESTRICTIONS MUD NOT LESS THAN ZERO,
1+MUD NOT GREATER THAN M,
ABS(IOP) NOT GREATER THAN 3.
NO ACTION BESIDES ERROR MESSAGE IER=-1 TAKES PLACE IF THESE
RESTRICTIONS ARE NOT SATISFIED.
THE PROCEDURE GIVES RESULTS IF THE RESTRICTIONS ON INPUT
PARAMETERS ARE SATISFIED, IF RADICANDS AT ALL FACTORIZATION
STEPS ARE POSITIVE AND/OR IF ALL DIAGONAL ELEMENTS OF
UPPER BAND FACTOR TU ARE NONZERO.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
NONE

METHOD

FACTORIZATION IS DONE USING CHOLESKY-S SQUARE-ROOT METHOD,
WHICH GENERATES THE UPPER BAND MATRIX TU SUCH THAT
TRANSPOSE(TU)*TU=A. TU IS RETURNED AS RESULT ON THE
LOCATIONS OF A. FURTHER, DEPENDENT ON THE ACTUAL VALUE OF
IOP, DIVISION OF R BY TRANSPOSE(TU) AND/OR TU IS PERFORMED
AND THE RESULT IS RETURNED ON THE LOCATIONS OF R.
FOR REFERENCE, SEE H. RUTISHAUSER, ALGORITHMUS 1 - LINEARES
GLEICHUNGSSYSTEM MIT SYMMETRISCHER POSITIV-DEFINITER
BANDMATRIX NACH CHOLESKY - , COMPUTING (ARCHIVES FOR
ELECTRONIC COMPUTING), VOL.1, ISS.1 (1966), PP.77-78.

.....
DIMENSION R(75), A(500)

TEST ON WRONG INPUT PARAMETERS

IF (IABS(IOP)=3) 100,100,520

IF (MUD) 520,110,110

MC = MUD+1

IF (M-MC) 520,120,120

120	MM = M-MUD	DMCHB	116
	IER = 0	DMCHB	117
C		DMCHB	118
C	MC IS THE MAXIMUM NUMBER OF ELEMENTS IN THE ROWS OF ARRAY A	DMCHB	119
C	MM IS THE INDEX OF THE LAST ROW IN ARRAY A WITH MC ELEMENTS	DMCHB	120
C		DMCHB	121
C	*****	DMCHB	122
C		DMCHB	123
C	START FACTORIZATION OF MATRIX A	DMCHB	124
	IF (IOP) 330,130,130	DMCHB	125
130	IEND = 0	DMCHB	126
	LLDST = MUD	DMCHB	127
	DO 320 K=1,M	DMCHB	128
	IST = IEND+1	DMCHB	129
	IEND = IST+MUD	DMCHB	130
	J = K-MM	DMCHB	131
	IF (J) 150,150,140	DMCHB	132
140	IEND = IEND-J	DMCHB	133
150	IF (J-1) 170,170,160	DMCHB	134
160	LLDST = LLDST-1	DMCHB	135
170	LMAX = MUD	DMCHB	136
	J = MC-K	DMCHB	137
	IF (J) 190,190,180	DMCHB	138
180	LMAX = LMAX-J	DMCHB	139
190	ID = 0	DMCHB	140
	TOL = A(IST)*EPS	DMCHB	141
C		DMCHB	142
C	START FACTORIZATION-LOOP OVER K-TH ROW	DMCHB	143
	DO 320 I=IST,IEND	DMCHB	144
	SUM = 0.00	DMCHB	145
	IF (LMAX) 230,230,200	DMCHB	146
C		DMCHB	147
C	PREPARE INNER LOOP	DMCHB	148
200	LL = IST	DMCHB	149
	LLD = LLDST	DMCHB	150
C		DMCHB	151
C	START INNER LOOP	DMCHB	152
	DO 220 L=1,LMAX	DMCHB	153
	LL = LL-LLD	DMCHB	154
	LLL = LL+ID	DMCHB	155
	SUM = SUM+A(LL)*A(LLL)	DMCHB	156
	IF (LLD=MUD) 210,220,220	DMCHB	157
210	LLD = LLD+1	DMCHB	158
220	CONTINUE	DMCHB	159
C	END OF INNER LOOP	DMCHB	160
C		DMCHB	161
C	TRANSFORM ELEMENT A(I)	DMCHB	162
230	SUM = A(I)-SUM	DMCHB	163
	IF (I=IST) 240,240,290	DMCHB	164
C		DMCHB	165
C	A(I) IS DIAGONAL ELEMENT. ERROR TEST.	DMCHB	166
240	IF (SUM) 520,520,250	DMCHB	167
C		DMCHB	168
C	TEST ON LOSS OF SIGNIFICANT DIGITS AND WARNING	DMCHB	169
250	IF (SUM=TOL) 260,260,280	DMCHB	170
260	IF (IER) 270,270,280	DMCHB	171
270	IER = K-1	DMCHB	172

C		DMCHB	173
C	COMPUTATION OF PIVOT ELEMENT	DMCHB	174
280	PIV = SQRT(SUM)	DMCHB	175
	A(I) = PIV	DMCHB	176
	PIV = 1.0/PIV	DMCHB	177
	GO TO 300	DMCHB	178
C		DMCHB	179
C	A(I) IS NOT DIAGONAL ELEMENT	DMCHB	180
290	A(I) = SUM*PIV	DMCHB	181
C		DMCHB	182
C	UPDATE ID AND LMAX	DMCHB	183
300	ID = ID+1	DMCHB	184
	IF (ID=J) 320,320,310	DMCHB	185
310	LMAX = LMAX+1	DMCHB	186
320	CONTINUE	DMCHB	187
C		DMCHB	188
C	END OF FACTORIZATION=LOOP OVER K-TH ROW	DMCHB	189
C	END OF FACTORIZATION OF MATRIX A	DMCHB	190
C		DMCHB	191
C	*****	DMCHB	192
C		DMCHB	193
C	PREPARE MATRIX DIVISIONS	DMCHB	194
	IF (IOP) 330,530,330	DMCHB	195
530	ID = N*M	DMCHB	196
	IEND = IABS(IOP)+2	DMCHB	197
	IF (IEND) 340,440,340	DMCHB	198
C		DMCHB	199
C	*****	DMCHB	200
C		DMCHB	201
C	START DIVISION BY TRANSPOSE OF MATRIX TU (TU IS STORED IN	DMCHB	202
C	LOCATIONS OF A)	DMCHB	203
340	IST = 1	DMCHB	204
	LMAX = 0	DMCHB	205
	J = -MR	DMCHB	206
	LLDST = MUD	DMCHB	207
	DO 450 K=1,M	DMCHB	208
	PIV = A(IST)	DMCHB	209
	IF (PIV) 350,520,350	DMCHB	210
350	PIV = 1.0/PIV	DMCHB	211
C		DMCHB	212
C	STA=T BACKSUBSTITUTION=LOOP FOR K-TH ROW OF MATRIX R	DMCHB	213
	DO 390 I=K,ID,M	DMCHB	214
	SUM = 0.00	DMCHB	215
	IF (LMAX) 390,390,360	DMCHB	216
C		DMCHB	217
C	PREPARE INNER LOOP	DMCHB	218
360	LL = IST	DMCHB	219
	LLL = I	DMCHB	220
	LLD = LLDST	DMCHB	221
C		DMCHB	222
C	START INNER LOOP	DMCHB	223
	DO 380 L=1,LMAX	DMCHB	224
	LL = LL+LLD	DMCHB	225
	LLL = LLL+1	DMCHB	226
	SUM = SUM+A(LL)*R(LL)	DMCHB	227
	IF (LLD=MUD) 370,380,380	DMCHB	228
370	LLD = LLD+1	DMCHB	229

380	CONTINUE	DMCHB	230
C	END OF INNER LOOP	DMCHB	231
C		DMCHB	232
C	TRANSFORM ELEMENT R(I)	DMCHB	233
C	R(I) = PIV*(R(I)-SUM)	DMCHB	234
390	END OF BACKSUBSTITUTION-LOOP FOR K-TH ROW OF MATRIX R	DMCHB	235
C		DMCHB	236
C	UPDATE PARAMETERS LMAX, IST AND LLDST	DMCHB	237
C	IF (MC=K) 410,410,400	DMCHB	238
400	LMAX = K	DMCHB	239
410	IST = IST+MC	DMCHB	240
	J = J+1	DMCHB	241
	IF (J) 430,430,420	DMCHB	242
420	IST = IST-J	DMCHB	243
	LLDST = LLDST-1	DMCHB	244
430	CONTINUE	DMCHB	245
C		DMCHB	246
C	END OF DIVISION BY TRANSPOSE OF MATRIX TU	DMCHB	247
C		DMCHB	248
C	*****	DMCHB	249
C		DMCHB	250
C	START DIVISION BY MATRIX TU (TU IS STORED ON LOCATIONS UP 4)	DMCHB	251
	IF (IEND) 440,440,530	DMCHB	252
440	IST = M+(MUD*(M+M-MC))/2+1	DMCHB	253
	LMAX = 0	DMCHB	254
	K = M	DMCHB	255
450	IEND = IST-1	DMCHB	256
	IST = IEND-LMAX	DMCHB	257
	PIV = A(IST)	DMCHB	258
	IF (PIV) 460,520,460	DMCHB	259
460	PIV = 1.0/PIV	DMCHB	260
	L = IST+1	DMCHB	261
C		DMCHB	262
C	START BACKSUBSTITUTION-LOOP FOR K-TH ROW OF MATRIX R	DMCHB	263
	DO 490 I=K, ID, M	DMCHB	264
	SUM = 0.00	DMCHB	265
	IF (LMAX) 490,490,470	DMCHB	266
470	LLL = I	DMCHB	267
C		DMCHB	268
C	START INNER LOOP	DMCHB	269
	DO 480 LL=L, IEND	DMCHB	270
	LLL = LLL+1	DMCHB	271
480	SUM = SUM+A(LL)*R(LL)	DMCHB	272
C	END OF INNER LOOP	DMCHB	273
C		DMCHB	274
C	TRANSFORM ELEMENT R(I)	DMCHB	275
C	R(I) = PIV*(R(I)-SUM)	DMCHB	276
490	END OF BACKSUBSTITUTION-LOOP FOR K-TH ROW OF MATRIX R	DMCHB	277
C		DMCHB	278
C	UPDATE PARAMETERS LMAX AND K	DMCHB	279
	IF (K=MH) 510,510,500	DMCHB	280
500	LMAX = LMAX+1	DMCHB	281
510	K = K-1	DMCHB	282
	IF (K) 530,530,450	DMCHB	283
C		DMCHB	284
C	END OF DIVISION BY MATRIX TU	DMCHB	285
C		DMCHB	286

SUBROUTINE DMCHB

74/74 OPT=1

FIN 4.6+452

15/02/78 17.25.31

C	*****	DMCHB	287
C		DMCHB	288
C	ERROR EXIT IN CASE OF WRONG INPUT PARAMETERS OR PIVOT ELEMENT	DMCHB	289
C	LESS THAN OR EQUAL TO ZERO	DMCHB	290
520	IER = -1	DMCHB	291
530	RETURN	DMCHB	292
	END	DMCHB	293

```

      SUBROUTINE EFFECT (EFSTR,TSTSR,NAXL,NRAD)
      C
      C PURPOSE OF THIS SUBROUTINE IS TO COMPUTE THE EFFECTIVE STRESS
      C THE EFFECTIVE STRESS IS COMPUTED A NEGLECTING SHEAR
      C
      COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IERSLM,IELSTN,IELSTR
      +,IXSAVE,IPRESS,ITELM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD
      +ET,IR,IHRK,ISA,ISAC,ISAF,ISB,ISBB,ISBHC,I
      +SBHF,ISBC,ISBF,ISK,ISKC,ISTRES,ITDELE,IXK,IXKC,IXKT
      +,IXLOAD,ICC,ITHET,ICA,ITMTAX,INADE1,IRA,IY,IAELM,I
      +AAAA,IBA,IHRA,IHRA,IXLODA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I
      +PSLA,IAK,IACN,IKAN,IKZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA,
      +IIDD,ITHE,IEFSTR,ITCRP,ICRPG,ICKATE
      +,IXKCC,IXKTT
      +,ITPB,ITCB,IBBSV,IESTSV,IESTNV
      +,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDSK,IPSSV,IPSSRT,JIIJ,IIPUS
      +,IBASE,ITFR,IYM,IZ
      +,IUSPSV
      COMMON /GEOPAR/ RFI,RF0,RCI,RCO,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUT,P
      +DEL
      COMMON /LIMITS/ NF,NC,NDEL,NDELF,NDELC,ICON,IOF,NFF,MAXDC,NETOT,NE
      +WF,NEWT,KA,NARF,NARC,NART,NABF,NABC,NAHT,NAEC,NAEF,NTAE,JAM,NAXB1,
      +NAXB2,NAXB4,NFF1,NABF2,NABT2,CP,CH,FRDEN,KZ1,ISSTOP,IPLAS,NKN
      +,ICRPP,MAXP,MAXC,MAXT,KRNCH,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N
      +MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NULD
      +,KPLAS,ICNT,HRAKR,IRAK,HAKBUM,IPLEND,ISP
      COMMON /LASTIC/ E,XNU,ALPHA
      COMMON /CONTNT/ PI
      COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2
      COMMON /CRP/ PSP(25),TPLASK
      DIMENSION EFSTR(NRAD,1)
      DIMENSION TSTSR(4,NRAD,1)
      C
      DO 110 J=1,NAXL
      DO 100 I=1,NRAD
      EFSTH(I,J) = SQRT((CR/(CH+1.))*(TSTSR(3,I,J)-TSTSR(2,I,J))**2+
      + (CR/(CP*(CH+1.)))*(TSTSR(2,I,J)-TSTSR(1,I,J))**2+1./(CH+1.)*
      + (TSTSR(1,I,J)-TSTSR(3,I,J))**2)
      C
      100 CONTINUE
      110 CONTINUE
      RETURN
      END

```

SUBROUTINE EXPAND

74/74

OPT=1

FIN 4.6+452

15/02/78

17.25.31

```

      SUBROUTINE EXPAND (NF, RD, B, BR, DELRT, RCU, RSIN)
      COMMON /BA/ AA(23,7), CF(7,8), RKS, DELZA(20), NN, NCONN
      +, NCNN, ISTRT, ITER
      +, FLXK
      COMMON /FEP/ MHF, NT, KKZZZ, ISTOP, RDOLD(20), RKSIN(20), ICNOLD(20)
      +, IDUOLD(20)
      +, PINOLD, POUTLD, POLD, NTOLD(20)
      +, ICHNG, CRPAD(20)
      +, STEPS
C      UPPER BOUNDARY DIMENSION OF THE AXIAL NODE.
C      THIS IS THE FREE THERMAL EXPANSION MODEL.
C
      DIMENSION B(6), BB(6), BR(6), RD(NF)
      TA(A,C) = C(1)+C(2)*A+C(3)+A**2+C(4)+A**3+C(5)+A**4+C(6)+A**5
      TB(D,E) = (D+E)/2.
      NF2 = NF*2
      NF2M2 = NF2-2
      DR = RD(NF)-RD(1)
      DRD = DR/(NF2-2)
      DELRT = 0.
      DO 130 I=1, NF2M2
        RIN = RD(NF)-DRD*I
        RUN = RIN+DRD
        R = TB(RUN, RIN)
        RN = R/RCU
        DO 100 JK=1, 6
          BB(JK) = B(JK)
          IF (RSIN.GT.RD(1).AND.R.LT.RSIN) BB(JK) = BR(JK)
100      CONTINUE
        TAC = TA(RN, BB)
        IF (NCONN.LE.0) GO TO 110
        TF = TAC*1.4+32.
        CST = TEMP(TF, CF, CF(1,6), 7)
        CST = CST*(TF-77.)
        GO TO 120
110      CONTINUE
        CST = FULEX(TAC)
120      CONTINUE
        DELL = CST*DRD
        IF (I.EQ.1) DELL = DELL+RD(1)*CST
        DELRT = DELRT+DELL
130      CONTINUE
      RETURN
      END

```

```

EXPAND      2
COMC        2
COMC        3
COMC        4
COMC        5
COMC        6
COMC        7
COMC        8
COMC        9
EXPAND      4
EXPAND      5
EXPAND      6
EXPAND      7
EXPAND      8
EXPAND      9
EXPAND     10
EXPAND     11
EXPAND     12
EXPAND     13
EXPAND     14
EXPAND     15
EXPAND     16
EXPAND     17
EXPAND     18
EXPAND     19
EXPAND     20
EXPAND     21
EXPAND     22
EXPAND     23
EXPAND     24
EXPAND     25
EXPAND     26
EXPAND     27
EXPAND     28
EXPAND     29
EXPAND     30
EXPAND     31
EXPAND     32
EXPAND     33
EXPAND     34
EXPAND     35
EXPAND     36
EXPAND     37
EXPAND     38

```

SUBROUTINE FISGAS

74/74

OPT=1

FIN 4.6+052

15/02/78

17.25.31

	SUBROUTINE FISGAS (P,T,X,Z,XEM,RKM)	FISGAS	2
C	FISGAS CALCULATES THE INCREMENTAL AMOUNT OF NOBLE GASES PRODUCED.	FISGAS	3
C	ASSUME 200 MEV PER FISSION = 3.4678 E 13 FISSIONS/KW-SEC, = 6.9356 E	FISGAS	4
C	13 PRODUCT ATOMS/KW-SEC	FISGAS	5
C	X IS THE GAS ATOM FRACTION OF FISSION PRODUCT ATOMS, DEFAULT=.28	FISGAS	6
C	Z IS THE KRYPTON FRACTION IN THE KR+XE GASEOUS FISSION PRODUCTS,	FISGAS	7
C	DEFAULT=0.125	FISGAS	8
C	P IS LOCAL POWER KW/FT	FISGAS	9
C	T IS DURATION OF TIME STEP AT POWER P, DAYS	FISGAS	10
C	NTS IS TIME STEP NUMBER	FISGAS	11
C	IPOW IS THE LOCAL AXIAL NODE IDENTIFICATION	FISGAS	12
C	MOLES OF KRYPTON PRODUCED DURING THE TIME STEP IS Z*X*6.9356 E13	FISGAS	13
C	6.023 E 23*SECDAY	FISGAS	14
	RKM = Z*X*P*T*0.993E-5	FISGAS	15
	XEM = (1.0-Z)/Z*RKM	FISGAS	16
	RETURN	FISGAS	17
	END	FISGAS	18

SUBROUTINE FLUXD

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

1		SUBROUTINE FLUXD (RV,RVE,FR35,FR38,RHU,IRL,DFS,A,B,C,TINLET)	FLUXD	2
	C	FLUXD CALCULATES THE FUEL RADIAL FLUX DEPRESSION.	FLUXD	3
	C		FLUXD	4
		DIMENSION RV(20,2), RVE(20,2)	FLUXD	5
		DATA SIGA5,SIGA8,SIGA0,SIGT5,SIGT8,SIGT0/578.,2.33,0.,586.5,10.8,4	FLUXD	6
		+.2/	FLUXD	7
		FR38 = 1.-FR35	FLUXD	8
		ENS = RHU*FR35*.00225427	FLUXD	9
		EN8 = RHU*FR38*.00223079	FLUXD	10
		END = (ENS+EN8)*2.	FLUXD	11
		SGA = (ENS*SIGA5+EN8*SIGA8+END*SIGA0)	FLUXD	12
		SGT = (ENS*SIGT5+EN8*SIGT8+END*SIGT0)	FLUXD	13
		CAPSQ = 3.*SGA*SGT*(1.-.8*SGA/SGT)	FLUXD	14
		CORR = .6*((TINLET-70.)/(394.))+1.0	FLUXD	15
		CAPPA = SQRT(CAPSQ)*CORR	FLUXD	16
		CAPSQ = CAPPA**2.	FLUXD	17
		A = 1.	FLUXD	18
		B = CAPSQ/4.	FLUXD	19
		C = B**2./4.	FLUXD	20
		RV(1,1) = 0.0	FLUXD	21
		RV(1,2) = 1.	FLUXD	22
		IZ = 11	FLUXD	23
		IF (IRL.GT.0) IZ = IRL+1	FLUXD	24
		DO 110 I=2,IZ	FLUXD	25
		J = I-1	FLUXD	26
		RV(I,1) = RV(I-1,1)+DFS/IZ	FLUXD	27
		R = RV(I,1)*2.54/4.*CAPPA	FLUXD	28
		RV(I,2) = 1.+R**2.+R**4./4.	FLUXD	29
		IF (IRL.GT.0) GO TO 100	FLUXD	30
		RVE(J,1) = (RV(J,1)+RV(I,1))/2.	FLUXD	31
		RVE(J,2) = (RV(J,2)+RV(I,2))/2.	FLUXD	32
100		CONTINUE	FLUXD	33
110		CONTINUE	FLUXD	34
		RETURN	FLUXD	35
		END	FLUXD	36

FUNCTION FULEX

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```
C      FUNCTION FULEX (T)
C
C      FULEX RETURNS THERMAL STRAIN(DELTA L OVER L).
C
      FULEX = 2.896E-9*(T*T-441.)+6.797E-6*(T-21.)
      RETURN
      END
```

```
FULEX      2
FULEX      3
FULEX      4
FULEX      5
FULEX      6
FULEX      7
FULEX      8
```

```

SUBROUTINE GAPCON (TELM,TDELE,TW,TWDEL,KAN,PRESS,YM,YD,IDO,TREF,RF GAPCON      2
+0,RCI,RCO,RFI,CP,CR,NOELF,TSTSR,TSTNR,NAH2D,TIME,DISPC,DISPF,HD,NF GAPCON      3
+FID,TFR,TFR,TCH,NART,NAHF,NAHT,NFO,NCD,NOELD,NARFO,NTAED,EFSTR,OLD GAPCON      4
+SE,DSPSV) GAPCON      5
C GAPCON      6
C SUBROUTINE GAPCON IS THE MAJOR THERMAL CALCULATOR. IT CONTAINS GAPCON      7
C ALL THE MAJOR LOOPS, TIME STEP, GAS CONTENT, AXIAL NODE, AND ITER GAPCON      8
C LOOP FOR GAP CONDUCTANCE. GAPCON      9
C GAPCON IS VERY SIMILAR TO GAPCON THERMAL 2. GAPCON     10
C GAPCON     11
COMMON /RSTR/DDZZ,KKNT,HHGG GAPCON     12
COMMON /RSTR/ RRCV(20),RASV(20),KKCSV(20),BBB(10,20),BROLD(10,20) GAPCON     13
+,BTOLD(10,20),TRULD(16,20),ANSV(6,6,20),VOID(20),KOUNT(20),DENCGRS, GAPCON     14
+DENCUS GAPCON     15
COMMON /PRSSV/ PRSINC(30),PRSS(30) GAPCON     16
COMMON /EA/ ZR(6,7),ZR4(6,7),ST(12,7),RV(20,2),UDZ(7,7) COME      2
COMMON /GA/ NPMK,NTIME,NFZ,NCZ,IPLNT,ATMOS,DCI,DCO,DFS,DSINZ,DTEMP COMA      2
+,DVOIDZ,FR3S,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRACN,FRACXE,FRU COMA      3
+,EN,FRPU02,FRSIN,IGAS,IPEAK,LFUEL,LVOIDZ, NFUEL,NQH,PRCDH,R COMA      4
+,ADS,RDUC,KOUP,S,TINLET,TM,TPLAS,VPLENZ,XCO,XH,XN,IRELOC,IHL,IDENSE COMA      5
+,FLXRAT,X,Z,ZZ,YY,WW,EXTP,ISTDR,NCLAD,ZLLAD,NSH,PROFIL(20),PSEUDO COMA      6
+(100),TIME(100),NFLX,TITLE(10),MXPWR,IQPT,JSTART,KSAVE,ISAVE,ITMCU COMA      7
+,T,ISTAND,ITAPE,NSTEPS,MINI(20),SUMARY,FRDEN2 COMA      8
INTEGER SUMARY COMA      9
REAL LFUEL,LVOIDZ,MXPWR COMA     10
COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN COMC      2
+, NCON,ISTRT,ITER COMC      3
+, FLXR COMC      4
COMMON/FEP/HMF,NT,KKZZZ,ISTOP,RDOLD(20),RRSIN(20),ICMOLD(20) COMC      5
+, IDCOLD(20) COMC      6
+,PINOLD,POUTLD,PULD,NTULD(20) COMC      7
+, ICHNG,CRPAD(20) COMC      8
+,STEPS COMC      9
COMMON/DA/ TABLE(80,2),GMWT(7),SIGLJ(7),EKLJ(7),HF(20),RTCU(5) COMD      2
+,RTCU(5),EPSIF,EPSIC COMD      3
COMMON /FA/ PI,CCPIN3,SECDAV,AVOGAD,HR,CONEN COMF      2
COMMON /AZ/ BURNUP(20),DELPI(20),DELR,DELRB(20),DELRCC(20),DELRCI COMI      2
+(20),DELRCT(20),DELRP(20),DELR(20),DSIN,DVOID,FRU02,FR3S,FR39,GAP, COMI      3
+HBC,KB,DELGDM(20) COMI      4
COMMON/LCMCOM/SAVNKE(50,20), SAVNKR(50,20), SUMNXE(50,20), GAPCON     23
+SUMNKR(50,20) GAPCON     24
C LEVEL 2,SAVNKE,SAVNKR,SUMNXE,SUMNKR GAPCON     25
REAL MOL,MOLEFR,LF GAPCON     26
INTEGER OLD GAPCON     27
REAL LFTS,MTONFT,MTM,HREACH,HREACH GAPCON     28
DIMENSION DHVOL(20), LSTHIF(20), E(7), TMA(7), TMFA(7), RVE(20,2) GAPCON     29
DIMENSION TW(NTAED,2), TWDEL(1), TELM(NOELD,NART,2), TDELE(NOELD,1 GAPCON     30
+) GAPCON     31
DIMENSION DISPF(NFFID,NABT), DISPC(NFFID,NABT), RD(NFFID,NABT) GAPCON     32
DIMENSION KNCN(20), KAN(1), TFR(NFO,NARFO), TFR(NFO,NABF), IDO(1), GAPCON     33
+TCH(NCD,NABT),YM(NABT,1),YD(NAB2D,1),TIME(3) GAPCON     34
DIMENSION TXMOLS(7), MOL(7), FRACIN(7), SUMOLS(7), OMEGA(7), VISCO GAPCON     35
+S(7), PHICAP(7,7), CON(7), CSUBP(7), MOLEFR(7) GAPCON     36
DIMENSION IFLAG(20), DVBX(20), DVBLST(20), AI(20), PITI(20), GAPCON     37
+DELVB(20), VULTX(20), TAVGXX(20), FXMOL(10,50), BI(10,20) GAPCON     38
COMMON /OUT/ POWER(20),RGAPX(20),RDOT(50,20),XMOLS(20),RCII,RDELST GAPCON     39
+(20),ISTART,CRELST(20),CRGRP(20),IACON(20,2),ESTORI(20) GAPCON     40

```

```

COMMON /CRP/ PSP(25),TPLASK
COMMON /AXL/ SUMM,SUMFF
DIMENSION PRESS(20), HGAS(20), HGC(20), HRAD(20), HSOLID(20)
DIMENSION OLOSE(NOELO,NARFD)
DIMENSION MREACU(20), MREACN(20), QG(20), DELHFT(20)
DIMENSION QOVRAC(20), G1PG2(20), GK(20), TCCOL(20), TCC(20)
DIMENSION BR(6,20),RSIN(20),EFSTR(NOELO,NART)
DIMENSION OLOPOM(20), KANOLD(20)
DIMENSION HGCOLD(20), ESUM(20), ESTSUM(20)
DIMENSION TSTR(4,NOELO,NART), TSTR(4,NOELO,NART)
DIMENSION ABTTRP(8), GKTRP(8), GJTRP(8)
DIMENSION PFACE(20), TTVOL(50), RVOL(50)
DIMENSION VAVGTC(20),DSPSV(NOELO,1)
NNOEL = NOELF+1
NARF = NARFD
C BEGIN TIME INDEPENDENT CALCULATIONS
ITRCNT = 0
ITWANS = 0
IUP = 0
IO = 0
LTLUS = 0
RATEC = 0.0
PSP(NPUW+2) = 0.
KONTMM = 0
PUNRLD = 0.
ISTEP = 0
IDU(1) = 1
IDU(NARF) = 1
IDU(NARF+1) = 1
FNPUW = FLOAT(NPUW)
NPOW1 = NPOW+1
DO 100 IJ=2,NART
  HGC(IJ) = 1000.
100 CONTINUE
TSINC = 2160.
LF = LFUEL/FNPUW
TPLASK = TPLAS+273.
TPLAS = TPLAS+1.8+32.
RVOIDZ = DVOIDZ/2.
RVOID = RVOIDZ
DVOID = DVOIDZ
RFS = DFS/2.
RFU = RFS
RCI = DCI/2.
RCO = DCO/2.
NFF1 = NOELO/2+2
NF = NFD
NC = NCD
RF1 = DVOIDZ/2.
TIMEN = 0.
TIMEO = 0.
TIMEF = 0.
CALL MOVEKA (0,ICNOLD,20)
CALL MOVEKA (0,IUNOLD,20)
CALL MOVEKA (0,DELGDM,20)
IF (NFI,LT,0.001) NFI = 0.001
DRC = (RCO-RCI)/FLOAT(NC-1)

```

```

GAPCON 41
GAPCON 42
GAPCON 43
GAPCON 44
GAPCON 45
GAPCON 46
GAPCON 47
GAPCON 48
GAPCON 49
GAPCON 50
GAPCON 51
GAPCON 52
GAPCON 53
GAPCON 54
GAPCON 55
GAPCON 56
GAPCON 57
GAPCON 58
GAPCON 59
GAPCON 60
GAPCON 61
GAPCON 62
GAPCON 63
GAPCON 64
GAPCON 65
GAPCON 66
GAPCON 67
GAPCON 68
GAPCON 69
GAPCON 70
GAPCON 71
GAPCON 72
GAPCON 73
GAPCON 74
GAPCON 75
GAPCON 76
GAPCON 77
GAPCON 78
GAPCON 79
GAPCON 80
GAPCON 81
GAPCON 82
GAPCON 83
GAPCON 84
GAPCON 85
GAPCON 86
GAPCON 87
GAPCON 88
GAPCON 89
GAPCON 90
GAPCON 91
GAPCON 92
GAPCON 93
GAPCON 94
GAPCON 95
GAPCON 96
GAPCON 97

```

```

      DRF = (RFS-RFI)/FLOAT(NF-1)
      IF (JSTART.GT.0) GO TO 160
C     THE FOLLOWING CALCULATIONS ARE NOT FOR THE RESTART RUN
      ISAVE = 0
      KTRAN = 1
      OLD = 1
      NEW = 2
      JCASE = 0
      TDAYS = 0.
      CALL MOVEKA (0,PITI,NPUN)
      CALL MOVEKA (1,LSTHIF,NPCW)
      DO 120 J=1,20
        DO 110 I=1,50
          SAVNKR(I,J) = 0.
          SAVNKE(I,J) = 0.
          SUMNKR(I,J) = 0.
          SUMNKE(I,J) = 0.
110      CONTINUE
120      CONTINUE
C
      CALL MOVEKA (0,KANOLD,20)
      CALL MOVEKA (0,PFACE,20)
      CALL MOVEKA (0,PRESS,20)
      CALL MOVEKA (0,PHSINC,30)
      CALL MOVEKA (0,PRSS,30)
      CALL MOVEKA (0,RDELST,20)
      CALL MOVEKA (0,RSIN,20)
      CALL MOVEKA (0,OLDPOW,20)
      CALL MOVEKA (0,KNCN,20)
      CALL MOVEKA (0,DHVOL,20)
      CALL MOVEKA (0,LSTHIF,20)
      CALL MOVEKA (0,DVBX,20)
      CALL MOVEKA (0,VOLTX,20)
      CALL MOVEKA (0,IFLAG,20)
      CALL MOVEKA (0,DVBLST,20)
      CALL MOVEKA (0,DELVB,20)
      CALL MOVEKA (0,TAVGXX,20)
      CALL MOVEKA (0,FXMDL,500)
      CALL MOVEKA (0,PITI,20)
      CALL MOVEKA (0,A1,20)
      CALL MOVEKA (0,MGCOLD,20)
      CALL MOVEKA (0,ESUM,20)
      CALL MOVEKA (0,ESTSUM,20)
      CALL MOVEKA (0,TTVOL,50)
      CALL MOVEKA (0,RVOL,50)
      CALL MOVEKA (0,VAVGTC,20)
      CALL MOVEKA (0,RSIN,20)
C
C     ZERO MOST OF COMMON BLOCK FEP
C
      ILEN1 = LOCF(KKZZZ)
      ILEN2 = LOCF(STEPS)
      ILEN = ILEN2-ILEN1+1
      CALL MOVEKA (0,KKZZZ,ILEN)
C
C     ZERO ENTIRE COMMON BLOCK OUT.
C

```

```

      GAPCON 98
      GAPCON 99
      GAPCON 100
      GAPCON 101
      GAPCON 102
      GAPCON 103
      GAPCON 104
      GAPCON 105
      GAPCON 106
      GAPCON 107
      GAPCON 108
      GAPCON 109
      GAPCON 110
      GAPCON 111
      GAPCON 112
      GAPCON 113
      GAPCON 114
      GAPCON 115
      GAPCON 116
      GAPCON 117
      GAPCON 118
      GAPCON 119
      GAPCON 120
      GAPCON 121
      GAPCON 122
      GAPCON 123
      GAPCON 124
      GAPCON 125
      GAPCON 126
      GAPCON 127
      GAPCON 128
      GAPCON 129
      GAPCON 130
      GAPCON 131
      GAPCON 132
      GAPCON 133
      GAPCON 134
      GAPCON 135
      GAPCON 136
      GAPCON 137
      GAPCON 138
      GAPCON 139
      GAPCON 140
      GAPCON 141
      GAPCON 142
      GAPCON 143
      GAPCON 144
      GAPCON 145
      GAPCON 146
      GAPCON 147
      GAPCON 148
      GAPCON 149
      GAPCON 150
      GAPCON 151
      GAPCON 152
      GAPCON 153
      GAPCON 154

```

	ILEN1 = LOCF(POWER)	GAPCON	155
	ILEN2 = LOCF(ESTORI(20))	GAPCON	156
	ILEN = ILEN2-ILEN1+1	GAPCON	157
	CALL MOVEKA (0,POWER,ILEN)	GAPCON	158
C		GAPCON	159
C	ZERO ENTIRE RSTRT COMMON BLOCK	GAPCON	160
C		GAPCON	161
	ILEN1 = LOCF(RKCV(1))	GAPCON	162
	ILEN2 = LOCF(DENCUS)	GAPCON	163
	ILEN = ILEN2-ILEN1+1	GAPCON	164
	CALL MOVEKA (0,RKCV,ILEN)	GAPCON	165
C		GAPCON	166
C		GAPCON	167
C	ZERO OUT COMMON AZ	GAPCON	168
C		GAPCON	169
	ILEN1 = LOCF(BURNUP(1))	GAPCON	170
	ILEN2 = LOCF(DELGDM(20))	GAPCON	171
	ILEN = ILEN2-ILEN1+1	GAPCON	172
	CALL MOVEKA (0,BURNUP,ILEN)	GAPCON	173
C		GAPCON	174
	DELOTO = 0.	GAPCON	175
	DELTAU = 0.	GAPCON	176
	DTOLD = 0.	GAPCON	177
	HATOLD = 0.	GAPCON	178
	SUMFF = 0.	GAPCON	179
	SUMH = 0.	GAPCON	180
	TIMEF = 0.	GAPCON	181
	W3 = 0.	GAPCON	182
	W5 = 0.	GAPCON	183
C	PRESTO IS NOW ESTIMATED (PIN INTERNAL PRESSURE) PSI	GAPCON	184
	PRESTO = ATMOS*14.696*2.	GAPCON	185
	TMX = 0.	GAPCON	186
	NX = 1	GAPCON	187
	NY = 1	GAPCON	188
	DO 150 I=1,NABT	GAPCON	189
	DO 140 J=1,NFF1	GAPCON	190
	IF (J.LE.NF) GO TO 130	GAPCON	191
	RD(J,I) = FLUAT(J-NF-1)*DRC+RCI	GAPCON	192
	GO TO 140	GAPCON	193
130	RD(J,I) = FLUAT(J-1)*DRF+RFI	GAPCON	194
140	CONTINUE	GAPCON	195
150	CONTINUE	GAPCON	196
160	CONTINUE	GAPCON	197
C	RESTART INFORMATION	GAPCON	198
	IF (JSTART.LE.0) GO TO 290	GAPCON	199
170	CONTINUE	GAPCON	200
C	THE FOLLOWING CARDS READ/WRITE FOR TRANSFER TO PERMANENT STORAGE.	GAPCON	201
	READ (2) TIMEN,PRESTO,DELGDM,DELPI,RSIN,PRESS,HGC,CRELST,CHGRP,RO,	GAPCON	202
	+TSTR,ROOT,SUMH,NT,PITI,BTOLD,SAVNKR,SAVNXE,SUMNKR,SUMNXE,LSTHIF,T	GAPCON	203
	+STNR,RKSV,RKCV,BBH,BHOLD,TROLD,NEW,OLD,TIMEF,TMX,NY,VOLTXS,RATE,KA	GAPCON	204
	+TEC,RATOLD,W1,W2,W3,W5,TW,TELM,DISPF,ROELST,KANOLD,ESUM,ESTSUM,KTR	GAPCON	205
	+AN,ESTORI,DTOLD,DELTAU,DELOTO,TDOYS,ARSV,BH,HGCOLD,OLDPIN,VOID,RKC	GAPCON	206
	+SV,KMUNT,DENCUS,DENCUS,PRSS,IACON,ISAVE,SUMFF,RKSIN,NX,DELVB,DVBX,	GAPCON	207
	+PIOLD,POUTLD,PINOLD,NTOLD,POWRLO,VAVGTC,PFACE,OLDSE,FLAG	GAPCON	208
	*WRITE (1) TIMEN,PRESTO,DELGDM,DELPI,RSIN,PRESS,HGC,CRELST,CHGRP,RO	GAPCON	209
	+TSTR,ROOT,SUMH,NT,PITI,BTOLD,SAVNKR,SAVNXE,SUMNKR,SUMNXE,LSTHIF,	GAPCON	210
	+STNR,RKSV,RKCV,BBH,BHOLD,TROLD,NEW,OLD,TIMEF,TMX,NY,VOLTXS,RATE,K	GAPCON	211

```

+ATEC,RATOLD,W1,W2,W3,W5,T*,TELM,DISPF,RDELST,KANOLD,ESUM,ESTSUM,KT GAPCON 212
+RAN,ESTORI,DTOLD,DELTAO,DELDT0,TDAYS,ARSV,BR,HGCOLD,MLGPOW,VOID,RK GAPCON 213
+CSV,KDUNT,DENCKS,DENCUS,PRSS,IACON,ISAVE,SUMFF,KRSIN,NX,DELYB,DVBX GAPCON 214
+,POLD,POULD,PINOLD,NTOLD,POWRLO,VAVGTC,PFACE,OLDSE,FLAG GAPCON 215
IF (IPLOT.EQ.0) GO TO 270 GAPCON 216
C THE FOLLOWING CARDS EFFECT PLOTS. CHANGES HERE MUST BE MADE IN PLO GAPCON 217
READ (2) POWER,BURNUP,HF,DELRB,DELRBI,DELRCC,DELRCT,DELRFT,KGAPX,T GAPCON 218
+COOL,TFB,TCH,MOL,HSOLID,HGAS,HHAD,GK,G1PG2,EFSTR,H1,DELRT GAPCON 219
WRITE (1) POWER,BURNUP,HF,DELRB,DELRBI,DELRCC,DELRCT,DELRFT,KGAPX, GAPCON 220
+TCOOL,TFB,TCH,MOL,HSOLID,HGAS,HHAD,GK,G1PG2,EFSTR,H1,DELRT GAPCON 221
C IF ITAPE.GT. 1 BUT .LT. NT WRITE DATA FROM TAPE USING OUTPUT GAPCON 222
IF (ITAPE.LE.1) GO TO 270 GAPCON 223
IF (ITAPE.LT.NT) GO TO 270 GAPCON 224
TTTT = TIMEN GAPCON 225
NFPNC = NF+NC GAPCON 226
NPOW1 = NPOW+1 GAPCON 227
JCASE = NT GAPCON 228
RCII = RCI GAPCON 229
NABT = NPOW+3 GAPCON 230
XMOTOT = 0. GAPCON 231
DO 180 I=1,7 GAPCON 232
XMOTOT = XMOTOT+MOL(I) GAPCON 233
180 CONTINUE GAPCON 234
ANKR = 0. GAPCON 235
ANXE = 0. GAPCON 236
DO 190 I=2,NPOW1 GAPCON 237
ANKR = ANKR+SAVNKR(NX,I)*DELZA(I)/12. GAPCON 238
ANXE = ANXE+SAVNXE(NX,I)*DELZA(I)/12. GAPCON 239
190 CONTINUE GAPCON 240
HL = (MOL(6)+MOL(7))/(ANKR+ANXE) GAPCON 241
PHAX = POWER(1) GAPCON 242
DO 200 I=1,NPOW1 GAPCON 243
PHAX = AMAX1(PHAX,POWER(I)) GAPCON 244
IF (POWER(I).EQ.PHAX) LPHAX = I GAPCON 245
200 CONTINUE GAPCON 246
DO 210 I=1,7 GAPCON 247
MOLEFR(I) = MOL(I)/XMOTOT GAPCON 248
210 CONTINUE GAPCON 249
DO 250 JPOW=1,NPOW1 GAPCON 250
JPOW = JPOW+1 GAPCON 251
IF (MINI(1).LE.0.AND.JPOW.NE.LPHAX) GO TO 240 GAPCON 252
IF (MINI(1).GT.0.AND.MINI(JPOW).EQ.0) GO TO 240 GAPCON 253
QS1 = 6518.35*POWER(JPOW)/RD(NF,JPOW) GAPCON 254
QS2 = 6518.35*POWER(JPOW)/RD(NF+1,JPOW) GAPCON 255
QG(JPOW) = 0.5*(QS1+QS2) GAPCON 256
QDVRAC(JPOW) = 6518.35*POWER(JPOW)/RD(NFPNC,JPOW) GAPCON 257
TCI = TCB(1,JPOW) GAPCON 258
TCO = TCB(NC,JPOW) GAPCON 259
TAVGC = (TCI+TCO)/2. GAPCON 260
TCC(JPOW) = TERP(TAVGC,AA,AA(1,2),NN) GAPCON 261
ARFO = RD(NF,JPOW) GAPCON 262
ARFI = RD(1,JPOW) GAPCON 263
ARFO = ARFO/RD(NF+NC,JPOW) GAPCON 264
ARFI = ARFI/RD(NC+NF,JPOW) GAPCON 265
AR SIN = RSIN(JPOW)/RD(NF+NC,JPOW) GAPCON 266
FVAT = 0.0 GAPCON 267
DO 230 IV=1,6 GAPCON 268

```

```

IX = IV+1
IF (RSIN(JPOW).GT.RD(1,JPOW)) GO TO 220
FVAT = FVAT+B1(IV,JPOW)*((ARFO**IX-ARFI**IX)/IX)
GO TO 230
220 CONTINUE
FVAT = FVAT+BR(IV,JPOW)*((ARSIN**IX-ARFI**IX)/IX)+B1(IV,JPOW)
+ )*((ARFO**IX-ARSIN**IX)/IX)
230 CONTINUE
VAVGT = FVAT*2./((ARFO**2-ARFI**2))
VAVGT = (VAVGT)+1.8+32.0
CALL OUTPUT (RL,GK(JPOW),G1PG2(JPOW),HGC(JPOW),HGAS(JPOW),HRAD(
+ JPOW),HSOLID(JPOW),JPOW,NX,JCASE,MOLEFR,LPMAX,PFACE(JPOW),PREST
+ (1,VAVGT,XMOTNT,QOVRAC(JPOW),QG(JPOW),TCC(JPOW),TCOL(JPOW),DELR
+ FT(JPOW),TTTT,TSTSR,NF,NC,NFPNC,TFB(1,JPOW),TCB(1,JPOW),RD(1,JP
+ OW),TSTNR,NNOEL,NOELD,EFSTR,NPOW1,SUMFF,SUMM)
240 CONTINUE
250 CONTINUE
DO 260 I=1,NABT
RDOLD(I) = RD(NF+1,I)
260 CONTINUE
270 CONTINUE
IF (NT.LT.JSTART) GO TO 170
IF (NT.GT.JSTART) GO TO 1440
DO 280 INT=1,NABT
RDOLD(INT) = RD(NF+1,INT)
280 CONTINUE
290 CONTINUE
NFPNC = NF+NC
TPLEN = (TINLET+DTEMP-22.)/1.8+273.
VVOIDZ = PI*HVOIDZ**2*LVOIDZ/FNPOW
VOLGAS = PI/4.*(DCI**2-DFS**2)*LF+VVOIDZ
FR38 = 1.0-FR35
FR39 = 1.-FR41-FR40
FRUO2 = 1.-FRPUO2
IF (FRPUO2.LE.1.E-10) FR39 = 0.
DENSIT = FRPUO2*11.46+FRUO2*10.97
PSP(21) = FR39
PSP(22) = FR38
PSP(23) = FRUO2
ANKR = 0.
ANXE = 0.
GAP = DCI-DFS
RCII = RCI
RHO = DENSIT*FRDEN
RHOS = DENSIT*FRDEN2
NFA = NFLX
C
C
C DETERMINE FLUX DEPRESSION
IF (NFLX) 300,310,320
300 RV(1,1) = 0.
RV(2,1) = 1.0
RV(1,2) = 1.0
RV(2,2) = 1.
ZZ = 1.0
YY = 0.0
WW = 0.0

```

GAPCON 269
GAPCON 270
GAPCON 271
GAPCON 272
GAPCON 273
GAPCON 274
GAPCON 275
GAPCON 276
GAPCON 277
GAPCON 278
GAPCON 279
GAPCON 280
GAPCON 281
GAPCON 282
GAPCON 283
GAPCON 284
GAPCON 285
GAPCON 286
GAPCON 287
GAPCON 288
GAPCON 289
GAPCON 290
GAPCON 291
GAPCON 292
GAPCON 293
GAPCON 294
GAPCON 295
GAPCON 296
GAPCON 297
GAPCON 298
GAPCON 299
GAPCON 300
GAPCON 301
GAPCON 302
GAPCON 303
GAPCON 304
GAPCON 305
GAPCON 306
GAPCON 307
GAPCON 308
GAPCON 309
GAPCON 310
GAPCON 311
GAPCON 312
GAPCON 313
GAPCON 314
GAPCON 315
GAPCON 316
GAPCON 317
GAPCON 318
GAPCON 319
GAPCON 320
GAPCON 321
GAPCON 322
GAPCON 323
GAPCON 324
GAPCON 325

	NFA = 2	GAPCON	326
	GO TO 320	GAPCON	327
310	CALL FLUXD (RV,RVE,FR35,FR38,RHO,IHL,DFS,ZZ,YY,*,*,TINLET)	GAPCON	328
	NFA = 11	GAPCON	329
	IF (IRL,GT,0) NFA = IRL+1	GAPCON	330
320	CONTINUE	GAPCON	331
	VOLUME = PI/4.*(DFS+DFS-DVOIDZ+DVOIDZ)*(1.-PRCDH/100.)	GAPCON	332
	VOLCC = VOLUME*CCPIN3	GAPCON	333
	W = RHO*VOLCC	GAPCON	334
	TDHVOL = PRCDH*(DFS**2-DVOIDZ**2)*PI*LFUEL/4./100.	GAPCON	335
	STP = 273./298.	GAPCON	336
	FILMOL = (VOLGAS*FNPOW+VPLENZ+TDHVOL)*STP*CCPIN3*ATMOS/RR	GAPCON	337
	Z1 = PI*DCI*LFUEL*2.54**2	GAPCON	338
	Z2 = W*S*LFUEL/RR	GAPCON	339
	ANH = XH*Z2	GAPCON	340
	ANN = XN*Z2	GAPCON	341
	ANCO = XCO*Z2	GAPCON	342
	REACT = 1./((TINLET+DTEMP/2.)/1.8+273.)	GAPCON	343
	Z3 = TERP(React,RTC,RTCO,5)	GAPCON	344
	Z3 = SQRT(Z3)	GAPCON	345
	Z4 = Z1/FNPOW	GAPCON	346
	XMOLS(NPOW1) = VPLENZ*CCPIN3*ATMOS/RR	GAPCON	347
	FRACTN(1) = FRACHE	GAPCON	348
	FRACTN(2) = FRACAR	GAPCON	349
	FRACTN(3) = FRACH	GAPCON	350
	FRACTN(4) = FRACN	GAPCON	351
	FRACTN(5) = 0.	GAPCON	352
	FRACTN(6) = FRACKR	GAPCON	353
	FRACTN(7) = FRACXE	GAPCON	354
	TOTN = ANN+FRACN*FILMOL	GAPCON	355
	MOL(5) = ANCO	GAPCON	356
	MTUNFT = RHO*VOLCC*12./1.E6	GAPCON	357
	UATWT = 235.*FR35+238.*FR38	GAPCON	358
	FRUU = UATWT/(UATWT+32.)	GAPCON	359
	PUATWT = 239.*FR39+240.*FR40+241.*FR41	GAPCON	360
	FRPUU = PUATWT/(PUATWT+32.)	GAPCON	361
	MTM = MTUNFT*(FRUU*FRUU2+FRPUU*FRPUU2)	GAPCON	362
	C = 0.	GAPCON	363
C	DETERMINE FUEL THERMAL CONDUCTIVITY	GAPCON	364
	IF (NFUEL) 330,350,370	GAPCON	365
330	CONTINUE	GAPCON	366
	DO 340 J=1,7	GAPCON	367
	TEMP = UU2(J,1)	GAPCON	368
	CF(J,1) = TEMP	GAPCON	369
	TK = (TEMP-32.)/1.8+273.	GAPCON	370
	CALL TCW1 (FRDEN,FRPUU2,TK,RK,DKDT,DENCH)	GAPCON	371
	CF(J,2) = 57.803*RK	GAPCON	372
340	CONTINUE	GAPCON	373
	GO TO 380	GAPCON	374
350	CONTINUE	GAPCON	375
	DO 360 J=1,7	GAPCON	376
	TEMP = UU2(J,1)	GAPCON	377
	CF(J,1) = TEMP	GAPCON	378
	TK = (TEMP-32.)/1.8+273.	GAPCON	379
	CALL TCOR (FRDEN,TK,RK,DKDT,DENCH)	GAPCON	380
	CF(J,2) = 57.803*RK	GAPCON	381
360	CONTINUE	GAPCON	382

	GO TO 380	GAPCON	383
370	CONTINUE	GAPCON	384
C		GAPCON	385
380	CALL INPUT (RV,RVE,NFA)	GAPCON	386
	PTEMP = DTEMP	GAPCON	387
C		GAPCON	388
C	*****	GAPCON	389
C		GAPCON	390
C	THE TIME STEP LOOP STARTS HERE.	GAPCON	391
C		GAPCON	392
C	*****	GAPCON	393
C		GAPCON	394
	ISTART = 1	GAPCON	395
	IF (JSTART.GT.ISTART) ISTART = JSTART+1	GAPCON	396
	ISTMT = ISTART	GAPCON	397
	NT = ISTART	GAPCON	398
390	CONTINUE	GAPCON	399
	JSAVE = NT-ISAVE	GAPCON	400
	IF (JSAVE.EQ.KSAVE) ISAVE = NT	GAPCON	401
	NEW = NEW+1	GAPCON	402
	IF (NEW.GT.2) NEW = 1	GAPCON	403
	OLD = OLD+1	GAPCON	404
	IF (OLD.GT.2) OLD = 1	GAPCON	405
	IF (IUP.EQ.0) GO TO 400	GAPCON	406
	READ (5,1500) EXTP,IPROFL,IMF,ITM,TOL,POWR	GAPCON	407
	IF (IPROFL.NE.0) READ (5,1510) (PROFIL(I),I=1,NPDM1)	GAPCON	408
	IF (IMF.NE.0) READ (5,1510) (HF(I),I=2,NAHF)	GAPCON	409
	IF (ITM.EQ.0) GO TO 400	GAPCON	410
	TIMED = TIMEN	GAPCON	411
	TIMEN = TIMEF	GAPCON	412
	TIMEF = TIMEN+TOL	GAPCON	413
	GO TO 410	GAPCON	414
400	TIMED = TIMEN	GAPCON	415
	TIMEN = TIME(NT)	GAPCON	416
	TIMEF = TIME(NT+1)	GAPCON	417
	IF (NT.EQ.NTIME) TIMEF = TIMEN+1.0	GAPCON	418
	POWR = PSEUDO(NT)	GAPCON	419
410	CONTINUE	GAPCON	420
	PEND = POWR	GAPCON	421
	TMN = TIMEN	GAPCON	422
	TMF = TIMEF	GAPCON	423
	TMD = TIMED	GAPCON	424
	KOUNTM = 0	GAPCON	425
	IF (NT.GT.1) ANKED = ANXE	GAPCON	426
	IF (NT.GT.1) ANKRD = ANKH	GAPCON	427
	GO TO 450	GAPCON	428
C	THIS PORTION IS ONLY REACHED IF A POWER STEP MUST BE SUBDIVIDED	GAPCON	429
420	CONTINUE	GAPCON	430
	KOUNTM = KOUNTM+1	GAPCON	431
	KONTMM = KONTMM+1	GAPCON	432
	IF (KONTMM.GT.ITMCUT) GO TO 1450	GAPCON	433
	POWR = POWRLD+1.	GAPCON	434
	IF (IUP.LT.0) POWR = POWRLD-1	GAPCON	435
	TIMED = TIMEN	GAPCON	436
	IF (KOUNTM.EQ.1) TIMEH = TMD	GAPCON	437
	IF (KOUNTM.EQ.1) GO TO 430	GAPCON	438
	NEW = NEW+1	GAPCON	439

	IF (NEW.GT.2) NEW = 1	GAPCON	440
	OLD = OLD+1	GAPCON	441
	IF (OLD.GT.2) OLD = 1	GAPCON	442
430	CONTINUE	GAPCON	443
	ADR = 0.00529	GAPCON	444
	IF (ITRANS.EQ.1) ADR = 1.157074E-6	GAPCON	445
	TIMEN = TIME0+ADR	GAPCON	446
	TIMEF = TIMEN+ADR	GAPCON	447
	IF (LTLOS.EQ.0) GO TO 440	GAPCON	448
	IF (IUP.GT.0.AND.POWER.LE.PEND) GO TO 450	GAPCON	449
	IF (IUP.LT.0.AND.POWER.GE.PEND) GO TO 450	GAPCON	450
440	CONTINUE	GAPCON	451
	TIMEN = TMN	GAPCON	452
	TIMEF = TMF	GAPCON	453
	TIME0 = TM0	GAPCON	454
	POWER = PEND	GAPCON	455
	KOUNTM = 0	GAPCON	456
450	CONTINUE	GAPCON	457
	IUP = 1	GAPCON	458
	IF (POWER.LT.POWERLD) IUP = -1	GAPCON	459
	IF (TIMEN.LT.TIME0.AND.TIMEN.GT.0) STOP = CHECK TIME ARRAYS	GAPCON	460
	TIME(3) = TIMEF	GAPCON	461
	TIME(2) = TIMEN	GAPCON	462
	TIME(1) = TIME0	GAPCON	463
	TTTT = TIMEN	GAPCON	464
	LPMAX = 0	GAPCON	465
	PMAX = 0.0	GAPCON	466
	DO 460 NP=1,NP0W1	GAPCON	467
	POWER(NP) = PROFIL(NP)	GAPCON	468
	PMAX = AMAX1(PMAX,POWER(NP))	GAPCON	469
	IF (POWER(NP).EQ.PMAX) LPMAX = NP	GAPCON	470
460	CONTINUE	GAPCON	471
	ID0(LPMAX) = 1	GAPCON	472
	IF (IPEAK) 470,480,470	GAPCON	473
470	CONTINUE	GAPCON	474
	PAVRG = POWER	GAPCON	475
	GO TO 490	GAPCON	476
480	PKPOWER = POWER	GAPCON	477
490	CALL POWDIS (POWER,PKPOWER,PAVRG,IPEAK,NP0W,AI,DELZA,LFUEL)	GAPCON	478
	DTEMP = PTEMP	GAPCON	479
	IF (MXPOWER.GT.POWER(LPMAX)) DTEMP = PTEMP*POWER(LPMAX)/MXPOWER	GAPCON	480
	IF (NT.NE.1.AND.KOUNTM.EQ.0) TDAYS = TDAYS+TIMEN-TIME0	GAPCON	481
	T = TDAYS*SECDAY	GAPCON	482
	JCASE = NT	GAPCON	483
	IF (T.LE.0.001) GO TO 500	GAPCON	484
	RTEM = 1.367E-5*ALOG10(T/3600.)*4.8E-4	GAPCON	485
	TSINC = 1./RTEM-273.	GAPCON	486
500	CONTINUE	GAPCON	487
	CALL MOVEAA (FRACTN,FXMDL(1,NP0W1),7)	GAPCON	488
	CALL TEMPI (AI,DTEMP,POWER,TINLET,TFB(1,1),TFB(1,NABF),TCB(1,1),TC	GAPCON	489
	+B(1,NABT),NF,NC,NP0W,NABT)	GAPCON	490
C	GAS RELEASE CAN NOT OCCUR WHEN THE CODE DOES A POWER OR TIME STEP	GAPCON	491
	IF (KOUNTM.EQ.0) NY = NX	GAPCON	492
	IF (NT.EQ.1) GO TO 510	GAPCON	493
	C = TIMEN-TIME0	GAPCON	494
	ITRANS = 0	GAPCON	495
	IF (C.LT.0.00347) ITRANS = 1	GAPCON	496

```

      KTRAN = KTRAN+1
      IF (ITRANS.EQ.0) KTRAN = 1
      STEP = 0.1
      SSTEP = C*B.64E5
      JEND = SSTEP
      STST = (SSTEP-FLUAT(JEND))*0.1
      IEND = JEND
      IF (STST.GT.0.) IEND = JEND+1
      IF (C.GT.1.) THX = TIMEN
      IF (C.GT.1.) NX = NY+1
C     GAS RELEASE CAN NOT OCCUR WHEN THE CODE DOES A POWER OR TIME STEP
      IF (KOUNTM.EQ.1) NX = NY
      IF (NY.GT.49) GO TO 1430
      IF (NX.LE.NY.AND.JSTART.EQ.NT-1) GO TO 510
      IF (NX.LE.NY) GO TO 540
510  CONTINUE
      ANXE = 0.
      ANKR = 0.
      DO 530 IPOW=2,NPOW1
        P = (POWER(IPOW)+POWER(IPOW+1))/2
        CALL F16GAS (P,C,X,Z,XEM,RKM)
        IF (NX.EQ.1) GO TO 520
        PXEM = SAVNXE(NX=1,IPOW)
        PRKM = SAVNKR(NX=1,IPOW)
520  IF (NX.EQ.1) PXEM = 0.
        IF (NX.EQ.1) PRKM = 0.
        SAVNXE(NX,IPOW) = XEM+PXEM
        SAVNKR(NX,IPOW) = RKM+PRKM
C     SAVN** VALUES ARE IN UNITS OF MOLES PER FT.,
C     THE FACTOR (DELZA/12) GIVES AN** IN TERMS OF MOLES
        ANKR = ANKR+SAVNKR(NX,IPOW)*DELZA(IPOW)/12.
        ANXE = ANXE+SAVNXE(NX,IPOW)*DELZA(IPOW)/12.
530  CONTINUE
540  CONTINUE
C     GAS **GEN** CAN NOT OCCUR WHEN THE CODE DOES A POWER OR TIME STEP
      IF (KOUNTM.EQ.1) ANXE = ANXEO
      IF (KOUNTM.EQ.1) ANKR = ANKRO
      LFTS = 0
      TAVGX = 0.
      VOLTXS = 0.
      K = 0
      ACELL = .8
      CALL MOVEKA (0,DISPC,NFF1*NABT)
      CALL MOVEKA (0,KAN,NABT)
      LPMAX = LPMAX+1
C
C     *****
C
C     GAS RELEASE LOOP STARTS HERE.
C
C     *****
C
      ISTOP = 11
      IF (KTRAN.GT.1) ISTOP = 2
      DO 1380 K=2,11
        ISTEP = K
        KKZZZ = K

```

```

      GAPCON 497
      GAPCON 498
      GAPCON 499
      GAPCON 500
      GAPCON 501
      GAPCON 502
      GAPCON 503
      GAPCON 504
      GAPCON 505
      GAPCON 506
      GAPCON 507
      GAPCON 508
      GAPCON 509
      GAPCON 510
      GAPCON 511
      GAPCON 512
      GAPCON 513
      GAPCON 514
      GAPCON 515
      GAPCON 516
      GAPCON 517
      GAPCON 518
      GAPCON 519
      GAPCON 520
      GAPCON 521
      GAPCON 522
      GAPCON 523
      GAPCON 524
      GAPCON 525
      GAPCON 526
      GAPCON 527
      GAPCON 528
      GAPCON 529
      GAPCON 530
      GAPCON 531
      GAPCON 532
      GAPCON 533
      GAPCON 534
      GAPCON 535
      GAPCON 536
      GAPCON 537
      GAPCON 538
      GAPCON 539
      GAPCON 540
      GAPCON 541
      GAPCON 542
      GAPCON 543
      GAPCON 544
      GAPCON 545
      GAPCON 546
      GAPCON 547
      GAPCON 548
      GAPCON 549
      GAPCON 550
      GAPCON 551
      GAPCON 552
      GAPCON 553

```

```

CALL MOVEKA (0,KAN,NABF)
IF (NT.EQ.1) RATE = 0.0
E(1) = 0.0
E(2) = 0.0
E(3) = RATE*ANH
IF (NOH.EQ.0) E(3) = 0.0
T1 = SUNT(T)
E(4) = TOTN=4.5662E-8*T1*Z1/GMWT(4)
IF (E(4).LT.0.0) E(4) = 0.0
E(5) = ANCO=Z3*T1*Z1/GMWT(5)
IF (E(5).LT.0.0) E(5) = 0.0
E(6) = ANKH*RATE
E(7) = ANXE*MATE
TOTMA = 0.
DO 550 I=1,7
  TMA(I) = FILMOL*FRACTN(I)+E(I)
  IF (FRACTN(6).GT.0.) TMA(4) = E(4)
  IF (NOH.EQ.0) TMA(3) = 0.0
550  TOTMA = TOTMA+TMA(I)
DO 560 I=1,7
  TMFA(I) = TMA(I)/TOTMA
560  CONTINUE
LFTS = 0
TAVGX = 0.
VOLTXS = 0.
DT = 0.
DELR = 0.
DELMC = 0
CALL MOVEKA (0,DELRP,NPOW1)
DTACEL = 0.8
DO 570 I=1,7
  SUMOLS(I) = FXMOL(I,NPOW1)*XMOLS(NPOW1)
570  CONTINUE
TCOOL(1) = TINLET
PSP(NABF) = POWER(NABF)*.1477/(FRDEN*RF5**2)
C
C *****
C
C AXIAL NODE LOOP STARTS HERE.
C
C *****
C
DO 1120 IPOW=1,NPOW1
  IOSC = 0
  KKNT = 0
  FDENS = FRDEN
  RHOD = DENSIT*FDENS
  JPOW = IPOW+1
  RADH = RADS
  IF (NT.NE.1) GO TO 580
  DVBX(JPOW) = 0.
  DVBLST(JPOW) = 0.0
C
C CALCULATE THE POWER AND TCOOL
580  CONTINUE
  BURNUP(IPOW) = PITI(IPOW)+POWER(IPOW)*C*.001/MTM
  P = 0.5*(POWER(IPOW+1)+POWER(IPOW))

```

```

GAPCON 554
GAPCON 555
GAPCON 556
GAPCON 557
GAPCON 558
GAPCON 559
GAPCON 560
GAPCON 561
GAPCON 562
GAPCON 563
GAPCON 564
GAPCON 565
GAPCON 566
GAPCON 567
GAPCON 568
GAPCON 569
GAPCON 570
GAPCON 571
GAPCON 572
GAPCON 573
GAPCON 574
GAPCON 575
GAPCON 576
GAPCON 577
GAPCON 578
GAPCON 579
GAPCON 580
GAPCON 581
GAPCON 582
GAPCON 583
GAPCON 584
GAPCON 585
GAPCON 586
GAPCON 587
GAPCON 588
GAPCON 589
GAPCON 590
GAPCON 591
GAPCON 592
GAPCON 593
GAPCON 594
GAPCON 595
GAPCON 596
GAPCON 597
GAPCON 598
GAPCON 599
GAPCON 600
GAPCON 601
GAPCON 602
GAPCON 603
GAPCON 604
GAPCON 605
GAPCON 606
GAPCON 607
GAPCON 608
GAPCON 609
GAPCON 610

```

	PSP(JPOW) = P*0.1477/(FRDEN*RES**2)	GAPCON	611
	MMF = MF(JPOW)	GAPCON	612
	TCOOL(JPOW) = TINLET+DTEMP*AI(JPOW)	GAPCON	613
	CALL MOVEKA (0,DELRP,NPOW1)	GAPCON	614
	DELGD = DELGDM(JPOW)	GAPCON	615
	IF (PFACE(JPOW).GT.0.) GO TO 600	GAPCON	616
	IF (IRELOC) 590,600,590	GAPCON	617
590	CONTINUE	GAPCON	618
C		GAPCON	619
C	RELOCATION CALCULATION	GAPCON	620
	CALL RELOC (DELGD,GAP,BURNUP(JPOW),P,IRELOC)	GAPCON	621
	DELGD = AMAX1(DELGD,DELGDM(JPOW))	GAPCON	622
600	CONTINUE	GAPCON	623
C		GAPCON	624
C	DENSIFICATION CALCULATION	GAPCON	625
	IF (IDENSF) 610,620,610	GAPCON	626
610	CALL DENSE (DELPI(JPOW),DFS,RHOD,RHOS,BURNUP(JPOW))	GAPCON	627
	FDENS = RHOD/DENSIT	GAPCON	628
C	DENSE SUBROUTINE RETURNS A NEGATIVE VALUE FOR THE	GAPCON	629
C	CHANGE IN FUEL RADIUS DUE TO DENSIFICATION (DELPI)	GAPCON	630
C		GAPCON	631
620	CONTINUE	GAPCON	632
630	CONTINUE	GAPCON	633
	KKKNT = 0	GAPCON	634
	IF (KKNT.EQ.1) KKKNT = 1	GAPCON	635
	DBU = 0.	GAPCON	636
	IF (NT.NE.1) DBU = .001*P*C/MTM	GAPCON	637
	ITER = 0	GAPCON	638
	IF (NFUEL) 640,650,640	GAPCON	639
640	TM = TM+5.*FRPUQ2=32.*BURNUP(JPOW)/10000.	GAPCON	640
C		GAPCON	641
C	*****	GAPCON	642
C		GAPCON	643
C	THIS IS THE START OF THE GAP CONDUCTANCE ITERATION LOOP.	GAPCON	644
C		GAPCON	645
C	*****	GAPCON	646
C		GAPCON	647
650	CONTINUE	GAPCON	648
	ITER = ITER+1	GAPCON	649
	IF (ITER.GT.1) GO TO 660	GAPCON	650
	TEND = TIMEN	GAPCON	651
	POWEREND = POWER(JPOW)	GAPCON	652
660	CONTINUE	GAPCON	653
	IF (ITRANS.NE.1) GO TO 670	GAPCON	654
C	SET-UP FOR A TRANSIENT STEP	GAPCON	655
	TIMEN = TIMEO+(TEND-TIMEO)*FLOAT(ITER)/FLOAT(IEND)	GAPCON	656
	POWER(JPOW) = OLDPOW(JPOW)+(POWEREND-OLDPOW(JPOW))*FLOAT(ITER)/	GAPCON	657
	FLOAT(IEND)	GAPCON	658
	IF (ITER.EQ.IEND) TIMEN = TEND	GAPCON	659
	IF (ITER.EQ.IEND) POWER(JPOW) = POWEREND	GAPCON	660
	IF (ITER.EQ.IEND.AND.JEND.NE.IEND) STEP = STST	GAPCON	661
	STEPS = STEP	GAPCON	662
	ITRCNT = ITRCNT+1	GAPCON	663
	IF (ITRCNT.GT.ITMCNT) GO TO 1450	GAPCON	664
670	CONTINUE	GAPCON	665
	IF (KKKNT.EQ.0) GO TO 690	GAPCON	666
	DIFF = (RD(NF,JPOW)-RD(1,JPOW))/FLOAT(NF-1)	GAPCON	667

```

C      CORRECT RD FOR THE FORMATION OF A VOID.
      DO 680 JJ=2,NF
      RD(JJ,JPOW) = RD(1,JPOW)+FLOAT(JJ-1)*DRFF
680    CONTINUE
690    CONTINUE
      QS1 = 6518.35*POWER(JPOW)/RD(NF,JPOW)
      QS2 = 6518.35*POWER(JPOW)/RD(NF+1,JPOW)
      IF (ITRANS.EQ.1) GO TO 700
      HMG = HGC(JPOW)
      QUVRAS = 0.5*(QS1+QS2)
      DTB = DT
      IF (NT.EQ.1.AND.K.EQ.2.AND.ITER.EQ.1) DTB = QUVRAS/1000.
      IF (NT.GT.1.AND.ITER.EQ.1) DTB = QUVRAS/HGC(JPOW)
700    CONTINUE
      HMG = HGC(JPOW)
C      GET BOUNDARY TEMPERATURES
      CALL TEMP2 (AI(JPOW),HF(JPOW),POWER(JPOW),RD(1,JPOW),TCB(1,J
+      POW),TFB(1,JPOW),DTEMP,ZZ,YY,WW,FDENS,DTB,NF,NC,NFPNC,TINLET
+      ,BI(1,JPOW),NFUEL,FRPU02,JPOW,RSIN(JPOW),TSINC,FRSIN,BH(1,JP
+      OW),C,NCLAD)
      USIN = 2.*RSIN(JPOW)
      RVOID = SQRT(RSIN(JPOW)*RSIN(JPOW)*(FRSIN+FDENS)/FRSIN+RVOID
+      Z*RVOIDZ*FDENS/FRSIN)
      IF (RVOID.LT.RD(1,JPOW)) RVOID = RD(1,JPOW)
      RD(1,JPOW) = RVOID
      DVOID = 2.*RVOID
      IF (ITRANS.EQ.0.AND.K.NE.ISTOP.AND.PRCOM.LE.0) GO TO 730
      VEX = 0.0
      RINT = RD(1,JPOW)
      RAD = (RD(NF,JPOW)-RD(1,JPOW))/50.
      RN = RD(NFPNC,JPOW)
      DO 720 IAD=1,50
      Y1 = (RINT+RAD*FLOAT(IAD=1))/RN
      Y2 = (RINT+RAD*FLOAT(IAD))/RN
      RVOL(IAD) = Y1
      VATF = 0.0
      TEST = Y2*RN
      YY2 = Y2
      YY1 = Y1
      DO 710 IV=1,6
      IX = IV+1
      YY1 = YY1*Y1
      YY2 = YY2*Y2
      BAD = BI(IV,JPOW)
      IF (RSIN(JPOW).GT.TEST) BAD = BR(IV,JPOW)
      VATF = VATF+BAD*(YY2-YY1)/FLOAT(IX)
710    CONTINUE
      ITVOL(IAD) = VATF
      IF (ITRANS.EQ.0.AND.K.NE.ISTOP) GO TO 720
C      CALCULATE STOKED ENERGY
      VIAD = VATF*2./(Y2**2-Y1**2)
      CALL HCAP (VIAD,CDT,CPNOM)
C      HCAP RETURNS INTEGRAL C DT IN JOULES PER GRAM.
      VIAD = FDENS*70.71*(Y2*Y2-Y1*Y1)*RN*RN*PI*CDT
C      THE CONSTANT 70.71 IS 2.54**2*10.96
C      VEX IS THE STOKED ENERGY IN THE FUEL RELATIVE TO TREF(70 F)
C      IN JOULES PER CM

```

```

      IF (RSIN(JPOW),GT,TEST) VIAD = VIAD*FRSIN/FRDEN
      VEX = VEX+VIAD
      CONTINUE
720  ESTORI(JPOW) = VEX
      CONTINUE
730  DO 740 I=1,NF
      TFR(I,IPOW) = 0.5*(TFB(I,IPOW)+TFB(I,JPOW))
740  CONTINUE
      GG(JPOW) = GQVRAS
      TFS = TFB(NF,JPOW)
      TCI = TCB(1,JPOW)
      TCU = TCB(NC,JPOW)
      CALL EXPAND (NF,RD(1,JPOW),BI(1,JPOW),BR(1,JPOW),DELRT(JPOW)
      ,RCU,RSIN(JPOW))
      CALL SWELL (TFB(1,JPOW),RD(1,JPOW),DRU,DVBX(JPOW),DELVB(JPOW)
      ,KAN(IPOW),ICHNG,NF,VOLUME,PI,FDENS,PHCDH)
      DVT = DELVB(JPOW)+DVBX(JPOW)
      DELRB(JPOW) = DVT/(3.*PI*RD(NF,JPOW))
      TAVGC = (TCI+TCU)/2.
      TCC(JPOW) = TERP(TAVGC,AA,AA(1,2),NN)
      IF (NT,GT,1) GO TO 750
      ALPHAQ = TERP(TAVGC,AA,AA(1,6),NN)
      DELRCI(JPOW) = RCI*ALPHAQ*(TAVGC-77.)
      DELR = DELRT(JPOW)+DELRB(JPOW)
750  DELRFT(JPOW) = DELRT(JPOW)+DELRB(JPOW)+DELGD+DELPI(JPOW)
      DELRFT(1) = DELRFT(2)
      DELRCT(JPOW) = RDELST(JPOW)-RCII
      IF (NT,EQ,1,AND,K,EQ,2) DELRCT(JPOW) = DELRCI(JPOW)
      DELRCT(1) = DELRCT(2)
      QUVKAC(JPOW) = 6518.35*POWER(JPOW)/RD(NF+NC,JPOW)
      IF (NT,GT,1,OR,K,GT,2) GO TO 760
      TH = GAP/2.+DELRCT(JPOW)*DELRFT(JPOW)
      GO TO 770
760  TH = RDELST(JPOW)-RFS+DELRFT(JPOW)
770  CONTINUE
      CRUF = 3.69*(RDUF+RUDC)
      ZZ1 = 0.
      IF (TH,GT,CRUF,OR,DELGD,LE,0) GO TO 800
      ZZ1 = DELGD-TH
      IF (ZZ1) 800,790,780
780  IF (TH,GT,0.) GO TO 800
790  TH = CRUF
800  CONTINUE
      AF = RFS+DELRFT(JPOW)
      AC = RCI+DELRG
      THR = TH+CRUF
      VRUF = PI*DFS*CRUF*DELZA(IPOW)
      TC1K = (TCI-32.)/1.8+273.

C
C   CALCULATE HOW MUCH SORBED GAS CAN REACT WITH THE CLAD
      RTCIK = 1/TCIK
      REACN = SQRT(2.082E-15)
      REACU = TERP(RTCIK,RTC,RTCO(1),5)
      REACN = SQRT(REACU)
      GRECN = REACN*T1*Z4
      GRECN = REACN*T1*Z4
      MREACN(IPOW) = GRECN/GHWT(4)

```

```

GAPCON 725
GAPCON 726
GAPCON 727
GAPCON 728
GAPCON 729
GAPCON 730
GAPCON 731
GAPCON 732
GAPCON 733
GAPCON 734
GAPCON 735
GAPCON 736
GAPCON 737
GAPCON 738
GAPCON 739
GAPCON 740
GAPCON 741
GAPCON 742
GAPCON 743
GAPCON 744
GAPCON 745
GAPCON 746
GAPCON 747
GAPCON 748
GAPCON 749
GAPCON 750
GAPCON 751
GAPCON 752
GAPCON 753
GAPCON 754
GAPCON 755
GAPCON 756
GAPCON 757
GAPCON 758
GAPCON 759
GAPCON 760
GAPCON 761
GAPCON 762
GAPCON 763
GAPCON 764
GAPCON 765
GAPCON 766
GAPCON 767
GAPCON 768
GAPCON 769
GAPCON 770
GAPCON 771
GAPCON 772
GAPCON 773
GAPCON 774
GAPCON 775
GAPCON 776
GAPCON 777
GAPCON 778
GAPCON 779
GAPCON 780
GAPCON 781

```



```

      MREACO(IPOW) = GRECO/GMWT(5)
      YGAS = (TCI+TFS)/2.
      ABTGAS = (TGAS-32.)/1.8+273.
      ABTGS = ABTGAS
      IF (JPOW.GT.2.OR.ITER.GT.1) GO TO 900
      DO 890 IG=1,8
      ABTGAS = 400.+FLUAT(IG-1)*200.
C      CALCULATE THERMAL CONDUCTIVITY OF THE GAS
C
      DO 810 I=1,7
      TKE = ABTGAS/EKLJ(I)
      OMEGA(I) = TEMP(TKE, TABLE, TABLE(1,2), 80)
      CONTINUE
810      DO 820 I=1,7
      VISCUS(I) = 2.67E-5*SQRT(ABTGAS*GMWT(I))/(OMEGA(I)*SIG
      +      LJ(I)**2)
      CONTINUE
820      DO 830 I=1,7
      DO 830 J=1,7
      PHICAP(I,J) = 1./SQRT(8.)/(SQRT(1.+GMWT(I)/GMWT(J))
      +      )*(1.+SQRT(VISCUS(I)/VISCUS(J))*((GMWT(J)/GMWT(I))*
      +      *.25)**2)
      CONTINUE
830      DO 840 I=1,7
      IF (I.EQ.3) GO TO 840
      IF (I.EQ.4) GO TO 840
      IF (I.EQ.5) GO TO 840
      CON(I) = 1.9891E-4*(SQRT(ABTGAS/GMWT(I)))/(SIGLJ(I)**2
      +      *OMEGA(I))
      CONTINUE
840      CSUBP(3) = 6.947-0.20E-3*ABTGAS+4.E08E-7*ABTGAS**2
      CSUBP(4) = 6.524+1.25E-3*ABTGAS-1.E-9*ABTGAS**2
      CSUBP(5) = CSUBP(4)
      DO 850 I=3,5
      CON(I) = (CSUBP(I)+1.25*1.987)*VISCUS(I)/GMWT(I)
850      CONTINUE
      GASKON = 0.
      VISMIX = 0.
      DO 870 I=1,7
      DENOM = 0.
      DO 860 J=1,7
      DENOM = DENOM+TMFA(J)*PHICAP(I,J)
      CONTINUE
860      GASKON = GASKON+TMFA(I)*CON(I)/DENOM
      VISMIX = VISMIX+TMFA(I)*VISCUS(I)/DENOM
870      CONTINUE
      GK(JPOW) = GASKON*241.9
      GKTRP(IG) = GK(JPOW)
      ABTCV = (TFB(1,IPOW)-32.)/1.8+273.
      EMMIX = 0.
      DO 880 I=1,7
      EMMIX = EMMIX+TMFA(I)*GMWT(I)
      CONTINUE
880      ABT1 = ABTGAS
C
C      CALCULATE THE TEMPERATURE JUMP DISTANCE. NOTE THE AVERAGE GAS
C      TEMP. IS USED, SINCE THE DEPENDENCE OF G ON T IS QUITE LINEAR.

```

```

C
      P1 = TMFA(1)*PRESTO+68947.
      P2 = TMFA(2)*PRESTO+68947.
      P3 = TMFA(3)*PNESTU+68947.
      P4 = TMFA(4)*PNESTU+68947.
      P5 = TMFA(5)*PNESTU+68947.
      P6 = TMFA(6)*PNESTU+68947.
      P7 = TMFA(7)*PNESTU+68947.

C
C   ACCUMULATION COEFFICIENTS FOR HE AND H2 BASED ON DATA OF ULLMAN
C   ET AL., CORR. FOR OTHER GASES ESTIMATED FROM MOLECULAR WEIGHT.
C
      ACHE1 = 0.025+2.5E-004BT1
      ACHE1 = 0.744+2.8E-004BT1
      GLOP = (ACHE1+ACHE1)/120.0
      HINT = ACHE1+0.8*GLOP
      ACAN1 = 30.0+GLOP*HINT
      ACN1 = 2.0+GLOP*HINT
      ACN1 = 20.0+GLOP*HINT
      ACCN1 = 20.0+GLOP*HINT
      ACN1 = 84.0+GLOP*HINT
      D11 = (4.0+BT1)/00.5
      D12 = (40.0+BT1)/00.5
      D13 = (2.0+BT1)/00.5
      D14 = (20.0+BT1)/00.5
      D15 = (20.0+BT1)/00.5
      D16 = (84.0+BT1)/00.5
      D17 = (151.3+BT1)/00.5
      SUM1 = P1+ACHE1/D11+ACAN1*P2/D12+APM1*P3/D13+ACN1*P4/D14+
      ACCN1*P5/D15+ACN1*P6/D16+ACHE1*P7/D17
      G1 = 2870.0+648KUN/SUM1
      G1P62(JPUN) = G1+1.417

C   1.417 = (2+1.8)/2.54
      GJTRP(16) = G1P62(JPUN)
      ABTTRP(16) = ABT648
      CONTINUE

490
900
      CONTINUE
      ABT648 = ABT68
      GK(JPUN) = TEMP(ABT68,ABTTRP,GKTRP,0)
      G1P62(JPUN) = TEMP(ABT68,ABTTRP,GJTRP,0)
      IF (THK) 030,030,010

C
910
      IL = NOELF+1
      TOTSP = 0.0
      TOTC = 0.0
      DO 920 IJ=IL,NOELD
        TOTC = TOTC+TSTSK(2,IJ,JPUN)
        TOTSP = TOTSP+TSTSK(2,IJ,NAHT)
920
      CONTINUE
C   UNLOCKING DECISION
      IF (TOTC.GT.TOTSP.AND.KANOLD(IPUN).EQ.1) KAN(IPUN) = 1
      IF (POWER(JPUN).GE.OLDPON(JPUN).AND.KANOLD(IPUN).EQ.1) GO TO
+
      930
      IF (POWER(JPUN).GE.OLDPON(JPUN).AND.KANOLD(JPUN).EQ.1) GO TO
+
      930
      IF (KAN(IPUN).EQ.1) GO TO 930
C   CALCULATION OF HGAS

```

```

GAPCON 839
GAPCON 840
GAPCON 841
GAPCON 842
GAPCON 843
GAPCON 844
GAPCON 845
GAPCON 846
GAPCON 847
GAPCON 848
GAPCON 849
GAPCON 850
GAPCON 851
GAPCON 852
GAPCON 853
GAPCON 854
GAPCON 855
GAPCON 856
GAPCON 857
GAPCON 858
GAPCON 859
GAPCON 860
GAPCON 861
GAPCON 862
GAPCON 863
GAPCON 864
GAPCON 865
GAPCON 866
GAPCON 867
GAPCON 868
GAPCON 869
GAPCON 870
GAPCON 871
GAPCON 872
GAPCON 873
GAPCON 874
GAPCON 875
GAPCON 876
GAPCON 877
GAPCON 878
GAPCON 879
GAPCON 880
GAPCON 881
GAPCON 882
GAPCON 883
GAPCON 884
GAPCON 885
GAPCON 886
GAPCON 887
GAPCON 888
GAPCON 889
GAPCON 890
GAPCON 891
GAPCON 892
GAPCON 893
GAPCON 894
GAPCON 895

```

```

      HGAS(JPOW) = GK(JPOW)/(TH+GIPG2(JPOW))*12.
      HSOLID(JPOW) = 0.
      KNCN(JPOW) = 0
      PFACE(JPOW) = 0.0
      GO TO 970
930  CONTINUE
      TEST = ABS(POWRLO-POWR)
      LTLOS = 1
C    DECISION ON SUBDIVIDING THE POWER STEP
      IF (TEST.GT.1.5) GO TO 420
      KNCN(JPOW) = 1
C    CALCULATION OF HSOLID
      PFACE(JPOW) = ABS(PRESS(IPOW))
      IF (KNCN(IPOW).EQ.0.AND.KNCN(JPOW).EQ.1) PFACE(JPOW) = ABS(P
+      RESS(JPOW))
      JPOW1 = JPOW+1
      IF (KNCN(IPOW).EQ.1.AND.KNCN(JPOW1).EQ.1) PFACE(JPOW) = ABS(
+      (PRESS(IPOW)+PRESS(JPOW))/2.)
      FK = KKS*57.803
      CK = TERP(TCI,AA,AA(1,2),NN)
      CFFACE = PFACE(JPOW)/14.223
      CEE = 3.69*EXP(-0.00124+CFFACE)
      HMEYER = TERP(TCI,AA,AA(1,7),NN)
      PREL = CFFACE/HMEYER
      RKM = 2.*FK*CK/(FK+CK)
      R1 = ROUF*1.0E+6
      R = (ROUC**2+ROUF**2)**0.5
C    BRANCH TO PROPER CALL OF HSOLID BASED ON PREC, AS PER BNWL-1894
      IF (PREL.GT.0.010) GO TO 940
      IF (PREL.LT.0.0001) GO TO 950
      HSOLID(JPOW) = 5.*RKM/(R*EXP(5.738-0.5285*ALOG(R1)))*0.01
      GO TO 960
940  HSOLID(JPOW) = 5.*RKM*PREL/(R*EXP(5.738-0.528*ALOG(R1)))
      GO TO 960
950  HSOLID(JPOW) = 5.*RKM*PREL**0.5/(R*EXP(5.738-0.5284*ALOG(R1)
+      ))
960  THC = CEE*(ROUF+ROUC)+GIPG2(JPOW)-5.5E-5
      IF (THC.LT.2.E-5) THC = 2.E-5
      HGAS(JPOW) = GK(JPOW)/THC*12.
970  CONTINUE
      KAN(IPOW) = 0.0
      IF (KNCN(IPOW).EQ.1.AND.KNCN(JPOW).EQ.1) KAN(IPOW) = 1
      IF (KAN(IPOW).EQ.1) IDG(JPOW) = 1
      IF (KAN(IPOW).EQ.1.AND.IDG(JPOW).GT.1) IDG(IPOW-1) = 1
      TFSR = TFS+460.
      TCISR = TCI+460.
      HRAD(JPOW) = .1713E+8/(1./EPSIF+AF/AC*(1./EPSIC-1.))*(TFSR**
+      2+TCISR**2)*(TFSR+TCISR)
      HGC(JPOW) = HSOLID(JPOW)+HGAS(JPOW)+HRAD(JPOW)
      IF (ITRANS.EQ.1) GO TO 1020
      DTC = QUVRAS/HGC(JPOW)
C    BEGIN CONVERGENCE OF DT
      DELTA = DTC-DTB
      DT = DTB
      DDZZ = ABS(DELTA)
      DELDT = DT-DTOLD
      DTOLD = DT

```

```

      GAPCON 898
      GAPCON 897
      GAPCON 898
      GAPCON 899
      GAPCON 900
      GAPCON 901
      GAPCON 902
      GAPCON 903
      GAPCON 904
      GAPCON 905
      GAPCON 906
      GAPCON 907
      GAPCON 908
      GAPCON 909
      GAPCON 910
      GAPCON 911
      GAPCON 912
      GAPCON 913
      GAPCON 914
      GAPCON 915
      GAPCON 916
      GAPCON 917
      GAPCON 918
      GAPCON 919
      GAPCON 920
      GAPCON 921
      GAPCON 922
      GAPCON 923
      GAPCON 924
      GAPCON 925
      GAPCON 926
      GAPCON 927
      GAPCON 928
      GAPCON 929
      GAPCON 930
      GAPCON 931
      GAPCON 932
      GAPCON 933
      GAPCON 934
      GAPCON 935
      GAPCON 936
      GAPCON 937
      GAPCON 938
      GAPCON 939
      GAPCON 940
      GAPCON 941
      GAPCON 942
      GAPCON 943
      GAPCON 944
      GAPCON 945
      GAPCON 946
      GAPCON 947
      GAPCON 948
      GAPCON 949
      GAPCON 950
      GAPCON 951
      GAPCON 952

```

```

          IF (ITER.EQ.1) GO TO 1000
          IF (NT.EQ.1) GO TO 980
          IF (KKKNT.EQ.1) GO TO 1050
          KKNT = 0
          IF (DDZZ.LT.10..AND.K.EQ.1STOP.AND.FHSIN.GT.FDENS) KKNT = 1
          IF (KKNT.EQ.1) GO TO 630
980      CONTINUE
          IF (ABS(DELTA).LE.10.) GO TO 1050
          IF (ITER.GT.35) GO TO 1040
C
C      TEST FOR SIMILAR SLOPES IN THE CONVERGENCE APPROACH AND
C      IF THE LAST TWO ITERATIONS HAVE HAD SIMILAR VALUES, USE
C      A DIFFERENT ITERATION SCHEME. GAPCON CAN ENCOUNTER
C      TWO BASIC TYPES AND THEY ARE (1) OSCILATING (UNDERDAMPED),
C      OR (2) OVERDAMPED.
          IF (IOSC.EQ.1) GO TO 990
          IF (DELTA*DELTA0.LT.0.) IOSC = 1
          IF (DELTA*DELTA0.LE.0.) GO TO 990
C
C      SINCE THE LAST TWO ITERATIONS HAS SIMILAR SLOPES,
C      I.E., BOTH POSITIVE OR NEGATIVE, USE AN EFFECTIVE ACCELERATION
C      PARAMETER OF 1.4.
          DT = DT0+0.4*DELTA
          IF (DT.GT.0.) GO TO 1010
          IOSC = 1
          DT = DT0
C
C      THE SOLUTION IS OSCILATING ABOUT WHAT SHOULD BECOME THE
C      CONVERGED SOLUTION AND THE CONVERGENCE MUST BE CONSTRAINED
C      TO FORCE A FASTER SOLUTION. THE EFFECTIVE ACCELERATION
C      PARAMETER USED IS REDUCED TO HALF OF ITS ORIGINAL
C      VALUE TO DAMPEN THE OSCILLATION.
990      CONTINUE
          IF (ITER.LE.5) GO TO 1000
          IF (ABS(DELOTD*DELDT).LT.1.2) DTACEL = 0.5*DTACEL
          IF (DTACEL.LT.0.05) DTACEL = 0.05
1000     CONTINUE
          DT = DT+DTACEL*DELTA
          IF ((DT+DT0).LT.1000.AND.ITER.LE.5) DTACEL = 0.5
1010     CONTINUE
          DELTA0 = DELTA
          DELDT0 = DELDT
1020     CONTINUE
          IF (ITHANS.EQ.0) GO TO 1030
          DTB = TFS*TCI
          IF (ITER.EQ.IEND) GO TO 1040
          IF (DTB.LT.0.0) STOP #NEG.DTB#
1030     CONTINUE
C
C      *****
C
C      THIS IS THE END OF THE GAP CONDUCTANCE ITERATION LOOP.
C
C      *****
          GO TO 650
1040     CONTINUE

```

GAPCON 953
 GAPCON 954
 GAPCON 955
 GAPCON 956
 GAPCON 957
 GAPCON 958
 GAPCON 959
 GAPCON 960
 GAPCON 961
 GAPCON 962
 GAPCON 963
 GAPCON 964
 GAPCON 965
 GAPCON 966
 GAPCON 967
 GAPCON 968
 GAPCON 969
 GAPCON 970
 GAPCON 971
 GAPCON 972
 GAPCON 973
 GAPCON 974
 GAPCON 975
 GAPCON 976
 GAPCON 977
 GAPCON 978
 GAPCON 979
 GAPCON 980
 GAPCON 981
 GAPCON 982
 GAPCON 983
 GAPCON 984
 GAPCON 985
 GAPCON 986
 GAPCON 987
 GAPCON 988
 GAPCON 989
 GAPCON 990
 GAPCON 991
 GAPCON 992
 GAPCON 993
 GAPCON 994
 GAPCON 995
 GAPCON 996
 GAPCON 997
 GAPCON 998
 GAPCON 999
 GAPCON 1000
 GAPCON 1001
 GAPCON 1002
 GAPCON 1003
 GAPCON 1004
 GAPCON 1005
 GAPCON 1006
 GAPCON 1007
 GAPCON 1008
 GAPCON 1009

```

1050      CONTINUE                                GAPCON 1010
C                                                GAPCON 1011
C                                                GAPCON 1012
C      CALCULATE FISSION GAS RELEASE          BEYER-HANN MODEL  GAPCON 1013
      IF (NX,LE,NY) GO TO 1060                    GAPCON 1014
      CALL GASREL (BURNUP(IPOW),FDENS,IGAS,DELZA(IPOW),LSTHIF(IPOW) GAPCON 1015
      ,NF,NFFID,NX,RD(1,JPOW),HDOT(1,IPOW),SAVNKR(1,IPOW),SAVNXE( GAPCON 1016
      ,1,IPOW),TFR(1,IPOW),SUMNKR(1,IPOW),SUMNXE(1,IPOW),RV,NFA,IFL GAPCON 1017
      ,AG(IPOW))                                GAPCON 1018
1060      CONTINUE                                GAPCON 1019
      UTACEL = 0.8                                GAPCON 1020
      DELRC = (DELRCT(JPOW)+DELRCT(IPOW))/2.        GAPCON 1021
      DELRF = (DELRFT(JPOW)+DELRFT(IPOW))/2.        GAPCON 1022
      V1 = PI*((DELRC+RCI)**2-(DELR+RFS)**2)*DELZA(IPOW) GAPCON 1023
      V2 = PI*((RD(1,IPOW)+RD(1,JPOW))/2.)**2*DELZA(IPOW) GAPCON 1024
      IF (V1,LE,VREF) V1 = VREF                    GAPCON 1025
      VOLTX(IPOW) = (V1/ABTGAS)+(V2/ABTCV)          GAPCON 1026
      IF (THH,GT,CHUF) PFACE(JPOW) = 0.0           GAPCON 1027
      IF (FUENS,GT,FHSIN) FRSIN = FOENS             GAPCON 1028
      IF (PHCDH,LE,0) GO TO 1110                   GAPCON 1029
C                                                GAPCON 1030
C      CALCULATE DISH EFFECTS AND AXIAL THERMAL EXPANSION  GAPCON 1031
      TRADM = TERP(RADH,RD(1,IPOW),TFH(1,IPOW),NF)  GAPCON 1032
      RPLAS = TERP(TPLAS,TFH(1,JPOW),RD(1,JPOW),NF) GAPCON 1033
      IF (TRADM,LE,TPLAS) GO TO 1070               GAPCON 1034
      RADH = RPLAS                                GAPCON 1035
      TRADM = TPLAS                                GAPCON 1036
1070      CONTINUE                                GAPCON 1037
      TRADHC = (TRADM-32.)/1.6                      GAPCON 1038
      CST1 = 2.896E-9*(TRADHC**2-25**2)+6.797E-6*(TRADHC-25.) GAPCON 1039
      DELTL = CST1*DELZA(IPOW)                     GAPCON 1040
      DHVOL(IPOW) = 0.0                             GAPCON 1041
      DO 1080 I=1,50                                GAPCON 1042
        RTEST = RVOL(I)*RD(NFPNC,JPOW)             GAPCON 1043
        IF (RTEST,GT,RADH) GO TO 1090              GAPCON 1044
      CONTINUE                                       GAPCON 1045
1080      CONTINUE                                GAPCON 1046
1090      M = I                                       GAPCON 1047
      IF (M,GT,50) M = 50                           GAPCON 1048
      VATF = 0.0                                     GAPCON 1049
      DO 1100 I=1,M                                  GAPCON 1050
        VATF = VATF+TTVOL(I)                       GAPCON 1051
1100      CONTINUE                                GAPCON 1052
      WD = (TDMVOL*DELZA(IPOW)/(DELZA(1)+LFUEL))/(PI*RADH**2) GAPCON 1053
      TAVGDC = VATF*2./((RVOL(M)**2-RVOL(1)**2))    GAPCON 1054
      CSTD = 2.896E-9*(TAVGDC**2-25**2)+6.797E-6*(TAVGDC-25.) GAPCON 1055
      DELTLD = CSTD*(DELZA(IPOW)-WD)               GAPCON 1056
      ABTDS = TAVGDC+273.                            GAPCON 1057
      DHVOL(IPOW) = (PI*(RADH**2-RVOLDZ**2)*(DELTLD+WD-DELTLD))/(AB GAPCON 1058
      TDS)                                           GAPCON 1059
      CONTINUE                                       GAPCON 1060
1110      IF (DHVOL(IPOW),LE,0) DHVOL(IPOW) = 0      GAPCON 1061
      VOLTXS = VOLTXS+VOLTX(IPOW)+DHVOL(IPOW)       GAPCON 1062
      IF (K,EW,ISTOP) DELGDM(JPOW) = DELGDM        GAPCON 1063
1120      CONTINUE                                GAPCON 1064
      IF (K,NE,ISTOP) GO TO 1170                   GAPCON 1065
      DO 1160 IPOW=1,NPOM                           GAPCON 1066

```

```

      JPOW1 = IPOW+2
      JPOW = IPOW+1
      DO 1150 J=2,NF
        RD(J,JPOW) = RD(J,JPOW)-DISPF(J,JPOW)
        DHF = (RFS-RD(1,JPOW))/FLOAT(NF-1)
        FAC = FLOAT(J-1)*DHF/(RFS-RD(1,JPOW))
        IF (KAN(IPOW).EQ.1) GO TO 1130
        DISPF(J,JPOW) = FAC*DELHFT(JPOW)
        GO TO 1140
      1130 MF = NF+1
        RD(J,JPOW) = FAC*(RD(MF,JPOW)-RD(1,JPOW)-CRUF)+RD(1,JPOW)
        RD(J,IPOW) = FAC*(RD(MF,IPOW)-RD(1,IPOW)-CRUF)+RD(1,IPOW)
        DISPF(J,JPOW) = (RD(MF,JPOW)-RFS-CRUF)*FAC
        DISPF(J,IPOW) = (RD(MF,IPOW)-RFS-LRUF)*FAC
        GO TO 1150
      1140 RD(J,JPOW) = RD(J,JPOW)+DISPF(J,JPOW)
      1150 CONTINUE
      1160 CONTINUE
      1170 CONTINUE
      CALL TEMP3 (NF,NC,NT,OLD,NEW,NABT,NFFID,NOELD,NTAED,TELM,THREF,T
+ INLET,TFB,TDELE,RD,TCH,TW,TWDEL,B1,NART,NARF,RSIN,BR,KAN)
      IF (IPOW.GT.NPOW) IPOW = NPOW
      TAVGXX(ISTEP+1) = TAVGX/FNPOW
      VPLENT = VPLENZ+(PI*RCI**2)*(SUMH-SUMHF)
      IF (VPLENT.LT.0) VPLENT = 0
      VOLTXS = VOLTXS+VPLENT/TPLENA
C
C   CALCULATE HOW MUCH SORBED GAS REMAINS IN THE PIN
      MOL(4) = TMA(4)
      TXMOLS(4) = 0.0
      TXMOLS(5) = 0.0
      DO 1180 IPOW=1,NPOW
        TXMOLS(4) = TXMOLS(4)+MREACN(IPOW)
        TXMOLS(5) = TXMOLS(5)+MREACO(IPOW)
      1180 CONTINUE
      MOL(4) = MOL(4)-TXMOLS(4)
      MOL(5) = MOL(5)-TXMOLS(5)
      IF (MOL(4).LE.0) MOL(4) = 0.0
      IF (MOL(5).LE.0) MOL(5) = 0.0
C
C   CALCULATE MOLE FRACTIONS OF GAS IN THE FUEL PIN.
      MOL(1) = TMA(1)
      MOL(2) = TMA(2)
      MOL(3) = TMA(3)
      MOL(6) = FRACKR*FILMOL
      MOL(7) = FRACXE*FILMOL
      DO 1190 I=1,NPOW1
        SUMNKR(NX,I) = AMAX1(SUMNKR(NX,I),SUMNKR(NX-1,I))
        SUMNXE(NX,I) = AMAX1(SUMNXE(NX,I),SUMNXE(NX-1,I))
        MOL(6) = MOL(6)+SUMNKR(NX,I)
        MOL(7) = MOL(7)+SUMNXE(NX,I)
      1190
      RL IS THE CUMULATIVE GAS RELEASE FRACTION
      RL = 0.0
      IF (ANXE.GT.0.0) RL = (MOL(6)+MOL(7))/(ANKR+ANXE)
      XMOTOT = 0
      DO 1200 I=1,7
        XMOTOT = XMOTOT+MOL(I)

```

```

      GAPCON 1067
      GAPCON 1068
      GAPCON 1069
      GAPCON 1070
      GAPCON 1071
      GAPCON 1072
      GAPCON 1073
      GAPCON 1074
      GAPCON 1075
      GAPCON 1076
      GAPCON 1077
      GAPCON 1078
      GAPCON 1079
      GAPCON 1080
      GAPCON 1081
      GAPCON 1082
      GAPCON 1083
      GAPCON 1084
      GAPCON 1085
      GAPCON 1086
      GAPCON 1087
      GAPCON 1088
      GAPCON 1089
      GAPCON 1090
      GAPCON 1091
      GAPCON 1092
      GAPCON 1093
      GAPCON 1094
      GAPCON 1095
      GAPCON 1096
      GAPCON 1097
      GAPCON 1098
      GAPCON 1099
      GAPCON 1100
      GAPCON 1101
      GAPCON 1102
      GAPCON 1103
      GAPCON 1104
      GAPCON 1105
      GAPCON 1106
      GAPCON 1107
      GAPCON 1108
      GAPCON 1109
      GAPCON 1110
      GAPCON 1111
      GAPCON 1112
      GAPCON 1113
      GAPCON 1114
      GAPCON 1115
      GAPCON 1116
      GAPCON 1117
      GAPCON 1118
      GAPCON 1119
      GAPCON 1120
      GAPCON 1121
      GAPCON 1122
      GAPCON 1123

```

```

1200      CONTINUE
          DO 1210 I=1,7
          FMOL(I,1) = MOL(I)/XMTOT
          MOLEFR(I) = FMOL(I,1)
1210      CONTINUE
C
C      CALCULATE THE PRESSURE FOR THE PIN
          PRESTO = 73.604*XMTOT/VOLTXS
          XMOLS(NPOM1) = PRESTO*VPLENT/(73.604*TRLENA)
          IF (ICHNG.EQ.1) GO TO 1230
          ICHNG = 0
          DO 1220 JK=1,NPOM1
          IF (KAN(JK).EQ.0) GO TO 1220
          ICHNG = 1
1220      CONTINUE
1230      CONTINUE
          IF (ICHNG.EQ.0.AND.K.GT.2.AND.K.NE.TSTOP) GO TO 1240
          CALL PELET (TELM,TDELE,TW,TWDEL,KAN,NEW,NABF,NSH,NTAED,TSTSR,TS
+          TNR,YM,NART,NARZD,YD,TIME,CP,CR,FRDEN,IUN,NDEL,RD,PRESTO,EXTP,
+          PRESS,K,DISPC,NFF1,EFSTR,DSPSV)
1240      CONTINUE
          IF (K.NE.TSTOP) GO TO 1290
          LTLUS = 0
          PUWRD = PUWR
          DO 1280 IPOW=1,NPOM1
          JPOW = IPOW+1
          KANOLD(IPOW) = KAN(IPOW)
          OLDPOW(JPOW) = POWER(JPOW)
          IF (KAN(IPOW).EQ.1) IDO(IPOW) = 1
          HGCOLD(JPOW) = HGC(JPOW)
          DVBX(JPOW) = DVBX(JPOW)+DELVB(JPOW)
          PITI(JPOW) = BURNUP(JPOW)
          IF (KAN(IPOW).EQ.1) LTLUS = 1
          IF (MINI(1).LE.0.AND.JPOW.NE.LPMAX) GO TO 1270
          IF (MINI(1).GT.0.AND.MINI(JPOW).EQ.0) GO TO 1270
          ARFO = RD(NF,JPOW)
          ARFI = RD(1,JPOW)
          ARFO = ARFO/RD(NF+NC,JPOW)
          ARFI = ARFI/RD(NC+NF,JPOW)
          ARSIN = RSIN(JPOW)/RD(NF+NC,JPOW)
          FVAT = 0.0
          DO 1260 IV=1,6
          IX = IV+1
          IF (RSIN(JPOW).GT.RD(1,JPOW)) GO TO 1250
          FVAT = FVAT+BI(IV,JPOW)*((ARFO**IX-ARFI**IX)/IX)
          GO TO 1260
1250      CONTINUE
          FVAT = FVAT+ER(IV,JPOW)*((ARSIN**IX-ARFI**IX)/IX)+BI(IV,J
+          POW)*((ARFO**IX-ARSIN**IX)/IX)
1260      CONTINUE
          VAVGT = FVAT*2./((ARFO**2-ARFI**2))
          VAVGT(IPOW) = VAVGT
          VAVGT = (VAVGT)*1.8+32.0
          CALL OUTPUT (RL,GK(JPOW),GIPG2(JPOW),HGC(JPOW),HGAS(JPOW),HR
+          AD(JPOW),HSOLID(JPOW),JPOW,NX,JCASE,MOLEFR,LPMAX,PFACE(JPOW)
+          ,PRESTO,VAVGT,XMTOT,QOVRAC(JPOW),RG(JPOW),TCG(JPOW),TCOOL(J
+          POW),DELWFT(JPOW),TTTT,TSTSR,NF,NC,NFPNC,TFB(1,JPOW),TCB(1,J

```

	+	POW),RD(1,JPOW),TSTNR,NNOEL,NNOELD,EPSR,NPOW1,SUMPF,SUMM)	GAPCON	1181
1270		CONTINUE	GAPCON	1182
1280		CONTINUE	GAPCON	1183
1290		CONTINUE	GAPCON	1184
C			GAPCON	1185
		DO 1300 I=1,NPOW	GAPCON	1186
		CALL MOVEAA (FXMOL(1,1),FXMOL(1,1),7)	GAPCON	1187
		XMOLS(I) = PRESTOAVULTX(1)/73.664	GAPCON	1188
1300		CONTINUE	GAPCON	1189
		IF (NT.EQ.1) GO TO 1310	GAPCON	1190
		IF (ANKH.LE.0.0.OR.ANXE.LE.0.0) GO TO 1310	GAPCON	1191
		RATEC = (XMOTOT-FILMOL)/(E(5)+ANKH+ANXE)	GAPCON	1192
		IF (K.EQ.2.AND.ISTEP.EQ.1) GO TO 1370	GAPCON	1193
		IF (RATE.LT.0.) RATE = 0.	GAPCON	1194
1310		CONTINUE	GAPCON	1195
		IF (K.EQ.ISTOP) GO TO 1390	GAPCON	1196
		IF (NT.EQ.1) GO TO 1360	GAPCON	1197
		IF (RATE.LE.0) RATE = 0.	GAPCON	1198
		W1 = RATEC-RATE	GAPCON	1199
		W2 = RATE-WATOLD	GAPCON	1200
		IF (RATE.LE.0..AND.RATEC.LE.0.) GO TO 1360	GAPCON	1201
		IF (RATE.LE.0.) GO TO 1370	GAPCON	1202
		IF (ABS(W1/RATE).LT..01) GO TO 1360	GAPCON	1203
		WATOLD = RATE	GAPCON	1204
		IF (K.GT.11) GO TO 1360	GAPCON	1205
		IF (K.GT.7) GO TO 1320	GAPCON	1206
		RATE = RATEC	GAPCON	1207
		GO TO 1350	GAPCON	1208
1320		CONTINUE	GAPCON	1209
		IF (IU.EQ.1) GO TO 1330	GAPCON	1210
		IF (W1*W3.LT.0) IU = 1	GAPCON	1211
		IF (W1*W3.LE.0) GO TO 1330	GAPCON	1212
		RATE = RATEC+.4*W1	GAPCON	1213
		IF (RATE.GT.0) GO TO 1350	GAPCON	1214
		IU = 1	GAPCON	1215
		RATE = WATOLD	GAPCON	1216
1330		CONTINUE	GAPCON	1217
		IF (K.LE.5) GO TO 1340	GAPCON	1218
		IF (ABS(W5/W2).LT.1.2) ACELL = .5*ACELL	GAPCON	1219
		IF (ACELL.LT..1) ACELL = .1	GAPCON	1220
1340		CONTINUE	GAPCON	1221
		RATE = RATE+ACELL*W1	GAPCON	1222
		IF (RATE+RATEC.LT.0.01.AND.K.LE.5) ACELL = .5	GAPCON	1223
1350		CONTINUE	GAPCON	1224
		W3 = W1	GAPCON	1225
		W5 = W2	GAPCON	1226
		GO TO 1380	GAPCON	1227
1360		CONTINUE	GAPCON	1228
		ISTOP = K+1	GAPCON	1229
1370		CONTINUE	GAPCON	1230
		RATE = RATEC	GAPCON	1231
		IF (RATE.LT.0) RATE = 0.0	GAPCON	1232
1380		CONTINUE	GAPCON	1233
1390		CONTINUE	GAPCON	1234
		ACELL = .8	GAPCON	1235
		RATE = RATEC	GAPCON	1236
		ISTOP = ISTEP	GAPCON	1237

SUBROUTINE GAPCON

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

      IF (NX.EQ.1) GO TO 1410
      DO 1400 I=1,NPOM
      K9 = LSTHIF(I)
1400   IF (KDOT(NX,I).GE.KDOT(K9,I)) LSTHIF(I) = NX
1410   CONTINUE
      RATE = KATEC
      IF (JSAVE.NE.KSAVE) GO TO 1420
      IF (KOUNTM.NE.0) GO TO 1420
C
C   IF ITAPE = 1, PROGRAM WILL READ FROM AND
C   THEN CONTINUE WRITING ON TAPE 2
C
      L = 1
      IF (ITAPE.EQ.1) L = 2
      WRITE (6,1430) NT
      WRITE (L) TIMEN,PRESTO,DELGDM,DELPI,MSIN,PRESS,HGC,CRELST,CRCRP,RD
      +,TSTSR,KDOT,SUMM,NT,PITI,BTOLD,SAVNXR,SAVNXE,SUMNXR,SUMNXX,LSTHIF,
      +,TSTNR,RKSV,RKCV,BBH,BROLD,TROLD,NEW,OLD,TIMEF,TMX,NY,VOLTXS,RATE,R
      +,ATEC,RATOLD,W1,W2,W3,W5,TW,TELM,DISPF,RDELST,KANOLD,ESUM,ESTSUM,KT
      +,KAN,ESTONI,DTOLD,DELTAO,DELTO,TODAYS,ARSV,BR,HGCOLD,OLDORX,VOID,RK
      +,CSV,KOUNT,DENCGR,DENCUS,PRSS,IACON,ISAVE,SUMFF,RHSIN,NX,DELVB,DVBX
      +,POLO,POUTLO,PINLO,NTOLD,POWRLO,VAVGTC,PFACE,OLDSE,FLAG
      IF (IPLUT.EQ.0) GO TO 1420
      WRITE (L) POWER,BURNUP,HF,DELRH,DELRCI,DELRCC,DELRCT,DELMFT,HGAPX,
      +,TCOUL,TFB,TCB,MUL,HSOLID,HGAS,HRAQ,GK,G1PG2,EFSTR,B1,DELRT
1420   CONTINUE
C
      IF (IUP.GT.0.AND.POWR.LT.PEND) GO TO 420
      IF (IUP.LT.0.AND.POWR.GT.PEND) GO TO 420
      IF (TIMEN.NE.TMN) GO TO 420
      NT = NT+1
      IF (NT.LE.NTIME) GO TO 390
      RETURN
C
1430   WRITE (6,1470)
      GO TO 1460
1440   WRITE (6,1480)
      GO TO 1460
1450   WRITE (6,1520)
1460   CONTINUE
      STOP
1470   FORMAT (1H0,#NY.GT.49 ENLARGE THE DIMENSION OF KDOT SAVNXX SUMNXX=
      +)
1480   FORMAT (1H0,#NT IS GREATER THAN JSTART#)
1490   FORMAT (20X,#WILL WRITE PLOT3 VARIABLES FOR NT=#,I3)
1500   FORMAT (F10.4,3I5,5X,2F10.4)
1510   FORMAT (8F10.2)
1520   FORMAT (5X,#ITHOUT EXCEEDED,PROGRAM STOPPED#)
      END

```

GAPCON 1238
 GAPCON 1239
 GAPCON 1240
 GAPCON 1241
 GAPCON 1242
 GAPCON 1243
 GAPCON 1244
 GAPCON 1245
 GAPCON 1246
 GAPCON 1247
 GAPCON 1248
 GAPCON 1249
 GAPCON 1250
 GAPCON 1251
 GAPCON 1252
 GAPCON 1253
 GAPCON 1254
 GAPCON 1255
 GAPCON 1256
 GAPCON 1257
 GAPCON 1258
 GAPCON 1259
 GAPCON 1260
 GAPCON 1261
 GAPCON 1262
 GAPCON 1263
 GAPCON 1264
 GAPCON 1265
 GAPCON 1266
 GAPCON 1267
 GAPCON 1268
 GAPCON 1269
 GAPCON 1270
 GAPCON 1271
 GAPCON 1272
 GAPCON 1273
 GAPCON 1274
 GAPCON 1275
 GAPCON 1276
 GAPCON 1277
 GAPCON 1278
 GAPCON 1279
 GAPCON 1280
 GAPCON 1281
 GAPCON 1282
 GAPCON 1283
 GAPCON 1284
 GAPCON 1285
 GAPCON 1286

```

      SUBROUTINE GASREL (BURNUP,FRDEN,IGAS,LF,LSTHIF,NF,NFF1D,NT,RD,RDDT GASREL      2
      +,SAVNKR,SAVNXE,TFB,SUMNKR,SUMNKE,RV,NFA,IFLAG) GASREL      3
C      GASREL CALCULATES THE FUEL LOCAL GAS RELEASE BASED ON THE BEYER= GASREL      4
C      MANN MODEL. GASREL      5
C      A NRC DEVELOPED, LATE 1977 VERSION, HIGH BURNUP ENHANCEMENT FACTOR GASREL      6
C      IS INCLUDED. GASREL      7
C      NOTE THE TFB'S USED HERE ARE AXIALLY AVERAGED FOR THE IPOW REGION GASREL      8
C      GASREL      9
C      DIMENSION RDDT(50), SAVNKR(50), SAVNKE(50), TFB(NF), SUMNKR(50), S GASREL     10
      +SUMNKE(50), RD(NFF1D), RV(20,2) GASREL     11
C      LEVEL 2,SAVNKE,SAVNKR,SUMNKE,SUMNKR GASREL     12
      REAL LF GASREL     13
      REAL LR,LGSR GASREL     14
C      LSTHIF=LAST HIGHEST RELEASED FRACTION GASREL     15
      DIMENSION SVT(9), TD(9) GASREL     16
      DATA SVT/1800.,1200.,850.,570.,300.,160.,75.,27.,7.5/ GASREL     17
      DATA TD/.01,.92,.93,.94,.95,.96,.97,.98,.99/ GASREL     18
      DATA VOL,LR/2.E-21,7.E-4/ GASREL     19
      LGSR = 0. GASREL     20
      SVI = TERP(FRDEN,TD,SVT,9) GASREL     21
      IF (BURNUP.LT.20000.) IFLAG = 0 GASREL     22
      IF (BURNUP.GE.20000) IFLAG = IFLAG+1 GASREL     23
      XPUSUR = BURNUP GASREL     24
      C = 2.6E+16*FRDEN GASREL     25
      IF (IFLAG.GT.1) GO TO 100 GASREL     26
      FT = C*BURNUP GASREL     27
      IF (IFLAG.EQ.1) BURNUP = 20000 GASREL     28
100  CONTINUE GASREL     29
      IF (IFLAG.GT.1) FT = C*BURNUP GASREL     30
      VFT = VOL*FT GASREL     31
      SV = SVI GASREL     32
      R1200 = TERP(2192.,TFB(1),RD(1),NF) GASREL     33
      R1700 = TERP(3092.,TFB(1),RD(1),NF) GASREL     34
      R1400 = TERP(2552.,TFB(1),RD(1),NF) GASREL     35
      R1214 = .5*(R1200+R1400) GASREL     36
      R1417 = .5*(R1400+R1700) GASREL     37
      R17 = .5*(R1700+RD(1)) GASREL     38
      X1417 = (R1400**2-R1700**2)/(RD(NF)**2-RD(1)**2) GASREL     39
      X1214 = (R1200**2-R1400**2)/(RD(NF)**2-RD(1)**2) GASREL     40
      X17 = (R1700**2-RD(1)**2)/(RD(NF)**2-RD(1)**2) GASREL     41
      D1214 = 2*R1214 GASREL     42
      D1417 = 2*R1417 GASREL     43
      D17 = 2*R17 GASREL     44
      RAT12 = TERP(D1214,RV(1,1),RV(1,2),NFA) GASREL     45
      RAT14 = TERP(D1417,RV(1,1),RV(1,2),NFA) GASREL     46
      RAT17 = TERP(D17,RV(1,1),RV(1,2),NFA) GASREL     47
      F1417 = X1417*RAT17 GASREL     48
      F17 = X17*RAT17 GASREL     49
      IF (BURNUP.LE.0) GO TO 110 GASREL     50
      LGSR = SV*LR*(.75-(3./VFT)*(1.-EXP(-VFT/4))) GASREL     51
C      LOW TEMP. GAS RELEASE MODEL LIMITED TO 8 PERCENT RELEASE GASREL     52
      IF (LGSR.GT..08) LGSR = .08 GASREL     53
110  CONTINUE GASREL     54
      RDDT(NT) = (.05*X1214*RAT12+.141*X1417*RAT14+.07*X17*RAT17)+LGSR GASREL     55
      IF (IGAS.EQ.0) GO TO 120 GASREL     56
      RDDT(NT) = RDDT(NT)+1.68*(.002112+.0052*F1417**2-(.00269*F1417*F17 GASREL     57

```

SUBROUTINE GASREL

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	+) +.00217*F17**2)**+.5+LGSR	GASREL	59
120	CONTINUE	GASREL	60
	IF (BURNUP.LT.20000.) GO TO 130	GASREL	61
	DELB = BURNUP-20000.	GASREL	62
	FRETN = 1-RODT(NT)	GASREL	63
	BRNEF = (1.-EXP(-.436E-4*DELB))/(1.+.665/RODT(NT)*EXP(-1.107E-4*DELB))	GASREL	64
	+LR))	GASREL	65
	RODT(NT) = RODT(NT)+BRNEF*FRETN	GASREL	66
130	CONTINUE	GASREL	67
	BURNUP = XPOSUR	GASREL	68
	IF (RODT(NT).GT.1.) RODT(NT) = 1.	GASREL	69
	SUM6 = 0.0	GASREL	70
	SUM7 = 0.0	GASREL	71
	NX = NT-1	GASREL	72
	IF (NT.EQ.1) GO TO 160	GASREL	73
	IF (RODT(NT).LE.RODT(NX)) GO TO 150	GASREL	74
	LSL = LSTHIF	GASREL	75
	NIX = NX	GASREL	76
	L = 0	GASREL	77
	DO 140 NLL=LSL,NIX	GASREL	78
	L = L+1	GASREL	79
	NL = NIX-L	GASREL	80
	IF (RODT(NT).GE.RODT(NL)) GO TO 140	GASREL	81
	NX = NL	GASREL	82
	GO TO 150	GASREL	83
140	CONTINUE	GASREL	84
	NX = 1	GASREL	85
C		GASREL	86
C	SAVNXX(A)=TOTAL MOLES/INCH IN AXIAL NODE IPOW THRU TIME STEP A	GASREL	87
150	CONTINUE	GASREL	88
	SUM6 = SAVNKR(NX)	GASREL	89
	SUM7 = SAVNKE(NX)	GASREL	90
C	SUMNXX(A,B)=ATOMS/NODE B RELEASED	GASREL	91
160	IF (NT.GT.1) GO TO 170	GASREL	92
	SUMNKR(NT) = RODT(NT)*SAVNKR(NT)*LF/12.	GASREL	93
	SUMNKE(NT) = RODT(NT)*SAVNKE(NT)*LF/12.	GASREL	94
	GO TO 180	GASREL	95
170	SUMNKR(NT) = RODT(NT)*(SAVNKR(NT)-SUM6)*LF/12.+SUMNKR(NX)	GASREL	96
	SUMNKE(NT) = RODT(NT)*(SAVNKE(NT)-SUM7)*LF/12.+SUMNKE(NX)	GASREL	97
180	RETURN	GASREL	98
	END	GASREL	99

SUBROUTINE HCAP

74/74 OPT=1

FTN 4.6+052

15/02/74 17.25.31

	SUBROUTINE HCAP (TTT,CDT,CPNOM)	HCAP	2
C		HCAP	3
C	HCAP CALCULATES THE FUEL HEAT CAPACITY, CPNOM, AND INTEGRAL HEAT	HCAP	4
C	CAPACITY, CDT SEE THE CALL HCAP IN GAPCON	HCAP	5
C		HCAP	6
	DIMENSION A(7)	HCAP	7
	DATA (A(1),I=1,7)/5.046E-2,9.2506E-5,-1.989E-7,2.5575E-10,-1.8582E	HCAP	8
	+13,6,907E-17,-9.75E-21/	HCAP	9
	CDT = 0.0	HCAP	10
	CPNOM = 0.0	HCAP	11
	TT = TTT	HCAP	12
	T = TTT	HCAP	13
	C = 0.	HCAP	14
	T1 = 0.	HCAP	15
	DO 100 I=1,7	HCAP	16
	T = T+TT	HCAP	17
	IF (I.EQ.1) T = TT	HCAP	18
	C = C+21.	HCAP	19
	IF (I.EQ.1) C = 21.	HCAP	20
	T1 = T1+TT	HCAP	21
	IF (I.EQ.1) T1 = 1.	HCAP	22
	IF (I.EQ.2) T1 = TT	HCAP	23
	CAD = (A(I)/FLOAT(I))*(T-C)	HCAP	24
	CPAD = A(I)*T1	HCAP	25
	CDT = CDT+CAD	HCAP	26
	CPNOM = CPAD+CPNOM	HCAP	27
100	CONTINUE	HCAP	28
	CDT = CDT*4.186	HCAP	29
	CPNOM = CPNOM*4.186	HCAP	30
	RETURN	HCAP	31
	END	HCAP	32

```

C      SUBROUTINE INPAXL (AELM,R,RA,Y,YACF,NADET,RO,NAB2D,NFF10)
C      PURPOSE IS TO CREATE NODAL AND ELEMENT INFORMATION FOR THE
C      AXIAL INTRACTION ANALYSIS
C      COMPUTE THE NUMBER OF ELEMENTS IN THE FUEL
C      NABC=NUMBER OF AXIAL BOUNDS IN THE CLAD
C      NARF=NUMBER OF AXIAL BOUNDS IN THE FUEL
C      NARF=NUMBER OF AXIAL REGIONS IN THE FUEL
C      NAEF=NUMBER OF ELEMENTS IN AXIAL FUEL MODEL
C
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IEPSLM,IELSTN,IELSTR
+,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INDC,INOF,INDT,INODE,INODEC,INOD
+,ET,IR,IRBR,ISA,ISAC,ISAF,ISR,ISRB,ISBHC,I
+,SBHF,ISBC,ISBF,ISK,ISKC,ISTRES,ITDELE,IXK,IXKC,IXKT
+,IXLOAD,ICC,ITHET,ICA,ITHTAX,INADET,IRA,IY,IAELM,I
+,ARAA,IBA,IRBA,IRBRA,IXLODA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I
+,PSLA,IAK,IACN,IKAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA,
+,IIDO,ITME,IEFSTR,ITCRP,ICRPG,ICRATE
+,IXKCC,IXKIT
+,ITFB,ITCB,IRBSV,IESTSV,IESTNV
+,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDS,IPSSV,IPSSRT,JIIJ,IIPDS
+,IBASE,ITFR,IYM,IZ
+,IDSPSV
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUT,P
+,DEL
COMMON /LIMITS/ NF,NC,NDEL,NDELF,NDELC,ICON,IDF,NFF,MAXDC,NETOT,NE
+,WF,NEWT,KA,NARF,NARC,NART,NABF,NARC,NART,NAEC,NAEF,NTAE,JAM,NAXB1,
+,NAXB2,NAXB4,NFF1,NABF2,NABT2,CP,CR,FRDEN,KZ1,ISSTOP,IPLAS,NKN
+,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N
+,MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD
+,KPLAS,ICNT,HRAKR,IRAK,RAKSUM,IPLND,ISP
COMMON /LASTIC/ E,XNU,ALPHA
COMMON /CONTNT/ PI
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2
COMMON /CRP/ PSP(25),TPLASK
COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN
+,NCNN,ISTR,ITER
+,FLXR
COMMON/FEP/HMF,NT,KKZZZ,ISTOP,ROOLD(20),RRSIN(20),ICNOLD(20)
+,IDOLD(20)
+,PINOLD,POUTLD,POLD,NTOLD(20)
+,ICHNG,CRPAD(20)
+,STEPS
DIMENSION AELM(3,1)
DIMENSION RA(1),YACF(1),NADET(1)
DIMENSION Y(NAB2D,1),RO(NFF10,1),R(1)
C
C      NARC=NUMBER OF ELEMENTS IN AXIAL CLAD MODEL
C
C      MODIFICATIONS NEEDED FOR ADDING 4 ELEMENT QUAD.
C      NATOT = 2*NABC+2*NARF
C      LL = 0
C      DO 110 K=1,NTAE,4
C      LL = LL+1
C      L = LL
C      LNF = NARF+L

```

	LMFF = NATOTF+L	INPAXL	28
	IF (L.LE,NARF) GO TO 100	INPAXL	29
	IF (L.EQ,NARF) LL = NABF+2+1	INPAXL	30
	L = LL	INPAXL	31
	LMF = L+NABC	INPAXL	32
	LMFF = LL+2*NABC+NARF	INPAXL	33
100	CONTINUE	INPAXL	34
	AELM(1,K) = L	INPAXL	35
	AELM(2,K) = LMF	INPAXL	36
	AELM(3,K) = LMFF	INPAXL	37
	AELM(1,K+1) = LMF	INPAXL	38
	AELM(2,K+1) = LMF+1	INPAXL	39
	AELM(3,K+1) = AELM(3,K)	INPAXL	40
	AELM(1,K+2) = AELM(2,K+1)	INPAXL	41
	AELM(2,K+2) = L+1	INPAXL	42
	AELM(3,K+2) = AELM(3,K)	INPAXL	43
	AELM(1,K+3) = L+1	INPAXL	44
	AELM(2,K+3) = L	INPAXL	45
	AELM(3,K+3) = AELM(3,K)	INPAXL	46
110	CONTINUE	INPAXL	47
C		INPAXL	48
C	DEFINE AN AXIAL ELEMENT ARRAY	INPAXL	49
C	AELM(K,I) K=ELEMENT NUMBER I=NODE NUMBERS ASSOCIATED WITH K	INPAXL	50
C	COMPUTE THE ELEMENT AND NODAL CONFIGURATION	INPAXL	51
C		INPAXL	52
C		INPAXL	53
C	CREATE THE DEGREE OF FREEDOM ARRAY FOR THE FUEL	INPAXL	54
C		INPAXL	55
C	CREATO NADET() FOR CORNER MODES	INPAXL	56
	NADET(1) = 1	INPAXL	57
	NADET(2) = 9	INPAXL	58
	DO 120 K=3,NABF	INPAXL	59
	NADET(K) = 25+(K-3)*12	INPAXL	60
120	CONTINUE	INPAXL	61
	L = NABF+3	INPAXL	62
	KK = NABF+NABF	INPAXL	63
	NADET(NABF+1) = 3	INPAXL	64
	NADET(NABF+2) = 7	INPAXL	65
	DO 130 K=L,KK	INPAXL	66
	NADET(K) = 23+(K-L)*12	INPAXL	67
130	CONTINUE	INPAXL	68
	L = 2*NABF+3	INPAXL	69
	KK = NABF+2+NABF	INPAXL	70
	IJ = NABF+2+1	INPAXL	71
	NADET(IJ) = 11	INPAXL	72
	NADET(IJ+1) = 19	INPAXL	73
	DO 140 K=L,KK	INPAXL	74
	NADET(K) = 31+(K-L)*12	INPAXL	75
140	CONTINUE	INPAXL	76
	NADET(KK+1) = NADET(KK)+6	INPAXL	77
	L = 2*NABF+NABC+3	INPAXL	78
	KK = NABF+2+2*NABC-1	INPAXL	79
	IJ = NABF+2+NABC+1	INPAXL	80
	NADET(IJ) = 13	INPAXL	81
	NADET(IJ+1) = 17	INPAXL	82
	DO 150 K=L,KK	INPAXL	83
	NADET(K) = 29+12*(K-L)	INPAXL	84

150	CONTINUE	INPAXL	85
	NADET(KK+1) = NADET(KK)+6	INPAXL	86
C	CREATE NADET FOR MID NODIS START WITH THE FUEL	INPAXL	87
	L = 2*NABF+2*NABC+2	INPAXL	88
	KK = NABF*2+2*NABC+NARF	INPAXL	89
	IJ = NABF*2+2*NABC+1	INPAXL	90
	NADET(IJ) = 5	INPAXL	91
	DO 160 K=L, KK	INPAXL	92
	NADET(K) = 21+12*(K-L)	INPAXL	93
160	CONTINUE	INPAXL	94
	L = 2*NABF+2*NABC+NARF+2	INPAXL	95
	KK = (NABF+NABC+NARF)*2	INPAXL	96
	IJ = L-1	INPAXL	97
	NADET(IJ) = 15	INPAXL	98
	DO 170 K=L, KK	INPAXL	99
	NADET(K) = 27+(K-L)*12	INPAXL	100
170	CONTINUE	INPAXL	101
	NADET(KK+1) = NADET(KK)+6	INPAXL	102
C	HAVE NOW CREATED NADET() FOR FOUR CST ELEMENTS	INPAXL	103
C	FOR EACH AXIAL REGION FOR BOTH THE FUEL AND THE CLAD	INPAXL	104
C		INPAXL	105
C	CREATE A RADIAL ARRAY WITH NODAL VALUES FOR THE AXIAL MODEL	INPAXL	106
C		INPAXL	107
	RA(4*NABT)	INPAXL	108
	Y(1,1) = 0.0	INPAXL	109
	DO 180 J=2, NABF2	INPAXL	110
	Y(J,1) = Y(J-1,1)+DELZA(J-1)	INPAXL	111
	NABF1 = J-NABF	INPAXL	112
	IF (J.GT.NABF) Y(J,1) = Y(NABF1,1)	INPAXL	113
180	CONTINUE	INPAXL	114
	Y(1,2) = 0.0	INPAXL	115
	DO 190 J=2, NABT2	INPAXL	116
	Y(J,2) = Y(J-1,2)+DELZA(J-1)	INPAXL	117
	NABT1 = J-NABT	INPAXL	118
	IF (J.GT.NABT) Y(J,2) = Y(NABT1,2)	INPAXL	119
190	CONTINUE	INPAXL	120
	K = 0	INPAXL	121
	DO 210 I=1,4	INPAXL	122
	NNN = NABF	INPAXL	123
	IF (I.GT.2) NNN = NABT	INPAXL	124
	NA = 1	INPAXL	125
	IF (I.EQ.2) NA = NF	INPAXL	126
	IF (I.EQ.3) NA = NF+1	INPAXL	127
	IF (I.EQ.4) NA = NF+NC	INPAXL	128
	DO 200 J=1, NNN	INPAXL	129
	K = K+1	INPAXL	130
	RA(K) = RD(NA, J)	INPAXL	131
	NUP = 1	INPAXL	132
	IF (I.GT.2) NUP = 2	INPAXL	133
	YACF(K) = Y(J, NUP)	INPAXL	134
200	CONTINUE	INPAXL	135
210	CONTINUE	INPAXL	136
C		INPAXL	137
	RETURN	INPAXL	138
	END	INPAXL	139

SUBROUTINE INPAXL	74/74	OPT=1	FTN 4.6+452	15/02/78	17.25.31
CARD NR.	SEVERITY	DETAILS	DIAGNOSIS OF PROBLEM		
88	1	MADET	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.		

	SUBROUTINE INPUT (RV,RVE,NF)	INPUT	2
C		INPUT	3
C	INPUT WRITES THE USER INPUT.	INPUT	4
C	DATE IS AN EXTERNAL FUNCTION.	INPUT	5
C		INPUT	6
	DIMENSION RV(20,2), RVE(20,2)	INPUT	7
	COMMON /FP/ D,BETA,E,G,H	INPUT	8
	COMMON /BA/ AA(23,7),CF(7,8),PKS,DELZA(20),NN,NCONN	COMC	2
	+, NCNN,ISTRT,ITER	COMC	3
	+, FLXR	COMC	4
	COMMON/FEP/HMF,NT,KKZZZ,ISTOP,ROOLD(20),RRSIN(20),ICNOLD(20)	COMC	5
	+, IDOOLD(20)	COMC	6
	+, PINOLD,POUTLD,POLD,NTOLD(20)	COMC	7
	+, ICHNG,CRPAD(20)	COMC	8
	+, STEPS	COMC	9
	COMMON /GA/ NPOW,NTIME,NFZ,NCZ,IPLDT,ATMOS,DCI,DCU,DFS,DSINZ,DIEMP	COMA	2
	+, DVOIDZ,FR35,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRACN,FRACXE,FRD	COMA	3
	+EN,FRPU02,FRPSIN,IGAS,IPEAK,LFUEL,LVOIDZ, NFUEL,NQH,PRCDH,R	COMA	4
	+ADS,ROUC,ROUF,S,TINLET,TM,TPLAS,VPLENZ,XCO,XH,XN,IRELUC,IRL,IDENSE	COMA	5
	+,FLXPAT,X,Z,ZZ,YY,WW,EXTP,ISTOR,NCLAD,ZCLAD,NSH,PROFIL(20),PSEUDO(	COMA	6
	+100),TIME(100),NFLX,TITLE(10),MXPOWR,IOPY,JSTART,KSAVE,ISAVE,ITMCU	COMA	7
	+T,ISTAND,ITAPE,NSTEPS,MINI(20),SUMARY,FRDENZ	COMA	8
	INTEGER SUMARY	COMA	9
	REAL LFUEL,LVOIDZ,MXPOWR	COMA	10
	COMMON /AZ/ BURNUP(20),DELPI(20),DELR,DELRB(20),DELRCC(20),DELRCI(	COMI	2
	+20),DELRCT(20),DELRP(20),DELR(20),DSIN,DVOID,FRU02,FR38,FR39,GAP,	COMI	3
	+HBC,K8,DELGDM(20)	COMI	4
	WRITE (6,260)	INPUT	12
	ADATE = DATE(L)	INPUT	13
	WRITE (6,270) ADATE	INPUT	14
	WRITE (6,280) TITLE	INPUT	15
	WRITE (6,290)	INPUT	16
	WRITE (6,300) FRPU02,FR39,FR40,FR41,FRU02,FR35,FR38,FRDEN,FRDEN2,F	INPUT	17
	+RSIN,DFS,DSINZ,DVOIDZ	INPUT	18
	WRITE (6,310) GAP,DCI,DCU,LFUEL,S,XH	INPUT	19
	WRITE (6,320) VPLENZ,TINLET,DIEMP,ROUF,ROUC	INPUT	20
	WRITE (6,330) ATMOS	INPUT	21
	WRITE (6,350) FRACHE,FRACAP,FRACH,FRACN,FRACKR,FRACXE	INPUT	22
	WRITE (6,470)	INPUT	23
	WRITE (6,470)	INPUT	24
	WRITE (6,470)	INPUT	25
	IF (NCLAD) 120,100,130	INPUT	26
100	CONTINUE	INPUT	27
	IF (ZCLAD.GT.0.) GO TO 110	INPUT	28
	WRITE (6,400)	INPUT	29
	GO TO 140	INPUT	30
110	CONTINUE	INPUT	31
	WRITE (6,410)	INPUT	32
	GO TO 140	INPUT	33
120	WRITE (6,420)	INPUT	34
	GO TO 140	INPUT	35
130	WRITE (6,430)	INPUT	36
140	WRITE (6,470)	INPUT	37
	IF (NFUEL) 170,150,160	INPUT	38
150	WRITE (6,440)	INPUT	39
	GO TO 200	INPUT	40
160	WRITE (6,450)	INPUT	41

```

      GO TO 180
170  WRITE (6,460)
      GO TO 200
180  CONTINUE
      IF (CF(1,2).GT.0.) GO TO 200
          DO 190 J=1,7
              TMPK = ((CF(J,1)-32.)/1.8)+273.
              PM = D*((FRDEN)/(1.+BETA*(1.-FRDEN)))
              TCWC = PM*(E/(G+TMPK)+H*TMPK**3.)
              CF(J,2) = 57.8*TCWC
190  CONTINUE
200  CONTINUE
      WRITE (6,470)
      WRITE (6,340)
      WRITE (6,240) (CF(J,1),CF(J,2),J=1,NCNN)
      WRITE (6,360)
      IF (IRL.GT.0) GO TO 220
      RV1 = RV(1,2)
          DO 210 I=1,NF
              RV(I,2) = RV(I,2)/RV1
210  CONTINUE
      WRITE (6,250) (RV(J,1),RV(J,2),J=1,NF)
      GO TO 230
220  WRITE (6,250) (RVE(J,1),RVE(J,2),J=1,IRL)
230  CONTINUE
      WRITE (6,380)
      MIN = MINI(1)
      WRITE (6,390) NPOW,IPEAK,IGAS,MIN,NTIME,IOPT,NFUEL,NFLX,ISTOR,NQH,
+IDENSF,NCLAD,IRL,IRELOC,ZCLAD,KSAVE,ITMCUT,JSTART,NSTEPS,IPLUT
      WRITE (6,370)
      RETURN
C
C
C
C
C
C
C
240  FORMAT (20X,*,18X,F6.0,31X,F5.3,31X,*)
250  FORMAT (20X,*,20X,F10.4,15X,F10.4,36X,*)
260  FORMAT (1H1)
270  FORMAT (//,115X,1A10)
280  FORMAT (1H0,20A4)
290  FORMAT (58X,*****X,/,58X,X INPUT VALUES *,/,20X,
+*****X,/,20X,X,X,91X,*)
300  FORMAT (20X,*,8X,*FUEL COMPOSITION*,67X,*/20X,X,X,10X,F6.4,X
+WEIGHT FRACTION PU02*,27X,*FRPU02*,20X,*/20X,X,X,13X,F6.4,X WEI
+GHT FRACTION PU239*,23X,*FR39 X,20X,*/20X,X,X,13X,F6.4,X WEIGH
+T FRACTION PU240*,23X,*FR40 X,20X,*/20X,X,X,13X,F6.4,X WEIGHT
+FRACTION PU241*,23X,*FR41 X,20X,*/20X,X,X,09X,F6.4,X WEIGHT FR
+ACTION U02*,29X,*FRU02 X,20X,*/20X,X,X,13X,F6.4,X WEIGHT FRACTI
+ON U235*,24X,*FR35 X,20X,*/20X,X,X,13X,F6.4,X WEIGHT FRACTION
+U238*,24X,*FR38 X,20X,*/20X,X,X,91X,*/20X,X,X,8X,*FUEL DENSIT
+Y*,28X,*(FRACTION TO) FRDEN*,6X,F5.3,11X,*/20X,X,X,8X,*DENSIF
+IFO FUEL DENSITY*,18X,*(FRACTION TO) FRDEN*,5X,F5.3,11X,*/20
+X,X,X,8X,*RESTRUCTURED FUEL DENSITY*,15X,*(FRACTION TO) FRSI*,6

```

```

      +X,F5.3,11X,###/20X,###,8X,#PELLET DIAMETER#,25X,#(INCHES)#,8X,#DFS INPUT 99
      +X,F6.4,10X,###/20X,###,8X,#INITIAL RESTRICTURED FUEL DIAM.#,9X, INPUT 100
      +X,(INCHES)#,8X,#DSIN#,7X,F6.4,10X,###/20X,###,8X,#INITIAL CENTER HO INPUT 101
      +LE DIAMETER#,12X,#(INCHES)#,8X,#DVOID#,5X,F6.4,10X,###) INPUT 102
310  FORMAT (20X,###,8X,#PELLET-TO-CLAD GAP#,22X,#(INCHES)#,8X,#GAP#,8X INPUT 103
      +,F6.4,10X,###/20X,###,8X,#CLAD INSIDE DIAMETER#,20X,#(INCHES)#,8X, INPUT 104
      +#DCI#,8X,F6.4,10X,###/20X,###,8X,#CLAD OUTSIDE DIAMETER#,19X,#(INC INPUT 105
      +#ES)#,8X,#DCU#,8X,F6.4,10X,###/20X,###,8X,#FUEL LENGTH#,29X,#(INCH INPUT 106
      +#S)#,8X,#LFUEL#,4X,F6.2,12X,###/20X,###,8X,#SORBED GAS CONTENT#,22 INPUT 107
      +X,#(CC/GRAM)#,7X,#S#,10X,F6.4,10X,###/20X,###,8X,#FRACTION OF SURB INPUT 108
      +ED GAS WHICH IS H2#,22X,###,9X,F6.4,10X,###) INPUT 109
320  FORMAT (20X,###,8X,#PLENUM VOLUME#,27X,#(CU. IN.)#,7X,#VPLENZ#,4X, INPUT 110
      +F5.3,12X,###/20X,###,8X,#COOLANT TEMPERATURE#,21X,#(DEG F)#,9X,#TI INPUT 111
      +NLET(1)#,F6.1,12X,###/20X,###,8X,#AXIAL TEMPERATURE GRADIENT ACROSS INPUT 112
      + CORE (DEG F)#,9X,#DTEMP#,3X,F6.1,13X,###/20X,###,8X,#FUEL SURFA INPUT 113
      +CE ROUGHNESS, ARITH. MEAN (INCH)#,10X,#ROUF#,7X,F6.6,8X,###/20 INPUT 114
      +X,###,8X,#CLAD ID SURFACE ROUGHNESS, ARITH. MEAN (INCH)#,10X,#ROU INPUT 115
      +C#,7X,F6.6,8X,###) INPUT 116
330  FORMAT (20X,###,8X,#FILL GAS PRESSURE#,23X,#(ATMOSPHERES) ATMOS#, INPUT 117
      +4X,F5.1,13X,###) INPUT 118
340  FORMAT (20X,###,91X,###/20X,###,8X,#FUEL THERMAL CONDUCTIVITY VALU INPUT 119
      +ES USED#,46X,###/20X,###,91X,###/20X,###,47X,#THERMAL CONDUCTIVITY INPUT 120
      +#,24X,###/20X,###,16X,#TEMPERATURE#,22X,#(BTU/HR-FT-DEG F)#,25X,## INPUT 121
      +#/20X,###,18X,#(DEG F)#,66X,###/20X,###,48X,#UNRESTRICTURED FUEL#, INPUT 122
      +24X,###/20X,###,15X,#####,20X,#####,23X INPUT 123
      +,###/20X,###,91X,###) INPUT 124
350  FORMAT (20X,###,8X,#FILL GAS COMPOSITION#,63X,###/20X,###,10X,F7.5 INPUT 125
      +, # MOLE FRACTION HELIUM#,52X,###/20X,###,10X,F7.5, # MOLE FRACTIO INPUT 126
      +N ARGON#,53X,###/20X,###,10X,F7.5, # MOLE FRACTION HYDROGEN#,50X, # INPUT 127
      +#/20X,###,10X,F7.5, # MOLE FRACTION NITROGEN#,50X,###/20X,###,10X INPUT 128
      +,F7.5, # MOLE FRACTION KRYPTON#,51X,###/20X,###,10X,F7.5, # MOLE F INPUT 129
      +RACTION XENON#,53X,###) INPUT 130
360  FORMAT (20X,###,91X,###/20X,###,8X,#FLUX DEPRESSION VALUES USED#,5 INPUT 131
      +6X,###/20X,###,91X,###/20X,###,21X,#DIAMETER (IN)#,12X,#FLUX RATIO INPUT 132
      +#,35X,###/20X,###,91X,###) INPUT 133
370  FORMAT (20X,###,91X,###/20X,##### INPUT 134
      +#####) INPUT 135
380  FORMAT (20X,###,91X,##,/,20X,###,10X,#### INPUT OPTIONS ####,58 INPUT 136
      +X,###,/,20X,###,91X,###) INPUT 137
390  FORMAT (20X,###,15X,#NPOW ==,13,15X,#IPEAK ==,13,15X,#IGAS ==,13 INPUT 138
      +,16X,###,/,20X,###,15X,#HINI ==,13,15X,#NTIME ==,13,41X,###,/,20X, # INPUT 139
      +#,15X,#IOPT ==,13,15X,#NFUEL ==,13,15X,#NFLX ==,13,16X,###,/,20 INPUT 140
      +X,###,15X,#ISTOR ==,13,15X,#NOH ==,13,15X,#IDENSP==,13,16X,###, / INPUT 141
      +,20X,###,15X,#NCLAD ==,13,15X,#IRL ==,13,15X,#IRELOC==,13,16X,## INPUT 142
      +,/,20X,###,15X,#ZCLAD ==,13,0,15X,#KSAVE ==,13,15X,#ITMCUT==,13,16 INPUT 143
      +X,###,/,20X,###,15X,#JSTART==,13,15X,#NSTEPS==,13,15X,#IPLT ==,13 INPUT 144
      +,16X,###) INPUT 145
400  FORMAT (20X,###,8X,#CLADDING IS ZIRCALOY-2#,61X,###) INPUT 146
410  FORMAT (20X,1H*,8X,#CLADDING IS ZIRCALOY-4#,61X,1H*) INPUT 147
420  FORMAT (20X,###,8X,#CLADDING IS 304 STAINLESS STEEL#,52X,###) INPUT 148
430  FORMAT (20X,###,8X,#CLADDING NOT SPECIFIED, PROPERTIES WERE INPUTT INPUT 149
      +EO#,35X,###) INPUT 150
440  FORMAT (20X,###,8X,#FUEL THERMAL CONDUCTIVITY VALUES DETERMINED FR INPUT 151
      +OM DATA OF LYONS, ET AL#,14X,###) INPUT 152
450  FORMAT (20X,###,8X,#FUEL THERMAL CONDUCTIVITY VALUES WERE INPUTTED INPUT 153
      +#,37X,###) INPUT 154
460  FORMAT (20X,###,8X,#FUEL CONTAINS RECYCLE ROD2#,57X,###,) INPUT 155
470  FORMAT (20X,###,91X,###) INPUT 156
      END INPUT 157

```

```

SUBROUTINE INPT (NABT,NABF,NF,NC,CP,CR)
C
C SUBROUTINE INPT READS TITLE AND VALUES FROM NAMELIST INPUT THEN
C IF NCLAD GT 0 READS USER CLADDING DATA THEN
C IF NFLX GT 1 READS IN FLUX DEPRESSION PARAMETERS ZZ,YY,WW THEN
C IF NFUEL GT 0 READS FUEL PROPERTIES
C WHEN KSAVE GT 0 PARAMETERS ARE WRITTEN ON RESTART TAPE 1
C WHEN JSTART GT 0 PARAMETERS ARE READ FROM RESTART TAPE 2
C A NONZERO VALUE OF JSTART CAUSES A RESTART. THE APPROPRIATE PARAME
C RS FOR TIME STEP NO. EQUAL TO THE VALUE OF JSTART ARE RETRIEVED
C FROM THE RESTART TAPE, THEN THE CALCULATIONS FOR TIME STEP JSTART+
C START AND CONTINUE THRU TIME STEP NTIME. FOR RESTARTS NTIME CAN BE
C INCREASED USING THE PARAMETER NSTEPS.
C A NONZERO VALUE OF IPLOT CAUSES ADDITIONAL OUTPUT ON THE RESTART
C TAPE. KSAVE MUST BE NONZERO. DATA FOR PLOTTING IS WRITTEN ON THE
C RESTART TAPE AT KSAVE INTERVALS
C
C DIMENSION DATMOS(1)
C DIMENSION RNEWK(5)
C EQUIVALENCE (NPOW,DATMOS)
C COMMON /FAD/ FF(5),NFLXN,FFT(5)
C COMMON/FP/BETA,D,E,G,H
C EQUIVALENCE (BETA,RNEWK)
C COMMON /GA/ NPOW,NTIME,NFZ,NCZ,IPLOT,ATMOS,DCI,DCQ,DFS,DSINZ,DTEMP
C +,DVOIDZ,FR35,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRACX,FRACD,FRAC
C +EN,FRPUO2,FRSIN,IGAS,IPEAK,LFUEL,LVOIDZ, NFUEL,NQH,PRCDH,R
C +ADS,POUC,ROUF,S,TINLET,TM,TPLAS,VPLENZ,XCO,XH,XN,IRELOC,IRL,IDENSF
C +,FLXRAT,X,Z,ZZ,YY,WW,EXTP,ISTOR,NCLAD,ZCLAD,NSH,PROFIL(20),PSEUDO(
C +100),TIME(100),NFLX,TITLE(10),MXPOWR,IOP,T,JSTART,KSARE,ISAVE,ITMCU
C +T,ISTAND,ITAPE,NSTEPS,MINI(20),SUMARY,FRDEN2
C INTEGER SUMARY
C REAL LFUEL,LVOIDZ,MXPOWR
C COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN
C +, NCNN,ISTRT,ITER
C +, FLXR
C COMMON/PEP/HHF,NT,KKZZZ,ISTOP,ROOLD(20),RRSIN(20),ICNOLD(20)
C +, IDOOLD(20)
C +,PINOLD,POUTLD,POLD,NTOLD(20)
C +, ICHNG,CRPAD(20)
C +,STEPS
C COMMON /EA/ ZR(6,7),ZR4(6,7),ST(12,7),RV(20,2),UO2(7,7)
C NAMELIST /INPUT/ATMOS,RNEWK,CP,CR, DCI,DCQ,DELZA,DFS,DSINZ,DTEMP,
C +DVOIDZ,EXTP,FLXRAT, FR35,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRAC
C +N,
C +XE,FRDEN,FRDEN2,FRPUO2,FRSIN,IDENSF,IGAS,IOP,T,IPEAK,IPLOT,IRELOC,
C +IRL,ISTOR,ITMCUT,JSTART,ITAPE,KSARE,LFUEL,LVOIDZ,MINI,MXPOWR,NC,
C +NCLAD
C +,NF,NFUEL,NFLX,NQH,NPOW,NSH,NSTEPS,NTIME,PRCDH,PSEUDO,PROFIL,RADS,
C +ROUC,ROUF,S,TINLET,TIME,TM,TPLAS,VPLENZ,X,XCO,XH,XN,Z,ZCLAD
C ILOC1 = LOC(NPOW)
C ILOC2 = LOC(NFLX)
C ILEN = ILOC2-ILOC1+1
C READ (5,310) TITLE,ISTAND
C IF (ISTAND.NE.0) GO TO 100
C CALL MOVEKA (0,DATMOS,ILEN)
C MXPOWR = 0.
C ITMCUT = 0.

```

```

    *X,F5,3,11X,##/20X,##,8X,PELLET DIAMETER#,25X,(INCHES)#,8X,DFS INPUT 99
    *X,F6,4,10X,##/20X,##,8X,INITIAL RESTRUCTURED FUEL DIAM.#,9X, INPUT 100
    *X,(INCHES)#,8X,DSIN#,7X,F6,4,10X,##/20X,##,8X,INITIAL CENTER HO INPUT 101
    +LE DIAMETER#,12X,(INCHES)#,8X,VOID#5X,F6,4,10X,## INPUT 102
310  FORMAT (20X,##,8X,PELLET-TO-CLAD GAP#,22X,(INCHES)#,8X,GAP#,8X INPUT 103
    +,F6,4,10X,##/20X,##,8X,CLAD INSIDE DIAMETER#,20X,(INCHES)#,8X, INPUT 104
    +DCI#,8X,F6,4,10X,##/20X,##,8X,CLAD OUTSIDE DIAMETER#,19X,(INC INPUT 105
    +MES)#,8X,DCUN#,8X,F6,4,10X,##/20X,##,8X,FUEL LENGTH#,29X,(INCH INPUT 106
    +ES)#,8X,LFUEL#,4X,F6,2,12X,##/20X,##,8X,SORBED GAS CONTENT#,22 INPUT 107
    +X,(CC/GRAM)#,7X,##,10X,F6,4,10X,##/20X,##,8X,FRACTION OF SURB INPUT 108
    +ED GAS WHICH IS H2#,22X,##,9X,F6,4,10X,## INPUT 109
320  FORMAT (20X,##,8X,PLENUM VOLUME#,27X,(CU. IN.)#.7X,KVLENZ#,4X, INPUT 110
    +F5,3,12X,##/20X,##,8X,COOLANT TEMPERATURE#,21X,(DEG F)#,9X,TI INPUT 111
    +NLET(1)#,F6,1,12X,##/20X,##,8X,AXIAL TEMPERATURE GRADIENT ACROSS INPUT 112
    + CORE (DEG F)#,9X,DTEMP#,3X,F6,1,13X,##/20X,##,8X,FUEL SURFA INPUT 113
    +CE ROUGHNESS, ARITH. MEAN (INCH)#,10X,ROUF#,7X,F6,6,8X,##/20 INPUT 114
    +X,##,8X,CLAD ID SURFACE ROUGHNESS, ARITH. MEAN (INCH)#,10X,ROU INPUT 115
    +C#,7X,F6,6,8X,## INPUT 116
330  FORMAT (20X,##,8X,FILL GAS PRESSURE#,23X,(ATMOSPHERES) ATMOS#, INPUT 117
    +4X,F5,1,13X,## INPUT 118
340  FORMAT (20X,##,91X,##/20X,##,8X,FUEL THERMAL CONDUCTIVITY VALU INPUT 119
    +ES USED#,46X,##/20X,##,91X,##/20X,##,47X,THERMAL CONDUCTIVITY INPUT 120
    +#,24X,##/20X,##,16X,TEMPERATURE#,22X,(BTU/HR-FT-DEG F)#,25X,## INPUT 121
    +#/20X,##,18X,(DEG F)#,66X,##/20X,##,48X,UNRESTRUCTURED FUEL#, INPUT 122
    +24X,##/20X,##,15X,##,20X,##,23X INPUT 123
    +,##/20X,##,91X,## INPUT 124
350  FORMAT (20X,##,8X,FILL GAS COMPOSITION#,63X,##/20X,##,10X,F7.5 INPUT 125
    +,MOLE FRACTION HELIUM#,52X,##/20X,##,10X,F7.5,MOLE FRACTIO INPUT 126
    +N ARGON#,53X,##/20X,##,10X,F7.5,MOLE FRACTION HYDROGEN#,50X,## INPUT 127
    +#/20X,##,10X,F7.5,MOLE FRACTION NITROGEN#,50X,##/20X,##,10X INPUT 128
    +,F7.5,MOLE FRACTION KRYPTON#,51X,##/20X,##,10X,F7.5,MOLE F INPUT 129
    +RACTION XENON#,53X,## INPUT 130
360  FORMAT (20X,##,91X,##/20X,##,8X,FLUX DEPRESSION VALUES USED#,5 INPUT 131
    +6X,##/20X,##,91X,##/20X,##,21X,DIAMETER (IN)#,12X,FLUX RATIO INPUT 132
    +#,35X,##/20X,##,91X,## INPUT 133
370  FORMAT (20X,##,91X,##/20X,##,***** INPUT 134
    +***** INPUT 135
380  FORMAT (20X,##,91X,##,/,20X,##,10X,***** INPUT (OPTIONS) *****#58 INPUT 136
    +X,##,/,20X,##,91X,## INPUT 137
390  FORMAT (20X,##,15X,NPOW ##,13,15X,PIPEAK ##,13,15X,IGAS ##,13 INPUT 138
    +,16X,##,20X,##,15X,MINI ##,13,15X,NTIME ##,13,41X,##,20X,## INPUT 139
    +##,15X,IOPT ##,13,15X,NFUEL ##,13,15X,NFLX ##,13,16X,##,20 INPUT 140
    +X,##,15X,ISTOR ##,13,15X,NQH ##,13,15X,IDENTS# ##,13,16X,##,20 INPUT 141
    +,20X,##,15X,NCLAD ##,13,15X,IRL ##,13,15X,IRELOC##,13,16X,## INPUT 142
    +##,20X,##,15X,ZCLAD ##,13,0,15X,MSAVE ##,13,15X,ITMCUT##,13,16 INPUT 143
    +X,##,20X,##,15X,JSTART##,13,15X,NSTEPS##,13,15X,IPLUT ##,13 INPUT 144
    +,16X,## INPUT 145
400  FORMAT (20X,##,8X,CLADDING IS ZIRCALOY-2#,61X,## INPUT 146
410  FORMAT (20X,1H,##,8X,CLADDING IS ZIRCALOY-4#,61X,1H# INPUT 147
420  FORMAT (20X,##,8X,CLADDING IS 304 STAINLESS STEEL#,52X,## INPUT 148
430  FORMAT (20X,##,8X,CLADDING NOT SPECIFIED, PROPERTIES WERE INPUT INPUT 149
    +ED#,35X,## INPUT 150
440  FORMAT (20X,##,8X,FUEL THERMAL CONDUCTIVITY VALUES DETERMINED FR INPUT 151
    +OM DATA OF LYONS, ET AL#,14X,## INPUT 152
450  FORMAT (20X,##,8X,FUEL THERMAL CONDUCTIVITY VALUES WERE INPUTED INPUT 153
    +#,37X,## INPUT 154
460  FORMAT (20X,##,8X,FUEL CONTAINS RECYCLE PUD2#,57X,##,) INPUT 155

```

```

470  FORMAT (20X,##,91X,##)
      END

```

```

INPUT 156
INPUT 157

```

```

C      SUBROUTINE INPT (NABT,NABF,NF,NC,CP,CR)
C
C      SUBROUTINE INPT READS TITLE AND VALUES FROM NAMELIST INPUT THEN
C      IF NCLAD GT 0 READS USER CLADDING DATA THEN
C      IF NFLX GT 1 READS IN FLUX DEPRESSION PARAMETERS ZZ,YY,WW THEN
C      IF NFUEL GT 0 READS FUEL PROPERTIES
C      WHEN KSAVE GT 0 PARAMETERS ARE WRITTEN ON RESTART TAPE 1
C      WHEN JSTART GT 0 PARAMETERS ARE READ FROM RESTART TAPE 2
C      A NONZERO VALUE OF JSTART CAUSES A RESTART. THE APPROPRIATE PARAME
C      RS FOR TIME STEP NO. EQUAL TO THE VALUE OF JSTART ARE RETRIEVED
C      FROM THE RESTART TAPE, THEN THE CALCULATIONS FOR TIME STEP JSTART+
C      START AND CONTINUE THRU TIME STEP NTIME. FOR RESTARTS NTIME CAN BE
C      INCREASED USING THE PARAMETER NSTEPS.
C      A NONZERO VALUE OF IPLOT CAUSES ADDITIONAL OUTPUT ON THE RESTART
C      TAPE. KSAVE MUST BE NONZERO. DATA FOR PLOTTING IS WRITTEN ON THE
C      RESTART TAPE AT KSAVE INTERVALS
C
C      DIMENSION DATMOS(1)
C      DIMENSION RNEWK(5)
C      EQUIVALENCE (NPOW,DATMOS)
C      COMMON /FAD/ FF(5),NFLXN,FFT(5)
C      COMMON/FP/BETA,D,E,G,H
C      EQUIVALENCE (BETA,RNEWK)
C      COMMON /GA/ NPOW,NTIME,NFZ,NCZ,IPLOT,ATMOS,DCI,DCO,DFS,DSINZ,DTEMP
C      +,DVOIDZ,FR35,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRACN,FRACXE,FRD
C      +EN,FRPUO2,FRSIN,IGAS,IPEAK,LFUEL,LVOIDZ, NFUEL,NOH,PRCDH,R
C      +ADS,ROUC,ROUF,S,TINLET,TM,TPLAS,VPLENZ,XCO,XH,XN,IRELOC,IRL,IDENSF
C      +,FLXRAT,X,Z,ZZ,YY,WW,EXTP,ISTOR,NCLAD,ZCLAD,NSH,PROFIL(20),PSEUDO(
C      +100),TIME(100),NFLX,TITLE(10),MXPOWR,IOPT,JSTART,KSARE,ISAVE,ITMCU
C      +T,ISTAND,ITAPE,NSTEPS,MINI(20),SUMARY,FRDEN2
C      INTEGER SUMARY
C      REAL LFUEL,LVOIDZ,MXPOWR
C      COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN
C      +, NCNN,ISTRT,ITER
C      +, FLXR
C      COMMON/PEP/HHF,NT,KKZZZ,ISTOP,ROOLD(20),RRSIN(20),ICNOLD(20)
C      +, IDOOLD(20)
C      +,PINOLD,POUTLD,POLD,NTOLD(20)
C      +, ICHNG,CRPAD(20)
C      +,STEPS
C      COMMON /EA/ ZR(6,7),ZR4(6,7),ST(12,7),RV(20,2),UO2(7,7)
C      NAMELIST /INPUT/ATMOS,RNEWK,CP,CR, DCI,DCO,DELZA,DFS,DSINZ,DTEMP,
C      +DVOIDZ,EXTP,FLXRAT, FR35,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRAC
C      +N,FRAC
C      +XE,FRDEN,FRDEN2,FRPUO2,FRSIN,IDENSF,IGAS,IOPT,IPEAK,IPLOT,IRELOC,
C      +IFL,ISTOR,ITMCUT,JSTART,ITAPE,KSARE,LFUEL,LVOIDZ,MINI,MXPOWR,NC,
C      +NCLAD
C      +,NF,NFUEL,NFLX,NOH,NPOW,NSH,NSTEPS,NTIME,PRCDH,PSEUDO,PROFIL,RADS,
C      +ROUC,ROUF,S,TINLET,TIME,TM,TPLAS,VPLENZ,X,XCO,XH,XN,Z,ZCLAD
C      ILOC1 = LOCF(NPOW)
C      ILOC2 = LOCF(NFLX)
C      ILEN = ILOC2-ILOC1+1
C      READ (5,310) TITLE,ISTAND
C      IF (ISTAND.NE.0) GO TO 100
C      CALL MOVEKA (0,DATMOS,ILEN)
C      MXPOWR = 0.
C      ITMCUT = 0.

```

```

      NF = 6
      NC = 4
      CP = 1.
      CR = 1.
      TM = 2790.
      X = 0.27
      TPLAS = 1200.
      Z = 0.125
      FLXRAT = .01
100  READ (5,INPUT)
      CONTINUE
      NABF = NPOW+2
      NABT = NPOW+3
      NCNN = 7
      IF (NFUEL.GT.0) NCNN = NFUEL
      FLXR = FLXRAT
      IF (JSTART.GT.1) GO TO 270
      IF (NCLAD) 130,110,150
110  DO 120 I=1,7
          DO 120 J=1,6
              AA(J,I) = ZR(J,I)
              IF (ZCLAD.GT.0) AA(J,I) = ZR4(J,I)
120  CONTINUE
      NN = 6
      GO TO 160
130  DO 140 I=1,7
          DO 140 J=1,12
              AA(J,I) = ST(J,I)
140  CONTINUE
      NN = 12
      GO TO 160
150  NN = NCLAD
      READ (5,330) ((AA(J,I),I=1,7),J=1,NCLAD)
160  CONTINUE
      IF (NFLX.LT.1) GO TO 190
      READ (5,320) ZZ,YY,WW
      WRITE (6,170) ZZ,YY,WW
170  FORMAT (1H,10X,10F12.4)
      NFLXN = NFLX
      DO 180 I=1,NFLX
          RDR = (DFS-DVOIDZ)/(NFLX-1)
          RR = (DVOIDZ+FLOAT(I-1)*RDR)*2.54
          RV(1,1) = RR/2.54
          RC = RR/2.
          RV(1,2) = ZZ+YY*RC**2+WW*RC**4
180  CONTINUE
      WRITE (6,170) RV
190  CONTINUE
      IF (NFUEL.LE.0) GO TO 200
      READ (5,330) ((CF(J,I),I=1,7),J=1,NFUEL)
      GO TO 220
200  CONTINUE
      DO 210 I=1,7
          DO 210 J=1,7
              CF(I,J) = U02(I,J)
210  CONTINUE
220  CONTINUE

```

```

      INPT      44
      INPT      45
      INPT      46
      INPT      47
      INPT      48
      INPT      49
      INPT      50
      INPT      51
      INPT      52
      INPT      53
      INPT      54
      INPT      55
      INPT      56
      INPT      57
      INPT      58
      INPT      59
      INPT      60
      INPT      61
      INPT      62
      INPT      63
      INPT      64
      INPT      65
      INPT      66
      INPT      67
      INPT      68
      INPT      69
      INPT      70
      INPT      71
      INPT      72
      INPT      73
      INPT      74
      INPT      75
      INPT      76
      INPT      77
      INPT      78
      INPT      79
      INPT      80
      INPT      81
      INPT      82
      INPT      83
      INPT      84
      INPT      85
      INPT      86
      INPT      87
      INPT      88
      INPT      89
      INPT      90
      INPT      91
      INPT      92
      INPT      93
      INPT      94
      INPT      95
      INPT      96
      INPT      97
      INPT      98
      INPT      99
      INPT     100

```

	NARF = NABT-2	INPT	101
	IF (DELZA(1).EQ.0.) GO TO 250	INPT	102
	AXCK = 0.0	INPT	103
	DO 230 I=2,NARF	INPT	104
	AXCK = AXCK+DELZA(I)	INPT	105
230	CONTINUE	INPT	106
	DIFF = ABS(AXCK-LFUEL)	INPT	107
	IF (DIFF.GT.0.1) GO TO 240	INPT	108
	GO TO 270	INPT	109
240	WRITE (6,300)	INPT	110
	STOP	INPT	111
250	DLZA = LFUEL/FLOAT(NPOW)	INPT	112
	DELZA(1) = 0.5	INPT	113
	DO 260 I=2,NARF	INPT	114
	DELZA(I) = DLZA	INPT	115
260	CONTINUE	INPT	116
	DELZA(NARF+1) = VPLENZ*4./((3.141592*DCI**2)	INPT	117
270	CONTINUE	INPT	118
	NFZ = NF	INPT	119
	NCZ = NC	INPT	120
	IF (JSTART.LE.0) GO TO 280	INPT	121
	REWIND 2	INPT	122
	READ (2) ILEN,(DATMOS(J),J=1,ILEN),DELZA,AA,NN,MXPQWR,CF,RV,NF,NC,	INPT	123
	+CP,CR,D,BETA,E,G,H,NABF,NABT	INPT	124
	NABF = NPOW+2	INPT	125
	NABT = NPOW+3	INPT	126
	IF (NSTEPS.GT.0) NTIME = NSTEPS	INPT	127
	INCINC = 1	INPT	128
	TEST = TERP(200.,AA,AA(1,4),NN)	INPT	129
	IF (TEST.GT.24.E6) INCINC = 10	INPT	130
280	CONTINUE	INPT	131
	IF (KSAVE.LE.0) GO TO 290	INPT	132
	REWIND 1	INPT	133
	WRITE (1) ILEN,(DATMOS(J),J=1,ILEN),DELZA,AA,NN,MXPQWR,CF,RV,NF,NC	INPT	134
	+CP,CR,D,BETA,E,G,H,NABF,NABT	INPT	135
290	CONTINUE	INPT	136
	NCONN = NFUEL	INPT	137
	WRITE (6,INPUT)	INPT	138
	RETURN	INPT	139
C		INPT	140
C		INPT	141
C		INPT	142
C		INPT	143
C		INPT	144
C		INPT	145
300	FORMAT (1H1,10X,'ERROR DETECTED IN INPUT DATA. SUM OF FUELED DELZ	INPT	146
	+A,NE,LFUEL=)	INPT	147
310	FORMAT (10A4,5X,1S)	INPT	148
320	FORMAT (3F10.5)	INPT	149
330	FORMAT (7E10.4)	INPT	150
	END	INPT	151

	SUBROUTINE INPUT (R,ELM,NDF,NODE,NDC,NODEC,NDOFT,NOT)	INPUT	2
	DIMENSION R(1),ELM(3,1),NDF(1),NODE(1),NDC(1),NODEC(1),N	INPUT	3
	+DT(1), NODET(1)	INPUT	4
C		INPUT	5
	COMMON /POINT/ IARE, IDISPC, IDISPF, IDISPT, IELM, IEPSLM, IELSTN, IELSTR	PELCOM	2
	+, IKSARE, IPRESS, ITEL, ILAST, ILIMIT, INDC, INDF, INDT, INODE, INODEC, INOD	PELCOM	3
	+ET, IR, IRBR, ISA, ISAC, ISAF, ISB, ISBB, ISBBC, I	PELCOM	4
	+SBBF, ISBC, ISBF, ISK, ISKC, ISTRES, ITDELE, IXK, IXKC, IXKT	PELCOM	5
	+, IXLOAD, ICC, ITHET, ICA, ITHAX, INADET, IRA, IY, IAE, I	PELCOM	6
	+ARAA, IRA, IRBA, IRRA, IXLOD, ITW, ITWDEL, IDISP, ILSTRA, ILSTNA, ISTPSA, I	PELCOM	7
	+PSLA, IAK, IACN, IAN, IRZ, IAX, IRD, IYACF, ITSTNR, ITSTNA, ITSTR, ITSTA, I	PELCOM	8
	+IIDU, ITME, IEFSTR, ITCRP, ICRPG, ICRATE	PELCOM	9
	+, IXKC, IXKT	PELCOM	10
	+, ITFB, ITCB, IBBSV, IESTSV, IESTNV	PELCOM	11
	+, ITHSAV, ITHXSV, IBASAV, IPSTS, IOLDSE, IPSSV, IPSST, JIJ, IIPUS	PELCOM	12
	+, IBASE, ITR, IYM, IZ	PELCOM	13
	+, IDSPSV	PELCOM	14
	COMMON /GEOPAR/ RFI, RFO, RCI, RCO, TREF, DELR, DELZ, ZDEL, KKA, PIN, POUT, P	PELCOM	15
	+DEL	PELCOM	16
	COMMON /LIMITS/ NF, NC, NOEL, NOELF, NOELC, ICON, IDF, NFF, MAXDC, NETOT, NE	PELCOM	17
	+WF, NEWT, KA, NARF, NARC, NART, NARF, NARC, NABT, NAEC, NAEF, NTAE, JAM, NAXBT,	PELCOM	18
	+NAXB2, NAXB4, NFF1, NABF2, NABT2, CP, CR, FROEN, KZ1, ISSTOP, IPLAS, NKN	PELCOM	19
	+, ICRPP, MAXF, MAXC, MAXT, KRNCHE, KRNCHE, KRNCHE, NFF2, NC4, NF4, MAXA, NEW, N	PELCOM	20
	+MX, KRNCHE, MCUD, IADJ, ICUT, TIMEC, TIMEF, ITI, STEP, LOCK(20,2), ILK, NOLD	PELCOM	21
	+, KPLAS, ICNT, RRAKR, IRAK, RAKSUM, IPLEND, ISP	PELCOM	22
	COMMON /LASTIC/ E, XNU, ALPHA	PELCOM	23
	COMMON /CONTNT/ PI	PELCOM	24
	COMMON /TAPES/ NIN, NOUT, NSCR1, NSCR2, ITPI, ITP2	PELCOM	25
	COMMON /CRP/ PSP(25), TPLASK	PELCOM	26
C		INPUT	7
C	COMPUTE ELEMENT INCIDENSE AND ELEMENT NUMBERING SCHEMES	INPUT	8
C	ELM(J,I), I=ELEMENT, J=COORDINATE NO.	INPUT	9
C		INPUT	10
C	COMPUTE NODE POINT NUMBERS FOR EACH ELEMENT	INPUT	11
C		INPUT	12
	L = 0	INPUT	13
	DO 100 K=1, NOEL, 2	INPUT	14
	L = L+1	INPUT	15
	IF (K.EQ.(NOELF+1)) L = L+1	INPUT	16
	ELM(1,K) = L	INPUT	17
	ELM(2,K) = NF+NC+L+1	INPUT	18
	ELM(3,K) = NF+NC+L	INPUT	19
	ELM(1,K+1) = L	INPUT	20
	ELM(2,K+1) = L+1	INPUT	21
	ELM(3,K+1) = ELM(2,K)	INPUT	22
100	CONTINUE	INPUT	23
C		INPUT	24
C	NOW HAVE ELEMENT NODAL INCIDENCES.	INPUT	25
C	CREATE DEGREE OF FREEDOM REFERENCE TABLE START WITH THE FUEL	INPUT	26
C		INPUT	27
	IN = ELM(1,1)	INPUT	28
	NDF(1) = IN	INPUT	29
	NODE(IN) = 1	INPUT	30
	NDF(2) = IN	INPUT	31
	IN = ELM(2,1)	INPUT	32
	NDF(3) = IN	INPUT	33
	NODE(IN) = 3	INPUT	34

```

      NDF(4) = IN
      IN = ELM(3,1)
      NDF(5) = IN
      NODE(IN) = 5
      NDF(6) = IN
      J = 6
C
      DO 110 I=2,NOELF
        J = J+1
        IN = ELM(2,I)
        NDF(J) = IN
        NODE(IN) = J
        J = J+1
        NDF(J) = IN
110      CONTINUE
C
C      NOW CALCULATE DEGREE OF FREEDOM FOR THE CLAD
C      NOTE THAT AT THIS TIME THE CLAD AND THE FACE ARE ASSUMED
C      TO BE SEPERATE      COMPUTE A SIMILAR ARRAY FOR THE CLAD
C
      IN = ELM(1,NOELF+1)
      NDC(1) = IN
      NODEC(IN) = 1
      NDC(2) = IN
      IN = ELM(2,NOELF+1)
      NDC(3) = IN
      NODEC(IN) = 3
      NDC(4) = IN
      IN = ELM(3,NOELF+1)
      NDC(5) = IN
      NODEC(IN) = 5
      NDC(6) = IN
      J = 6
      IB = NOELF+2
      DO 120 I=IB,NOEL
        J = J+1
        IN = ELM(2,I)
        NDC(J) = IN
        NODEC(IN) = J
        J = J+1
        NDC(J) = IN
120      CONTINUE
C
      MAXDC = J
C
C      SEPERATE DEGREES OF FREEDOM ARE WRITTEN FOR THE CLAD AND THE FUEL
C      IF CLAD AND FUEL ARE IN CONTACT THE DEGREES OF FREEDOM FOR THE
C      FUEL AND CLAD MUST BE CHANGED
C      IF CLAD IS CONTACTED TOTAL DEGREES OF FREEDOM MUST BE USED
C
      NEWT = (NF+NC)*4+2+2
      NETOT = (NF+NC)*4
      NEWF = NF*4
      I = 0
C
      DO 170 III=1,NETOT

```

```

      INPUT 35
      INPUT 36
      INPUT 37
      INPUT 38
      INPUT 39
      INPUT 40
      INPUT 41
      INPUT 42
      INPUT 43
      INPUT 44
      INPUT 45
      INPUT 46
      INPUT 47
      INPUT 48
      INPUT 49
      INPUT 50
      INPUT 51
      INPUT 52
      INPUT 53
      INPUT 54
      INPUT 55
      INPUT 56
      INPUT 57
      INPUT 58
      INPUT 59
      INPUT 60
      INPUT 61
      INPUT 62
      INPUT 63
      INPUT 64
      INPUT 65
      INPUT 66
      INPUT 67
      INPUT 68
      INPUT 69
      INPUT 70
      INPUT 71
      INPUT 72
      INPUT 73
      INPUT 74
      INPUT 75
      INPUT 76
      INPUT 77
      INPUT 78
      INPUT 79
      INPUT 80
      INPUT 81
      INPUT 82
      INPUT 83
      INPUT 84
      INPUT 85
      INPUT 86
      INPUT 87
      INPUT 88
      INPUT 89
      INPUT 90
      INPUT 91

```

SUBROUTINE INPUT

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	IF (III.GT.NEWF) GO TO 130	INPUT	92
	I = I+1	INPUT	93
	IN = NDF(I)	INPUT	94
	NDT(I) = IN	INPUT	95
	NODET(IN) = NODE(IN)	INPUT	96
	GO TO 170	INPUT	97
130	CONTINUE	INPUT	98
	II = III-NEWF	INPUT	99
	IN = NDC(II)	INPUT	100
	IF (II.LE.2) GO TO 140	INPUT	101
	IF (II.GT.4.AND,II.LE.6) GO TO 150	INPUT	102
	GO TO 160	INPUT	103
140	CONTINUE	INPUT	104
	NODET(IN) = NODE(NF)	INPUT	105
	GO TO 170	INPUT	106
150	CONTINUE	INPUT	107
	NODET(IN) = NODE(2*NF+NC)	INPUT	108
	GO TO 170	INPUT	109
160	CONTINUE	INPUT	110
	I = I+1	INPUT	111
	NDT(I) = IN	INPUT	112
	NODET(IN) = I-1	INPUT	113
170	CONTINUE	INPUT	114
C		INPUT	115
C	HAVE NOW SET UP ARRAY TO COMPUTE POSITION AND D O F FOR BOTH FUEL	INPUT	116
C	AND CLAD AS WELL AS FOR CONTACTING FUEL AND CLAD	INPUT	117
C		INPUT	118
C		INPUT	119
C	RETURN	INPUT	120
C		INPUT	121
	END	INPUT	122

SEVERITY DETAILS DIAGNOSIS OF PROBLEM

I	NDF	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDF	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDF	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDF	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDF	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDC	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDC	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDC	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDC	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.
I	NDC	ARRAY REFERENCE OUTSIDE DIMENSION BOUNDS.

```

SUBROUTINE LOAD (ARE, B, BB, BC, BF, BRC, BBF, NODE, NODEC, NODET, R, RBR, CC, LOAD 2
+THETEO, TDFLE, TELM, XLOAD, ELM, NOELD, TCRP, NABFO, KAN, IDO, THSAV) LOAD 3
C LOAD 4
C PURPOSE OF THIS SUBROUTINE IS TO COMPUTE THE COMBINED LOAD VECTOR LOAD 5
C FOR SOLUTION LOAD 6
C FOLLOWING LOADS ARE ALL COMBINED INTO THE BASED VECTOR LOAD 7
C APPLIED LOADS LOAD 8
C THERMAL LOADS LOAD 9
C CREEP LOADS LOAD 10
C PLASTIC LOADS LOAD 11
C LOAD 12
C DIMENSION TCRP(4, NOELD, 1) LOAD 13
C DIMENSION THRMVS(50), THSAV(4, NOELD, 1) LOAD 14
C DIMENSION ARE(1), B(1), BB(1), BRC(1), NODE(1), NODEC(1), NODET(1) LOAD 15
+ , R(1), RBR(1), TDELE(NOELD), TELM(NOELD, NABFO, 2) LOAD 16
+ , XLOAD(6, 4, 1) LOAD 17
+ , BC(1), BF(1), BBF(1), THERM(6), ELM(3, 1), CC(4, 4, 1), THETEO(4, 1) LOAD 18
+ ) LOAD 19
C DIMENSION KAN(1), IDO(1) LOAD 20
C COMMON /POINT/ IARE, IDISPC, IDISPF, IDISPT, IELM, IEPSTM, IELSTN, IELSTR PELCOM 2
+ , IKSAYE, IPRESS, ITLM, ILAST, ILMIT, INDC, INDF, INOT, INODE, INODEC, INOD PELCOM 3
+ ET, IR, IRBR, ISA, ISAC, ISAF, ISB, ISBB, ISBBC, I PELCOM 4
+ SBBF, ISBC, ISBF, ISK, ISKC, ISTRES, ITDELE, IXX, IXXC, IXXT PELCOM 5
+ , IXLOAD, ICC, ITHET, ICA, ITHTX, INADET, IRA, IY, IAEHM, I PELCOM 6
+ ARAA, ISA, IRBA, IRBRA, IXLDA, ITW, ITWDEL, IDISP, ILSTRA, ILSTNA, ISTRSA, I PELCOM 7
+ PSLA, IAK, IACN, IKAN, IPZ, IAXX, IRD, IYACF, ITSTNR, ITSTNA, ITSTSR, ITSTSA, I PELCOM 8
+ IDO, ITME, IEFSTR, ITCRP, ICRPG, ICRATE PELCOM 9
+ , IXXCC, IXXTT PELCOM 10
+ , ITFB, ITCB, IBBVS, IESTSV, IESTNV PELCOM 11
+ , ITHSAV, ITHXSV, IBASAV, IPSTS, IULDSE, IPSSV, IPSSRT, JIJU, IIPDS PELCOM 12
+ , IBASE, ITFR, IYM, IZ PELCOM 13
+ , IDSPSV PELCOM 14
C COMMON /GEOPAR/ RFI, RFO, RCI, RCO, TREF, DELR, DELZ, ZDEL, KKA, PIN, POUT, P PELCOM 15
+ DEL PELCOM 16
C COMMON /LIMITS/ NF, NC, NOEL, NOELF, NOELC, ICON, IDF, NFF, MAXDC, NETOT, NE PELCOM 17
+ WF, NEWT, KA, NARF, NARC, NART, NARF, NABC, NABT, NAEC, NAEF, NTAE, JAM, NAXBT, PELCOM 18
+ NAXB2, NAXB4, NFF1, NABF2, NABT2, CP, CR, FRDEN, KZ1, ISSTOP, IPLAS, NKN PELCOM 19
+ , ICRPP, MAXF, MAXC, MAXT, KRNCHE, KRNCHE, KRNCHE, NFF2, NC4, NF4, MAXA, NEW, N PELCOM 20
+ HX, KRNCHE, MCUD, IADJ, ICUT, TIMEC, TIMEF, ITI, STEP, LOCK(20, 2), ILK, NOLD PELCOM 21
+ , KPLAS, ICNT, RRAKR, IRAK, RAKSUM, IPLEND, ISP PELCOM 22
C COMMON /LASTIC/ E, XNU, ALPHA PELCOM 23
C COMMON /CONTNT/ PI PELCOM 24
C COMMON /TAPES/ NIN, NOUT, NSCR1, NSCR2, ITP1, ITP2 PELCOM 25
C COMMON /CRP/ PSP(25), TPLASK PELCOM 26
C COMMON /BA/ AA(23, 7), CF(7, 8), RKS, DELZA(20), NN, NCONN COMC 2
+ , NCNN, ISTRY, ITER COMC 3
+ , FLXR COMC 4
C COMMON /FEP/ HMF, NT, KKZZZ, ISTOP, RDOLD(20), RRSIN(20), ICNOLD(20) COMC 5
+ , IDOOLD(20) COMC 6
+ , PINOLD, POUTLD, POLD, NTOLD(20) COMC 7
+ , ICHNG, CRPAD(20) COMC 8
+ , STEPS COMC 9
C LOAD 24
C DIMENSION NNA(3) LOAD 24
C LOAD 25
C CALL MOVEKA (0, THETEO, 4*NOEL) LOAD 26
NFF2 = 4*(NF+NC) LOAD 27

```

```

      NC4 = NC*4
      NF4 = NF*4
      DO 100 I=1,6
      THERM(I) = 0.0
100  CONTINUE
      IF (NKN,EQ,2) GO TO 120
      FRAO = PI*2.0*R(NF+NC)*POUT*DELZ/2.0
      FRAI = PI*2.0*R(NF+1)*PIN*DELZ/2.0
      C
      NFF1 = NFF/2
      II = NODEC(NFF1)
      IJ = NODEC(NFF)
      C
      C
      C COMPUTE THE RADIAL LOAD VECTORS FOR THE PRESSURE DIFFERENTIAL
      RBC(II) = -FRAO
      BBC(IJ) = -FRAO
      BC(II) = -FRAO
      BC(IJ) = -FRAO
      II = NODEC(NF+1)
      IJ = NODEC(NFF1+NF+1)
      BBC(II) = FRAI
      BBC(IJ) = FRAI
      BC(II) = FRAI
      BC(IJ) = FRAI
      C
      C
      C COMPUTE THE COMBINED LOAD VECTOR FOR CLAD CONTACT-CONTINUED SYSTEM
      II = NODET(NFF1)
      IJ = NODET(NFF)
      BB(II) = -FRAO
      BB(IJ) = -FRAO
      B(II) = -FRAO
      B(IJ) = -FRAO
      II = NODET(NF+1)
      IJ = NODET(NFF1+NF+1)
      BB(II) = FRAI
      BB(IJ) = FRAI
      B(II) = FRAI
      B(IJ) = FRAI
      C
      IF (KKA,EQ,2) GO TO 120
      C
      C IN A SIMILAR MANNER THE AXIAL LOAD VECTOR FOR THE FUEL AND CLAD
      C MUST BE CREATED
      C PAXL= AXIAL LOAD IN CLAD DUE TO EXTERNAL PRESSURE
      C
      PAXL = ((POUT*PI*RCO**2)-(PIN*PI*RCI**2))
      C
      C CLAD ELEMENTS DUE TO EXTERNAL PRESSURE NOW DIVIDE THE AXIAL LOADS
      C OVER THE NUMBER OF ELEMENTS ALONG THE TOP SURFACE OF THE CLAD
      C MODEL. THIS CAN BEST BE DONE BY COMPUTING NODAL LOAD CIRCLES.
      C DEFINE LIMITS FOR THE DEGREES OF FREEDOM IN THE LOAD VECTOR
      C
      LL = 2*NF+NC+1
      LU = 2*(NF+NC)-1
      C
      C COMPUTE LOAD VECTOR FOR THE CLAD ALONG NODAL CIRCLES BASE STRESS
      C CALCULATIONS ON THE RADIAL POSITION

```

```

      LOAD 28
      LOAD 29
      LOAD 30
      LOAD 31
      LOAD 32
      LOAD 33
      LOAD 34
      LOAD 35
      LOAD 36
      LOAD 37
      LOAD 38
      LOAD 39
      LOAD 40
      LOAD 41
      LOAD 42
      LOAD 43
      LOAD 44
      LOAD 45
      LOAD 46
      LOAD 47
      LOAD 48
      LOAD 49
      LOAD 50
      LOAD 51
      LOAD 52
      LOAD 53
      LOAD 54
      LOAD 55
      LOAD 56
      LOAD 57
      LOAD 58
      LOAD 59
      LOAD 60
      LOAD 61
      LOAD 62
      LOAD 63
      LOAD 64
      LOAD 65
      LOAD 66
      LOAD 67
      LOAD 68
      LOAD 69
      LOAD 70
      LOAD 71
      LOAD 72
      LOAD 73
      LOAD 74
      LOAD 75
      LOAD 76
      LOAD 77
      LOAD 78
      LOAD 79
      LOAD 80
      LOAD 81
      LOAD 82
      LOAD 83
      LOAD 84

```

C	TOT = PI*(R(NFF)**2-R(LL)**2)	LOAD	85
	SIG = -PAXL/TOT	LOAD	86
C		LOAD	87
	DO 110 I=LL,LU	LOAD	88
	AREA = PI*(R(I+1)**2-R(I)**2)	LOAD	89
	XLINE = PI*(2.*R(I+1)+2.*R(I))	LOAD	90
C		LOAD	91
C	COMPUTE LINE LOAD FOR NODE 1	LOAD	92
C		LOAD	93
	ZF = SIG*AREA/XLINE	LOAD	94
	ZFS = ZF*PI*2.*R(I)	LOAD	95
	ZFE = ZF*PI*2.*R(I+1)	LOAD	96
C		LOAD	97
C	NOTE MUST INDEX D.O.F. TO AXIAL DIRECTION	LOAD	98
C		LOAD	99
	IN = NODEC(I)+1	LOAD	100
	INN = NODEC(I+1)+1	LOAD	101
	BBC(IN) = BBC(IN)+ZFS	LOAD	102
	BC(IN) = BBC(IN)	LOAD	103
	BBC(INN) = BBC(INN)+ZFE	LOAD	104
	BC(INN) = BBC(INN)	LOAD	105
	IN = NODET(I)+1	LOAD	106
	INN = NODET(I+1)+1	LOAD	107
	BB(IN) = BB(IN)+ZFS	LOAD	108
	B(IN) = BB(IN)	LOAD	109
	BB(INN) = BB(INN)+ZFE	LOAD	110
	B(INN) = BB(INN)	LOAD	111
110	CONTINUE	LOAD	112
C		LOAD	113
C	HAVE COMPUTED AXIAL LOADS DUE TO PRESSURE DIFFERENTIAL	LOAD	114
C		LOAD	115
C	NOW COMPUTE THERMAL LOAD VECTOR. USING XLOAD(I,J,KI)	LOAD	116
C		LOAD	117
C	CONTINUE	LOAD	118
120	SUMTHF = 0.0	LOAD	119
	DO 210 KI=1,NOEL	LOAD	120
C		LOAD	121
	IF (KI.LE.NOELF.AND.ICON.EQ.0) GO TO 210	LOAD	122
	IF (NKN.EQ.2) GO TO 200	LOAD	123
C	CALL TERP FOR ALPHA	LOAD	124
C		LOAD	125
	TC = TELM(KI,KA,NEW)	LOAD	126
	IF (KI.GT.NOELF) GO TO 180	LOAD	127
	IF (NCONN.GT.0) GO TO 130	LOAD	128
	TCC = (TC-32.)/1.8	LOAD	129
	ALPHA = FULEX(TCC)	LOAD	130
	GO TO 140	LOAD	131
130	CONTINUE	LOAD	132
	ALPHA = TERP(TC,CF,CF(1,6),NCONN)	LOAD	133
140	CONTINUE	LOAD	134
	IF (IRZ.GT.1) GO TO 160	LOAD	135
	IF (NCONN.GT.0) GO TO 150	LOAD	136
	THET = ALPHA	LOAD	137
	GO TO 200	LOAD	138
150	CONTINUE	LOAD	139
	THET = ALPHA+TDELE(KI)	LOAD	140
		LOAD	141

	GO TO 200	LOAD	142
160	ALPH1 = ALPHA	LOAD	143
	IF (NCONN.GT.0) GO TO 170	LOAD	144
	TLD = TCG-TDELE(KI)/1.8	LOAD	145
	ALPH2 = FULEX(TLD)	LOAD	146
	THET = ALPH1-ALPH2	LOAD	147
	GO TO 200	LOAD	148
170	CONTINUE	LOAD	149
	TLD = TC-TDELE(KI)	LOAD	150
	ALPH2 = TERP(TLD,CF,CF(1,6),NCONN)	LOAD	151
	THET = ALPH1*(TC-TREF)-ALPH2*(TLD-TREF)	LOAD	152
	GO TO 200	LOAD	153
180	CONTINUE	LOAD	154
	ALPHA = TERP(TC,AA,AA(1,6),NN)	LOAD	155
	IF (IRZ.GT.1) GO TO 190	LOAD	156
	THET = ALPHA*TDELE(KI)	LOAD	157
	GO TO 200	LOAD	158
190	ALPH1 = ALPHA	LOAD	159
	TLD = TC-TDELE(KI)	LOAD	160
	ALPH2 = TERP(TLD,AA,AA(1,6),NN)	LOAD	161
	THET = ALPH1*(TC-TREF)-ALPH2*(TLD-TREF)	LOAD	162
200	CONTINUE	LOAD	163
	THRMSV(KI) = THET	LOAD	164
	IF (KI.GT.NOELF) GO TO 210	LOAD	165
	SUMTHF = SUMTHF+THET	LOAD	166
210	CONTINUE	LOAD	167
	THAVF = SUMTHF/LOAT(NOELF)	LOAD	168
	DO 320 KI=1,NOEL	LOAD	169
	THET = THAVF	LOAD	170
	IF (KI.GT.NOELF) THET = THRMSV(KI)	LOAD	171
	IF (NKN.EQ.2) THET = 0.0	LOAD	172
C		LOAD	173
	PR = 2.0*PI*ARE(KI)*RBR(KI)	LOAD	174
C		LOAD	175
	DO 240 I=1,6	LOAD	176
	SUM = 0.0	LOAD	177
	DO 220 J=1,3	LOAD	178
	S = PR*XLOAD(I,J,KI)*THET	LOAD	179
	SUM = SUM+S	LOAD	180
220	CONTINUE	LOAD	181
	SCUM = 0.0	LOAD	182
	DO 230 J=1,4	LOAD	183
	SC = PR*XLOAD(I,J,KI)*TERP(J,KI,KA)	LOAD	184
	SCUM = SCUM+SC	LOAD	185
230	CONTINUE	LOAD	186
	SUM = SUM+SCUM	LOAD	187
	THERM(I) = SUM	LOAD	188
240	CONTINUE	LOAD	189
C		LOAD	190
C	NOW COMPUTE THE CORRECT DEGREES OF FREEDOM AND POSITION IN THE	LOAD	191
C	DISPLACEMENT ARRAY OR D.O.F. TO MERGE THE INTO THE LOAD VECTORS	LOAD	192
	DO 250 I=1,3	LOAD	193
	THETEO(I,KI) = THET	LOAD	194
	IF (NKN.EQ.1.AND.ISSTOP.EQ.1.AND.KKA.EQ.2) THSAV(I,KI,KA) =	LOAD	195
	THETEO(I,KI)	LOAD	196
250	CONTINUE	LOAD	197
	THETEO(4,KI) = 0.0	LOAD	198

C	ADD CREEP STRAIN TO STRESS CORRECTION TERM	LOAD	199
	DO 260 K=1,4	LOAD	200
	THETEO(K,KI) = THETEO(K,KI)+TCPP(K,KI,KA)	LOAD	201
260	CONTINUE	LOAD	202
C	NOW HAVE STRAIN CORRECTION TERM FOR LOAD AND STRESS	LOAD	203
C	FOR THE FUEL, THE CLAD, AND THE COMBINED CASE	LOAD	204
C		LOAD	205
	NNA(1) = ELM(1,KI)	LOAD	206
	NNA(2) = ELM(2,KI)	LOAD	207
	NNA(3) = ELM(3,KI)	LOAD	208
C		LOAD	209
	IF (ICON.EQ.1) GO TO 300	LOAD	210
	IF (KI.GT.NOELF) GO TO 280	LOAD	211
C		LOAD	212
	IF (KI.LE.NOELF.AND.ICON.EQ.0) GO TO 320	LOAD	213
C	COMPUTE LOAD FOR THE FUEL	LOAD	214
	DO 270 I=1,3	LOAD	215
C		LOAD	216
	J = 2*I-1	LOAD	217
	NNI = NNA(I)	LOAD	218
	II = NODE(NNI)	LOAD	219
	BBF(II) = BBF(II)+THERM(J)	LOAD	220
	BBF(II+1) = BBF(II+1)+THERM(J+1)	LOAD	221
	BF(II) = BBF(II)	LOAD	222
	BF(II+1) = BBF(II+1)	LOAD	223
270	CONTINUE	LOAD	224
C		LOAD	225
280	CONTINUE	LOAD	226
C		LOAD	227
C	COMPUTE THE LOAD FOR THE CLAD	LOAD	228
C		LOAD	229
	DO 290 I=1,3	LOAD	230
	J = 2*I-1	LOAD	231
	NNI = NNA(I)	LOAD	232
	II = NODE(NNI)	LOAD	233
	BBC(II) = BBC(II)+THERM(J)	LOAD	234
	BBC(II+1) = BBC(II+1)+THERM(J+1)	LOAD	235
	BC(II) = BBC(II)	LOAD	236
	BC(II+1) = BBC(II+1)	LOAD	237
290	CONTINUE	LOAD	238
C		LOAD	239
300	CONTINUE	LOAD	240
	IF (ICON.NE.1) GO TO 320	LOAD	241
C		LOAD	242
C	COMPUTE LOAD VECTOR FOR CONTACT BETWEEN FUEL AND CLAD	LOAD	243
C		LOAD	244
	DO 310 I=1,3	LOAD	245
	J = 2*I-1	LOAD	246
	NNI = NNA(I)	LOAD	247
	II = NODE(NNI)	LOAD	248
	BB(II) = BB(II)+THERM(J)	LOAD	249
	BB(II+1) = BB(II+1)+THERM(J+1)	LOAD	250
	B(II) = BB(II)	LOAD	251
	B(II+1) = BB(II+1)	LOAD	252
310	CONTINUE	LOAD	253
320	CONTINUE	LOAD	254
	IF (KAN(KA).EQ.0.AND.IDO(KA).EQ.1) GO TO 340	LOAD	255

SUBROUTINE LOAD

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

C
  IF (ICON.EQ.0) GO TO 340
  II = NF+NC
  DO 330 I=1,II
    IN = NODET(I)+1
    BB(IN) = 0.0
    B(IN) = 0.0
330   CONTINUE
    GO TO 370
340   CONTINUE
    DO 350 I=1,NF
      IN = NODE(I)+1
      BBF(IN) = 0.0
      BF(IN) = 0.0
350   CONTINUE
      DO 360 I=1,NC
        JJ = NF+I
        IN = NODEC(JJ)+1
        BBC(IN) = 0.0
        BC(IN) = 0.0
360   CONTINUE
370   CONTINUE
      RETURN
      END

```

```

LOAD 256
LOAD 257
LOAD 258
LOAD 259
LOAD 260
LOAD 261
LOAD 262
LOAD 263
LOAD 264
LOAD 265
LOAD 266
LOAD 267
LOAD 268
LOAD 269
LOAD 270
LOAD 271
LOAD 272
LOAD 273
LOAD 274
LOAD 275
LOAD 276
LOAD 277
LOAD 278
LOAD 279

```

SUBROUTINE MOVEAA 74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```
      SUBROUTINE MOVEAA (A,B,N)
      DIMENSION A(N), B(N)
      ENTRY MOVEAB
      DO 100 I=1,N
      B(I) = A(I)
100    CONTINUE
      RETURN
      END
```

```
MOVEAA 2
MOVEAA 3
MOVEAA 4
MOVEAA 5
MOVEAA 6
MOVEAA 7
MOVEAA 8
MOVEAA 9
```

SUBROUTINE MOVEKA 74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```
      SUBROUTINE MOVEKA (K,IA,N)
      DIMENSION IA(N)
      DO 100 I=1,N
        IA(I) = K
      100 CONTINUE
      RETURN
      END
```

```
MOVEKA 2
MOVEKA 3
MOVEKA 4
MOVEKA 5
MOVEKA 6
MOVEKA 7
MOVEKA 8
```

```

SUBROUTINE OUTPUT (CGR,GK,G1PG2,HG,HGAS,HRRAD,HBSOLID,IPOW,NY,JC4SE, OUTPUT      2
+MOLEFR,LPMAX,PFACE,PRESTO,VAVG,XTOTOT,QOVRAC,QOVRAS,TCC,TCOOL,DEL OUTPUT      3
+RFT,TTTT,TSTSR,NF,NC,NFPNC,TFB,TCB,RD,TSTNR,NNOEL,NOELD,EFSTR,NPOW OUTPUT      4
+1,SUMFF,SUMM) OUTPUT      5
COMMON /GA/ NPOW,NTIME,NFZ,NCZ,IPLDT,ATMOS,DCI,DCO,DFS,DSINZ,DTEMP COMA      2
+,DVOIDZ,FR35,FR40,FR41,FRACAR,FRACH,FRACHE,FRACKR,FRACN,FRACX,FRD COMA      3
+EN,FRPUO2,FRBIN,IGAS,IPEAK,LFUEL,LVOIDZ, NFUEL,NNH,PRCDH,R COMA      4
+ADS,ROUC,ROUF,S,TINLET,TM,TPLAS,VPLENZ,XCO,XH,XN,IRELOC,IRL,IDENSF COMA      5
+,FLXRAT,X,Z,ZZ,YY,WW,EXTP,ISTDR,NCLAD,ZCLAD,NSH,PROFIL(20),PSEUDO( COMA      6
+100),TIME(100),NFLX,TITLE(10),MXPOWR,IQPT,JSTART,KSAVE,ISAVE,ITMCU COMA      7
+T,ISTAND,ITAPE,NSTEPS,MINI(20),SUMARY,FRDENZ COMA      8
INTEGER SUMARY COMA      9
REAL LFUEL,LVOIDZ,MXPOWR COMA      10
COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN COMC      2
+, NCNN,ISTRT,ITER COMC      3
+, FLXR COMC      4
COMMON/PEP/HMF,NT,KKZZZ,ISTOP,RDOLD(20),RRSIN(20),ICNOLD(20) COMC      5
+, IDOOLD(20) COMC      6
+,PINOLD,POUTLD,POLD,NTOLD(20) COMC      7
+, ICHNG,CRPAD(20) COMC      8
+,STEPS COMC      9
COMMON/DA/ TABLE(80,2),GMWT(7),SIGLJ(7),EKLJ(7),HF(20),RTCO(5) COMD      2
+,RTC(5),EPSIF,EP8IC COMD      3
COMMON /FA/ PI,CCPIN3,SECDAY,AVOGAD,RR,CONEN COMF      2
COMMON /AZ/ BURNUP(20),DELPI(20),DELR,DELRB(20),DELRCC(20),DELRCI( COMI      2
+20),DELRCT(20),DELRP(20),DELR(20),DSIN,DVOID,FRUO2,FR38,FR39,GAP, COMI      3
+HBC,KB,DELGDH(20) COMI      4
COMMON /OUT/ POWER(20),RGAPX(20),RDOT(50,20),XMOLS(20),RCII,RDELST OUTPUT      12
+(20),ISTART,CRELST(20),CRCRP(20),IACON(20,2),ESTORI(20) OUTPUT      13
REAL MOLEFR,MLFR OUTPUT      14
DIMENSION MOLEFR(7) OUTPUT      15
DIMENSION TSTSR(4,NOELD,15), TSTNR(4,NOELD,15) OUTPUT      16
DIMENSION MLFR(7), EFSTR(NOELD,1) OUTPUT      17
DIMENSION TFB(NF), TCB(NC), RD(NFPNC) OUTPUT      18
DIMENSION WRT(12) OUTPUT      19
C OUTPUT      20
C CONVERT AVG. AND PEAK LINEAR POWER FROM KW/FT TO W/M OUTPUT      21
PEAK = POWER(LPMAX) OUTPUT      22
P = POWER(IPOW) OUTPUT      23
SIP = P*3280.84 OUTPUT      24
SIPK = PEAK*3280.84 OUTPUT      25
C OUTPUT      26
C CONVERT STORED ENERGY FROM J/CM TO J/M OUTPUT      27
ESTORJ = ESTORI(IPOW)*100. OUTPUT      28
C OUTPUT      29
C CONVERT TIME IN-REACTOR FROM DAYS TO SECONDS OUTPUT      30
SITIM = TTTT*SECDAY OUTPUT      31
C OUTPUT      32
C CONVERT BURNUP FROM MWD/MTM TO J/KGM OUTPUT      33
SIBRN = BURNUP(IPOW)*A.64E+7 OUTPUT      34
C OUTPUT      35
C CONVERT HEAT FLUXES FROM BTU/HR-FT2 TO W/M2 OUTPUT      36
SIGAC = QOVRAC*3.15248 OUTPUT      37
SIGAS = QOVRAS*3.15248 OUTPUT      38
C OUTPUT      39
C CONVERT FILM COEFFICIENT AND CONDUCTANCES FROM BTU/HR-FT2-DEG.F TO OUTPUT      40
W/M2-DEG.C(INTERNATIONAL TABLES) OUTPUT      41

```

	SIMF = HF(IPOW)*5.678263	OUTPUT	42
	SIHG = HG*5.678263	OUTPUT	43
	MSLD = (MSQLJD/HG)*100.	OUTPUT	44
	HGS = (HGAS/HG)*100.	OUTPUT	45
	HRD = (HRAD/HG)*100.	OUTPUT	46
C		OUTPUT	47
C	CONVERT RADIAL DIMENSIONS FROM INCHES TO METERS	OUTPUT	48
	IF (PFACE.GT.0.0) DELRFT = RD(NF)-DF9/2.	OUTPUT	49
	IF (PFACE.GT.0.) DELRT(IPOW) = DELRFT-DELRB(IPOW)-DELGDM(IPOW)-DEL	OUTPUT	50
	+PI(IPOW)	OUTPUT	51
	SIDLRT = DELRT(IPOW)*0.0254	OUTPUT	52
	SIDLRB = DELRB(IPOW)*0.0254	OUTPUT	53
	SIDLR = DELRFT*0.0254	OUTPUT	54
	RGAPX(IPOW) = RDELST(IPOW)-RD(NF)	OUTPUT	55
	SIRGAP = RGAPX(IPOW)*0.0254	OUTPUT	56
	SIGT = GIPG2*0.0254	OUTPUT	57
	SIRLT = DELGDM(IPOW)*0.0254	OUTPUT	58
	SIDFT = DELPI(IPOW)*0.0254	OUTPUT	59
	SICRP = DELRCC(IPOW)*0.0254	OUTPUT	60
C		OUTPUT	61
C	CONVERT DEG.F TO DEG.C	OUTPUT	62
	SITCL = (TCOOL-32.)/1.8	OUTPUT	63
	VAVGTC = (VAVGT-32.)/1.8	OUTPUT	64
C		OUTPUT	65
C	CONVERT CLAD AND GAS THERMAL CONDUCTIVITY FROM BTU/HR-FT-DEG.F	OUTPUT	66
C	TO W/M-DEG.C*(INTERNATIONAL TABLES)	OUTPUT	67
	SICC = TCC*1.73073	OUTPUT	68
	SIGK = GK*1.73073	OUTPUT	69
C		OUTPUT	70
C	CONVERT INTERFACIAL AND GAS PRESSURES FROM PSI TO PA	OUTPUT	71
	PFACE = ABS(PFACE)	OUTPUT	72
	SIPFC = PFACE*6894.757	OUTPUT	73
	SIPRES = PRESTO*6894.757	OUTPUT	74
C		OUTPUT	75
C	CONVERT MOLES FROM GRAM-MOLES TO KILOGRAM-MOLES	OUTPUT	76
	SIXMT = XMOTOT/1000.0	OUTPUT	77
	NPOW2 = NPOW1+1	OUTPUT	78
	WRITE (6,340) JCASE,IPOW,NPOW2,TITLE,ISAVE	OUTPUT	79
	IF (ISTOR.NE.0) GO TO 100	OUTPUT	80
	WRITE (6,350) SIP,P,SIPK,PEAK,SITIM,TTTT,SIBRN,BURNUP(IPOW)	OUTPUT	81
	GO TO 110	OUTPUT	82
100	WRITE (6,360) SIP,P,SIPK,PEAK,ESTORJ,SITIM,TTTT,SIBRN,BURNUP(IPOW)	OUTPUT	83
110	CONTINUE	OUTPUT	84
	WRITE (6,370) SIGAC,QUVRAC,SIGAS,QUVRAS,SITCL,TCOOL	OUTPUT	85
	WRITE (6,380) SIMF,HF(IPOW),SICC,TCC	OUTPUT	86
	VAVGTC = (VAVGT-32.)/1.8	OUTPUT	87
	WRITE (6,390) VAVGTC,VAVGT	OUTPUT	88
	IF (ITAPE.GT.1) GO TO 140	OUTPUT	89
	IF (NT.GT.1) GO TO 120	OUTPUT	90
	CRELST(IPOW) = CRELST(IPOW)+RDELST(IPOW)-RCII	OUTPUT	91
	GO TO 130	OUTPUT	92
120	CRELST(IPOW) = CRELST(IPOW)+RDELST(IPOW)-RDDLO(IPOW)	OUTPUT	93
130	CONTINUE	OUTPUT	94
	CRCRP(IPOW) = CRCRP(IPOW)+CRPAD(IPOW)	OUTPUT	95
140	CONTINUE	OUTPUT	96
	CRTOT = CRELST(IPOW)+CRCRP(IPOW)	OUTPUT	97
	SICRSI = CRELST(IPOW)*0.0254	OUTPUT	98

```

      SICRP = CRCRP(IPOW)*0.0254          OUTPUT 99
      SICRTT = CRTOT*0.0254             OUTPUT 100
      RDOLD(IPOW) = RD(NF+1)            OUTPUT 101
      WRITE (6,410) SIDLRT,DELRT(IPOW),SIRLT,DELGDM(IPOW),SIDFT,DELP1(IP
+OW),SIDLRB,DELRB(IPOW),SIDLR,DELRFT,SICRST,CRELST(IPOW),SICRP,CRCR
+P(IPOW),SICRTT,CRTOT                   OUTPUT 102
      WRITE (6,280)                      OUTPUT 103
      DO 150 I=1,NF                     OUTPUT 104
      TFBC = (TFB(I)-32.)/1.8           OUTPUT 105
      RDM = RD(I)*0.0254                OUTPUT 106
      WRITE (6,290) TFBC,TFB(I),RDM,RD(I) OUTPUT 107
150  CONTINUE                          OUTPUT 108
      TCBC = (TCB(I)-32.)/1.8           OUTPUT 109
      I = NF+1                          OUTPUT 110
      RDM = RDELST(IPOW)*0.0254         OUTPUT 111
      WRITE (6,300) TCBC,TCB(I),RDM,RDELST(IPOW) OUTPUT 112
      I = NF+NC                         OUTPUT 113
      RDM = RD(I)*0.0254               OUTPUT 114
      TCBC = (TCB(NC)-32.)/1.8         OUTPUT 115
      WRITE (6,310) TCBC,TCB(NC),RDM,RD(I) OUTPUT 116
      WRITE (6,420) SIRGAP,RGAPX(IPOW),SIPFC,PFACE OUTPUT 117
      WRITE (6,430) SHG,HG              OUTPUT 118
      WRITE (6,440) HSLD,HGS,HRD        OUTPUT 119
      WRITE (6,460) SIGK,GK             OUTPUT 120
      WRITE (6,470) SIGT,G1PG2          OUTPUT 121
      DO 160 I=1,7                     OUTPUT 122
160  MLFR(I) = MOLEFR(I)               OUTPUT 123
      WRITE (6,400) RDUT(NY,IPOW),CGR,SIPRES,PRESTO,MIXMT OUTPUT 124
      WRITE (6,450) (MLFR(I),I=1,7)     OUTPUT 125
      WRITE (6,480) IPOW                OUTPUT 126
      WRITE (6,320)                     OUTPUT 127
      WRITE (6,330)                     OUTPUT 128
      C                                OUTPUT 129
      C                                OUTPUT 130
      C                                OUTPUT 131
      CONVERT STRESSES TO KSI AND STRAINS TO PERCENT. OUTPUT 132
      C                                OUTPUT 133
      C                                OUTPUT 134
      DO 170 I=1,4                     OUTPUT 135
170  CONTINUE                          OUTPUT 136
      NTEST = NOELD=NNOEL+1            OUTPUT 137
      NOELF = NNOEL-1                  OUTPUT 138
      DO 260 IN=1,NOELF                OUTPUT 139
      INC = IN+NOELF                   OUTPUT 140
      WRT(1) = TSTNR(1,IN,IPOW)*100.   OUTPUT 141
      WRT(2) = TSTNR(2,IN,IPOW)*100.   OUTPUT 142
      WRT(3) = TSTSR(1,IN,IPOW)/1000.  OUTPUT 143
      WRT(4) = TSTSR(2,IN,IPOW)/1000.  OUTPUT 144
      WRT(5) = EFSTR(IN,IPOW)/1000.    OUTPUT 145
      IF (IN.GT,NTEST) GO TO 180        OUTPUT 146
      WRT(6) = TSTNR(1,INC,IPOW)*100.   OUTPUT 147
      WRT(7) = TSTNR(2,INC,IPOW)*100.   OUTPUT 148
      WRT(8) = TSTSR(1,INC,IPOW)/1000.  OUTPUT 149
      WRT(9) = TSTSR(2,INC,IPOW)/1000.  OUTPUT 150
      WRT(10) = EFSTR(INC,IPOW)/1000.   OUTPUT 151
      IF (IN.GT,2) GO TO 220            OUTPUT 152
      IF (IN.EQ,1) WRT(11) = TSTNR(3,IN,IPOW)*100. OUTPUT 153
      IF (IN.EQ,1) WRT(12) = TSTSR(3,INC,IPOW)/1000. OUTPUT 154
      IF (IN.EQ,2) WRT(11) = TSTSR(3,IN,IPOW)/1000. OUTPUT 155
      IF (IN.EQ,2) WRT(12) = TSTSR(3,INC,IPOW)/1000.

```

```

      WRITE (6,270) (WRT(I),I=1,12)
      GO TO 260
180   IF (IN.EQ.4) WRITE (6,190) (WRT(I),I=1,5)
190   FORMAT (5X,2(F6.4,2X),2X,3(F7.2,2X),53X,#PIN AXIAL DISPLACEMENT#)
      IF (IN.EQ.5) WRITE (6,200) (WRT(I),I=1,5)
200   FORMAT (5X,2(F6.4,2X),2X,3(F7.2,2X),49X,9X,#FUEL CLAD#)
      IF (IN.EQ.6) WRITE (6,210) (WRT(I),I=1,5),SUMFF,SUMH
      IF (IN.GT.3.AND.IN.LT.7) GO TO 260
      WRITE (6,270) (WRT(I),I=1,5)
210   FORMAT (5X,2(F6.4,2X),2X,3(F7.2,2X),49X,2X,2(F9.4,2X))
      GO TO 260
220   IF (IN.EQ.4) WRITE (6,230) (WRT(I),I=1,10)
      IF (IN.EQ.5) WRITE (6,240) (WRT(I),I=1,10)
      IF (IN.EQ.6) WRITE (6,250) (WRT(I),I=1,10),SUMFF,SUMH
      IF (IN.GT.3.AND.IN.LT.7) GO TO 260
      WRITE (6,270) (WRT(I),I=1,10)
230   FORMAT (5X,2(F6.4,2X),2X,3(F7.2,2X),5X,F7.4,2X,F7.4,1X,3(F7.2,2X),
+4X,#PIN AXIAL DISPLACEMENT#)
240   FORMAT (5X,2(F6.4,2X),2X,3(F7.2,2X),5X,F7.4,2X,F7.4,1X,3(F7.2,2X),
+9X,#FUEL CLAD#)
250   FORMAT (5X,2(F6.4,2X),2X,3(F7.2,2X),5X,F7.4,2X,F7.4,1X,3(F7.2,2X),
+2X,2(F9.4,2X))
260   CONTINUE
      RETURN
C
C
C
C
C
C
C
270   FORMAT (5X,2(F6.4,2X),2X,3(F7.2,2X),5X,F7.4,2X,F7.4,1X,3(F7.2,2X),
+2X,2(F9.4,2X))
280   FORMAT (1H0,24X,#..... RADIAL TEMPERATURE ..... #,20X,#...
+... RADIUS .....#)
290   FORMAT (25X,F6.0,# DEG.C #,7X,#(F7.0,# DEG.F)#,10X,F10.6,# METER
+8X,9X,#(F7.5,# INCH)#)
300   FORMAT (1H0,24X,F6.0,# DEG.C #,7X,#(F7.0,# DEG.F)#,10X,F10.6,# M
+ETERS#,9X,#(F7.5,# INCH)#)
310   FORMAT (25X,F6.0,# DEG.C #,7X,#(F7.0,# DEG.F)#,10X,F10.6,# METER
+3X,9X,#(F7.5,# INCH)#)
320   FORMAT (4X,#FUEL STRAINS,PCT#,9X,#FUEL STRESSES,KSI#,9X,#CLAD#,1X,
+#STRAINS,PCT#,8X,#CLAD STRESSES,KSI#,8X,#AXIAL STRAIN/STRESS#,/)
330   FORMAT (5X,#RADIAL#,3X,#HOOP#,5X,#RADIAL#,4X,#HOOP#,4X,#EFFECTIVE#
+,6X,#RADIAL#,4X,#HOOP#,4X,#RADIAL#,4X,#HOOP#,4X,#EFFECTIVE#,7X,#FU
+EL#,4X,#CLAD#)
340   FORMAT (1H1,2X,#STEP#,13,5X,#BOUNDARY NUMBER#,13,2X,#OF#,13,10X,10
+A4,20X,#ISAVE #,14)
350   FORMAT (1H0,8X,#LOCAL LINEAR HEAT RATING#,5X,1PE10.4,# W/M (
+X,OPF6.2,# KW/FT)#,/,9X,#PEAK LINEAR HEAT RATING#,6X,1PE10.4,# W/M
+ ( X,OPF6.2,# KW/FT)#,/,9X,#TIME IN-REACTOR#,14X,1PE10.4,# SE
+C. (X,OPF11.6,# DAYS)#,/,9X,#BURNUP#,23X,1PE10.4,# J/KGM (X,OPF8
+,1,# MWU/MTM)#,/)
360   FORMAT (1H0,8X,#LOCAL LINEAR HEAT RATING#,5X,1PE10.4,# W/M (
+X,OPF6.2,# KW/FT)#,/,9X,#PEAK LINEAR HEAT RATING#,6X,1PE10.4,# W/M
+ ( X,OPF6.2,# KW/FT)#,11X,#STORED ENERGY#,1PE12.3,2X,#(J/M)#,
+/,9X,#TIME IN-REACTOR#,14X,1PE10.4,# SEC. (X,OPF11.6,# DAYS)#,/,9

```

```

OUTPUT 156
OUTPUT 157
OUTPUT 158
OUTPUT 159
OUTPUT 160
OUTPUT 161
OUTPUT 162
OUTPUT 163
OUTPUT 164
OUTPUT 165
OUTPUT 166
OUTPUT 167
OUTPUT 168
OUTPUT 169
OUTPUT 170
OUTPUT 171
OUTPUT 172
OUTPUT 173
OUTPUT 174
OUTPUT 175
OUTPUT 176
OUTPUT 177
OUTPUT 178
OUTPUT 179
OUTPUT 180
OUTPUT 181
OUTPUT 182
OUTPUT 183
OUTPUT 184
OUTPUT 185
OUTPUT 186
OUTPUT 187
OUTPUT 188
OUTPUT 189
OUTPUT 190
OUTPUT 191
OUTPUT 192
OUTPUT 193
OUTPUT 194
OUTPUT 195
OUTPUT 196
OUTPUT 197
OUTPUT 198
OUTPUT 199
OUTPUT 200
OUTPUT 201
OUTPUT 202
OUTPUT 203
OUTPUT 204
OUTPUT 205
OUTPUT 206
OUTPUT 207
OUTPUT 208
OUTPUT 209
OUTPUT 210
OUTPUT 211
OUTPUT 212

```

SUBROUTINE OUTPUT

74/74 OPT=1

FTN 4.6+452

15/02/78 17.23.31

```

      +X, #BURNUP#, 23X, 1PE10.4, # J/KGM (#, OPFB.1, # MWD/MTM)#, /) OUTPUT 213
370  FORMAT (20X, #CLAD OD SURFACE HEAT FLUX#, 10X, 1PE10.4, # W/M2#, 14X, # ( OUTPUT 214
      +X, 1PE10.4, # BTU/HR-FT2)#, /, 20X, #FUEL SURFACE HEAT FLUX#, 13X, 1PE10. OUTPUT 215
      +4, # W/M2#, 14X, #X, 1PE10.4, # BTU/HR-FT2)#, /, 20X, #COOLANT TEMPERATUR OUTPUT 216
      +E#, 16X, OPF10.2, # DEG.C#, 13X, #X, F10.2, # DEG.F)#) OUTPUT 217
380  FORMAT (20X, #FILM COEFFICIENT#, 19X, 1PE10.4, # W/M2-DEG.C#8X, #X, 1PE OUTPUT 218
      +10.4, # BTU/HR-FT2-DEG.F)#, /, 20X, #CLAD THERMAL CONDUCTIVITY#, 10X, 1P OUTPUT 219
      +E10.4, # W/M-DEG.C#, 9X, #X, 1PE10.4, # BTU/HR-FT, DEG.F)#) OUTPUT 220
390  FORMAT (20X, #VOLUME AVERAGE FUEL TEMPERATURE#, 4X, OPF10.1, # DEG.C#, OUTPUT 221
      +13X, #X, F10.1, # DEG.F)#) OUTPUT 222
400  FORMAT (1H0, 19X, #GAS RELEASE FRACTION DURING CURRENT TIME STEP#, 5X OUTPUT 223
      +, F10.4, 5X, #CUMULATIVE GAS RELEASE#, F10.4, /, 20X, #INTERNAL GAS PRESS OUTPUT 224
      +URE#, 32X, 1PE10.4, # PA #, 8X, #X, OPF5.0, # PSI#, /25X, #GAS CONTENT#, 3 OUTPUT 225
      +7X, 1PE10.4, # KG-MOLE#) OUTPUT 226
410  FORMAT (20X, #CHANGE IN FUEL RADIUS DUE TO THERMAL EXPANSION OUTPUT 227
      +X, 1PE10.4, # METERS#, 9X, #X, OPF10.6, # INCH)#, /, 20X, #CHANGE IN FUEL OUTPUT 228
      + RADIUS DUE TO RELOCATION#, 13X, 1PE11.4, # METERS#, 9X, #X, OPF10.6, # OUTPUT 229
      + INCH)#, /, 20X, #CHANGE IN FUEL RADIUS DUE TO DENSIFICATION#, 10X, 1PE OUTPUT 230
      +11.4, # METERS#, 9X, #X, OPF10.6, # INCH)#, /, 20X, #CHANGE IN FUEL RADI OUTPUT 231
      +US DUE TO SWELLING #, 14X, 1PE10.4, # METERS#, 9X, #X, OPF10.6, # INCH OUTPUT 232
      +)#, /, 20X, #TOTAL CHANGE IN FUEL RADIUS #, 23X, 1PE11.4, # METERS#, 9X OUTPUT 233
      +, #X, OPF10.6, # INCH)#, /, 20X, #ELASTIC CHANGE IN CLAD RADIUS#, 23X, 1 OUTPUT 234
      +PE11.4, # METERS#, 9X, #X, OPF10.6, # INCH)#, /, 20X, #CREEP-#, #PLASTIC OUTPUT 235
      +CHANGE IN CLAD RADIUS#, 17X, 1PE11.4, # METERS#, 9X, #X, OPF10.6, # INCH OUTPUT 236
      +)#, /, 20X, #TOTAL CHANGE IN CLAD RADIUS #, 23X, 1PE11.4, # METERS#, 9X OUTPUT 237
      +, #X, OPF10.6, # INCH)#, ) OUTPUT 238
420  FORMAT (/, 20X, #HOT GAP (RADIAL)#, 36X, 1PE11.4, # METERS#, 9X, #X, OPF OUTPUT 239
      +10.6, # INCH)#, /, 20X, #FUEL-CLAD INTERFACIAL PRESSURE#, 23X, 1PE10.4, # OUTPUT 240
      + PA#, 13X, #X, 1PE10.4, # PSI#)#/) OUTPUT 241
430  FORMAT (1H , 19X, #FUEL TO CLAD GAP CONDUCTANCE#, 20X, 1PE10.4, # W/M2- OUTPUT 242
      +DEG.C (#, OPF7.1, # BTU/HR-FT2-DEG.F)#) OUTPUT 243
440  FORMAT (1H , 19X, #CONDUCTANCE COMPONENTS #, F3.0, # SOLID CONTACT, # OUTPUT 244
      +, F4.0, #GAS CONDUCTION, #, F3.0, #RADIATION#) OUTPUT 245
450  FORMAT (1H , 19X, #MOLE FRACTIONS #, F4.2, # HE, #, F4.2, # AR, #, F4.2 OUTPUT 246
      +, # H, #, F4.2, # N, #, F4.2, # CU, #, F4.2, # KR, #, F4.2, # XE#, /) OUTPUT 247
460  FORMAT (1H0, 19X, #THERMAL CONDUCTIVITY OF FILL GAS#, 5X, 1PE10.4, # W/ OUTPUT 248
      +M-DEG.C#, 8X, #X, 1PE10.4, # BTU/HR-FT-DEG.F)#, ) OUTPUT 249
470  FORMAT (20X, #TEMPERATURE JUMP DISTANCE#, 12X, 1PE10.4, # METERS#, 11X, OUTPUT 250
      +X, 1PE10.4, # INCH)#) OUTPUT 251
480  FORMAT (5X, #***** MECHANICS RE OUTPUT 252
      +ULTS FOR AXIAL REGION #, I2, # ***** OUTPUT 253
      +*****#) OUTPUT 254
      END OUTPUT 255

```


D-105

	KPLAS = 0	PELET	28
	RAKSUM = 0.	PELET	29
	ISP = 0	PELET	30
	NFF2 = (NF+NC)*4	PELET	31
	NEW = NEWD	PELET	32
	NOLD = 2	PELET	33
	IF (NEW.EQ.2) NOLD = 1	PELET	34
	CALL SECOND (TM1)	PELET	35
C		PELET	36
C	CALL INPUT AND INITIALIZE CALCULATIONS.	PELET	37
	IRZ = NT	PELET	38
	NC4 = NC*4	PELET	39
	CALL MOVEAA (TTIME,D(ITIME),15)	PELET	40
	CP = CPP	PELET	41
	CR = CRR	PELET	42
	FRDEN = FFDEN	PELET	43
	IF (IRZ.EQ.1) GO TO 100	PELET	44
	PINNEW = PRESTO	PELET	45
	POUTNW = EXTP	PELET	46
	PNEW = POUTNW-PINNEW	PELET	47
	PIN = PINNEW-PINOLD	PELET	48
	POUT = POUTNW-POUTLD	PELET	49
	PDEL = PNEW-POLD	PELET	50
	IF (KZ.NE.ISTOP) GO TO 110	PELET	51
	POLD = PNEW	PELET	52
	POUTLD = POUTNW	PELET	53
	PINOLD = PINNEW	PELET	54
	GO TO 110	PELET	55
100	PIN = PRESTO	PELET	56
	POUT = EXTP	PELET	57
	PDEL = EXTP-PRESTO	PELET	58
	POLD = PDEL	PELET	59
	PINOLD = PIN	PELET	60
	POUTLD = POUT	PELET	61
110	CONTINUE	PELET	62
C		PELET	63
	IPREV = 0	PELET	64
	ICUT = 0	PELET	65
	IADJ = 1	PELET	66
	TIMEF = TTIME(3)	PELET	67
	TIMEN = TTIME(2)	PELET	68
	IDF = 0	PELET	69
	IF (NT.GT.ISTART.OR.KZ.GT.2) GO TO 120	PELET	70
	JAH = 1	PELET	71
	DELF = RD(NF,1)	PELET	72
	DELC = RD(NFF1,1)-RD(NF+1,1)	PELET	73
	DELZ = (DELF+DELC)/2.	PELET	74
	CALL INPAXL (D(IAELM),D(IR),D(IRA),D(IY),D(IYACF),D(INADET),RD,NAB	PELET	75
	+T2,NFF1)	PELET	76
	CALL INPUT (D(IR),D(IELM),D(INDF),D(INODE),D(INDC),D(INODEC),D(INO	PELET	77
	+DET),D(INOT))	PELET	78
120	CONTINUE	PELET	79
C		PELET	80
C		PELET	81
C	THE NKN LOOP CONTROLS THE CALCULATION AND UPGRADING OF THE TOTAL S	PELET	82
C	AND STPAINS. AFTER TEMPERATURE CONVERGENCE THIS LOOP IS	PELET	83
C	EXECUTED TWICE. THE FIRST ITERATION AFTER CONVERGENCE CALCULATES	PELET	84

```

C      AND ADDS IN THE CURRENT ELASTIC INCREMENTS. THEN AND ONLY THEN
C      IS SUBROUTINE CREEP CALLED TO CALCULATE THE CREEP-PLASTIC
C      INCREMENTS.
C
C      ILK = -1
C      DO 420 NKN=1,2
C      IKL = 1
130    CONTINUE
C      IF (ICUT, EQ, 1) IKL = IADJ
C      STEP = (TIMEF-TIMEN)/FLOAT(IADJ)
C
C      TIME CUTTING LOOP FOR CREEP
C
C      DO 340 ITI=1, IKL
C      CALL MOVEKA (0, D(ITCRP), 4*NOEL+NARC)
C      ISSTOP = 0
C      KZ1 = KZ
C      IPLAS = 0
C      IRAK = 0
C      IF (KZ1, NE, ISSTOP) GO TO 180
C      ISSTOP = 1
C      IF (NKN, EQ, 1) GO TO 140
140    CONTINUE
C      IF (ICUT, EQ, 0) GO TO 150
C      TIME(2) = TIMEN+FLOAT(ITI-1)+STEP
C      TIMEC = TIME(2)
C      TIME(3) = TIME(2)+STEP
150    CONTINUE
C      IPLAS = 0
C      CALL SECOND (TM2)
C      TMDL = TM2-TM1
C      WRITE (6, 450) TMDL
C      IF (TMDL, LT, 100.) GO TO 160
C      WRITE (6, 460) TMDL
C      STOP
160    CONTINUE
C      CALL CREEP (D(ITSTR), D(IEFSTR), D(ITSTNR), TELM, TIME, D(ITCRP)
C      +      , D(ICRPG), D(ICRATE), NOEL, NARC, NSH)
C      DO 170 I=1, NARC
C      DO 170 J=1, NOEL
C      EFSTR(J, I) = D(IEFSTR+NOEL+(I-1)+J-1)
170    CONTINUE
180    CONTINUE
C
C      THE KKA LOOP CONTROLS THE RADIAL-AXIAL-RADIAL CHAINED SOLUTION
C      THE KA LOOP RUNS THROUGH ALL AXIAL REGIONS, CALLING FOR
C      RADIAL CALCULATIONS FOR CONTACT AND PRESELECTED REGIONS
C
C      DO 230 KKA=1,2
C      DO 210 KA=1, NARC
C      IF (KAN(KA), EQ, 0, AND, IDU(KA), EQ, 0) GO TO 210
C      ICON = 1
C      IF (KAN(KA), EQ, 0, AND, IDU(KA), EQ, 1) ICON = 0
C      IF (ILK, GT, 0, AND, KKA, EQ, 1) GO TO 190
C      CALL STIFF (D(IFLM), D(IR), D(IZ), D(INODE), D(IXK), D(IXKC)
C      +      , D(INODEC), RD, NFF1, D(INODET), D(IXKT), D(IEPSLM), D(ISTR)
C      +      , ES), D(IARE), D(IHRH), D(IXLOAD), D(IXSAVE), D(ICC), D(IKAN)

```

```

PELET 85
PELET 86
PELET 87
PELET 88
PELET 89
PELET 90
PELET 91
PELET 92
PELET 93
PELET 94
PELET 95
PELET 96
PELET 97
PELET 98
PELET 99
PELET 100
PELET 101
PELET 102
PELET 103
PELET 104
PELET 105
PELET 106
PELET 107
PELET 108
PELET 109
PELET 110
PELET 111
PELET 112
PELET 113
PELET 114
PELET 115
PELET 116
PELET 117
PELET 118
PELET 119
PELET 120
PELET 121
PELET 122
PELET 123
PELET 124
PELET 125
PELET 126
PELET 127
PELET 128
PELET 129
PELET 130
PELET 131
PELET 132
PELET 133
PELET 134
PELET 135
PELET 136
PELET 137
PELET 138
PELET 139
PELET 140
PELET 141

```

```

+      ,IDO,NOEL,TELM,NABF) PELET 142
CALL BOUND (D(IXK),D(IXKC),D(IXKT),D(INODE),D(INODEC), PELET 143
+      D(INODET),D(IKAN),IDO,MAXC,MAXT) PELET 144
190 CONTINUE PELET 145
      IPREV = ICUT PELET 146
      IF (ILK.GT.0.AND.KKA.EQ.1) GO TO 200 PELET 147
      CALL SOLVET (D(ISB),D(ISBB),D(INODE),D(INODEC),D(INODE PELET 148
+      T),D(IXK),D(IXKC),D(IXKT),D(ICC),D(ITHET),KAN,IDO,D(IS PELET 149
+      BF),D(ISHC),D(ISBBF),D(ISBBC),D(ILSTRA),TELM,TOELE(1,K PELET 150
+      A),D(IR),D(IARAA),D(IRBRA),NABF,NOEL,MAXC,D(IXKCC),D(I PELET 151
+      XKTT),D(IBBSV),D(IESTSV),NFF2,MAXT,NABF,D(IESTNV)) PELET 152
200 CONTINUE PELET 153
      IF (ICUT.EQ.1.AND.IPREV.EQ.0) GO TO 130 PELET 154
210 CONTINUE PELET 155
      IF (KKA.EQ.2) GO TO 230 PELET 156
      IF (ICHNG.EQ.0.AND.KZ1.GT.2.AND.IPLAS.EQ.0.AND.ICUT.EQ.0) PELET 157
+      GO TO 220 PELET 158
      CALL STFAXL (D(IAELM),D(IAK),D(IARAA),D(IPSLA),IACON,D(IN PELET 159
+      ADET),D(IRA),D(IRBRA),D(ISTRSA),TW,NTAE,D(IXL04),D(IYACF PELET 160
+      ),D(ICA),KAN) PELET 161
      CALL ABOUND (D(IAK),D(IDISP),D(INADET)) PELET 162
220 CONTINUE PELET 163
      CALL ASOLVE (D(IBA),D(IBBA),D(INADET),D(IAK),IACON,NABT,D PELET 164
+      (IKAN),D(IPRESS),TW,D(IBASAV)) PELET 165
230 CONTINUE PELET 166
      CALL MOVEKA (0,YM,(NOEL+1)*NARC) PELET 167
C PELET 168
C INTERPOLATE AND ADD IN RADIAL DISPLACEMENTS TO THE RD ARRAY PELET 169
C PELET 170
      K = 0 PELET 171
      LU = NF+1 PELET 172
      NFP1 = NF+1 PELET 173
      DO 250 I=1,NARC PELET 174
        IF (KAN(I).EQ.0.AND.IDO(I).EQ.0) GO TO 250 PELET 175
        K = K+1 PELET 176
        YM(K,1) = (Y(I,2)+Y(I+1,2))/2.0 PELET 177
        DO 240 J=LU,NFF1 PELET 178
          YM(K,J+1) = DISPC(J,I) PELET 179
240 CONTINUE PELET 180
250 CONTINUE PELET 181
      KMAX = K PELET 182
      DO 280 I=1,NABT PELET 183
        IF (NKN.EQ.1) CRPAD(I) = 0.0 PELET 184
        DO 260 J=LU,NFF1 PELET 185
          DSP = TERP(Y(I,2),YM,YM(J+1),KMAX) PELET 186
          IF (J.EQ.NFP1) ADR = DSP PELET 187
          IF (ISSTOP.NE.1) GO TO 260 PELET 188
          IF (ISSTOP.EQ.1.AND.NKN.EQ.1) DSPSV(J,I) = DSP PELET 189
          IF (NKN.EQ.2.AND.KPLAS.EQ.1) RD(J,I) = RD(J,I)-DSPSV(J PELET 190
+          ,I) PELET 191
          RD(J,I) = RD(J,I)+DSP PELET 192
260 CONTINUE PELET 193
          IF (NKN.EQ.2.AND.KPLAS.NE.1) CRPAD(I) = CRPAD(I)+ADR PELET 194
          IF (ISSTOP.EQ.1) GO TO 270 PELET 195
          RDELST(I) = RD(NF+1,I)+ADR PELET 196
270 CONTINUE PELET 197
          IF (NKN.EQ.2.AND.KPLAS.EQ.1) RDELST(I) = RDELST(I)-DSPSV( PELET 198

```

	+	NFF1,I)	PELET	199
C		DECISION POINT FOR PLASTICITY ITERATION	PELET	200
		IF (NKN.EQ.2.AND.KPLAS.EQ.1) RDELST(I) = RDELST(I)+ADR	PELET	201
280		CONTINUE	PELET	202
		IF (ISSTOP.NE.1) GO TO 430	PELET	203
		IF (KPLAS.GT.0.AND.KPLAS.LT.IPLEND) GO TO 140	PELET	204
		IF (KPLAS.EQ.IPLEND.AND.IPLAS.EQ.1) GO TO 140	PELET	205
		IF (KPLAS.EQ.IPLEND.AND.ISP.EQ.1) GO TO 140	PELET	206
C			PELET	207
C		INTERPOLATE THE INCREMENTAL STRAINS AND STRESSES	PELET	208
C			PELET	209
		CALL MOVEKA (0,D(IDISPC),NFF1*NARC)	PELET	210
		CALL MOVEKA (0,YM,(NOEL+1)*NARC)	PELET	211
		DO 330 M=1,4	PELET	212
		DO 320 J=1,NOEL	PELET	213
		K = 0	PELET	214
		DO 290 I=1,NARC	PELET	215
		YAV(I) = (Y(I,2)+Y(I+1,2))/2.0	PELET	216
		IF (KAN(I).EQ.0.AND.IDO(I).EQ.0) GO TO 290	PELET	217
		K = K+1	PELET	218
		YD(K) = TSISR(M,J,I)	PELET	219
		YI(K) = YAV(I)	PELET	220
		YN(K) = TSINR(M,J,I)	PELET	221
290		CONTINUE	PELET	222
		DO 300 I=1,NARC	PELET	223
		IF (IDO(I).EQ.1.OR.KAN(I).EQ.1) GO TO 300	PELET	224
		TSISR(M,J,I) = TERP(YAV(I),YI,YD,K)	PELET	225
		TSINR(M,J,I) = TERP(YAV(I),YI,YN,K)	PELET	226
300		CONTINUE	PELET	227
		DO 310 I=1,NARC	PELET	228
		YN(I) = 0.0	PELET	229
		YD(I) = 0.0	PELET	230
310		CONTINUE	PELET	231
320		CONTINUE	PELET	232
330		CONTINUE	PELET	233
340		CONTINUE	PELET	234
		TIME(2) = TIMEN	PELET	235
		IF (ILK.EQ.-1) GO TO 350	PELET	236
		GO TO 410	PELET	237
C			PELET	238
C		UNLOCKING INFORMATION	PELET	239
350		CONTINUE	PELET	240
		ILK = 0	PELET	241
		DO 400 I=2,NARC	PELET	242
		IF (IACON(I,NEW).EQ.0.AND.IACON(I,NOLD).EQ.1) GO TO 360	PELET	243
		GO TO 400	PELET	244
360		IF (IACON(I-1,NEW).EQ.0.AND.IACON(I-1,NOLD).EQ.1) GO TO 390	PELET	245
		DO 370 IJ=2,I	PELET	246
		IMJ = I-IJ+1	PELET	247
		KSTART = IMJ	PELET	248
		IF (IACON(IMJ,NEW).EQ.1) GO TO 380	PELET	249
		IF (IACON(IMJ,NEW).EQ.0.AND.IACON(IMJ,NOLD).EQ.1) GO TO 3	PELET	250
		80	PELET	251
370	+	CONTINUE	PELET	252
380		CONTINUE	PELET	253
390		IF (IACON(I+1,NEW).EQ.0.AND.IACON(I+1,NOLD).EQ.1) GO TO 400	PELET	254
		KSTOP = I	PELET	255

SUBROUTINE PELET

74/74 OPT=1

FTN 4.6+452

15/02/78 17.23.31

	ILK = ILK+1	PELET	256
	LOCK(ILK,1) = KSTART	PELET	257
	LOCK(ILK,2) = KSTOP	PELET	258
400	CONTINUE	PELET	259
C	HAVE CREATED AN ARRAY TO COMPUTE THE SIZES OF THE	PELET	260
C	CANTACT REGIONS	PELET	261
C	ILK=THE NUMBER OF SEGMENTS ALONG THE FUEL PIN WHERE	PELET	262
C	ADJUSTMENTS IN AXIAL LOAD MUST BE MADE	PELET	263
C	LOCK(ILK,1)=STARTING REGION	PELET	264
C	LOCK(ILK,2)=STOP REGION	PELET	265
C		PELET	266
	IF (ILK.GT.0) GO TO 130	PELET	267
410	CONTINUE	PELET	268
	ILK = 0	PELET	269
420	CONTINUE	PELET	270
430	CONTINUE	PELET	271
	CALL MOVEKA (0,D(IDISPC),NFF1*NABT)	PELET	272
	DO 440 I=1,NARF	PELET	273
	IF (KAN(I).EQ.0) D(IPRESS+I-1) = 0.0	PELET	274
	IF (KAN(I).EQ.0) PRSS(I) = 0.0	PELET	275
	PRSS(I) = D(IPRESS+I-1)	PELET	276
440	CONTINUE	PELET	277
	RETURN	PELET	278
C		PELET	279
450	FORMAT (25X,TIME EXPIRED IN PELET =,F8.3)	PELET	280
460	FORMAT (5X,PELET TIME LIMIT = CREEP CORRECTION TOO LARGE,5X,TIME	PELET	281
	EXPIRED =,F8.3)	PELET	282
	END	PELET	283

SEVERITY DETAILS DIAGNOSIS OF PROBLEM

I	CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE INHIBITED.
I	CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE INHIBITED.
I	CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE INHIBITED.
I	CONTROL VARIABLE IN COMMON OR EQUIVALENCED, OPTIMIZATION MAY BE INHIBITED.

SUBROUTINE POWDIS

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	SUBROUTINE POWDIS (POWER,PKPOWER,PAVRG,IPEAK,NPOW,AI,DELZA,FULEN)	POWDIS	2
	DIMENSION DELZA(20), PST(20)	POWDIS	3
C	INTEGRATE THE LOCAL POWER DISTRIBUTION AND NORMALIZE	POWDIS	4
C	THE RESULT TO 1.0. THUS AINCR REPRESENTS THE .5 OF THE	POWDIS	5
C	AVERAGING TIMES THE X INCREMENT WHICH IS 1.0/NPOW. NPOW IS	POWDIS	6
	DIMENSION POWER(20), AI(20)	POWDIS	7
	NPOW1 = NPOW+1	POWDIS	9
	AI(1) = 0.0	POWDIS	10
	AI(2) = 0.0	POWDIS	11
	NPOW2 = NPOW+2	POWDIS	12
	IF (NPOW.LE.1) GO TO 110	POWDIS	13
	PST(2) = POWER(1)	POWDIS	14
	DO 100 I=3,NPOW2	POWDIS	15
	AI(I) = (POWER(I-2)+POWER(I-1))*DELZA(I-1)/(2.*FULEN)+AI(I-1)	POWDIS	16
	PST(I) = POWER(I-1)	POWDIS	17
100	CONTINUE	POWDIS	18
110	CONTINUE	POWDIS	19
	AITOT = AI(NPOW2)	POWDIS	20
	IF (AITOT.LE.0.) GO TO 180	POWDIS	21
C	RENORMALIZE THE POWER AND AI SUCH THAT AI(1)=1.0.	POWDIS	22
C	IF (IPEAK) 120,140,120	POWDIS	23
120	CONTINUE	POWDIS	24
	DO 130 I=2,NPOW2	POWDIS	25
	POWER(I) = PST(I)*PAVRG/AITOT	POWDIS	26
	AI(I) = AI(I)/AITOT	POWDIS	27
130	CONTINUE	POWDIS	28
	GO TO 170	POWDIS	29
C	CALCULATE POWER DISTRIBUTION BASED ON PEAK POWER (KW/FT)	POWDIS	30
140	CONTINUE	POWDIS	31
	PEAK = POWER(1)	POWDIS	32
	DO 150 I=2,NPOW1	POWDIS	33
	PEAK = AMAX1(PEAK,POWER(I))	POWDIS	34
150	CONTINUE	POWDIS	35
C	NORMALIZE THE POWER DISTRIBUTION TO PKPOWER	POWDIS	36
	DO 160 I=2,NPOW2	POWDIS	37
	POWER(I) = PST(I)*PKPOWER/PEAK	POWDIS	38
	AI(I) = AI(I)/AITOT	POWDIS	39
160	CONTINUE	POWDIS	40
	PAVRG = AITOT*PKPOWER/(PEAK)	POWDIS	41
170	CONTINUE	POWDIS	42
	RETURN	POWDIS	43
180	CONTINUE	POWDIS	44
	WRITE (6,190) AI	POWDIS	45
	STOP	POWDIS	46
190	FORMAT (1H1,17TROUBLE IN POWDIS CHECK PROFILE,/(1X,10F12.6))	POWDIS	47
	END	POWDIS	48
		POWDIS	49
		POWDIS	50
		POWDIS	51
		POWDIS	52

```

SUBROUTINE RATE (NSH,CRATE,DTIME,TELM,SE,CRPG,NAXL,NRAD,PSTS,PSSV, RATE      2
+PSSRT,IJ,IPOS,OLDSE,BASE) RATE      3
C                                     RATE      4
C                                     RATE      5
C                                     RATE      6
C                                     RATE      7
C                                     RATE      8
C                                     RATE      9
C                                     RATE     10
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IPELTM,IELSTN,IELSTR PELCOM    2
+,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD PELCOM    3
+ET, IR,IRBR, ISA,ISAC,ISAF,ISB,ISBB,ISBRC,I PELCOM    4
+BBBF,ISBC,ISBF,ISK,ISKC,ISTRES, ITDELE, IXK,IXKC,IXKT PELCOM    5
+,IXLOAD, ICC,ITHET,ICA,ITHTAX, INADET,IRA,IY,IAELM,I PELCOM    6
+ARAA,IRA,IBRA,IRBRA,IXLDA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I PELCOM    7
+PSLA,IAK,IACN,IKAN,IRZ,IAAX,TRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM    8
+IIDU,ITME,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM    9
+, IXKCC,IXKTT PELCOM   10
+,ITFB, ITCB, IBBSV,IESTSV,IESTNV PELCOM   11
+,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLOSE,IPSSV,IPSSRT,JIIJ,IIPDS PELCOM   12
+,IBASE,ITFR,IYM,IZ PELCOM   13
+,IDSPSV PELCOM   14
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM   15
+DEL PELCOM   16
COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IDF,NFF,MAXDC,NETOT,NE PELCOM   17
+WF,NEWT,KA,NARF,NARC,NART,NABF,NARC,NABT,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM   18
+NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CR,FRDEN,KZ1,ISSTOP,IPLAS,NKN PELCOM   19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM   20
+MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD PELCOM   21
+, KPLAS,ICNT,RRAKR,IRAK,RAKSUM,IPLEND,ISP PELCOM   22
COMMON /LASTIC/ E,XNU,ALPHA PELCOM   23
COMMON /CONST/ PI PELCOM   24
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM   25
COMMON /CRP/ PSP(25),TPLASK PELCOM   26
COMMON /BA/ AA(23,7),CF(7,8),HKS,DELZA(20),NN,NCONN COMC      2
+, NCNN,ISTR,ITER COMC      3
+, FLXR COMC      4
COMMON/FEP/HWF,NT,KKZZZ,ISTOP,RDOLD(20),RRSIN(20),ICNOLD(20) COMC      5
+, IDOOLD(20) COMC      6
+,PINOLD,POUTLO,POLD,NTOLD(20) COMC      7
+, ICHNG,CRPAD(20) COMC      8
+,STEPS COMC      9
DIMENSION TELM(NRAD,NAXL,2), CRATE(NRAD,1), TIME(5), SE(NRAD,1) RATE     13
DIMENSION CRPG(NRAD,1) RATE     14
DIMENSION PSTS(NRAD,1),OLDSE(NRAD,1),PSSV(1),PSSRT(1),IJ(1), RATE     15
+ IPOS(1),BASE(1), DTIME(5) RATE     16
C *** PNL CREEP MODEL JANUARY 1978 VERSION *** RATE     17
DATA C1/3310./,C2/.0044/,C3/5.96E14/,C5/1.05E-4/,C6/6.36E4/, RATE     18
+ C7/1.0E-5/,C8/9.5E3/,C9/65.77/,C10/638.5/,C11/7.18E-7/ RATE     19
C                                     RATE     20
C                                     RATE     21
C                                     RATE     22
C                                     RATE     23
C                                     RATE     24
C DETECTION OF PLASTIC ELEMENTS. PSTS(I,J) IS THE YIELD SURFACE RATE     25
C FOR ELEMENT I,J. SF(I,J) IS THE CURRENT GENERALIZED STRESS,AND RATE     26
C CRPG(I,J) IS THE CURRENT GENERALIZED STRAIN RATE     27

```


C	SKIP FOR STAINLESS STEEL	RATE	28
	TEST = TERP(200,AA,AA(1,4),NN)	RATE	29
	IF (TEST,GT,24,E6) GO TO 140	RATE	30
	DO 110 J=1,NAXL	RATE	31
	DO 110 I=LOW,NRAD	RATE	32
	TEMPK = (TELM(I,J,NEW)-32.0)/1.8+273.0	RATE	33
	CWK = 120260,-146.5*TEMPK+.1378*TEMPK**2-.0728E-3*TEMPK**3	RATE	34
	CWN = -1.715E-2+.3583E-3*TEMPK-.7638E-6*TEMPK**2+.4954E-9*TE	RATE	35
	MPK**3	RATE	36
	PSTS(I,J) = CWK*(CRPG(I,J))*CWN	RATE	37
	IF (PSTS(I,J),GT,8E(I,J)) GO TO 110	RATE	38
	IPLAS = 1	RATE	39
	REST = SE(I,J)-PSTS(I,J)	RATE	40
	IF (BIGT,LT,REST) BIGT = REST	RATE	41
C	CALCULATION OF INITIAL CONDITIONS FOR PLASTICITY	RATE	42
	IF (KPLAS,NE,0) GO TO 100	RATE	43
	ICNT = ICNT+1	RATE	44
	IJ(ICNT) = (J-1)*NRAD+I	RATE	45
	PSSV(ICNT) = SE(I,J)-PSTS(I,J)	RATE	46
	BASE(ICNT) = SE(I,J)-OLDSE(I,J)	RATE	47
	GO TO 110	RATE	48
100	CONTINUE	RATE	49
	TF = TELM(I,J,NEW)	RATE	50
	E = TERP(TF,AA,AA(1,4),NN)	RATE	51
	CRATE(I,J) = ((SE(I,J)-PSTS(I,J))/E)*1.6	RATE	52
110	CONTINUE	RATE	53
	IRAK = 0	RATE	54
C	DECISION POINT FOR PSEUDO-ELASTIC LOAD STEPS. IPLAS=1 IS THE	RATE	55
C	SIGNAL THAT PLASTICITY HAS OCCURRED. KPLAS IS A COUNTER,AND	RATE	56
C	KPLAS=IPLAS SIGNALS THE END OF THE PLASTICITY ITERATIONS.	RATE	57
	IPLAS = ICNT+1	RATE	58
	IF (IPLAS,EQ,1,AND,BIGT,LT,500.) IPLAS = 0	RATE	59
	IF (KPLAS,GE,IPLAS,AND,IPLAS,EQ,0) CALL MOVEKA (0,CRATE,NOEL*NARC	RATE	60
	*)	RATE	61
	IF (KPLAS,EQ,0,AND,IPLAS,EQ,0) GO TO 140	RATE	62
	IF (KPLAS,GE,2,AND,IPLAS,EQ,1) GO TO 140	RATE	63
	IF (KPLAS,GE,2,AND,IPLAS,EQ,0) KPLAS = KPLAS+1	RATE	64
	IF (KPLAS,GT,IPLAS) GO TO 140	RATE	65
	IF (KPLAS,EQ,1) KPLAS = 2	RATE	66
	IF (KPLAS,EQ,0,AND,IPLAS,EQ,1) KPLAS = 1	RATE	67
C	SORT PLASTIC ELEMENTS IN ASCENDING ORDER OF PLASTICITY.	RATE	68
	IF (KPLAS,EQ,1) CALL SORTP (PSSV,ICNT,PSSRT,IPOS)	RATE	69
C	START OF LOAD STEP CALCULATION. THE QUANTITY (1.-RRAKR) IS THE	RATE	70
C	FRACTION OF TOTAL LOAD STEP TO BE TAKEN. IRAK=1 IS THE SIGNAL	RATE	71
C	TO TAKE A PSEUDO-ELASTIC LOAD STEP. RAKSUM IS THE CUMULATIVE FRACT	RATE	72
C	OF THE TOTAL LOAD STEP.	RATE	73
	IRAK = 1	RATE	74
	IT = ICNT-KPLAS+1	RATE	75
	IF (IT,LE,0) IT = 1	RATE	76
	IG = IPOS(IT)	RATE	77
	DO 120 J=1,NAXL	RATE	78
	DO 120 I=LOW,NRAD	RATE	79
	ITEST = (J-1)*NRAD+I	RATE	80
	IF (ITEST,EQ,IJ(IG)) GO TO 130	RATE	81
120	CONTINUE	RATE	82
130	CONTINUE	RATE	83
	IF (KPLAS,EQ,1) RAKR = PSSRT(IT)/BASE(IG)	RATE	84

	IF (KPLAS.GT.1) RRAKR = 1.-(PSTS(I,J)-SE(I,J))/BASE(IG)	RATE	85
	RAKSUM = RAKSUM+(1.-RRAKR)	RATE	86
	IF (KPLAS.EQ.IPLEND) RRAKR = RAKSUM	RATE	87
	IF (KPLAS.EQ.IPLEND.AND.RAKSUM.GT.1.0) RRAKR = 1.0	RATE	88
140	CONTINUE	RATE	89
C		RATE	90
C	DECISION TO SKIP THE CREEP CALCULATION. ISP=1 IS THE SIGNAL	RATE	91
C	THAT THIS IS THE LAST PSEUDO-ELASTIC STEP.	RATE	92
C		RATE	93
	IF (KPLAS.EQ.IPLEND) ISP = ISP+1	RATE	94
	IF (KPLAS.GT.0.AND.KPLAS.LT.IPLEND) GO TO 360	RATE	95
	IF (KPLAS.EQ.IPLEND.AND.IPLAS.EQ.1) GO TO 360	RATE	96
	IF (KPLAS.EQ.IPLEND.AND.ISP.EQ.1) GO TO 360	RATE	97
C		RATE	98
C	GAS = UNIVERSAL GAS CONSTANT, CAL/MOLE-DEG F	RATE	99
C		RATE	100
	GAS = 1.987	RATE	101
	NNN = (NF+NC-2)*2	RATE	102
	N = (NABT-1)	RATE	103
	NRF = (NF-1)*2	RATE	104
	NA = NABF-1	RATE	105
	M = IRZ	RATE	106
	DO 150 I=1,3	RATE	107
	TIME(I) = DTIME(I)*24.0	RATE	108
150	CONTINUE	RATE	109
	DLTM = TIME(3)-TIME(2)	RATE	110
	FISR = 1.0E13	RATE	111
C	STRAIN HARDENING OPTION DECISION POINT	RATE	112
C	STRAIN HARDENING OPTION VARIABLE = NSH	RATE	113
C	NSH = 1 MEANS THE STRAIN HARDENING OPTION IS USED	RATE	114
C		RATE	115
	IF (NSH.EQ.1.AND.M.GT.1) GO TO 230	RATE	116
C		RATE	117
	DO 220 J=1,N	RATE	118
C	FLUX CALC. BASED ON SPECIFIC POWER(WATTS/GM FUEL),AND ENRICHMENT	RATE	119
C	C REF GLASSSTONE AND SESONSKE, 75-90	RATE	120
C	EQUATION IS BASED ON A FLUX OF 2.4E13 AT 16.53 W/G FOR 3.4	RATE	121
C	PERCENT ENRICHMENT	RATE	122
	ENR = 1.0-PSP(22)+1.3*(1.-PSP(23))*PSP(21)	RATE	123
	FFLUX = 4.936E10*PSP(J)/ENR	RATE	124
	FFLUX = FFLUX*FLXR	RATE	125
	DO 220 I=1,NNN	RATE	126
	TEMK = (TELM(I,J,NEW)-32.0)/1.8+273.0	RATE	127
	IF (I.LE.NRF.AND.J.LE.NA) GO TO 200	RATE	128
C	TEST FOR STAINLESS STEEL VS. ZIRC.	RATE	129
	TEST = TERP(200.,AA,AA(1,4),NN)	RATE	130
	IF (TEST.GT.24.E6) GO TO 180	RATE	131
C		RATE	132
C	ZR CREEP CALCULATION (P.J. PANKASKIE)	RATE	133
C	*** PNL CREEP MODEL JANUARY 1978 VERSION ***	RATE	134
C		RATE	135
C	TIME HARDENING	RATE	136
	CTU = C10*(1+.021*ALOG(1.+FFLUX/6.E12))*EXP(-C11*SE(I,J))	RATE	137
	DELT = CTU-TEMK	RATE	138
	TNH = TANH(100.*DELT)	RATE	139
	GA = C6+.061*SE(I,J)	RATE	140
	QI = C8+.034*SE(I,J)	RATE	141

D-114

SUBROUTINE RATE

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	TAE = -(C6*(1.-TNH)+Q1*(1.+TNH))/(2.*GAS*TEMK)	RATE	142
	IF (FFLUX,LE,10.) GO TO 160	RATE	143
	CFF = 0.85*ALOG(FFLUX)	RATE	144
	GO TO 170	RATE	145
160	CONTINUE	RATE	146
	CFF = (QA/(4.*GAS*CTO))* (3.-CTO/TEMK)	RATE	147
170	CONTINUE	RATE	148
	FAE = (CFF-C9)*(1.+TNH)/2.	RATE	149
	ATERM = EXP(TAE+FAE)	RATE	150
	SCR = C3*ATERM*((1.-TNH)*SINH(C5*SE(I,J))+(1.+TNH)*SINH(C7*SE(I,J)))/2.	RATE	151
	DLTC = C1*(EXP(-C2*TIME(2))-EXP(-C2*TIME(3)))	RATE	152
	CRATE(I,J) = SCR*(DLTC+DLTM)	RATE	153
	IWRT = NNN-3	RATE	154
	IF (I,LT,IWRT) GO TO 190	RATE	155
	IF (J,NE,4) GO TO 190	RATE	156
	GO TO 190	RATE	157
180	CONTINUE	RATE	158
C		RATE	159
C	NUCLEAR SYSTEMS MATERIALS HANDBOOK	RATE	160
C	PROPERTY 2206 .20 CW 316 SS REV 0	RATE	161
C	CREEP FOR STAINLESS STEEL(	RATE	162
C		RATE	163
	AN = 24.608-0.01923*TEMK	RATE	164
	A = 1.609E5*EXP(14633./TEMK)	RATE	165
	ALF = (-5.225E-4+A.716E-7*TEMK)/(23.55-.01796*TEMK)	RATE	166
	SRATE = A*(SINH(ALF*SE(I,J)))+AN*EXP(-32209./TEMK)	RATE	167
	CRATE(I,J) = SRATE*(TIME(3)-TIME(2))/100.	RATE	168
190	CONTINUE	RATE	169
C		RATE	170
	GO TO 220	RATE	171
200	CONTINUE	RATE	172
C	START THE FUEL CREEP CALCULATION	RATE	173
C		RATE	174
	IF (TEMK,LT,1000.0) GO TO 210	RATE	175
	IF (TEMK,GT,TPLASK) TEMK = TPLASK	RATE	176
	TEM = TEMK	RATE	177
	SRATE = 510.*SE(I,J)**1.3*EXP(-45798./TEM)	RATE	178
	CRATE(I,J) = SRATE*(TIME(3)-TIME(2))	RATE	179
	GO TO 220	RATE	180
210	CRATE(I,J) = 0.0	RATE	181
220	CONTINUE	RATE	182
	GO TO 340	RATE	183
C		RATE	184
C	CHANGES TO ALLOW FOR STRAIN HARDENING OPTION	RATE	185
C		RATE	186
C	START STRAIN HARDENING OPTION	RATE	187
C		RATE	188
230	CONTINUE	RATE	189
	DO 330 J=1,N	RATE	190
	DO 330 I=1,NNN	RATE	191
	ENR = 1.-PSP(22)+1.3*(1.-PSP(23))*PSP(21)	RATE	192
	FFLUX = 4.934E10*PSP(J)/ENR	RATE	193
	FFLUX = FFLUX*FLXR	RATE	194
	TEMK = (TELM(I,J,NEW)-32.0)/1.R+273.0	RATE	195
	IF (I,LE,NRF,AND,J,LE,NA) GO TO 300	RATE	196
	TEST = TERP(200.,AA,AA(1,4),NN)	RATE	197
		RATE	198

```

      IF (TEST.GT.24,E6) GO TO 280
C      ZETA = TIME(3)
240  CONTINUE
C      STRAIN HARDENING
C      CTO = C10*(1+.021*ALOG(1+FFLUX/6.E12))*EXP(-C11*SE(I,J))
      DELT = CTO-TEMK
      TNH = TANH(100.*DELT)
      QA = C6-.061*SE(I,J)
      QI = C8-.038*SE(I,J)
      TAE = -(C6*(1.-TNH)+QI*(1.+TNH))/(2.*GAS*TEMK)
      IF (FFLUX.LE.10.) GO TO 250
      CFF = 0.85*ALOG(FFLUX)
      GO TO 260
250  CONTINUE
      CFF = (QA/(4.*GAS*CTO))*(3.-CTO/TEMK)
260  CONTINUE
      FAE = (CFF-C9)*(1.+TNH)/2.
      ATERM = EXP(TAE+FAE)
      SCR = C3*ATERM*((1.-TNH)*SINH(C5*SE(I,J))+(1.+TNH)*SINH(C7*SE(I,J)))/2.
      TC1 = C1*(1.-EXP(-C2*ZETA))
      TRC1 = C1*C2*EXP(-C2*ZETA)
      CSTR1 = SCR*(ZETA+TC1)
      DCST1 = SCR*(1.0+TRC1)
      ERROR = CRPG(I,J)-CSTR1
      XFACT = 0.001*CRPG(I,J)
      ABERR = ABS(ERROR)
      IF (ABERR.LT.XFACT) GO TO 270
      H = ERROR/DCST1
      ZETA = ZETA+H
      GO TO 240
270  CONTINUE
C      NOW COMPARE INCREMENTAL STRAIN INCREMENT FOR STRAIN HARDENING
C      MODE
C      ZETAN = ZETA+DLTM
C      COMPUTE INCREMENTAL STRAIN FOR TIME STEP
C      TC2 = C1*(1.-EXP(-C2*ZETAN))
      CSTR2 = SCR*(ZETAN+TC2)
      CRATE(I,J) = CSTR2-CSTR1
C      GO TO 290
280  CONTINUE
      A = 1.609E5*EXP(14633./TEMK)
      AN = 24.608-0.01923*TEMK
      ALF = (-5.225E-4+8.716E-7*TEMK)/(23.55-.01796*TEMK)
      SRATE = A*(SINH(ALF*SE(I,J)))+AN*EXP(-32209./TEMK)
      CRATE(I,J) = SRATE*(TIME(3)-TIME(2))/100.
290  CONTINUE
C      GO TO 330
300  CONTINUE

```

```

RATE 199
RATE 200
RATE 201
RATE 202
RATE 203
RATE 204
RATE 205
RATE 206
RATE 207
RATE 208
RATE 209
RATE 210
RATE 211
RATE 212
RATE 213
RATE 214
RATE 215
RATE 216
RATE 217
RATE 218
RATE 219
RATE 220
RATE 221
RATE 222
RATE 223
RATE 224
RATE 225
RATE 226
RATE 227
RATE 228
RATE 229
RATE 230
RATE 231
RATE 232
RATE 233
RATE 234
RATE 235
RATE 236
RATE 237
RATE 238
RATE 239
RATE 240
RATE 241
RATE 242
RATE 243
RATE 244
RATE 245
RATE 246
RATE 247
RATE 248
RATE 249
RATE 250
RATE 251
RATE 252
RATE 253
RATE 254
RATE 255

```

SUBROUTINE RATE

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

C		RATE	256
C	START FUEL CREEP CALCULATION NOTE NO PRIMARY CREEP FOR FUEL	RATE	257
C		RATE	258
	IF (TEMK.LT.1000.0) GO TO 310	RATE	259
	IF (TEMK.GT.TPLASK) TEMK = TPLASK	RATE	260
	SRATE = 510.*SE(I,J)+1.3*EXP(-45798./TEMK)	RATE	261
	CRATE(I,J) = SRATE*(TIME(3)-TIME(2))	RATE	262
	GO TO 320	RATE	263
310	CRATE(I,J) = 0.0	RATE	264
320	CONTINUE	RATE	265
330	CONTINUE	RATE	266
340	CONTINUE	RATE	267
	DO 350 I=LOW,NRAD	RATE	268
	DO 350 J=1,NAXL	RATE	269
	OLDSE(I,J) = SE(I,J)	RATE	270
350	CONTINUE	RATE	271
360	CONTINUE	RATE	272
	RETURN	RATE	273
	END	RATE	274

SUBROUTINE RELOC

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

C      SUBROUTINE RELOC (DELGO,GAP,BURNUP,P,IFLAG)
C      RELOC CALCULATES THE FUEL RADIAL EXPANSION CAUSED BY CRACK
C      FORMATION.
C      B = EXP(-4.+BURNUP**+.25)
      DELGD = (42.+(B/(1.+B))+0.9*P+3.)*GAP/100.
      IF (IFLAG.LT.0) DELGD = DELGD*.25*GAP
      IF (DELGD.LT.0) DELGD = 0.0
      DELGD = DELGD/2.
      RETURN
      END

```

```

RELCL 2
RELCL 3
RELCL 4
RELCL 5
RELCL 6
RELCL 7
RELCL 8
RELCL 9
RELCL 10
RELCL 11
RELCL 12
RELCL 13

```

SUBROUTINE SIMQ (A,B,N)

PURPOSE

OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS,
 $AX=B$

USAGE

CALL SIMQ(A,B,N,KS)

DESCRIPTION OF PARAMETERS

A - MATRIX OF COEFFICIENTS STORED COLUMNWISE. THESE ARE
 DESTROYED IN THE COMPUTATION. THE SIZE OF MATRIX A IS
 N BY N.

B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE ARE
 REPLACED BY FINAL SOLUTION VALUES, VECTOR X.

N - NUMBER OF EQUATIONS AND VARIABLES. N MUST BE .GT. ONE.

KS - OUTPUT DIGIT

0 FOR A NORMAL SOLUTION

1 FOR A SINGULAR SET OF EQUATIONS

REMARKS

MATRIX A MUST BE GENERAL.

IF MATRIX IS SINGULAR, SOLUTION VALUES ARE MEANINGLESS.

AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX
 INVERSION (MINV) AND MATRIX PRODUCT (GMPRO).

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

METHOD

METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL
 DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGING
 ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL
 ELEMENTS.

THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN
 N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES IS
 CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL SOLUTION
 VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),
 VARIABLE 2 IN B(2),....., VARIABLE N IN B(N).

IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.0,
 THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1. THIS
 TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT.

.....
 DIMENSION A(1), B(1)

FORWARD SOLUTION

TOL = 0.0

KS = 0

JJ = -N

DO 170 J=1,N

JY = J+1

JJ = JJ+N+1

BIGA = 0

IT = JJ-J

SIMQ	2
SIMQ	3
SIMQ	4
SIMQ	5
SIMQ	6
SIMQ	7
SIMQ	8
SIMQ	9
SIMQ	10
SIMQ	11
SIMQ	12
SIMQ	13
SIMQ	14
SIMQ	15
SIMQ	16
SIMQ	17
SIMQ	18
SIMQ	19
SIMQ	20
SIMQ	21
SIMQ	22
SIMQ	23
SIMQ	24
SIMQ	25
SIMQ	26
SIMQ	27
SIMQ	28
SIMQ	29
SIMQ	30
SIMQ	31
SIMQ	32
SIMQ	33
SIMQ	34
SIMQ	35
SIMQ	36
SIMQ	37
SIMQ	38
SIMQ	39
SIMQ	40
SIMQ	41
SIMQ	42
SIMQ	43
SIMQ	44
SIMQ	45
SIMQ	46
SIMQ	47
SIMQ	48
SIMQ	49
SIMQ	50
SIMQ	51
SIMQ	52
SIMQ	53
SIMQ	54
SIMQ	55
SIMQ	56
SIMQ	57
SIMQ	58

```

      DO 110 I=J,N
C      SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN
C      IJ = IT+1
      IF (ABS(BIGA)-ABS(A(IJ))) 100,110,110
100    BIGA = A(IJ)
      IMAX = I
110    CONTINUE
C      TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)
C      IF (ABS(BIGA)=TOL) 120,120,130
120    KS = 1
      RETURN
C      INTERCHANGE ROWS IF NECESSARY
C      I1 = J+N*(J-2)
130    IT = IMAX-J
      DO 140 K=J,N
      I1 = I1+N
      I2 = I1+IT
      SAVE = A(I1)
      A(I1) = A(I2)
      A(I2) = SAVE
C      DIVIDE EQUATION BY LEADING COEFFICIENT
C      A(I1) = A(I1)/BIGA
140    SAVE = B(IMAX)
      B(IMAX) = B(J)
      B(J) = SAVE/BIGA
C      ELIMINATE NEXT VARIABLE
C      IF (J=N) 150,180,150
150    IQS = N*(J-1)
      DO 170 IX=JY,N
      IXJ = IQS+IX
      IT = J-IX
      DO 160 JX=JY,N
      IXJX = N*(JX-1)+IX
      JJX = IXJX+IT
160    A(IXJX) = A(IXJX)-(A(IXJ)*A(JJX))
170    B(IX) = B(IX)-(B(J)*A(IXJ))
C      BACK SOLUTION
C      NY = N-1
180    IT = N*N
      DO 190 J=1,NY
      IA = IT-J
      IB = N-J
      IC = N
      DO 190 K=1,J
      B(IB) = B(IB)-A(IA)*B(IC)
      IA = IA-N
190    IC = IC-1
      RETURN
      END

```

```

      SIMQ      59
      SIMQ      60
      SIMQ      61
      SIMQ      62
      SIMQ      63
      SIMQ      64
      SIMQ      65
      SIMQ      66
      SIMQ      67
      SIMQ      68
      SIMQ      69
      SIMQ      70
      SIMQ      71
      SIMQ      72
      SIMQ      73
      SIMQ      74
      SIMQ      75
      SIMQ      76
      SIMQ      77
      SIMQ      78
      SIMQ      79
      SIMQ      80
      SIMQ      81
      SIMQ      82
      SIMQ      83
      SIMQ      84
      SIMQ      85
      SIMQ      86
      SIMQ      87
      SIMQ      88
      SIMQ      89
      SIMQ      90
      SIMQ      91
      SIMQ      92
      SIMQ      93
      SIMQ      94
      SIMQ      95
      SIMQ      96
      SIMQ      97
      SIMQ      98
      SIMQ      99
      SIMQ     100
      SIMQ     101
      SIMQ     102
      SIMQ     103
      SIMQ     104
      SIMQ     105
      SIMQ     106
      SIMQ     107
      SIMQ     108
      SIMQ     109
      SIMQ     110
      SIMQ     111
      SIMQ     112
      SIMQ     113
      SIMQ     114
      SIMQ     115
      SIMQ     116
      SIMQ     117
      SIMQ     118
      SIMQ     119

```



```

SUBROUTINE SOLVET (B,BB,NODE,NODEC,NODET,XK,XKC,XKT,CC,THETED,KAN, SOLVET      2
+IDU,RF,BC,BBF,BBC,ELSTSA,TELM,TDELE,R,ARAA,RRRA,NABFD,NOELD,MAXCD, SOLVET      3
+XKCC,XKTT,BBSV,ESTSV,NFF2D,MAXTD,NARFD,ESTNV) SOLVET      4
  INTEGER BASE,BASS,HASN SOLVET      5
  DIMENSION ELSTSA(4,1), R(1) SOLVET      6
  DIMENSION IDU(1), KAN(1) SOLVET      7
  DIMENSION TELM(NOELD,NABFD,2) SOLVET      8
  DIMENSION XKCC(MAXCD,NABFD), XKTT(MAXTD,NARFD) SOLVET      9
  DIMENSION XKT(1), XK(1), XKC(1), NODEC(1), NODE(1), NODET(1), SOLVET     10
+BF(1), BC(1),BBF(1), BBC(1), B(1), BB(1), CC(4,4,1), THETED(4 SOLVET     11
+,1) SOLVET     12
  DIMENSION ARAA(1), RRRA(1) SOLVET     13
  DIMENSION TDELE(NOELD) SOLVET     14
  DIMENSION BBSV(NFF2D,NARFD),ESTSV(4,NOELD,NARFD),ESTNV(4,NOELD, SOLVET     15
+ NARFD) SOLVET     16
C SOLVET     17
  COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IPLSM,IELSTN,IELSTR PELCOM      2
+,IKSAVE,IPRESS,ITEM,ILAST,ILIMIT,INDC,INOF,INDT,INODE,INODEC,INOD PELCOM      3
+ET, IR,IRBR, ISA,ISAC,ISAP,ISR,ISBB,ISBBC,I PELCOM      4
+SBBF,ISBC,ISBF,ISK,ISKC,ISTRES, ITDELE, IXK,IXKC,IXKI PELCOM      5
+,IXLOAD, ICC,ITHET,ICA,ITHTAX, INADET,IRA,IY,IAELM,I PELCOM      6
+ARAA,IRA,IRBA,IRBRA,IXLDA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I PELCOM      7
+PSLA,IAK,IACN,IKAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM      8
+IIDO,ITME,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM      9
+, IXKCC,IXKTT PELCOM     10
+,ITFB, IYCB, IBBSV,IESTSV,IESTNV PELCOM     11
+,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDSE,IPSSV,IPSSRT,JIIJ,IIPDS PELCOM     12
+,IBASE,ITFR,IYM,IZ PELCOM     13
+,IDSPSV PELCOM     14
  COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM     15
+DEL PELCOM     16
  COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IOF,NFF,MAXDC,NETOT,NE PELCOM     17
+WF,NEHT,KA,NARF,NARC,NART,NARF,NARC,NART,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM     18
+NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CR,FRUEN,KZI,ISSTOP,IPLAS,NKN PELCOM     19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM     20
+MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20.2),ILK,NOLD PELCOM     21
+, KPLAS,ICNT,RRAKR,IPAK,RAKSUM,IFLEND,ISP PELCOM     22
  COMMON /LASTIC/ E,XNU,ALPHA PELCOM     23
  COMMON /CONTNT/ PI PELCOM     24
  COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM     25
  COMMON /CRP/ PSP(25),TPLASK PELCOM     26
  COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN COMC      2
+, NCNN,ISTR,ITER COMC      3
+, FLXR COMC      4
  COMMON/FEP/HMF,NT,KKZZZ,ISTOP,RDOLD(20),RRSIN(20),ICNOLD(20) COMC      5
+, IODOLD(20) COMC      6
+,PINOLD,POUTLD,POLD,NTOLD(20) COMC      7
+, ICHNG,CRPAD(20) COMC      8
+,STEPS COMC      9
C SOLVET     20
  COMMON D(1) SOLVET     21
C SOLVET     22
  CALL MOVEKA (0,B,NFF2) SOLVET     23
  CALL MOVEKA (0,BB,NFF2) SOLVET     24
  CALL MOVEKA (0,D(IELSTR),NOEL*4) SOLVET     25
  CALL MOVEKA (0,D(IELSTN),NOEL*4) SOLVET     26
  CALL MOVEKA (0,BBF,NF4) SOLVET     27

```

	CALL MOVEKA (0,BF,NF4)	SOLVET	28
	CALL MOVEKA (0,BBC,NC4)	SOLVET	29
	CALL MOVEKA (0,BC,NC4)	SOLVET	30
	IF (KKA,EQ,2) GO TO 100	SOLVET	31
	CALL MOVEKA (0,ELSTSA,4*NTAE)	SOLVET	32
100	CONTINUE	SOLVET	33
	MIN = NF+NC+1	SOLVET	34
	MAX = 2*NF+NC-1	SOLVET	35
	MM = 4*KA	SOLVET	36
C	SUBTRACT OFF THE LAST ELASTIC STRESS-STRAIN INCREMENTS TO	SOLVET	37
C	PREPARE FOR PLASTIC CALCULATION.	SOLVET	38
	IF (KPLAS,NE,1.OR,KKA,NE,1) GO TO 120	SOLVET	39
	DO 110 J=1,NOEL	SOLVET	40
	BASN = ITSTNR-1+(KA-1)*NOEL+4+4*(J-1)	SOLVET	41
	BASS = ITSTR-1+(KA-1)*NOEL+4+4*(J-1)	SOLVET	42
	DO 110 K=1,4	SOLVET	43
	D(BASS+K) = D(BASS+K)-ESTSV(K,J,KA)	SOLVET	44
	D(BASN+K) = D(BASN+K)-ESTNV(K,J,KA)	SOLVET	45
110	CONTINUE	SOLVET	46
120	CONTINUE	SOLVET	47
	IF (IRAK,EQ,0) GO TO 180	SOLVET	48
C	PSEUDO=ELASTIC LOAD STEP.	SOLVET	49
	IF (ICON,EQ,0) GO TO 140	SOLVET	50
	DO 130 J=1,NMX	SOLVET	51
	B(J) = BBSV(J,KA)*(1.-RRAKH)	SOLVET	52
130	CONTINUE	SOLVET	53
	GO TO 160	SOLVET	54
140	CONTINUE	SOLVET	55
	DO 150 J=1,NC4	SOLVET	56
	BC(J) = BBSV(NF4+J,KA)*(1.-RRAKR)	SOLVET	57
150	CONTINUE	SOLVET	58
160	CONTINUE	SOLVET	59
	DO 170 KK=1,NOEL	SOLVET	60
	IBASTH = NOEL+4*(KA-1)+4*(KK-1)+ITHSAV-1	SOLVET	61
	DO 170 JJ=1,4	SOLVET	62
	THETEO(JJ,KK) = D(IBASTH+JJ)*(1.-RRAKR)	SOLVET	63
170	CONTINUE	SOLVET	64
	GO TO 270	SOLVET	65
180	CONTINUE	SOLVET	66
C	IMPOSE AXIAL STRESSES (SKIP FOR KKA=1)	SOLVET	67
	IF (KKA,EQ,1) GO TO 260	SOLVET	68
	IF (KA,EQ,NARC) GO TO 230	SOLVET	69
	IF (ICON,EQ,0) GO TO 200	SOLVET	70
	KU = KA+1	SOLVET	71
C	SEARCH FOR NEXT CONTACT REGION ABOVE CURRENT (KA) REGION	SOLVET	72
	DO 190 I=KU,NARC	SOLVET	73
	IF (KAN(I),EQ,1) GO TO 190	SOLVET	74
	MM = 4*I	SOLVET	75
	GO TO 200	SOLVET	76
190	CONTINUE	SOLVET	77
200	CONTINUE	SOLVET	78
C	AVERAGE THE AXIAL STRESS IN THE FUEL	SOLVET	79
	AVGSTS = 0.0	SOLVET	80
	ITST = 4*NARC	SOLVET	81
	IF (MM,EQ,ITST) GO TO 230	SOLVET	82
	TA = ARAA(MM)+RBRA(MM)+ARAA(MM-1)+RBRA(MM-1)+ARAA(MM-2)+RBRA(MM-2)	SOLVET	83
	+4*ARAA(MM-3)+RBRA(MM-3)	SOLVET	84

	DO 210 II=1,4	SOLVET	85
	MK = MM-(II-1)	SOLVET	86
	WI = ARAA(MK)*RRRA(MK)/TA	SOLVET	87
	AVGSTS = AVGSTS+WI*ELSTSA(3,MK)	SOLVET	88
210	CONTINUE	SOLVET	89
C	IMPOSE AXIAL STRESSES ON THE FUEL	SOLVET	90
	DO 220 I=MIN,MAX	SOLVET	91
	AREA = R(I+1)**2-R(I)**2	SOLVET	92
	XLINE = 2.0*(R(I)+R(I+1))	SOLVET	93
	ZINT = AVGSTS*AREA/XLINE	SOLVET	94
	ZFS = ZINT*PI*2.0*R(I)	SOLVET	95
	ZFE = ZINT*R(I+1)*2.0*PI	SOLVET	96
	IN = NODET(I)+1	SOLVET	97
	INN = NODET(I+1)+1	SOLVET	98
	BB(IN) = BB(IN)+ZFS	SOLVET	99
	B(IN) = BB(IN)	SOLVET	100
	BB(INN) = BB(INN)+ZFE	SOLVET	101
	B(INN) = BB(INN)	SOLVET	102
	IFF = NODE(I)+1	SOLVET	103
	INF = NODE(I+1)+1	SOLVET	104
	BBF(IFF) = BBF(IFF)+ZFS	SOLVET	105
	BF(IFF) = BBF(IFF)	SOLVET	106
	BBF(INF) = BBF(INF)+ZFE	SOLVET	107
	BF(INF) = BBF(INF)	SOLVET	108
220	CONTINUE	SOLVET	109
230	CONTINUE	SOLVET	110
C	AVERAGE THE AXIAL STRESS FOR THE CLAD AND IMPOSE IT	SOLVET	111
	MM = MM+NAEF	SOLVET	112
	AVGSTS = 0.0	SOLVET	113
	TA = ARAA(MM)*RRRA(MM)+ARAA(MM-1)*RRRA(MM-1)+ARAA(MM-2)*RRRA(MM-2)	SOLVET	114
	++ARAA(MM-3)*RRRA(MM-3)	SOLVET	115
	DO 240 II=1,4	SOLVET	116
	MK = MM-(II-1)	SOLVET	117
	WI = ARAA(MK)*RRRA(MK)/TA	SOLVET	118
	AVGSTS = AVGSTS+WI*ELSTSA(3,MK)	SOLVET	119
240	CONTINUE	SOLVET	120
	MAX = 2*NF+NC+1	SOLVET	121
	MAXX = NFF+1	SOLVET	122
	DO 250 I=MAX,MAXX	SOLVET	123
	AREA = R(I+1)**2-R(I)**2	SOLVET	124
	XLINE = 2.0*(R(I)+R(I+1))	SOLVET	125
	ZINT = AVGSTS*AREA/XLINE	SOLVET	126
	ZFS = ZINT*PI*2.0*R(I)	SOLVET	127
	ZFE = ZINT*R(I+1)*2.0*PI	SOLVET	128
	IN = NODET(I)+1	SOLVET	129
	INN = NODET(I+1)+1	SOLVET	130
	BB(IN) = BB(IN)+ZFS	SOLVET	131
	B(IN) = BB(IN)	SOLVET	132
	BB(INN) = BB(INN)+ZFE	SOLVET	133
	B(INN) = BB(INN)	SOLVET	134
	IN = NODEC(I)+1	SOLVET	135
	INN = NODEC(I+1)+1	SOLVET	136
	BBC(IN) = BBC(IN)+ZFS	SOLVET	137
	BC(IN) = BBC(IN)	SOLVET	138
	BBC(INN) = BBC(INN)+ZFS	SOLVET	139
	BC(INN) = BBC(INN)	SOLVET	140
250	CONTINUE	SOLVET	141

260	CONTINUE	SOLVET	142
C		SOLVET	143
C	PREPARE INPUT ARRAY FOR SOLUTION	SOLVET	144
C	WE FIRST ASSUME THAT FUEL AND CLAD IS NOT IN CONTACT	SOLVET	145
C		SOLVET	146
C	CREATE LOAD VECTOR	SOLVET	147
C		SOLVET	148
	CALL LOAD (D(IARE),B,RR,BC,RF,BBC,BBF,NODE,NODEC,NODET,D(IR),D(IRB	SOLVET	149
	+R),D(ICC),D(ITHET),TDELE,TELM,D(IXLOAD),D(IELM),NOEL,D(ITCRP),NABF	SOLVET	150
	+KAN,IDO,D(ITHSAV))	SOLVET	151
270	CONTINUE	SOLVET	152
	IF (ICON.EQ.1) GO TO 330	SOLVET	153
	IF (ISSTOP.NE.1) GO TO 290	SOLVET	154
	IF (KKA.NE.2) GO TO 290	SOLVET	155
	IF (NKN.NE.1) GO TO 290	SOLVET	156
	DO 280 J=1,NC4	SOLVET	157
	BBSV(J+NF4,KA) = BC(J)	SOLVET	158
280	CONTINUE	SOLVET	159
290	CONTINUE	SOLVET	160
C		SOLVET	161
	IOP = 1	SOLVET	162
	IF (ICUT.NE.0.OR.IPLAS.NE.0.OR.KZ1.EQ.2.OR.ICNOLD(KA).NE.ICON) GO	SOLVET	163
	+TO 310	SOLVET	164
	IF (IDOLD(KA).NE.IDO(KA)) GO TO 310	SOLVET	165
	IOP = -1	SOLVET	166
	DO 300 IJ=1,MAXC	SOLVET	167
	XKC(IJ) = XKCC(IJ,KA)	SOLVET	168
300	CONTINUE	SOLVET	169
310	CONTINUE	SOLVET	170
	CALL DMCHB (BC,XKC,NC4,1,7,IOP,1,E=14,IER)	SOLVET	171
	DO 320 IJ=1,MAXC	SOLVET	172
	XKCC(IJ,KA) = 0.0	SOLVET	173
	XKCC(IJ,KA) = XKC(IJ)	SOLVET	174
320	CONTINUE	SOLVET	175
C		SOLVET	176
	GO TO 390	SOLVET	177
C		SOLVET	178
C	FUEL AND CLAD ARE NOW IN CONTACT	SOLVET	179
C		SOLVET	180
330	CONTINUE	SOLVET	181
	IF (ISSTOP.NE.1) GO TO 350	SOLVET	182
	IF (NKN.NE.1) GO TO 350	SOLVET	183
	IF (KKA.NE.2) GO TO 350	SOLVET	184
	DO 340 J=1,NFF2	SOLVET	185
	BBSV(J,KA) = BB(J)	SOLVET	186
340	CONTINUE	SOLVET	187
350	CONTINUE	SOLVET	188
	IDF = 1	SOLVET	189
C		SOLVET	190
	IOP = 1	SOLVET	191
	IF (ICUT.NE.0.OR.IPLAS.NE.0.OR.KZ1.EQ.2.OR.ICNOLD(KA).NE.ICON) GO	SOLVET	192
	+TO 370	SOLVET	193
	IOP = -1	SOLVET	194
	DO 360 IJ=1,MAXT	SOLVET	195
	XKT(IJ) = XKTT(IJ,KA)	SOLVET	196
360	CONTINUE	SOLVET	197
370	CONTINUE	SOLVET	198

SUBROUTINE SOLVET

74/74

OPT=1

FTN 4.6+452

15/02/78

17.25.31

C	CALL DMCHB (8,XKT,NMX,1,7,IOP,1.E-14,IER)	SOLVET	199
	DO 380 IJ=1,MAXT	SOLVET	200
	XKIT(IJ,KA) = 0.0	SOLVET	201
	XKIT(IJ,KA) = XKT(IJ)	SOLVET	202
380	CONTINUE	SOLVET	203
390	CONTINUE	SOLVET	204
	ICNOLD(KA) = ICON	SOLVET	205
	IDOOLD(KA) = IDO(KA)	SOLVET	206
	CALL STRES (8,HC,BF,D(IELM),D(IELSTN),D(IELSTR),D(IEPSLM),NODE,NOD	SOLVET	207
	+EC,NODET,CC,THETEO,D(ISTRES),D(ITSTNR),D(ITSTSR),D(ILSTRA),D(ILSTN	SOLVET	208
	+A),NOEL,D(IDISPC),D(IDISPF),KAN,IDU,NFF1,D(IARAA),D(IRBRA))	SOLVET	209
C		SOLVET	210
	IF (NKN,NE,1,OR,ISSTOP,NE,1,OR,KKA,NE,2) GO TO 410	SOLVET	211
	DO 400 J=1,NOEL	SOLVET	212
	BASE = IELSTR-1+4*(J-1)	SOLVET	213
	BASN = IELSTN-1+4*(J-1)	SOLVET	214
	DO 400 K=1,4	SOLVET	215
	ESTSV(K,J,KA) = D(BASE+K)	SOLVET	216
	ESTNV(K,J,KA) = D(BASN+K)	SOLVET	217
400	CONTINUE	SOLVET	218
410	CONTINUE	SOLVET	219
	CALL XFACE (D(IELSTR),D(IPRESS))	SOLVET	220
C		SOLVET	221
C		SOLVET	222
C		SOLVET	223
C		SOLVET	224
C		SOLVET	225
	RETURN	SOLVET	226
C		SOLVET	227
	END	SOLVET	228

```

SUBROUTINE SORTP (X,N,Y,IPOS)
C THIS ROUTINE SORTS THE ELEMENTS OF THE INPUT VECTOR X AND PUTS THE
C ELEMENTS INTO THE VECTOR Y. IT ALSO CARRIES ALONG THE INDEX NUMBE
C OF EACH ORDERED OBSERVATION--THAT IS, IT CARRIES ALONG THE POSITIO
C THE I-TH ORDERED OBSERVATION (FOR EACH I) AS IT WAS IN THE ORIGINA
C UNORDERED DATA VECTOR X. THESE POSITIONS ARE PLACED IN THE VECTOR
C THIS ROUTINE IS USEFUL IN ATTEMPTING TO LOCATE THE MINIMUM, THE MA
C OR SOME OTHER ORDERED OBSERVATION OF INTEREST IN THE ORIGINAL UNOR
C INPUT VECTOR X.
C THE INPUT TO THIS ROUTINE IS THE SINGLE PRECISION VECTOR X OF
C (UNSORTED) OBSERVATIONS, THE INTEGER VALUE N (= SAMPLE SIZE),
C AN EMPTY SINGLE PRECISION VECTOR Y INTO WHICH THE SORTED OBSERVATI
C WILL BE PLACED, AND AN EMPTY SINGLE PRECISION VECTOR XPOS INTO WHI
C POSITIONS OF THE SORTED OBSERVATIONS WILL BE PLACED.
C THE OUTPUT FROM THIS ROUTINE IS THE SINGLE PRECISION VECTOR Y INTO
C THE SORTED OBSERVATIONS HAVE BEEN PLACED, AND THE SINGLE PRECISION
C XPOS INTO WHICH THE POSITIONS OF THE SORTED OBSERVATIONS HAVE BEEN
C RESTRICTIONS ON THE MAXIMUM ALLOWABLE VALUE OF N--THE DIMENSIONS
C OF VECTORS IU AND IL (DEFINED AND USED INTERNALLY WITHIN THIS ROUT
C DETERMINE THE MAXIMUM ALLOWABLE VALUE OF N FOR THIS
C ROUTINE. IF IU AND IL EACH HAVE DIMENSION K, THEN N MAY NOT EXCEE
C 2**(K+1) - 1. FOR THIS ROUTINE AS WRITTEN, THE DIMENSIONS OF IU A
C HAVE BEEN SET TO 36, THUS THE MAXIMUM ALLOWABLE VALUE OF N IS
C APPROXIMATELY 137 BILLION. SINCE THIS EXCEEDS THE MAXIMUM ALLOWAB
C VALUE FOR AN INTEGER VARIABLE IN MANY COMPUTERS, AND SINCE A SORT
C BILLION ELEMENTS IS PRESENTLY IMPRACTICAL AND UNLIKELY, THEREFORE
C TEST FOR WHETHER THE INPUT SAMPLE SIZE N EXCEEDS 137 BILLION HAS B
C INCORPORATED INTO THIS ROUTINE. IT IS THUS ASSUMED THAT THERE IS
C (PRACTICAL) RESTRICTION ON THE MAXIMUM VALUE OF N FOR THIS ROUTINE
C PRINTING--NONE UNLESS AN ERROR CONDITION EXISTS
C THIS ROUTINE IS SINGLE PRECISION IN INTERNAL OPERATION.
C SUBROUTINES NEEDED--NONE
C SORTING METHOD--BINARY SORT
C REFERENCE--CACH MARCH 1969, PAGE 186 (BINARY SORT ALGORITHM BY RIC
C C. SINGLETON.
C --CACH JANUARY 1970, PAGE 54.
C --CACH OCTOBER 1970, PAGE 624.
C --JACM JANUARY 1961, PAGE 41.
C WRITTEN BY JAMES J. FILLIBEN, STATISTICAL ENGINEERING LABORATORY (
C NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. 20234 JUNE 19
C DIMENSION X(1),Y(1),IPOS(1)
C COMMON /IODEV/ IPR,IRD
C DIMENSION IU(36),IL(36)
C CHECK THE INPUT ARGUMENTS FOR ERRORS
C IPR = 6
C IF (N.LT.1) GO TO 120
C IF (N.EQ.1) GO TO 130
C HOLD = X(1)
C DO 100 I=2,N
C IF (X(I).NE.HOLD) GO TO 140
C 100 CONTINUE
C WRITE (IPR,150) HOLD
C DO 110 I=1,N
C Y(I) = X(I)
C IPOS(I) = I
C 110 CONTINUE
C RETURN

```

```

SORTP 2
SORTP 3
SORTP 4
SORTP 5
SORTP 6
SORTP 7
SORTP 8
SORTP 9
SORTP 10
SORTP 11
SORTP 12
SORTP 13
SORTP 14
SORTP 15
SORTP 16
SORTP 17
SORTP 18
SORTP 19
SORTP 20
SORTP 21
SORTP 22
SORTP 23
SORTP 24
SORTP 25
SORTP 26
SORTP 27
SORTP 28
SORTP 29
SORTP 30
SORTP 31
SORTP 32
SORTP 33
SORTP 34
SORTP 35
SORTP 36
SORTP 37
SORTP 38
SORTP 39
SORTP 40
SORTP 41
SORTP 42
SORTP 43
SORTP 44
SORTP 45
SORTP 46
SORTP 47
SORTP 48
SORTP 49
SORTP 50
SORTP 51
SORTP 52
SORTP 53
SORTP 54
SORTP 55
SORTP 56
SORTP 57
SORTP 58

```

120	WRITE (IPR,160)	SORTP	59
	WRITE (IPR,180) N	SORTP	60
	RETURN	SORTP	61
130	WRITE (IPR,170)	SORTP	62
	Y(1) = X(1)	SORTP	63
	IPOS(1) = 1.0	SORTP	64
	RETURN	SORTP	65
140	CONTINUE	SORTP	66
150	FORMAT (1H,108H***** NON-FATAL DIAGNOSTIC--THE FIRST INPUT ARGUM	SORTP	67
	+ENT (A VECTOR) TO THE SORTP SUBROUTINE HAS ALL ELEMENTS X,E15.8,6	SORTP	68
	+H *****)	SORTP	69
160	FORMAT (1H,91H***** FATAL ERROR--THE SECOND INPUT ARGUMENT TO THE	SORTP	70
	+ SORTP SUBROUTINE IS NON-POSITIVE *****)	SORTP	71
170	FORMAT (1H,100H***** NON-FATAL DIAGNOSTIC--THE SECOND INPUT ARGUM	SORTP	72
	+ENT TO THE SORTP SUBROUTINE HAS THE VALUE 1 *****)	SORTP	73
180	FORMAT (1H,135H***** THE VALUE OF THE ARGUMENT IS ,18.6H *****)	SORTP	74
C		SORTP	75
C	COPY THE VECTOR X INTO THE VECTOR Y	SORTP	76
	DO 190 I=1,N	SORTP	77
	Y(I) = X(I)	SORTP	78
190	CONTINUE	SORTP	79
C		SORTP	80
C	DEFINE THE IPOS (POSITION) VECTOR. BEFORE SORTING, THIS WILL	SORTP	81
C	BE A VECTOR WHOSE I-TH ELEMENT IS EQUAL TO I.	SORTP	82
C		SORTP	83
	DO 200 I=1,N	SORTP	84
	IPOS(I) = I	SORTP	85
200	CONTINUE	SORTP	86
C		SORTP	87
C	CHECK TO SEE IF THE INPUT VECTOR IS ALREADY SORTED	SORTP	88
C		SORTP	89
	NM1 = N-1	SORTP	90
	DO 210 I=1,NM1	SORTP	91
	IP1 = I+1	SORTP	92
	IF (Y(I).LE.Y(IP1)) GO TO 210	SORTP	93
	GO TO 220	SORTP	94
210	CONTINUE	SORTP	95
	RETURN	SORTP	96
220	M = 1	SORTP	97
	I = 1	SORTP	98
	J = N	SORTP	99
230	IF (I.GE.J) GO TO 300	SORTP	100
240	K = I	SORTP	101
	MID = (I+J)/2	SORTP	102
	AMED = Y(MID)	SORTP	103
	IMED = IPOS(MID)	SORTP	104
	IF (Y(I).LE.AMED) GO TO 250	SORTP	105
	Y(MID) = Y(I)	SORTP	106
	IPOS(MID) = IPOS(I)	SORTP	107
	Y(I) = AMED	SORTP	108
	IPOS(I) = IMED	SORTP	109
	AMED = Y(MID)	SORTP	110
	IMED = IPOS(MID)	SORTP	111
250	L = J	SORTP	112
	IF (Y(J).GE.AMED) GO TO 270	SORTP	113
	Y(MID) = Y(J)	SORTP	114
	IPOS(MID) = IPOS(J)	SORTP	115

```

      Y(J) = AMED
      IPOS(J) = IMED
      AMED = Y(MID)
      IMED = IPOS(MID)
      IF (Y(I).LE.AMED) GO TO 270
      Y(MID) = Y(I)
      IPOS(MID) = IPOS(I)
      Y(I) = AMED
      IPOS(I) = IMED
      AMED = Y(MID)
      IMED = IPOS(MID)
      GO TO 270
260   Y(L) = Y(K)
      IPOS(L) = IPOS(K)
      Y(K) = TT
      IPOS(K) = ITT
270   L = L+1
      IF (Y(L).GT.AMED) GO TO 270
      TT = Y(L)
      ITT = IPOS(L)
280   K = K+1
      IF (Y(K).LT.AMED) GO TO 280
      IF (K.LE.L) GO TO 260
      LMI = L-1
      JMK = J-K
      IF (LMI.LE.JMK) GO TO 290
      IL(M) = I
      IU(M) = L
      I = K
      M = M+1
      GO TO 310
290   IL(M) = K
      IU(M) = J
      J = L
      M = M+1
      GO TO 310
300   M = M-1
      IF (M.EQ.0) RETURN
      I = IL(M)
      J = IU(M)
310   JMI = J-1
      IF (JMI.GE.11) GO TO 240
      IF (I.EQ.1) GO TO 230
      I = I-1
320   I = I+1
      IF (I.LE.J) GO TO 300
      AMED = Y(I+1)
      IMED = IPOS(I+1)
      IF (Y(I).LE.AMED) GO TO 320
      K = I
330   Y(K+1) = Y(K)
      IPOS(K+1) = IPOS(K)
      K = K+1
      IF (AMED.LT.Y(K)) GO TO 330
      Y(K+1) = AMED
      IPOS(K+1) = IMED
      GO TO 320

      END

```

```

SORTP      116
SORTP      117
SORTP      118
SORTP      119
SORTP      120
SORTP      121
SORTP      122
SORTP      123
SORTP      124
SORTP      125
SORTP      126
SORTP      127
SORTP      128
SORTP      129
SORTP      130
SORTP      131
SORTP      132
SORTP      133
SORTP      134
SORTP      135
SORTP      136
SORTP      137
SORTP      138
SORTP      139
SORTP      140
SORTP      141
SORTP      142
SORTP      143
SORTP      144
SORTP      145
SORTP      146
SORTP      147
SORTP      148
SORTP      149
SORTP      150
SORTP      151
SORTP      152
SORTP      153
SORTP      154
SORTP      155
SORTP      156
SORTP      157
SORTP      158
SORTP      159
SORTP      160
SORTP      161
SORTP      162
SORTP      163
SORTP      164
SORTP      165
SORTP      166
SORTP      167
SORTP      168
SORTP      169
SORTP      170
SORTP      171
SORTP      172

SORTP      173

```



```

      SUBROUTINE STFAXL (AELM,AK,ARAA,EP3LA,IACON,NADET,RA,RRRA,STRS4,TW STFAXL      2
      +,NTAED,XLOADA,YACF,CA,KAN) STFAXL      3
C      PURPOSE OF THE SUBROUTINE IS TO COMPUTE THE AXIAL STIFFNESS STFAXL      4
C      MODEL FOR THE FURL AND FOR THE CLAD. STFAXL      5
C      FORM THE ELEMENT STIFFNESS MATRIX AND MERGE THEM INTO THE TOTAL STFAXL      6
C      MATRIX BASED UPON CONTACT CRITERIA(ICUN,EQ.1) NOT THAT CONTACT STFAXL      7
C      ALONG THE AXIS OF THE FUEL PIN MUST BE KNOWN INORDER TO DEFINE STFAXL      8
C      THE MERGING OPERATION OF THE TOTAL STIFFNESS MATRIX. STFAXL      9
C      STFAXL     10
C      COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IEPSLM,IELSTN,IELSTR PELCOM     11
      +,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD PELCOM     12
      +ET, IR,IRBR, ISA,ISAC,ISAF,ISB,ISBB,ISBBC,I PELCOM     13
      +SBBF,ISBC,ISBF,ISK,ISKC,ISTRES, ITDELE, IXX,IXKC,IXKT PELCOM     14
      +,IXLOAD, ICC,ITHET,ICA,ITHTAX, INADET,IRA,IY,IAELM,I PELCOM     15
      +ARAA,IBA,IBBA,IRBRA,IXLODA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I PELCOM     16
      +PSLA,IAK,IACN,IKAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSP,ITSTSA, PELCOM     17
      +IIDU,ITME,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM     18
      +, IXKCC,IXKTT PELCOM     19
      +,ITFB, ITCB, IBBSV,IESTSV,IESTNV PELCOM     20
      +,ITHSAV,ITHXSV,IBASAV,IPSTS,IULOSE,IPRSV,IPSSRT,JIIJ,IIPDS PELCOM     21
      +,IBASE,ITPR,IYM,IZ PELCOM     22
      +,IDSPSV PELCOM     23
      COMMON /GEOPAR/ PFI,RFO,RCI,RCU,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM     24
      +DEL PELCOM     25
      COMMON /LIMITS/ NF,NC,NDEL,NOELF,NOELC,ICON,IDF,NFF,HAXDC,NETOT,NE PELCOM     26
      +WF,NEWT,KA,NARF,NARC,NART,NARF,NABC,NABT,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM     27
      +NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CR,FRDEN,KZ1,ISSTOP,IPLAS,NKN PELCOM     28
      +,ICRPP,HAXF,HAXC,HAXT,KRNCH,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM     29
      +MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD PELCOM     30
      +, KPLAS,ICNT,RRAKR,IRAK,RAKSUM,IPLEND,ISP PELCOM     31
      COMMON /LASTIC/ E,XNU,ALPHA PELCOM     32
      COMMON /CONTNT/ PI PELCOM     33
      COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM     34
      COMMON /CRP/ PSP(25),TPLASK PELCOM     35
C      DIMENSION AKBAR(6,6) STFAXL     36
      DIMENSION AELM(3,1), RA(1), TW(NTAED,2), YACF(1 STFAXL     37
      +,IACON(20,2),KAN(1),NADET(1),AK(1),EP3LA(4,6,1),STRS STFAXL     38
      +A(4,6,1), XLOADA(6,4,1), RRRA(1), ARAA(1) STFAXL     39
      COMMON /RA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN COMC      2
      +, NCNN,ISTR1,ITER COMC      3
      +, FLXR COMC      4
      COMMON/FEF/HMF,NT,KKZZZ,ISTOP,HDDLD(20),RRSIN(20),ICNOLD(20) COMC      5
      +, IDOOLD(20) COMC      6
      +,PINOLD,POUTLD,POLD,NTOLD(20) COMC      7
      +, ICHNG,CRPAD(20) COMC      8
      +,STEPS COMC      9
      COMMON /STFMAT/ B(4,6),C(4,6),BT(6,4),PROD(4,6),XKBAR(6,6) STFAXL     19
      EQUIVALENCE (AKBAR(1,1),XKBAR(1,1)) STFAXL     20
      DIMENSION CA(4,4,1) STFAXL     21
      DIMENSION LKNT(26) STFAXL     22
      PI = 3.141592 STFAXL     23
      CALL MOVEKA (0,AK,MAXA) STFAXL     24
C      DEFINE CORRECTION ARRAY STFAXL     25
      LKNT(1) = 1 STFAXL     26
      DO 100 I=2,26 STFAXL     27

```

```

      LKNT(I) = LKNT(I-1)+1
100  CONTINUE
      KRNGH = NAXB4-24
      MCUD = 26
      IF (NAXB4,LT,37) MCUD = NAXB4
      IF (NAXB4,LT,37) KRNGH = MCUD-2
      DO 360 K=1,NTAE
        N1 = AELM(1,K)
        N2 = AELM(2,K)
        N3 = AELM(3,K)
        R1 = RA(N1)
        R2 = RA(N2)
        Z1 = YACF(N1)
        Z2 = YACF(N2)
C      COMPUTE LOCATION OF QUADRILATRIAL MID NODE BASED ON CORNER
C      COORDINATES.
        LL = (K-(K/4)*4)
        KL = LL
        IF (KL.EQ.0) KL = 4
        IF (KL.GT.1) GO TO 110
        R3 = (R1+R2)/2.0
        NN2 = AELM(2,K+1)
        ZZ2 = YACF(NN2)
        Z3 = (ZZ2+Z2)/2.0
110  CONTINUE
C
C      FORM MEAN VALUE LOCATION COORDINATES CENTROID OF EACH ELEMENT
C
        RBAR = (R1+R2+R3)/3.0
        AREA = (R2*Z3+R1*Z2+R3*Z1-R2*Z1-R3*Z2-R1*Z3)/2.0
C
C      COMPUTE AREA COORDINATE AND COEF.
C
        B1 = Z2-Z3
        B2 = Z3-Z1
        B3 = Z1-Z2
        A1 = R3-R2
        A2 = R1-R3
        A3 = R2-R1
        XL1 = 1.0/3.0
        XL2 = 1./3.
        XL3 = 1./3.
C
C      COMPUTE THE B MATRIX
C
        B(1,1) = B1
        B(2,1) = 2.*AREA*XL1/RBAR
        B(3,1) = 0.0
        B(4,1) = A1
        B(1,2) = 0.0
        B(2,2) = 0.0
        B(3,2) = A1
        B(4,2) = B1
        B(1,3) = B2
        B(2,3) = 2.*AREA*XL2/RBAR
        B(3,3) = 0.0
        B(4,3) = A2

```

```

STFAXL 28
STFAXL 29
STFAXL 30
STFAXL 31
STFAXL 32
STFAXL 33
STFAXL 34
STFAXL 35
STFAXL 36
STFAXL 37
STFAXL 38
STFAXL 39
STFAXL 40
STFAXL 41
STFAXL 42
STFAXL 43
STFAXL 44
STFAXL 45
STFAXL 46
STFAXL 47
STFAXL 48
STFAXL 49
STFAXL 50
STFAXL 51
STFAXL 52
STFAXL 53
STFAXL 54
STFAXL 55
STFAXL 56
STFAXL 57
STFAXL 58
STFAXL 59
STFAXL 60
STFAXL 61
STFAXL 62
STFAXL 63
STFAXL 64
STFAXL 65
STFAXL 66
STFAXL 67
STFAXL 68
STFAXL 69
STFAXL 70
STFAXL 71
STFAXL 72
STFAXL 73
STFAXL 74
STFAXL 75
STFAXL 76
STFAXL 77
STFAXL 78
STFAXL 79
STFAXL 80
STFAXL 81
STFAXL 82
STFAXL 83
STFAXL 84

```

```

      B(1,4) = 0.0
      B(2,4) = 0.0
      B(3,4) = A2
      B(4,4) = B2
      B(1,5) = B3
      B(2,5) = 2.*AREA*XL3/RBAR
      B(3,5) = 0.0
      B(4,5) = A3
      B(1,6) = 0.0
      B(2,6) = 0.0
      B(3,6) = A3
      B(4,6) = B3
      DO 120 J=1,6
        DO 120 I=1,4
          B(I,J) = B(I,J)/(2.*AREA)
        CONTINUE
120   C
      C
      C      NOW FOR THE C MATRIX
      C
      TF = TW(K,NEW)
      IF (K,LE,NAEF) E = TERP(TF,CF,CF(1,4),NCNN)
      IF (K,LE,NAEF) XNU = TERP(TF,CF,CF(1,5),NCNN)
      IF (K,GT,NAEF) E = TERP(TF,AA,AA(1,4),NN)
      IF (K,GT,NAEF) XNU = TERP(TF,AA,AA(1,5),NN)
      COEF = E/((1.+XNU)*(1.-2.*XNU))
      DO 140 J=1,4
        DO 130 I=1,3
          C(I,J) = XNU*COEF
          IF (I.EQ,J) C(I,J) = (1.-XNU)*COEF
          IF (J.EQ,4) C(I,J) = 0.0
        CONTINUE
130   C(4,J) = 0.0
140   CONTINUE
      C(4,4) = ((1.-2.*XNU)/2.)*COEF
      C
      C      NOW COMPUTE THE TRANSPOSE OF B
      C
      DO 150 J=1,4
        DO 150 I=1,6
          BT(I,J) = B(J,I)
        CONTINUE
150   C
      C
      C      NOW COMPUTE ELEMENTAL STIFFNESS MATRIX FIRST FORM MATRIX
      C      PRODUCT B**T+C*B
      C
      DO 170 JJ=1,6
        DO 170 III=1,4
          SUM = 0.0
          PROD(III,JJ) = 0.0
          DO 160 JJJ=1,4
            S = C(III,JJJ)*B(JJJ,JJ)
            SUM = SUM+S
          CONTINUE
160   PROD(III,JJ) = SUM
170   CONTINUE
      C
      C      NOW FORM THE FINAL PRODUCT

```

```

STFAXL      85
STFAXL      86
STFAXL      87
STFAXL      88
STFAXL      89
STFAXL      90
STFAXL      91
STFAXL      92
STFAXL      93
STFAXL      94
STFAXL      95
STFAXL      96
STFAXL      97
STFAXL      98
STFAXL      99
STFAXL     100
STFAXL     101
STFAXL     102
STFAXL     103
STFAXL     104
STFAXL     105
STFAXL     106
STFAXL     107
STFAXL     108
STFAXL     109
STFAXL     110
STFAXL     111
STFAXL     112
STFAXL     113
STFAXL     114
STFAXL     115
STFAXL     116
STFAXL     117
STFAXL     118
STFAXL     119
STFAXL     120
STFAXL     121
STFAXL     122
STFAXL     123
STFAXL     124
STFAXL     125
STFAXL     126
STFAXL     127
STFAXL     128
STFAXL     129
STFAXL     130
STFAXL     131
STFAXL     132
STFAXL     133
STFAXL     134
STFAXL     135
STFAXL     136
STFAXL     137
STFAXL     138
STFAXL     139
STFAXL     140
STFAXL     141

```

```

C
      DO 200 J=1,6
        DO 190 I=1,6
          AKBAR(I,J) = 0.0
          DO 180 II=1,4
            AKBAR(I,J) = AKBAR(I,J)+BT(I,II)*PROD(II,J)
180      CONTINUE
          AKBAR(I,J) = AKBAR(I,J)+RBAR*2.*PI*AREA
190      CONTINUE
200      CONTINUE
C
C      FORM AN ARRAY OF ALL THE AXIAL ELEMENTS, THE MERGING OPERATION
C      WILL BE DONE FOR EACH ELEMENT BASED UPON THE CONTACT RELATIONSHIP
C      IACON(KK). IACON(KK) DEFINES WHETHER NODE K IS IN CONTACT WITH
C      THE ADJACENT NODE.
C
      IF (K.GT.NAEF) GO TO 240
      L = 0
      LL = 0
C
      DO 230 I=1,6
        IF (L.GE.2) L = 0
        NNN = NADET(N1)
        IF (I.GT.2) NNN = NADET(N2)
        IF (I.GT.4) NNN = NADET(N3)
        LL = 0
C
      DO 220 J=1,6
        IF (LL.GE.2) LL = 0
        ND = NADET(N1)
        IF (J.GT.2) ND = NADET(N2)
        IF (J.GT.4) ND = NADET(N3)
C
C      COMPUTE COORDINATES OF THE MASTER STIFFNESS MATRIX FOR THE FUEL
C
      II = NNN+L
      JJ = ND+LL
      IF (II.GT.JJ) GO TO 210
      III = MCUD*(II-1)+JJ-II+1
      IF (II.GT.KRNCH) III = III-LKNT(II-KRNCH)
      AK(III) = AK(III)+AKBAR(I,J)
210      CONTINUE
      LL = LL+1
220      CONTINUE
      L = L+1
230      CONTINUE
      GO TO 280
240      CONTINUE
      L = 0
      LL = 0
      DO 270 I=1,6
        IF (L.GE.2) L = 0
        NNN = NADET(N1)
        IF (I.GT.2) NNN = NADET(N2)
        IF (I.GT.4) NNN = NADET(N3)
        LL = 0
        DO 260 J=1,6

```

```

STFAXL 142
STFAXL 143
STFAXL 144
STFAXL 145
STFAXL 146
STFAXL 147
STFAXL 148
STFAXL 149
STFAXL 150
STFAXL 151
STFAXL 152
STFAXL 153
STFAXL 154
STFAXL 155
STFAXL 156
STFAXL 157
STFAXL 158
STFAXL 159
STFAXL 160
STFAXL 161
STFAXL 162
STFAXL 163
STFAXL 164
STFAXL 165
STFAXL 166
STFAXL 167
STFAXL 168
STFAXL 169
STFAXL 170
STFAXL 171
STFAXL 172
STFAXL 173
STFAXL 174
STFAXL 175
STFAXL 176
STFAXL 177
STFAXL 178
STFAXL 179
STFAXL 180
STFAXL 181
STFAXL 182
STFAXL 183
STFAXL 184
STFAXL 185
STFAXL 186
STFAXL 187
STFAXL 188
STFAXL 189
STFAXL 190
STFAXL 191
STFAXL 192
STFAXL 193
STFAXL 194
STFAXL 195
STFAXL 196
STFAXL 197
STFAXL 198

```

	IF (LL.GE.2) LL = 0	STFAXL	199
	ND = NADET(N1)	STFAXL	200
	IF (J.GT.2) ND = NADET(N2)	STFAXL	201
	IF (J.GT.4) ND = NADET(N3)	STFAXL	202
C		STFAXL	203
C	COMPUTE THE COORDINATES OF MASTER AXIAL STIFFNESS FOR THE CLAD	STFAXL	204
C		STFAXL	205
	II = NNN+L	STFAXL	206
	JJ = ND+LL	STFAXL	207
	IF (II.GT.JJ) GO TO 250	STFAXL	208
	III = MCUD*(II-1)+JJ-II+1	STFAXL	209
	IF (II.GT.KRNCH) III = III-LKNT(II-KRNCH)	STFAXL	210
	AK(III) = AK(III)+AKBAR(I,J)	STFAXL	211
250	CONTINUE	STFAXL	212
	LL = LL+1	STFAXL	213
260	CONTINUE	STFAXL	214
	L = L+1	STFAXL	215
270	CONTINUE	STFAXL	216
280	CONTINUE	STFAXL	217
C	HAVE MERGED TOTAL STIFFNESS FOR THE FUEL AND CLADAXIAL MODEL	STFAXL	218
C		STFAXL	219
C	NOW DEVELOPE THE STEESS AND STRAIN MATHIX FOR EACH AXIAL	STFAXL	220
C	ELEMENT	STFAXL	221
C		STFAXL	222
	DO 300 J=1,6	STFAXL	223
	DO 290 I=1,4	STFAXL	224
	EPSLA(I,J,K) = B(I,J)	STFAXL	225
	STRSA(I,J,K) = PROD(I,J)	STFAXL	226
290	CONTINUE	STFAXL	227
300	CONTINUE	STFAXL	228
C		STFAXL	229
C	DEFINE LOAD VECTOR INNER PRODUCT XLOADA	STFAXL	230
C		STFAXL	231
	DO 330 J=1,4	STFAXL	232
	DO 320 I=1,6	STFAXL	233
	SUM = 0.0	STFAXL	234
	DO 310 II=1,4	STFAXL	235
	SUM = SUM+BT(I,II)*C(II,J)	STFAXL	236
310	CONTINUE	STFAXL	237
	XLOADA(I,J,K) = SUM	STFAXL	238
320	CONTINUE	STFAXL	239
330	CONTINUE	STFAXL	240
C		STFAXL	241
C	COMPUTE THE AREA	STFAXL	242
C		STFAXL	243
	ARAA(K) = AREA	STFAXL	244
C		STFAXL	245
C	COMPUTE MEAN RADIUS	STFAXL	246
C		STFAXL	247
	RBRA(K) = RBAR	STFAXL	248
	DO 350 J=1,4	STFAXL	249
	DO 340 I=1,4	STFAXL	250
	CA(I,J,K) = C(I,J)	STFAXL	251
340	CONTINUE	STFAXL	252
350	CONTINUE	STFAXL	253
360	CONTINUE	STFAXL	254
C		STFAXL	255

	DO 370 I=1,NARF	STFAXL 256
	II = I	STFAXL 257
	IACON(II,NEW) = 0	STFAXL 258
	III = II+1	STFAXL 259
	IACON(III,NEW) = 0	STFAXL 260
370	CONTINUE	STFAXL 261
	DO 380 I=1,NARF	STFAXL 262
	IF (KAN(I).EQ.0) GO TO 380	STFAXL 263
	IACON(I,NEW) = 1	STFAXL 264
	IF (ILK.GT.0) IACON(I,NEW) = 0	STFAXL 265
	IACON(I+1,NEW) = 1	STFAXL 266
	IF (ILK.GT.0) IACON(I+1,NEW) = 0	STFAXL 267
380	CONTINUE	STFAXL 268
	DO 450 I=1,NABF	STFAXL 269
	IF (IACON(I,NEW).EQ.0) GO TO 450	STFAXL 270
	JJ = I+NABF	STFAXL 271
	N1 = NADET(JJ)+1	STFAXL 272
	J = 2*NABF+I	STFAXL 273
	N2 = NADET(J)+1	STFAXL 274
	ISAV1 = MCUD*(N1-1)+1	STFAXL 275
	IF (N1.GT.KRNCH) ISAV1 = ISAV1-LKNT(N1-KRNCH)	STFAXL 276
	ISAV2 = MCUD*(N2-1)+1	STFAXL 277
	IF (N2.GT.KRNCH) ISAV2 = ISAV2-LKNT(N2-KRNCH)	STFAXL 278
	SAV1 = AK(ISAV1)	STFAXL 279
	SAV2 = AK(ISAV2)	STFAXL 280
C	INTRODUCE CONSTRAINTS ON CLADDING	STFAXL 281
	MAX = N2-N1-1	STFAXL 282
	DO 390 KM=1,MAX	STFAXL 283
	IJ = (N1+KM-1)*MCUD+N2-N1+1-KM	STFAXL 284
	IM = (N1-1)*MCUD+KM+1	STFAXL 285
	NM = N1-KRNCH	STFAXL 286
	IF (NM.GT.0) IM = IM-LKNT(NM)	STFAXL 287
	NJ = N2-KRNCH	STFAXL 288
	II = N1+KM	STFAXL 289
	IF (II.GT.KRNCH) IJ = IJ-LKNT(II-KRNCH)	STFAXL 290
	AK(IM) = AK(IM)+AK(IJ)	STFAXL 291
390	CONTINUE	STFAXL 292
	IN = MCUD*(N2-N1)-1	STFAXL 293
	II = IN+N2	STFAXL 294
	IF (II.GT.NAXB4) IN = IN-(II-NAXB4)	STFAXL 295
	DO 400 KM=1,IN	STFAXL 296
	IM = (N1-1)*MCUD+N2-N1+1+KM	STFAXL 297
	IJ = (N2-1)*MCUD+1+KM	STFAXL 298
	NM = N1-KRNCH	STFAXL 299
	IF (NM.GT.0) IM = IM-LKNT(NM)	STFAXL 300
	NJ = N2-KRNCH	STFAXL 301
	IF (NJ.GT.0) IJ = IJ-LKNT(NJ)	STFAXL 302
	AK(IM) = AK(IM)+AK(IJ)	STFAXL 303
400	CONTINUE	STFAXL 304
C		STFAXL 305
C	NOW INTERCHANGE COL,S	STFAXL 306
	INN = N1-1	STFAXL 307
	IN = MCUD*(N2-N1)-1	STFAXL 308
	IF (INN.LT.IN) IN = INN	STFAXL 309
	IF (IN.LE.0) GO TO 420	STFAXL 310
C		STFAXL 311
	DO 410 KM=1,IN	STFAXL 312

SUBROUTINE STFAXL

74/74 OPT=1

FIN 4.6+452

15/02/78 17.25.31

```

      II = N1-KM
      IM = MCUD*(II-1)+1+KM
      IF (II.GT.KRNCH) IM = IM-LKNT(II-KRNCH)
      IJ = IM+(N2-N1)
      AK(IM) = AK(IM)+AK(IJ)
410    CONTINUE
420  CONTINUE
      C
      C
      C      NOW ZERO OUT TERMS ROW
      MM = MCUD
      IF (N2.GE.KRNCH) MM = MCUD-(N2-KRNCH)-1
      DO 430 KM=1,MM
      IM = (N2-1)*MCUD+KM
      NM = N2-KRNCH
      IF (NM.GT.0) IM = IM-LKNT(NM)
      AK(IM) = 0.0
430    CONTINUE
      C      ZERO OUT COL
      MA = MCUD-1
      IF (N2.LE.MA) MA = N2-1
      DO 440 KM=1,MA
      II = N2-KM
      IM = MCUD*(II-1)+N2-II+1
      IF (II.GT.KRNCH) IM = IM-LKNT(II-KRNCH)
      AK(IM) = 0.0
440    CONTINUE
      C      NOW PUT A 1.0 ON DIAGONAL
      AK(ISAV2) = 1.0
      AK(ISAV1) = SAV1+SAV2
450    CONTINUE
      RETURN
      END

```

```

STFAXL 313
STFAXL 314
STFAXL 315
STFAXL 316
STFAXL 317
STFAXL 318
STFAXL 319
STFAXL 320
STFAXL 321
STFAXL 322
STFAXL 323
STFAXL 324
STFAXL 325
STFAXL 326
STFAXL 327
STFAXL 328
STFAXL 329
STFAXL 330
STFAXL 331
STFAXL 332
STFAXL 333
STFAXL 334
STFAXL 335
STFAXL 336
STFAXL 337
STFAXL 338
STFAXL 339
STFAXL 340
STFAXL 341
STFAXL 342
STFAXL 343
STFAXL 344
STFAXL 345

```

```

SUBROUTINE STIFF (ELM,R,Z,NODE,XK,XKC,NODEC,RD,NFFID,NODET,XKT,EPS STIFF 2
+LM,STRESS,ARE,RBR,XLOAD,XKSAVE,CC,KAN,IDU,NODEL,TELM,NABFD) STIFF 3
C STIFF 4
C PURPOSE OF THIS SUBROUTINE IS TO CALCULATE THE STIFFNESS MATRICES STIFF 5
C FOR FUEL, CLAD, AND COMBINED SYSTEM. (XK,XKC,XKT,RESP,) STIFF 6
C STIFF 7
C STIFF 8
C DIMENSION NODE(1),XK(1),R(1),Z(1),LKNT(6),XKC(1),ELM(3,1),N STIFF 9
+NODEC(1), NODET(1), XKT(1), TELM(NODEL,NABFD,2), EPBLM(4,6,1), STRE STIFF 10
+SS(4,6,1) STIFF 11
C DIMENSION ARE(1), RBR(1), XLOAD(6,4,1), XKSAVE(6,6,1), CC(4,4,1) STIFF 12
DIMENSION KAN(1), IDU(1) STIFF 13
C DIMENSION RD(NFFID,1) STIFF 14
C STIFF 15
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IPELTM,IELSTN,IELSTR PELCOM 2
+,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD PELCOM 3
+ET, IR,IRBR, ISA,ISAC,ISAF,ISB,ISBB,ISBBC,I PELCOM 4
+SBBF,ISBC,ISBF,ISK,ISKC,ISTRES, ITDELE, IXK,IXKC,IXKT PELCOM 5
+,IXLOAD, ICC,ITHET,ICA,ITHYAX, INADET,IRA,IY,IAELM,I PELCOM 6
+ARAA,IBA,IBBA,IRBRA,IXLDA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I PELCOM 7
+PBLA,IAC,IACN,IKAN,IRZ,IAAX,IRC,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM 8
+IIDO,ITME,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM 9
+, IXKCC,IXKIT PELCOM 10
+,ITFB, ITCB, IBRSV,IESTSV,IESTNV PELCOM 11
+.ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDS,IPSSV,IPSSRT,JIIJ,IIPUS PELCOM 12
+,IBASE,ITFR,IYM,IZ PELCOM 13
+,IDSPSV PELCOM 14
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELR,DELZ,ZOEL,KKA,PIN,POUT,P PELCOM 15
+DEL PELCOM 16
COMMON /LIMITS/ NF,NC,NODEL,NODELF,NODELC,ICN,IDF,NFF,MAXDC,NETOT,NE PELCOM 17
+WF,NEWT,KA,NARF,NARC,NART,NABF,NABC,NABT,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM 18
+NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CR,FROEN,KZ1,ISSTOP,IPLAS,NKN PELCOM 19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM 20
+MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD PELCOM 21
+, KPLAB,ICNT,RRAKR,IRAK,RAKSUM,IPLEND,ISP PELCOM 22
COMMON /LASTIC/ E,XNU,ALPHA PELCOM 23
COMMON /CONTNT/ PI PELCOM 24
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM 25
COMMON /CRP/ PSP(25),TPLASK PELCOM 26
COMMON /STFMAT/ B(4,6),C(4,6),BT(6,4),PRUD(4,6),XKBAR(6,6) STIFF 17
C STIFF 18
COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN COMC 2
+, NCNN,ISTRT,ITER COMC 3
+, FLXR COMC 4
COMMON/FEP/HMF,NT,KKZZZ,ISTOP,RDOLD(20),RRSIN(20),ICNOLD(20) COMC 5
+, IDOLD(20) COMC 6
+,PINOLD,POUTLD,POLD,NTOLD(20) COMC 7
+, ICHNG,CRPAD(20) COMC 8
+,STEPS COMC 9
C STIFF 20
PI = 3.14159 STIFF 21
IF (JAM.NE.1) GO TO 110 STIFF 22
DO 100 J=1,NFF1 STIFF 23
Z(J) = 0.0 STIFF 24
Z(J+NFF1) = DELZ STIFF 25
R(J) = (RD(J,KA)+RD(J,KA+1))/2. STIFF 26
R(J+NFF1) = R(J) STIFF 27

```


100	CONTINUE	STIFF	28
110	CONTINUE	STIFF	29
	LKNT(1) = 1	STIFF	30
	LKNT(2) = 3	STIFF	31
	LKNT(3) = 6	STIFF	32
	LKNT(4) = 10	STIFF	33
	LKNT(5) = 15	STIFF	34
	LKNT(6) = 21	STIFF	35
	KRNCHF = NF4-6	STIFF	36
	KRNCHC = NC4-6	STIFF	37
	NMX = 4+NF1-4	STIFF	38
	KRNCHT = NMX-6	STIFF	39
	NF4 = NF*4	STIFF	40
C		STIFF	41
C	FORM STIFFNESS MATRIX ON ELEMENT BY ELEMENT BASIS AND MERGE INTO	STIFF	42
C	TOTAL STIFFNESS MATRIX. STRESS MATRIX WILL ALSO BE FORMED	STIFF	43
C		STIFF	44
	CALL MOVEKA (0,STRESS,24*NOEL)	STIFF	45
	CALL MOVEKA (0,XKT,MAXT)	STIFF	46
	CALL MOVEKA (0,XKC,MAXC)	STIFF	47
	CALL MOVEKA (0,XK,MAXF)	STIFF	48
C		STIFF	49
	DO 460 KI=1,NOEL	STIFF	50
	N1 = ELM(1,KI)	STIFF	51
	N2 = ELM(2,KI)	STIFF	52
	N3 = ELM(3,KI)	STIFF	53
	R1 = R(N1)	STIFF	54
	R2 = R(N2)	STIFF	55
	R3 = R(N3)	STIFF	56
	Z1 = Z(N1)	STIFF	57
	Z2 = Z(N2)	STIFF	58
	Z3 = Z(N3)	STIFF	59
C	FORM MEAN VALUE LOCATION COORDINATES (CENTROID OF EACH ELEMENT)	STIFF	60
	RBAR = (R1+R2+R3)/3.0	STIFF	61
	AREA = (R2*Z3+R1*Z2+R3*Z1-R2*Z1-R3*Z2-R1*Z3)/2.0	STIFF	62
C	COMPUTE AREA COORDINATE AND COEFFICIENT	STIFF	63
	B1 = Z2-Z3	STIFF	64
	B2 = Z3-Z1	STIFF	65
	B3 = Z1-Z2	STIFF	66
	A1 = R3-R2	STIFF	67
	A2 = R1-R3	STIFF	68
	A3 = R2-R1	STIFF	69
	XL1 = 1.0/3.0	STIFF	70
	XL2 = 1.0/3.0	STIFF	71
	XL3 = 1.0/3.0	STIFF	72
C		STIFF	73
C	COMPUTE THE B MATRIX	STIFF	74
C		STIFF	75
	B(1,1) = B1	STIFF	76
	B(2,1) = 2.0*AREA*XL1/RBAR	STIFF	77
	B(3,1) = 0.0	STIFF	78
	B(4,1) = A1	STIFF	79
	B(1,2) = 0.0	STIFF	80
	B(2,2) = 0.0	STIFF	81
	B(3,2) = A1	STIFF	82
	B(4,2) = B1	STIFF	83
	B(1,3) = B2	STIFF	84

B(2,3) = 2.0*AREA*XL2/RBAR	STIFF	85
B(3,3) = 0.0	STIFF	86
B(4,3) = A2	STIFF	87
B(1,4) = 0.0	STIFF	88
B(2,4) = 0.0	STIFF	89
B(3,4) = A2	STIFF	90
B(4,4) = B2	STIFF	91
B(1,5) = B3	STIFF	92
B(2,5) = 2.0*AREA*XL3/RBAR	STIFF	93
B(3,5) = 0.0	STIFF	94
B(4,5) = A3	STIFF	95
B(1,6) = 0.0	STIFF	96
B(2,6) = 0.0	STIFF	97
B(3,6) = A3	STIFF	98
B(4,6) = B3	STIFF	99
DO 130 J=1,6	STIFF	100
DO 120 I=1,4	STIFF	101
B(I,J) = B(I,J)/(2.0*AREA)	STIFF	102
CONTINUE	STIFF	103
120 CONTINUE	STIFF	104
130	STIFF	105
C	STIFF	106
C	STIFF	107
C	STIFF	108
TF = TELM(KI,KA,NEW)	STIFF	109
IF (KI,GT,NOELF) GO TO 140	STIFF	110
E = TERP(TF,CF,CF(1,4),NCNN)	STIFF	111
XNU = TERP(TF,CF,CF(1,5),NCNN)	STIFF	112
GO TO 150	STIFF	113
C	STIFF	114
140 CONTINUE	STIFF	115
E = TERP(TF,AA,AA(1,4),NN)	STIFF	116
XNU = TERP(TF,AA,AA(1,5),NN)	STIFF	117
150 CONTINUE	STIFF	118
COEF = E/((1.+XNU)*(1.-2.*XNU))	STIFF	119
DO 170 J=1,3	STIFF	120
C(4,J) = 0.0	STIFF	121
C(J,4) = 0.0	STIFF	122
DO 160 I=1,3	STIFF	123
C(I,J) = XNU*COEF	STIFF	124
IF (I.EQ.J) C(I,J) = (1.0-XNU)*COEF	STIFF	125
160 CONTINUE	STIFF	126
170 CONTINUE	STIFF	127
C(4,4) = ((1.0-2.0*XNU)/2.0)*COEF	STIFF	128
C	STIFF	129
C	STIFF	130
C	STIFF	131
DO 190 J=1,4	STIFF	132
DO 180 I=1,6	STIFF	133
BT(I,J) = B(J,I)	STIFF	134
180 CONTINUE	STIFF	135
190 CONTINUE	STIFF	136
C	STIFF	137
C	STIFF	138
C	STIFF	139
C	STIFF	140
C	STIFF	141
DO 220 JJ=1,6		

	DO 210 III=1,4	STIFF	142
	SUM = 0.0	STIFF	143
	DO 200 JJJ=1,4	STIFF	144
	S = C(III,JJJ)*B(JJJ,JJ)	STIFF	145
	SUM = SUM+S	STIFF	146
200	CONTINUE	STIFF	147
	PROD(III,JJ) = SUM	STIFF	148
210	CONTINUE	STIFF	149
220	CONTINUE	STIFF	150
C		STIFF	151
C	NOW FORM FINAL PRODUCT	STIFF	152
C		STIFF	153
	DO 250 J=1,6	STIFF	154
	DO 240 I=1,6	STIFF	155
	SUM = 0.0	STIFF	156
	DO 230 II=1,4	STIFF	157
	S = BT(I,II)*PROD(II,J)	STIFF	158
	SUM = SUM+S	STIFF	159
230	CONTINUE	STIFF	160
	XKBAR(I,J) = SUM	STIFF	161
	XKBAR(I,J) = XKBAR(I,J)*RBAR*2.0*PI*AREA	STIFF	162
240	CONTINUE	STIFF	163
250	CONTINUE	STIFF	164
C		STIFF	165
C	BEFORE A GLOBAL STIFFNESS MATRIX WILL BE ASSEMBLED CHECK TO	STIFF	166
C	SEE IF CONTACT BETWEEN THE FUEL AND CLAD HAS OCCURRED	STIFF	167
	IF (ICON.EQ.1) GO TO 310	STIFF	168
	IF (KI.GT.NOELF) GO TO 280	STIFF	169
C		STIFF	170
C		STIFF	171
	K = 0	STIFF	172
	KK = 0	STIFF	173
	DO 270 I=1,6	STIFF	174
	IF (K.GE.2) K = 0	STIFF	175
	NNN = NODE(N1)	STIFF	176
	IF (I.GT.2) NNN = NODE(N2)	STIFF	177
	IF (I.GT.4) NNN = NODE(N3)	STIFF	178
	KK = 0	STIFF	179
	DO 260 J=1,6	STIFF	180
	IF (KK.GE.2) KK = 0	STIFF	181
	ND = NODE(N1)	STIFF	182
	IF (J.GT.2) ND = NODE(N2)	STIFF	183
	IF (J.GT.4) ND = NODE(N3)	STIFF	184
C		STIFF	185
C		STIFF	186
	II = NNN+K	STIFF	187
	JJ = ND+KK	STIFF	188
	KK = KK+1	STIFF	189
	IF (JJ.LT.II) GO TO 260	STIFF	190
	III = 8+(II-1)+JJ-II+1	STIFF	191
	IF (II.GT.KRNCHF) III = III-LKNT(II-KRNCHF)	STIFF	192
	XK(III) = XK(III)+XKBAR(I,J)	STIFF	193
260	CONTINUE	STIFF	194
	K = K+1	STIFF	195
270	CONTINUE	STIFF	196
C		STIFF	197
C	HAVE NOW COMPUTED AND MERGED THE LOCAL STIFFNESS MATRIX	STIFF	198

C	INTO GLOBAL	STIFF	199
C		STIFF	200
	IF (K1.LE.NOE1F) GO TO 340	STIFF	201
280	CONTINUE	STIFF	202
C		STIFF	203
C	NOW DEVELOP STIFFNESS FOR CLAD	STIFF	204
C		STIFF	205
	K = 0	STIFF	206
	KK = 0	STIFF	207
	DO 300 I=1,6	STIFF	208
	IF (K.GE.2) K = 0	STIFF	209
	NNN = NODEC(N1)	STIFF	210
	IF (I.GT.2) NNN = NODEC(N2)	STIFF	211
	IF (I.GT.4) NNN = NODEC(N3)	STIFF	212
	KK = 0	STIFF	213
	DO 290 J=1,6	STIFF	214
	IF (KK.GE.2) KK = 0	STIFF	215
	ND = NODEC(N1)	STIFF	216
	IF (J.GT.2) ND = NODEC(N2)	STIFF	217
	IF (J.GT.4) ND = NODEC(N3)	STIFF	218
C		STIFF	219
C	COMPUTE COORDINATES OF MASTER STIFFNESS MATRIX	STIFF	220
C		STIFF	221
	II = NNN+K	STIFF	222
	JJ = ND+KK	STIFF	223
	KK = KK+1	STIFF	224
	IF (JJ.LT.II) GO TO 290	STIFF	225
	III = 8*(II-1)+JJ-II+1	STIFF	226
	IF (II.GT.KRNCHC) III = III-LKNT(II-KRNCHC)	STIFF	227
	XKC(III) = XKC(III)+XKBAR(I,J)	STIFF	228
290	CONTINUE	STIFF	229
	K = K+1	STIFF	230
300	CONTINUE	STIFF	231
	GO TO 340	STIFF	232
310	CONTINUE	STIFF	233
C		STIFF	234
C	SET UP STIFFNESS FOR CONTACT CORRELATION BETWEEN THE FUEL AND	STIFF	235
C	CLAD	STIFF	236
C		STIFF	237
	K = 0	STIFF	238
	KK = 0	STIFF	239
	DO 330 I=1,6	STIFF	240
	IF (K.GE.2) K = 0	STIFF	241
	NNN = NODET(N1)	STIFF	242
	IF (I.GT.2) NNN = NODET(N2)	STIFF	243
	IF (I.GT.4) NNN = NODET(N3)	STIFF	244
	KK = 0	STIFF	245
	DO 320 J=1,6	STIFF	246
	IF (KK.GE.2) KK = 0	STIFF	247
	ND = NODET(N1)	STIFF	248
	IF (J.GT.2) ND = NODET(N2)	STIFF	249
	IF (J.GT.4) ND = NODET(N3)	STIFF	250
C		STIFF	251
C	COMPUTE COORDINATES FOR THE MASTER STIFFNESS MATRIX FOR THE TOTAL	STIFF	252
C		STIFF	253
	II = NNN+K	STIFF	254
	JJ = ND+KK	STIFF	255

	KK = KK+1	STIFF	256
	IF (JJ.LT.II) GO TO 320	STIFF	257
	III = 8*(II-1)+JJ-II+1	STIFF	258
	IF (II.GT.KRNCHT) III = III-LKNT(II-KRNCHT)	STIFF	259
	XKT(III) = XKT(III)+XKBAR(I,J)	STIFF	260
320	CONTINUE	STIFF	261
	K = K+1	STIFF	262
330	CONTINUE	STIFF	263
340	CONTINUE	STIFF	264
C	DEVELOPE STRESS AND STRAIN MATRIX USING PROD(I,J) AND B(I,J)	STIFF	265
C	WHERE STRESSES COME FROM S=PROD(I,J)*DISPLACEMENTS AND STRAINS	STIFF	266
C	ARE EPSOLON=PROD(I,J)	STIFF	267
C		STIFF	268
C	CREATE STRAIN MATRIX EPSLM(I,J,K) WHERE K IS THE VALUE OF	STIFF	269
C	THE ELEMENT NO.	STIFF	270
C		STIFF	271
C	CREATE STRESS MATRIX STRESS(I,J,K)	STIFF	272
C		STIFF	273
	DO 360 J=1,6	STIFF	274
	DO 350 I=1,4	STIFF	275
	EPSLM(I,J,KI) = B(I,J)	STIFF	276
	STRESS(I,J,KI) = PROD(I,J)	STIFF	277
350	CONTINUE	STIFF	278
360	CONTINUE	STIFF	279
C		STIFF	280
C	DEFINE THE ARRAY XLOAD(I,J,KI).	STIFF	281
C		STIFF	282
	DO 390 J=1,4	STIFF	283
	DO 380 I=1,6	STIFF	284
	SUM = 0.	STIFF	285
	DO 370 II=1,4	STIFF	286
	SUM = SUM+BT(I,II)*C(II,J)	STIFF	287
370	CONTINUE	STIFF	288
	XLOAD(I,J,KI) = SUM	STIFF	289
380	CONTINUE	STIFF	290
390	CONTINUE	STIFF	291
C		STIFF	292
C	COMPUTE AREA ARRAY FOR EACH ELEMENT.	STIFF	293
C		STIFF	294
	ARE(KI) = AREA	STIFF	295
C		STIFF	296
C	COMPUTE MEAN RADIUS ARRAY	STIFF	297
C		STIFF	298
	RBR(KI) = RBAR	STIFF	299
C		STIFF	300
	IF (KI.EQ.NOELF+1) GO TO 400	STIFF	301
	GO TO 430	STIFF	302
400	CONTINUE	STIFF	303
C		STIFF	304
C	SAVE THE STIFFNESS MATRIX FOR THE INTERFACIAL ELEMENT.	STIFF	305
C		STIFF	306
	DO 420 I=1,6	STIFF	307
	DO 410 J=1,6	STIFF	308
	XKSAVE(I,J,1) = XKBAR(I,J)	STIFF	309
410	CONTINUE	STIFF	310
420	CONTINUE	STIFF	311
C		STIFF	312

SUBROUTINE STIFF

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

430      CONTINUE
C      MATERIALS INFORMATION FOR THERMAL STRESS CORRECTION
          DO 450 J=1,4
          DO 440 I=1,4
            CC(I,J,KI) = C(I,J)
440      CONTINUE
450      CONTINUE
460      CONTINUE
C      RETURN
      END

```

```

STIFF      313
STIFF      314
STIFF      315
STIFF      316
STIFF      317
STIFF      318
STIFF      319
STIFF      320
STIFF      321
STIFF      322
STIFF      323

```

```

SUBROUTINE STRES (B,BC,BF,ELM,ELSTN,ELSTRS,EPSLM,NODE,NODEC,NODET, STRES      2
+CC,THETEO,STRESS,TSTNR,TSTSR,ELSTSA,ELSTNA,NOELD,DISPC,DISPF,KAN,I STRES      3
+DO,NFF10,ARAA,RBRA) STRES      4
C STRES      5
C THIS SUBROUTINE CALCULATES THE INCREMENTAL ELEMENT STRAINS (ELSTN) STRES      6
C AND STRESSES (ELSTRS) FROM THE NODAL DISPLACEMENTS (BB OR BC) STRES      7
C NOTE THAT THE STRESS EQUIVALENT OF THE ELEMENT THERMAL STRAINS STRES      8
C (THETEO) ARE SUBTRACTED OFF. THE INCREMENTAL STRESSES AND STRAINS STRES      9
C ARE ADDED TO THE TOTAL STRESS-STRAIN ARRAY-TSTSR, TSTNR. STRES     10
C STRES     11
C BEGIN LOOP FOR ADDING THE INCREMENTAL STRESSES AND STRAINS TO THE STRES     12
C TOTAL ARRAY STRES     13
COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IEPSLM,IELSTN,IELSTR PELCOM      2
+,IKSAVE,IPRESS,ITELM,ILAST,ILIMIT,INOC,INDF,INOT,INODE,INODEC,INOD PELCOM      3
+ET, IR,IRBR, ISA,ISAC,ISAF,ISR,ISRB,ISBRC,I PELCOM      4
+88BF,ISBC,ISBF,ISK,ISKC,ISTRES, IDELE, IXX,IXKC,IXKT PELCOM      5
+,IXLOAD, ICC,ITHET,ICA,ITHIAX, INADET,IRA,IY,IAELM,I PELCOM      6
+ARAA,IBA,IBRA,IRRA,IXLUDA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I PELCOM      7
+PSLA,IAK,IACN,IKAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTSR,ITSTSA, PELCOM      8
+IDDO,ITME,IEFSTR, ITCRP,ICRPG,ICRATE PELCOM      9
+, IXKCC,IXKTT PELCOM     10
+,ITFB, ITCB, IBBSV,IESTSV,IESTNV PELCOM     11
+,ITHSAV,ITHXSV,IBASAV,IPSTG,IULDSI,IPSSV,IPSSRT,JIIJ,IIPDS PELCOM     12
+,IBASE,ITFR,IYM,IZ PELCOM     13
+,IDSPSV PELCOM     14
COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELFR,DELZ,ZDEL,KKA,PIN,POUT,P PELCOM     15
+DEL PELCOM     16
COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IDF,NFF,MAXOC,NETQT,NE PELCOM     17
+WF,NEWT,KA,NARF,NARC,NART,NARF,NABC,NABT,NAEC,NAEF,NTAE,JAM,NAXBT, PELCOM     18
+NAXB2,NAXB4,NFF1,NABF2,NABT2, CP,CR,FRDEN,KZ1,ISSTOP,IPLAS,NKN PELCOM     19
+,ICRPP,MAXF,MAXC,MAXT,KRNCHF,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N PELCOM     20
+MX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD PELCOM     21
+, KPLAS,ICNT,RRAKR,IRAK,RAKSUM,IPLEND,ISP PELCOM     22
COMMON /LASTIC/ E,XNU,ALPHA PELCOM     23
COMMON /CONTNT/ PI PELCOM     24
COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2 PELCOM     25
COMMON /CRP/ PSP(25),TPLASK PELCOM     26
C STRES     15
C DIMENSION B(1), BC(1), BF(1), ELM(3,1), EPSLM(4,6,1), ELSTN(4,1), STRES     16
+ELSTRS(4,1), NODEC(1), NODET(1), STRESS(4,6,1), NODE(1), IDO(1), K STRES     17
+AN(1) STRES     18
C DIMENSION NN(3), CC(4,4,1), THETEO(4,1) STRES     19
C DIMENSION DISP(6), DISPC(NFF10,1), DISPF(NFF10,1) STRES     20
C DIMENSION TSTNR(4,NOELD,1), TSTSR(4,NOELD,1), ELSTSA(4,1), ELSTNA( STRES     21
+4,1) STRES     22
C DIMENSION ARAA(1), RBRA(1) STRES     23
C STRES     24
C DO 280 KI=1,NOEL STRES     25
C NN(1) = ELM(1,KI) STRES     26
C NN(2) = ELM(2,KI) STRES     27
C NN(3) = ELM(3,KI) STRES     28
C STRES     29
C SKIP OVER SEPERATE FUEL AND CLAD STRUCTURE IF CONTACT HAS STRES     30
C OCCURED STRES     31
C STRES     32
C IF (ICON.EQ.1) GO TO 130 STRES     33
C STRES     34

```

C	SKIP OVER FUEL PORTION WHEN KI IS GREATER THAN THE NUMBER OF	STRES	35
C	ELEMENTS IN THE FUEL	STRES	36
C		STRES	37
	IF (KI.LE.NOELF.AND.ICON.EQ.0) GO TO 200	STRES	38
	ISKIP = 0	STRES	39
	IF (ISKIP.EQ.0) GO TO 110	STRES	40
C	FUEL	STRES	41
	DO 100 I=1,3	STRES	42
	J = I*2-1	STRES	43
	NNI = NN(I)	STRES	44
	IN = NODE(NNI)	STRES	45
	DISP(J) = BF(IN)	STRES	46
	DISP(J+1) = BF(IN+1)	STRES	47
	IDN = NNI	STRES	48
	IF (NNI.GT.NFFID) IDN = NNI-NF-NC	STRES	49
	DISPC(IDN,KA) = BF(IN)	STRES	50
100	CONTINUE	STRES	51
	GO TO 160	STRES	52
C		STRES	53
110	CONTINUE	STRES	54
C	CLAD	STRES	55
	DO 120 I=1,3	STRES	56
	J = I*2-1	STRES	57
	NNI = NN(I)	STRES	58
	IN = NODE(NNI)	STRES	59
	DISP(J) = BC(IN)	STRES	60
	DISP(J+1) = BC(IN+1)	STRES	61
	IDN = NNI	STRES	62
	IF (NNI.GT.NFFID) IDN = NNI-NF-NC	STRES	63
	DISPC(IDN,KA) = BC(IN)	STRES	64
120	CONTINUE	STRES	65
	GO TO 160	STRES	66
C		STRES	67
C	COMBINED FUEL AND CLAD IN CONTACT	STRES	68
C		STRES	69
130	CONTINUE	STRES	70
	DO 150 I=1,3	STRES	71
	J = I*2-1	STRES	72
	NNI = NN(I)	STRES	73
	IN = NODE(NNI)	STRES	74
	DISP(J) = B(IN)	STRES	75
	DISP(J+1) = B(IN+1)	STRES	76
	IDN = NNI	STRES	77
	IF (NNI.GT.NFFID) IDN = NNI-NFFID	STRES	78
	IF (NKN.NE.2.AND.KI.LE.NOELF) GO TO 140	STRES	79
	DISPC(IDN,KA) = B(IN)	STRES	80
140	CONTINUE	STRES	81
150	CONTINUE	STRES	82
160	CONTINUE	STRES	83
C		STRES	84
C	USE ELEMENT NODAL DISPLACEMENTS TO OBTAIN ELEMENT STRAINS	STRES	85
C	AND STRESSES	STRES	86
C		STRES	87
	DO 190 I=1,4	STRES	88
	IF (ICON.EQ.0.AND.KI.LE.NOELF) GO TO 190	STRES	89
	S = 0.0	STRES	90
	SS = 0.0	STRES	91


```

      DO 170 J=1,6
C
C      COMPUTE STRAIN
C
      S = S+EPSLM(I,J,KI)*DISP(J)
C
C      CPMPUTE STRESSES
C
      SS = SS+STRESS(I,J,KI)*DISP(J)
170  CONTINUE
C
C      ELEMENT STRAIN
C      ELSTN(I,KI) = S
C
C      ELEMENT STRESS
C
      SIGU = 0.0
      DO 180 II=1,4
      SIGU = SIGU+CC(I,II,KI)*THETEU(II,KI)
180  CONTINUE
      ELSTRS(I,KI) = SS-SIGU
190  CONTINUE
200  CONTINUE
      IF (ISSTOP.NE.1.OR.KKA.EQ.1) GO TO 280
      LD = 1
      IMI = 4
      IF (KI.LE.NOELF.AND.ICON.EQ.0) LD = 3
      IF (KI.LE.NOELF.AND.ICON.EQ.0) IMI = 3
      DO 270 I=LD,IMI
      IF (IRZ.EQ.1.AND.NKN.EQ.1.AND.ISSTOP.EQ.1) GO TO 210
      GO TO 240
210  CONTINUE
      IF (I.EQ.3) GO TO 220
      TSTNR(I,KI,KA) = ELSTN(I,KI)
      TSTSR(I,KI,KA) = ELSTRS(I,KI)
      GO TO 270
220  MM = 4+KA
      IF (KI.GT.NOELF) MM = 4+KA+NAEF
C
C
C      MUST DEFINE A VOLUME WEIGHT FOR AXIAL STRESS TO MESH WITH TOTAL ST
      STS = 0.0
      STN = 0.0
      TA = ARAA(MM)+RBRA(MM)+ARAA(MM-1)*RBRA(MM-1)+ARAA(MM-2)*RBRA
      (MM-2)+ARAA(MM-3)*RBRA(MM-3)
      DO 230 II=1,4
      MK = MM-(II-1)
      WI = ARAA(MK)*RBRA(MK)/TA
      STS = STS+WI*ELSTSA(3,MK)
      STN = STN+WI*ELSTNA(3,MK)
230  CONTINUE
      TSTNR(3,KI,KA) = STN
      TSTSR(3,KI,KA) = STS
      ELSTN(3,KI) = STN
      ELSTRS(3,KI) = STS
      GO TO 270
240  IF (I.EQ.3) GO TO 250

```

```

      STRES 92
      STRES 93
      STRES 94
      STRES 95
      STRES 96
      STRES 97
      STRES 98
      STRES 99
      STRES 100
      STRES 101
      STRES 102
      STRES 103
      STRES 104
      STRES 105
      STRES 106
      STRES 107
      STRES 108
      STRES 109
      STRES 110
      STRES 111
      STRES 112
      STRES 113
      STRES 114
      STRES 115
      STRES 116
      STRES 117
      STRES 118
      STRES 119
      STRES 120
      STRES 121
      STRES 122
      STRES 123
      STRES 124
      STRES 125
      STRES 126
      STRES 127
      STRES 128
      STRES 129
      STRES 130
      STRES 131
      STRES 132
      STRES 133
      STRES 134
      STRES 135
      STRES 136
      STRES 137
      STRES 138
      STRES 139
      STRES 140
      STRES 141
      STRES 142
      STRES 143
      STRES 144
      STRES 145
      STRES 146
      STRES 147
      STRES 148

```

	TSTNR(I,KI,KA) = TSTNR(I,KI,KA)+ELSTN(I,KI)	STRES	149
	TSTSR(I,KI,KA) = TSTSR(I,KI,KA)+ELSTRS(I,KI)	STRES	150
	GO TO 270	STRES	151
250	MM = 4*KA	STRES	152
	IF (KI.GT.NOELF) MM = 4*KA+NAEP	STRES	153
	STS = 0.0	STRES	154
	STN = 0.0	STRES	155
	TA = ARAA(MM)*RBRA(MM)+ARAA(MM-1)*RBRA(MM-1)+ARAA(MM-2)*RBRA	STRES	156
	(MM-2)+ARAA(MM-3)*RBRA(MM-3)	STRES	157
	DO 260 I=1,4	STRES	158
	MK = MM-(I-1)	STRES	159
	WI = ARAA(MK)*RBRA(MK)/TA	STRES	160
	STS = STS+WI*ELSTSA(3,MK)	STRES	161
	STN = STN+WI*ELSTNA(3,MK)	STRES	162
260	CONTINUE	STRES	163
	TSTNR(3,KI,KA) = TSTNR(I,KI,KA)+STN	STRES	164
	ELSTRS(3,KI) = STS	STRES	165
	ELSTN(3,KI) = STN	STRES	166
	TSTSR(3,KI,KA) = TSTSR(I,KI,KA)+STS	STRES	167
270	CONTINUE	STRES	168
	IF (KA.EQ.NARC.AND.KI.LE.NOELF) TSTSR(3,KI,KA) = 0.0	STRES	169
280	CONTINUE	STRES	170
C	TIME CUTTING FOR CREEP CALCULATIONS	STRES	171
C	SKIP TIME CUTTING FOR ELASTIC AND PLASTIC CALCULATIONS.	STRES	172
C	TIME CUTTING FOR CREEP CALCULATIONS	STRES	173
C	SKIP TIME CUTTING FOR ELASTIC AND PLASTIC CALCULATIONS	STRES	174
	IF (KPLAS.GT.0.AND.KPLAS.LT.IPLEND) GO TO 320	STRES	175
	IF (KPLAS.EQ.IPLEND.AND.IPLAS.EQ.1) GO TO 320	STRES	176
	IF (KPLAS.EQ.IPLEND.AND.ISP.EQ.1) GO TO 320	STRES	177
	IF (ISSTOP.NE.1) GO TO 320	STRES	178
	IF (ICUT.EQ.1.AND.KKA.EQ.2) GO TO 290	STRES	179
	IF (NKN.NE.2.OR.KKA.NE.1) GO TO 310	STRES	180
	IF (ICUT.EQ.1) GO TO 310	STRES	181
290	CONTINUE	STRES	182
	IF (KA.EQ.1) BSCHK = 0.0	STRES	183
	IF (KA.EQ.1) BIG = 0.0	STRES	184
	LO = NOELF+1	STRES	185
	IF (ICON.EQ.1) LO = 1	STRES	186
	DO 300 I=LO,NOEL	STRES	187
	BIGCHK = ABS(TSTSR(2,I,KA))	STRES	188
	CHK = ABS(ELSTRS(2,I)/TSTSR(2,I,KA))	STRES	189
	IF (CHK.LT.BSCHK.OR.BIGCHK.LT.2000.) GO TO 300	STRES	190
	IF (BIGCHK.LT.5000..AND.I.LE.NOELF) GO TO 300	STRES	191
	BSCHK = CHK	STRES	192
	BIG = BIGCHK	STRES	193
	MAXI = I	STRES	194
	IF (BIG.LT.BIGCHK) BIG = BIGCHK	STRES	195
300	CONTINUE	STRES	196
	IF (KA.NE.NARC) GO TO 310	STRES	197
	IF (BIG.LT.2000) GO TO 320	STRES	198
	IF (ICUT.EQ.1) GO TO 310	STRES	199
	IF (BSCHK.LT.05) GO TO 310	STRES	200
	IADJ = BSCHK/0.05	STRES	201
	ICUT = 1	STRES	202
	IF (IADJ.GT.0) WRITE (6,330) KA,MAXI,IADJ	STRES	203
310	CONTINUE	STRES	204
	IF (ICUT.EQ.0) GO TO 320	STRES	205

SUBROUTINE STRES

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	IF (KKA.EQ.1.OR.NKN.EQ.1) GO TO 320	STRES	206
	IF (KA.NE.NARC) GO TO 320	STRES	207
	ILFT = IADJ-ITI	STRES	208
	IF (ILFT.LE.3) GO TO 320	STRES	209
	ZCEL = 0.05/BSCHK	STRES	210
	NWT = (TIMEF-TIMEC)/(STEP+ZCEL)	STRES	211
	IADJ = 1+NWT+ITI	STRES	212
	STEP = (TIMEF-TIMEC)/FLOAT(IADJ-ITI+1)	STRES	213
320	CONTINUE	STRES	214
	RETURN	STRES	215
C		STRES	216
C		STRES	217
330	FORMAT (5X,'CREEP CORRECTION IS TOO LARGE IN REGION',IS,'ELEMENT',	STRES	218
	+IS,'REQUIRED SUBSTEPS=',IS)	STRES	219
	END	STRES	220

SUBROUTINE SWELL

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	SUBROUTINE SWELL (TFR,RD,DBU,SDV,DELYB,KAN,ICHNG,NF,VOLUME,PI,FRDE	SWELL	2
	+N,PRCDH)	SWELL	3
C		SWELL	4
C	SWELL CALCULATES FUEL RADIAL EXPANSION DUE TO FISSION PRODUCT	SWELL	5
C	FORMATION.	SWELL	6
C		SWELL	7
	DIMENSION RD(NF),TFR(NF)	SWELL	8
	DATA SDOY,SDOY/0.016,0.004/	SWELL	9
	R1700 = TERP(3092.,TFR,RD,NF)	SWELL	10
	R1350 = TERP(2462.,TFR,RD,NF)	SWELL	11
	VC = (R1350-R1350-R1700-R1700)*PI	SWELL	12
	DVN1 = SDOY*VOLUME+FRDEN*DBU/1.E4	SWELL	13
	DVN2 = SDOY*VC*DBU/1.E4	SWELL	14
	DELYB = DVN1+DVN2	SWELL	15
	IF (ICHNG.EQ.0) GO TO 100	SWELL	16
	IF (KAN.EQ.0) GO TO 100	SWELL	17
	VL = PI*(RD(NF)*RD(NF)-R1350*R1350)	SWELL	18
	DISHV = 1.00*PRCDH/100.*VOLUME	SWELL	19
	VOIDV = 1.00*PI*RD(1)*RD(1)	SWELL	20
	VH = PI*R1700*R1700-VOIDV-DISHV	SWELL	21
	IF (VH.LT.0.) VH = 0.	SWELL	22
	ADR = (1.-FRDEN)*(0.8*VH+.5*VC+.3*VL)+0.0*DISHV+0.0*VOIDV	SWELL	23
	XXX = SDV+DELYB	SWELL	24
	TEST = XXX-ADR	SWELL	25
	IF (TEST.LE.0.0) DELYB = 0.	SWELL	26
100	CONTINUE	SWELL	27
	RETURN	SWELL	28
	END	SWELL	29

SUBROUTINE TCOR

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25:31

```

C      SUBROUTINE TCOR (F,TK,RK,DKDT,DENCR)
C
C      TCOR CALCULATES UO2 THERMAL CONDUCTIVITY, RK AND DIFFERENTIAL
C      CONDUCTIVITY, WITH RESPECT TO TEMPERATURES DKDT
C      BASED ON LYONS DATA.
C
      DENCR = 1.0789*(F/(1.0+(1.0-F)*0.5))
      TKK = TK+129.4
      RK = (38.24/TKK+6.1256E-13*TK**3)*DENCR
      DKDT = (-38.24/(TKK**2)+1.8377E-12*TK**2)*DENCR
      RETURN
      END

```

```

TCOR      2
TCOR      3
TCOR      4
TCOR      5
TCOR      6
TCOR      7
TCOR      8
TCOR      9
TCOR     10
TCOR     11
TCOR     12
TCOR     13

```

SUBROUTINE TCW1

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

      SUBROUTINE TCW1 (F,FRPUO2,TK,RK,DKDT,DENCR)
      C
      C TCW1 CALCULATES MIXED OXIDE FUEL THERMAL CONDUCTIVITY, RK AND
      C DIFFERENTIAL CONDUCTIVITY WITH RESPECT TO TEMPERATURE, DKDT
      C
      TKK = TK+129.4
      FRP = FRPUO2
      IF (FRPUO2.GT.0.05) PRP = 0.05
      PCOR = 1./(1.+FRP)
      DENCR = 1.0789*(F/(1.0+(1.0-F)*0.5))
      TKK = TK+129.4
      RK = (38.24/TKK+6.1256E-13*TK**3)+DENCR
      DKDT = (-38.24/(TKK**2)+1.8377E-12*TK**2)+DENCR
      RK = RK*PCOR
      DKDT = DKDT*PCOR
      RETURN
      END

```

```

TCW1      2
TCW1      3
TCW1      4
TCW1      5
TCW1      6
TCW1      7
TCW1      8
TCW1      9
TCW1     10
TCW1     11
TCW1     12
TCW1     13
TCW1     14
TCW1     15
TCW1     16
TCW1     17
TCW1     18

```

SUBROUTINE TEMPI

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	SUBROUTINE TEMPI (AI,DTEMP,POWER,TINLET,TLFB,TUFB,TLCB,TUCB,NF,NC,	TEMP1	2
	+NPOW,NABT)	TEMP1	3
C		TEMP1	4
C	TEMP1 SETS BOUNDARY FUEL ROD TEMPERATURES THAT NEED NOT BE INSIDE	TEMP1	5
C	ADDITIONAL LOOPS.	TEMP1	6
C		TEMP1	7
	DIMENSION AI(NABT), POWER(NABT), TKB(15), TLFB(NF), TUFB(NF), TLCB	TEMP1	8
	+ (NC), TUCB(NC)	TEMP1	9
	TKB(1) = TINLET	TEMP1	10
	TKB(NABT) = TINLET+DTEMP	TEMP1	11
	DO 100 J=1,NF	TEMP1	12
	TLFB(J) = TINLET	TEMP1	13
	TUFB(J) = TINLET+DTEMP	TEMP1	14
100	CONTINUE	TEMP1	15
	DO 110 J=1,NC	TEMP1	16
	TLCB(J) = TINLET	TEMP1	17
	TUCB(J) = TINLET+DTEMP	TEMP1	18
110	CONTINUE	TEMP1	19
	AI(1) = 0.0	TEMP1	20
	NPOW2 = NPOW+2	TEMP1	21
	AI(NABT) = AI(NPOW2)	TEMP1	22
	POWER(1) = 0.0	TEMP1	23
	POWER(NABT) = 0.0	TEMP1	24
	RETURN	TEMP1	25
	END	TEMP1	26

```

      SUBROUTINE TEMP2 (AI,HF,P,RD,TCB,TFB,DTEMP,ZZ,YY,WW,FRDEN,DT1,NF,N
      +C,NFPNC,TINLET,B1,NFUEL,FRPUO2,IPOW,RSIN,TSINC,FRSIN,BR,C,NCLAD)
C
C      TEMP2 CALCULATES THE LOCAL FUEL BOUNDARY RADIAL TEMPERATURE
C      PROFILE.
C
      COMMON /BA/ AA(23,7),CF(7,8),RKS,DELZA(20),NN,NCONN
      +, NCNN,ISTRT,ITER
      +, FLXR
      COMMON/FEP/HHF,NT,KKZZZ,ISTOP,RDOLD(20),RRSIN(20),ICNOLD(20)
      +, IDOOLD(20)
      +, PINOLD,POUTLD,POLD,NTOLD(20)
      +, ICHNG,CRPAD(20)
      +, STEPS
      DIMENSION RD(NFPNC), TCB(NC), TFB(NF), B1(6), B2(4),X(10), BR(6)
      TKB = TINLET+AI*DTEMP
      A = 3413.*12./3.14159
      B = A/2.*P
      QCO = 8/RD(NFPNC)
      TCB(NC) = TKB+QCO/HF
      TCS = TCB(NC)
      DIF = 2.*RD(1)
      DIA = 2.*RD(NF)
      DIC = 2.*RD(NF+1)
      DCO = 2.*RD(NF+NC)
      CALL DDLR (TCS,DIF,DIA,DIC,DCO,P,ZZ,YY,WW,FRDEN,B1,DT1,NFUEL,FRPUO
      +2,IPOW,RSIN,TSINC,FRSIN,B2,BR,C,NCLAD)
      DO 110 J=1,NF
        TFB(J) = 0.0
        X(1) = 1.
        X(2) = RD(J)/RD(NF+NC)
        X(3) = X(2)*X(2)
        X(4) = X(2)*X(3)
        X(5) = X(2)*X(4)
        X(6) = X(2)*X(5)
        DO 100 K=1,6
          BB = B1(K)
          IF (RSIN.GT.RD(1).AND.RD(J).LT.RSIN) BB = BR(K)
          TFB(J) = TFB(J)+BB*X(K)
        CONTINUE
      100   TFB(J) = TFB(J)+1.8+32.
      110   CONTINUE
      IF (ITER.GT.1.OR.KKZZZ.GT.2) GO TO 140
      DO 130 J=1,NC
        TCB(J) = 0.0
        X(7) = 1.
        X(8) = RD(J+NF)/RD(NF+NC)
        X(9) = X(8)*X(8)
        X(10) = X(9)*X(8)
        DO 120 K=7,10
          TCB(J) = TCB(J)+B2(K-6)*X(K)
        CONTINUE
      120   TCB(J) = TCB(J)+1.8+32.
      130   CONTINUE
      140   CONTINUE
      RETURN
      END

```



```

SUBROUTINE TEMP3 (NF,NC,NT,K,N,NABT,NFF1D,NOELD,NTAED,TELM,TREF,TI TEMP3      2
+NLET,TFB,TDELE,RD,TCB,TW,TWDEL,B,NART,NABF,RSIN,BR,KAN) TEMP3      3
DIMENSION TELM(NOELD,NABF,2), RD(NFF1D,1), TCB(NC,NAHT), TW(NTAED, TEMP3      4
+2), TWDEL(NTAED), TFB(NF,NABF), TDELE(NOELD,NABF), B(10,NABT) TEMP3      5
C TEMP3      6
C TEMP3 CALCULATES FUEL AND CLADDING ELEMENT TEMPERATURES FOR USE BY TEMP3      7
C PELET. TEMP3      8
C TEMP3      9
DIMENSION BR(6,NART),BB(6),KAN(NABF) TEMP3     10
RTEST = RSIN/RD(NF,1) TEMP3     11
T1 = TINLET TEMP3     12
NAB1 = NABT-1 TEMP3     13
NAB2 = NABT-2 TEMP3     14
NF1 = NF-1 TEMP3     15
NC1 = NC-1 TEMP3     16
IF (NT.NE.1) GO TO 130 TEMP3     17
DO 120 I=1,NART TEMP3     18
DO 100 M=1,NF1 TEMP3     19
J = 2*M-1 TEMP3     20
TELM(J,I,K) = TREF TEMP3     21
TELM(J+1,I,K) = TREF TEMP3     22
CONTINUE TEMP3     23
TELM(J,I,N) = TREF TEMP3     24
TELM(J+1,I,N) = TREF TEMP3     25
NO = 2*NF1 TEMP3     26
DO 110 M=1,NC1 TEMP3     27
J = 2*M-1+NO TEMP3     28
TELM(J,I,K) = TREF TEMP3     29
TELM(J+1,I,K) = TREF TEMP3     30
TELM(J,I,N) = TREF TEMP3     31
TELM(J+1,I,N) = TREF TEMP3     32
CONTINUE TEMP3     33
110 CONTINUE TEMP3     34
120 CONTINUE TEMP3     35
130 CONTINUE TEMP3     36
DO 170 I=1,NABF TEMP3     37
IF (I.EQ.NABF) GO TO 150 TEMP3     38
DO 140 M=1,NF1 TEMP3     39
J = 2*M-1 TEMP3     40
A1 = (TFB(M,I)+TFB(M,I+1))/2.0 TEMP3     41
A2 = (TFB(M+1,I+1)+TFB(M+1,I))/2.0 TEMP3     42
TELM(J,I,N) = (2.*A1+A2)/3.0 TEMP3     43
TELM(J+1,I,N) = (A1+2.*A2)/3.0 TEMP3     44
TDELE(J,I) = TELM(J,I,N)-TELM(J,I,K) TEMP3     45
TDELE(J+1,I) = TELM(J+1,I,N)-TELM(J+1,I,K) TEMP3     46
CONTINUE TEMP3     47
140 CONTINUE TEMP3     48
150 CONTINUE TEMP3     49
NO = 2*NF1 TEMP3     50
DO 160 M=1,NC1 TEMP3     51
A1 = (TCB(M,I)+TCB(M,I+1))/2.0 TEMP3     52
A2 = (TCB(M+1,I+1)+TCB(M+1,I))/2.0 TEMP3     53
J = 2*M-1+NO TEMP3     54
TELM(J,I,N) = (2.*A1+A2)/3.0 TEMP3     55
TELM(J+1,I,N) = (A1+2.*A2)/3.0 TEMP3     56
TDELE(J+1,I) = TELM(J+1,I,N)-TELM(J+1,I,K) TEMP3     57
CONTINUE TEMP3     58
160 CONTINUE
170 CONTINUE

```

SUBROUTINE TEMP3

74/74 OPT=1

FIN 4,6+452

15/02/78 17.25.31

```

      IF (NT.NE.1) GO TO 190
      DO 180 J=1,NTAED
        TW(J,K) = TREF
      CONTINUE
180   CONTINUE
190   DO 200 I=1,NAB1
        A1 = TCB(NC,I)
        A2 = TCB(NC,I+1)
        A3 = TCB(1,I+1)
        A4 = TCB(1,I)
        TAV = 0.25*(A1+A2+A3+A4)
        J = 1+(I-1)*4+NABM2
        TW(J,N) = (TAV+A4+A1)/3.
        TW(J+1,N) = (TAV+A1+A2)/3.
        TW(J+2,N) = (TAV+A2+A3)/3.
        TW(J+3,N) = (TAV+A3+A4)/3.
200   CONTINUE
        DO 220 I=1,NABM2
          F2 = (RD(NF,I)+RD(1,I))/2./RD(NF+NC,I)
          F3 = F2+F2
          F4 = F3+F2
          F5 = F4+F2
          J = I+1
          F6 = F5+F2
          DO 210 JK=1,6
            BB(JK) = B(JK,J)
            IF (RBIN.GT.RD(1,I).AND.RTEST.GT.5.) BB(JK) = BR(JK,J)
          CONTINUE
210   T2 = BB(1)+BB(2)+F2+BB(3)+F3+BB(4)+F4+BB(5)+F5+BB(6)+F6
          T2 = T2*1.8+32.
          TAV = (T1+T2)/2.
          T1 = T2
          L = 1+(I-1)*4
          A1 = TFB(NF,I)
          A2 = TFB(NF,I+1)
          A3 = TFB(1,I+1)
          A4 = TFB(1,I)
          TW(L,N) = (TAV+A4+A1)/3.
          TW(L+1,N) = (TAV+A1+A2)/3.
          TW(L+2,N) = (TAV+A2+A3)/3.
          TW(L+3,N) = (TAV+A3+A4)/3.
220   CONTINUE
        L = NTAED
        DO 230 I=1,L
          TWDEL(I) = TW(I,N)-TW(I,K)
        CONTINUE
230   CONTINUE
      RETURN
C
      END

```

```

TEMP3  59
TEMP3  60
TEMP3  61
TEMP3  62
TEMP3  63
TEMP3  64
TEMP3  65
TEMP3  66
TEMP3  67
TEMP3  68
TEMP3  69
TEMP3  70
TEMP3  71
TEMP3  72
TEMP3  73
TEMP3  74
TEMP3  75
TEMP3  76
TEMP3  77
TEMP3  78
TEMP3  79
TEMP3  80
TEMP3  81
TEMP3  82
TEMP3  83
TEMP3  84
TEMP3  85
TEMP3  86
TEMP3  87
TEMP3  88
TEMP3  89
TEMP3  90
TEMP3  91
TEMP3  92
TEMP3  93
TEMP3  94
TEMP3  95
TEMP3  96
TEMP3  97
TEMP3  98
TEMP3  99
TEMP3 100
TEMP3 101
TEMP3 102
TEMP3 103
TEMP3 104
TEMP3 105
TEMP3 106
TEMP3 107

```

FUNCTION TERP

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

      FUNCTION TERP (TT, TI, TD, N)
C      DIMENSION TI(1), TD(1)
C
      NO2 = N/2
      IF (TI(1).GT.TI(N)) GO TO 170
      IF (TT.LE.TI(1)) GO TO 130
      IF (TT.GE.TI(N)) GO TO 140
      IF (TT.GE.TI(NO2)) GO TO 100
      J1 = 1
      J2 = NO2
      GO TO 110
100    CONTINUE
      J1 = NO2
      J2 = N
110    CONTINUE
      DO 120 J=J1,J2
        IF (TT-TI(J)) 160,150,120
120    CONTINUE
      GO TO 140
130    TERP = TD(1)
      RETURN
140    TERP = TD(N)
      RETURN
150    TERP = TD(J)
      RETURN
160    CONTINUE
      TERP = TD(J-1)+(TD(J)-TD(J-1))*(TT-TI(J-1))/(TI(J)-TI(J-1))
      RETURN
170    IF (TT.GE.TI(1)) GO TO 130
      IF (TT.LE.TI(N)) GO TO 140
      IF (TT.LE.TI(NO2)) GO TO 180
      J1 = 1
      J2 = NO2
      GO TO 190
180    CONTINUE
      J1 = NO2
      J2 = N
190    CONTINUE
      DO 200 J=J1,J2
        IF (TT-TI(J)) 200,150,160
200    CONTINUE
      GO TO 140
      END

```

```

      TERP      2
      TERP      3
      TERP      4
      TERP      5
      TERP      6
      TERP      7
      TERP      8
      TERP      9
      TERP     10
      TERP     11
      TERP     12
      TERP     13
      TERP     14
      TERP     15
      TERP     16
      TERP     17
      TERP     18
      TERP     19
      TERP     20
      TERP     21
      TERP     22
      TERP     23
      TERP     24
      TERP     25
      TERP     26
      TERP     27
      TERP     28
      TERP     29
      TERP     30
      TERP     31
      TERP     32
      TERP     33
      TERP     34
      TERP     35
      TERP     36
      TERP     37
      TERP     38
      TERP     39
      TERP     40
      TERP     41
      TERP     42
      TERP     43
      TERP     44
      TERP     45

```

SUBROUTINE WARR

74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

	SUBROUTINE WARR (A,N,NN)	WARR	2
C		WARR	3
C	WARR IS USED TO PRINT TWO DIMENSION ARRAYS DIMENSIONED A(N,NN).	WARR	4
C		WARR	5
C	THE HEADING MUST BE PRINTED BY THE CALLING PROGRAM.	WARR	6
C		WARR	7
	DIMENSION A(N,NN)	WARR	8
C		WARR	9
	DO 100 I=1,NN	WARR	10
	WRITE (6,120) (A(J,I),J=1,N)	WARR	11
100	CONTINUE	WARR	12
	RETURN	WARR	13
	ENTRY WIARR	WARR	14
	DO 110 I=1,NN	WARR	15
	WRITE (6,130) (A(J,I),J=1,N)	WARR	16
110	CONTINUE	WARR	17
	RETURN	WARR	18
C		WARR	19
120	FORMAT (5X,10(1PE12.3))	WARR	20
130	FORMAT (5X,10(15,5X))	WARR	21
C		WARR	22
	END	WARR	23

```

      SUBROUTINE XFACE (ELSTRS,PRESS)
C
C      SUBROUTINE XFACE
C      PURPOSE OF THIS SUBROUTINE IS TO COMPUTE THE INTERFACIAL PRESSURE
C      BETWEEN THE FUEL CLADDING. THIS SUBROUTINE USES AN ELEMENT
C      STIFFNERS TO COMPUTE THE INTERFACIAL LOAD.
C
      COMMON /POINT/ IARE,IDISPC,IDISPF,IDISPT,IELM,IEPSLM,IELSTN,IELSTR
      *,IKSAVE,IPRESS,ITEM,ILAST,ILIMIT,INDC,INDF,INDT,INODE,INODEC,INOD
      *,ET,IR,IRBR,ISA,ISAC,ISAF,ISB,ISBB,ISBBC,I
      *,ISBF,ISBC,ISBF,ISK,ISKC,ISTRES,ITDELE,IXK,IXKC,IXKT
      *,IXLOAD,ICC,ITHET,ICA,ITHTAX,INADET,IRA,IY,IAELM,I
      *,ARAA,IRA,IRBA,IRBRA,IXLODA,ITW,ITWDEL,IDISP,ILSTRA,ILSTNA,ISTRSA,I
      *,PSLA,IAK,IACN,IKAN,IRZ,IAAX,IRD,IYACF,ITSTNR,ITSTNA,ITSTRR,ITSTSA,
      *,IIDD,ITHE,IEFSTR,ITCRP,ICRPG,ICRATE
      *,IXKCC,IXKTT
      *,ITFB,ITCB,IBBSV,IESTSV,IESTNV
      *,ITHSAV,ITHXSV,IBASAV,IPSTS,IOLDSE,IPSSV,IPSSRT,JIIJ,IIPDS
      *,IBASE,ITFR,IYM,IZ
      *,IOSPSV
      COMMON /GEOPAR/ RFI,RFO,RCI,RCO,TREF,DELR,DELZ,ZDEL,KKA,PIN,POUF,P
      *,OEL
      COMMON /LIMITS/ NF,NC,NOEL,NOELF,NOELC,ICON,IDF,NFF,MAXDC,NETOT,NE
      *,WF,NEWT,KA,NARF,NARC,NART,NARF,NABC,NABT,NAEC,NAEF,NTAE,JAM,NAXB1,
      *,NAXB2,NAXB4,NFF1,NABF2,NABT2,CP,CR,FRDEN,KZ1,ISSTOP,IPLAS,NKN
      *,ICRPP,MAXF,MAXC,MAXT,KRNCH,KRNCHC,KRNCHT,NFF2,NC4,NF4,MAXA,NEW,N
      *,KX,KRNCH,MCUD,IADJ,ICUT,TIMEC,TIMEF,ITI,STEP,LOCK(20,2),ILK,NOLD
      *,KPLAS,ICNT,RRAKR,IRAK,RAKSUM,IPLND,ISP
      COMMON /LASTIC/ E,XNU,ALPHA
      COMMON /CONTNT/ PI
      COMMON /TAPES/ NIN,NOUT,NSCR1,NSCR2,ITP1,ITP2
      COMMON /CRP/ PSP(25),TPLASK
C
      DIMENSION ELSTRS(4,1),PRESS(1)
      COMMON /PRSSV/ PRSINC(30),PRSS(30)
C
      IF (ICON.EQ.0) GO TO 110
      SUM = 0.0
      DO 100 I=1,4
        II = NOELF-2+I
        SUM = SUM+ELSTRS(1,II)
      100 CONTINUE
      IF (ISSTOP.EQ.1.AND.KKA.EQ.2) PRSS(KA) = PRSS(KA)+SUM/4.0
      PRSS(KA) = PRSS(KA)+SUM/4.0
      IF (ISSTOP.EQ.1.AND.KKA.EQ.2) PRESS(KA) = PRSS(KA)
      PRSINC(KA) = SUM/4.0
      GO TO 120
C
      110 CONTINUE
      PRESS(KA) = 0.0
      PRSINC(KA) = 0.0
      PRSS(KA) = 0.0
C
      120 CONTINUE
      RETURN
      END

```

SUBROUTINE HEADER 74/74 OPT=1

FTN 4.6+452

15/02/78 17.25.31

```

      SUBROUTINE HEADER
C
C      HEADER WRITE (THE BANNER PAGE.
C
      CALL DATIM (DATE,TIME)
C
C
      WRITE (6,100)
      CALL FHLPR (1H0,8HGAPCON-3,6)
      WRITE (6,110)
      CALL FHLPR (1H0,DATE,6)
C
      WRITE (6,110)
      CALL FHLPR (1H0,TIME,6)
      RETURN
C
C
C
C
C
C
100  FORMAT (1H1/)
110  FORMAT (1H0/1H0/)
      END

```

```

HEADER 2
HEADER 3
HEADER 4
HEADER 5
HEADER 6
HEADER 7
HEADER 8
HEADER 9
HEADER 10
HEADER 11
HEADER 12
HEADER 13
HEADER 14
HEADER 15
HEADER 16
HEADER 17
HEADER 18
HEADER 19
HEADER 20
HEADER 21
HEADER 22
HEADER 23
HEADER 24
HEADER 25

```

C	SUBROUTINE FHLPR (CC,A,NOU)	HEADING	2
C		HEADING	3
C	SUBROUTINE FHLPR IS USED TO CREATE A BANNER FORM 8 CHARACTER	HEADING	4
C	MESSAGE ON COMPUTER PRINTOUT. IT CAN BE USED TO GENERATE A TITLE	HEADING	5
C	PAGE.	HEADING	6
C		HEADING	7
C	FHLPR WAS CONVERTED TO UNIVAC 1108 BY T J TRAPP AND KENT STEWART	HEADING	8
C	BATTELLE NORTHWEST - MAY 1975.	HEADING	9
C		HEADING	10
C	COMMON /LETTER/ CHR(2,12,50)	HEADING	11
C	DIMENSION A(2), S(49), XMASK(10), NC(10)	HEADING	12
C	EQUIVALENCE (XCHAR,ICAR)	HEADING	13
C	DIMENSION IARR(49)	HEADING	14
C	EQUIVALENCE (S,IARR)	HEADING	15
C		HEADING	16
C	DATA S/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO	HEADING	17
C	+,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ,1H,1H+,1H-,1H/,1H0,1	HEADING	18
C	+H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H=,1H.,1H),1H*,1H,,1H(,1H#,1H+	HEADING	19
C	+,1H\$	HEADING	20
C	DATA XMASK/77000000000000000000B,77000000000000000000B,00007700000000	HEADING	21
C	+00000000B,00007700000000000000B,00000000770000000000B,0000000077000000	HEADING	22
C	+000B,0000000000000770000000B,0000000000007700000B,0000000000000000770	HEADING	23
C	+0B,000000000000000000077B/	HEADING	24
C	DATA XBLANK/55555555555555555555B/	HEADING	25
C	DATA IMULT/6/	HEADING	26
C		HEADING	27
C	J = 0	HEADING	28
C	IW = 1	HEADING	29
C	MULT = 0	HEADING	30
C	DO 120 IC=1,8	HEADING	31
C	XCHAR = XMASK(IC).AND.A(IW)	HEADING	32
C	ICAR = SHIFT(ICAR,MULT)	HEADING	33
C	XCHAR = XCHAR.OR.XBLANK	HEADING	34
C	MULT = MULT+IMULT	HEADING	35
C		HEADING	36
C	DO 100 L=1,49	HEADING	37
C	IF (XCHAR.EQ.S(L)) GO TO 110	HEADING	38
C	IF (ICAR.EQ.IARR(L)) GO TO 110	HEADING	39
100	CONTINUE	HEADING	40
C	L = 50	HEADING	41
110	CONTINUE	HEADING	42
C	J = J+1	HEADING	43
C	NC(J) = L	HEADING	44
120	CONTINUE	HEADING	45
C		HEADING	46
C	WRITE (NOU,130) CC,(((CHR(I,J,NC(K)),I=1,2),K=1,8),J=1,12)	HEADING	47
C	RETURN	HEADING	48
C		HEADING	49
C		HEADING	50
C		HEADING	51
C		HEADING	52
C		HEADING	53
C		HEADING	54
C		HEADING	55
130	FORMAT (A1,2X,16AB/(3X,16AB))	HEADING	56
C	END	HEADING	57

```

BLOCK DATA CHARS
COMMON /LETTER/ A(2,12),B(2,12),C(2,12),D(2,12),E(2,12),F(2,12),G(
+2,12),H(2,12),I(2,12),J(2,12),K(2,12),L(2,12),M(2,12),N(2,12),O(2,
+12),P(2,12),Q(2,12),R(2,12),S(2,12),T(2,12),U(2,12),V(2,12),W(2,12
+),X(2,12),Y(2,12),Z(2,12),BLANK(2,12),PLUS(2,12),MINUS(2,12),SLASH
+(2,12),ZERO(2,12),ONE(2,12),TWO(2,12),THREE(2,12),FOUR(2,12),FIVE(
+2,12),SIX(2,12),SEVEN(2,12),EIGHT(2,12),NINE(2,12),EQUAL(2,12),PER
+IOD(2,12),RPAREN(2,12),ASTRISK(2,12),COMMA(2,12),LPAREN(2,12),APOS(
+2,12),AMPSND(2,12),DOLLAR(2,12),ILLEGAL(2,12)
DATA A/8H AAAAAA,8HAAA ,8H AAAAAA,8HAAAA ,8HAA ,8H
+ AA ,8HAA ,8H AA ,8HAA ,8H AA ,8HAAAAAA,8HAA
+AAA ,8HAAAAAA,8HAAAA ,8HAA ,8H AA ,8HAA ,8H
+ AA ,8HAA ,8H AA ,8HAA ,8H AA ,8HAA ,8H
+ AA /
DATA B/8HBBBBBBBB,8HBBB ,8HBBBBBBBB,8HBBBBB ,8HBB ,8H
+ BB ,8HBB ,8H BB ,8HBB ,8H BB ,8HBBBBBBBB,8HBB
+BB ,8HBBBBBBBB,8HBBB ,8HBB ,8H BB ,8HBB ,8H
+ BB ,8HBB ,8H BB ,8HBBBBBBBB,8HBBBBB ,8HBB
+BB /
DATA C/8H CCCCCC,8HCCC ,8HCCCCCCCC,8HCCCCC ,8HCC ,8H
+ CC ,8HCC ,8H ,8HCC ,8H ,8HCC ,8H
+ ,8HCC ,8H ,8HCC ,8H CC ,8HCCCCCCCC,8HCCCCC ,8H CCCCCC,8HCC
+CC /
DATA D/8HDDDDDDDD,8HDDDD ,8HDDDDDDDD,8HDDDDDD ,8HDD ,8H
+ DD ,8HDD ,8H DD ,8HDD ,8H DD ,8HDD ,8H
+ DD ,8HDD ,8H DD ,8HDDDDDDDD,8HDDDDDD ,8HDDDDDDDD,8HDD
+DD /
DATA E/8HEEEEEEEE,8HEEEEE ,8HEEEEEEEE,8HEEEEE ,8HEE ,8H
+ ,8HEE ,8H ,8HEE ,8H ,8HEEEEEEEE,8HE
+ ,8HEE ,8H ,8HEEEEEEEE,8HEEEEE ,8HEEEEEEEE,8HEE
+EEE /
DATA F/8HFFFFFFFF,8HFFFF ,8HFFFFFFFF,8HFFFF ,8HFF ,8H
+ ,8HFF ,8H ,8HFF ,8H ,8HFFFFFFFF,8HF
+ ,8HFF ,8H ,8HFF ,8H ,8HFF ,8H
+ /
DATA G/8H GGGGGGG,8HG GGG ,8HG GGGGGGG,8HG GGGG ,8HGG ,8H
+ GG ,8HGG ,8H ,8HGG ,8H GG,8HG GGGG
+GG ,8HGG ,8H GG ,8HG GGGGGGG,8HG GGGG ,8H GGGGGGG,8HGG
+GG /
DATA H/8HHH ,8H HH ,8HHH ,8H HH ,8HHH ,8H
+ HH ,8HHH ,8H HH ,8HHH ,8H HH ,8HHHHHHHH,8HHH
+HHH ,8HHHHHHHH,8HHHHH ,8HHH ,8H HH ,8HHH ,8H
+ HH ,8HHH ,8H HH ,8HHH ,8H HH ,8HHH ,8H
+ HH /
DATA I/8HIIIIIIII,8HIIII ,8HIIIIIIII,8HIIII ,8H II ,8H
+ ,8H II ,8H ,8H II ,8H ,8H II ,8H
+ ,8H II ,8H ,8HIIIIIIII,8HIIII ,8HIIIIIIII,8HII
+II /
DATA J/8H ,8H JJ ,8H ,8H JJ ,8H ,8H
+ JJ ,8H ,8H JJ ,8H ,8H JJ ,8H ,8H
+ JJ ,8H ,8H JJ ,8H ,8H JJ ,8H ,8H

```


+ JJ	,8HJJ	,8H	JJ	,8HJJJJJJJJ,8HJJJJJJ	,8H JJJJJJJJ,8HJJ	HEADING	115	
+JJ	/					HEADING	116	
DATA K/8HKK	,8H	KK	,8HKK	,8H KK	,8HKK	,8H K	HEADING	117
+K	,8HKK	,8HKK	,8HKK	K,8HKK	,8HKKKKKKKK,8H		HEADING	118
+	,8HKKKKKKKK,8H		,8HKK	K,8HKK	,8HKK	,8HKK	HEADING	119
+	,8HKK	,8H KK	,8HKK	,8H KK	,8HKK	,8H	HEADING	120
+ KK	/						HEADING	121
DATA L/8HLL	,8H		,8HLL	,8H	,8HLL	,8H	HEADING	122
+	,8HLL	,8H	,8HLL	,8H	,8HLL	,8H	HEADING	123
+	,8HLL	,8H	,8HLL	,8H	,8HLL	,8H	HEADING	124
+	,8HLL	,8H	,8HLLLLLLLL,8HLLLL		,8HLLLLLLLL,8HLL		HEADING	125
+LLL	/						HEADING	126
DATA M/8HMM	,8H	MM	,8HMMM	,8H MMM	,8HMMM	,8H M	HEADING	127
+MM	,8HMM MM	,8HMM MM	,8HMM MM	M,8HMM MM	,8HMM	MMM,8H	HEADING	128
+ MM	,8HMM	M,8H MM	,8HMM	,8H MM	,8HMM	,8H	HEADING	129
+ MM	,8HMM	,8H MM	,8HMM	,8H MM	,8HMM	,8H	HEADING	130
+ MM	/						HEADING	131
DATA N/8HNN	,8H	NN	,8HNNN	,8H NN	,8HNNNN	,8H	HEADING	132
+ NN	,8HNN NN	,8H NN	,8HNN NN	,8H NN	,8HNN NN	,8H NN	HEADING	133
+ NN	,8HNN NN,8H NN		,8HNN	N,8H NN	,8HNN	,8HNN	HEADING	134
+ NN	,8HNN	,8H NNNN	,8HNN	,8H NNN	,8HNN	,8H	HEADING	135
+ NN	/						HEADING	136
DATA O/8H 0000000,8H0000			,8H00000000,8H000000		,8H00	,8H	HEADING	137
+ 00	,8H00	,8H 00	,8H00	,8H 00	,8H00	,8H	HEADING	138
+ 00	,8H00	,8H 00	,8H00	,8H 00	,8H00	,8H	HEADING	139
+ 00	,8H00	,8H 00	,8H00000000,8H000000		,8H 0000000,8H00		HEADING	140
+00	/						HEADING	141
DATA P/8HPPPPPPPP,8HPPPP			,8HPPPPPPPP,8HPPPPPP		,8HPP	,8H	HEADING	142
+ PP	,8HPP	,8H PP	,8HPP	,8H PP	,8HPPPPPPPP,8HPP		HEADING	143
+PPP	,8HPPPPPPPP,8HPPPP		,8HPP	,8H	,8HPP	,8H	HEADING	144
+	,8HPP	,8H	,8HPP	,8H	,8HPP	,8H	HEADING	145
+	/						HEADING	146
DATA Q/8H 00000,8H00			,8H 000000,8H0000		,8H 00	,8H	HEADING	147
+QQ	,8H00	,8H 00	,8H00	,8H 00	,8H00	,8H	HEADING	148
+ QQ	,8H00	,8H 00	,8H00	,8H 00	,8H00	,8H00	HEADING	149
+ QQ	,8H 00	,8H 0000	,8H 000000,8H0000		,8H 00000,8H00		HEADING	150
+ QQ	/						HEADING	151
DATA R/8HRRRRRRRR,8HRRRR			,8HRRRRRRRR,8HRRRRR		,8HRR	,8H	HEADING	152
+ RR	,8HRR	,8H RR	,8HRR	,8H RR	,8HRRRRRRRR,8HRR		HEADING	153
+RRR	,8HRRRRRRRR,8HRRRR		,8HRR	,8HRR	,8HRR	,8H R	HEADING	154
+ R	,8HRR	,8H RR	,8HRR	,8H RR	,8HRR	,8H	HEADING	155
+ RR	/						HEADING	156
DATA S/8H 8888888,8H8888			,8H88888888,8H888888		,8H88	,8H	HEADING	157
+ SS	,8H88	,8H	,8H88	,8H	,8H88888888,8H88		HEADING	158
+SS	,8H 8888888,8H888888		,8H	,8H SS	,8H	,8H	HEADING	159
+ SS	,8H88	,8H SS	,8H88888888,8H888888		,8H 8888888,8H88		HEADING	160
+SS	/						HEADING	161
DATA T/8HTTTTTTTTT,8HTTTT			,8HTTTTTTTTT,8HTTTT		,8H	TT,8H	HEADING	162
+	,8H	TT,8H	,8H	TT,8H	,8H	TT,8H	HEADING	163
+	,8H	TT,8H	,8H	TT,8H	,8H	TT,8H	HEADING	164
+	,8H	TT,8H	,8H	TT,8H	,8H	TT,8H	HEADING	165
+	/						HEADING	166
DATA U/8HUU	,8H	UU	,8HUU	,8H UU	,8HUU	,8H	HEADING	167
+ UU	,8HUU	,8H UU	,8HUU	,8H UU	,8HUU	,8H	HEADING	168
+ UU	,8HUU	,8H UU	,8HUU	,8H UU	,8HUU	,8H	HEADING	169
+ UU	,8HUU	,8H UU	,8HUUUUUUUU,8HUUUUUU		,8H UUUUUUU,8HUU		HEADING	170
+UU	/						HEADING	171

```

DATA V/8HVV ,8H VV ,8HVV ,8H VV ,8HVV ,8H HEADING 172
+ VV ,8HVV ,8H VV ,8HVV ,8H VV ,8HVV ,8H HEADING 173
+ VV ,8H VV ,8H VV ,8H VV ,8H VV ,8H VV ,8H HEADING 174
+ ,8H VV V ,8HVV ,8H VVV ,8H V ,8H HEADING 175
+ / HEADING 176
DATA W/8HWW ,8H WW ,8HWW ,8H WW ,8HWW ,8H HEADING 177
+ WW ,8HWW ,8H WW ,8HWW ,8H WW ,8HWW ,8H HEADING 178
+ WW ,8HWW WW ,8H WW ,8HWW WW ,8HWW WW ,8HWW HEADING 179
+ WW ,8HWWWW ,8H WWWW ,8HWWWW ,8H WWW ,8HWWWW ,8H HEADING 180
+ / HEADING 181
DATA X/8HXX ,8H XX ,8H XX ,8H XX ,8H XX ,8H X HEADING 182
+ X ,8H XX ,8HXX ,8H XX X ,8HXX ,8H XXX ,8H HEADING 183
+ ,8H XXX ,8H ,8H XX X ,8HXX ,8H XX ,8HXX HEADING 184
+ ,8H XX ,8H XX ,8H XX ,8H XX ,8H XX HEADING 185
+ XX / HEADING 186
DATA Y/8HYY ,8H YY ,8H YY ,8H YY ,8H YY ,8HYY HEADING 187
+ ,8H YY Y ,8HY ,8H YY ,8H YY ,8H YY ,8H HEADING 188
+ ,8H YY ,8H ,8H YY ,8H ,8H YY ,8H HEADING 189
+ ,8H YY ,8H ,8H YY ,8H ,8H YY ,8H HEADING 190
+ / HEADING 191
DATA Z/8HZZZZZZZ ,8HZZZZZ ,8HZZZZZZZZ ,8HZZZZZ ,8H ,8H Z HEADING 192
+ Z ,8H ,8HZZ ,8H Z ,8HZZ ,8H ZZ ,8H HEADING 193
+ ,8H ZZ ,8H ,8H ZZ ,8H ,8H ZZ ,8H HEADING 194
+ ,8H ZZ ,8H ZZZZZZZZ ,8HZZZZZ ,8HZZZZZZZZ ,8HZZ HEADING 195
+ ZZZ / HEADING 196
DATA BLANK/8H ,8H ,8H ,8H ,8H ,8H HEADING 197
+ 8H ,8H ,8H ,8H ,8H ,8H HEADING 198
+ 8H ,8H ,8H ,8H ,8H ,8H HEADING 199
+ 8H ,8H ,8H ,8H ,8H ,8H HEADING 200
+ 8H / HEADING 201
DATA PLUS/8H ++ ,8H ,8H ++ ,8H ,8H ++ ,8 HEADING 202
+ H ,8H ++ ,8H ,8H ++ ,8H ,8H ++ ,8 HEADING 203
+ H++++ ,8H++++ ,8H++++ ,8H ++ ,8H ,8H ++ ,8 HEADING 204
+ H ,8H ++ ,8H ,8H ++ ,8H ,8H ++ ,8 HEADING 205
+ / HEADING 206
DATA MINUS/8H ,8H ,8H ,8H ,8H ,8H HEADING 207
+ 8H ,8H ,8H ,8H ,8H ,8H HEADING 208
+ 8H----- ,8H----- ,8H----- ,8H ,8H HEADING 209
+ 8H ,8H ,8H ,8H ,8H ,8H HEADING 210
+ 8H / HEADING 211
DATA SLASH/8H ,8H // ,8H ,8H // ,8H ,8H // HEADING 212
+ 8H // ,8H ,8H// ,8H ,8H // ,8H ,8H // HEADING 213
+ 8H ,8H // ,8H ,8H // ,8H ,8H // HEADING 214
+ 8H ,8H // ,8H ,8H // ,8H ,8H // HEADING 215
+ 8H / HEADING 216
DATA ZERO/8H 00000 ,8H00 ,8H 000000 ,8H000 ,8H 00 ,8 HEADING 217
+ H 00 ,8H00 ,8H 00 ,8H00 ,8H 00 ,8H00 ,8 HEADING 218
+ H 00 ,8H00 ,8H 00 ,8H00 ,8H 00 ,8H00 ,8 HEADING 219
+ H 00 ,8H 00 ,8H 00 ,8H 000000 ,8H000 ,8H 00000 ,8 HEADING 220
+ H00 / HEADING 221
DATA ONE/8H 11 ,8H ,8H 111 ,8H ,8H 1111 ,8H HEADING 222
+ ,8H 11 ,8H ,8H 11 ,8H ,8H 11 ,8H HEADING 223
+ ,8H 11 ,8H ,8H 11 ,8H ,8H 11 ,8H HEADING 224
+ ,8H 11 ,8H ,8H 11111 ,8H11 ,8H 11111 ,8H HEADING 225
+ 11 / HEADING 226
DATA TWO/8H 222222 ,8H2222 ,8H222222 ,8H222222 ,8H22 ,8H HEADING 227
+ 22 ,8H ,8H 22 ,8H ,8H 22 ,8H ,8H HEADING 228

```

```

+ 22 ,8H ,8H22 ,8H 22,8H ,8H 22 ,8H HEADING 229
+ ,8H 22 ,8H ,8H22222222,8H22222 ,8H22222222,8H HEADING 230
+22222 / HEADING 231
DATA THREE/8H 333333,8H3333 ,8H3333333,8H33333 ,8H33 , HEADING 232
+8H 33 ,8H ,8H 33 ,8H ,8H 33 ,8H , HEADING 233
+8H 333 ,8H ,8H 333 ,8H ,8H 33 ,8H , HEADING 234
+8H 33 ,8H33 ,8H 33 ,8H3333333,8H33333 ,8H 3333333, HEADING 235
+8H3333 / HEADING 236
DATA FOUR/8H ,8H44 ,8H ,8H44 ,8H 44,8H44 ,8H 44,8 HEADING 237
+8H44 ,8H 44 ,8H44 ,8H 44 ,8H44 ,8H 44,8 HEADING 238
+8H44 ,8H 44 ,8H44 ,8H 4444444,8H44444 ,8H4444444,8 HEADING 239
+8H4444 ,8H ,8H44 ,8H ,8H44 ,8H , HEADING 240
+8H44 / HEADING 241
DATA FIVE/8H5555555,8H55555 ,8H5555555,8H55555 ,8H55 ,8 HEADING 242
+8H ,8H55 ,8H ,8H55 ,8H ,8H5555555,8 HEADING 243
+8H5555 ,8H55555555,8H555555 ,8H ,8H 55 ,8H ,8 HEADING 244
+8H 55 ,8H55 ,8H 55 ,8H55555555,8H555555 ,8H 5555555,8 HEADING 245
+8H5555 / HEADING 246
DATA SIX/8H 6666666,8H66666 ,8H66666666,8H66666 ,8H66 ,8H HEADING 247
+ ,8H66 ,8H ,8H66 ,8H ,8H66666666,8H HEADING 248
+6666 ,8H66666666,8H66666 ,8H66 ,8H 66 ,8H66 ,8H HEADING 249
+66 ,8H66 ,8H 66 ,8H66666666,8H66666 ,8H 6666666,8H HEADING 250
+6666 / HEADING 251
DATA SEVEN/8H7777777,8H77777 ,8H7777777,8H7777 ,8H77 , HEADING 252
+8H 77 ,8H ,8H77 ,8H ,8H 77,8H7 ,8H HEADING 253
+8H ,8H 77,8H ,8H 77 ,8H ,8H 77 , HEADING 254
+8H ,8H 77 ,8H ,8H 77 ,8H ,8H 77 , HEADING 255
+8H / HEADING 256
DATA EIGHT/8H 8888888,8H88888 ,8H88888888,8H88888 ,8H88 , HEADING 257
+8H 88 ,8H88 ,8H 88 ,8H88 ,8H 88 ,8H 8888888, HEADING 258
+8H8888 ,8H 8888888,8H88888 ,8H88 ,8H 88 ,8H88 , HEADING 259
+8H 88 ,8H88 ,8H 88 ,8H88888888,8H88888 ,8H 8888888, HEADING 260
+8H8888 / HEADING 261
DATA NINE/8H 9999999,8H99999 ,8H99999999,8H99999 ,8H99 ,8 HEADING 262
+8H 99 ,8H99 ,8H 99 ,8H99 ,8H 99 ,8H99999999,8 HEADING 263
+8H9999 ,8H 9999999,8H99999 ,8H ,8H 99 ,8H HEADING 264
+8H 99 ,8H ,8H 99 ,8H99999999,8H99999 ,8H99999999,8 HEADING 265
+8H9999 / HEADING 266
DATA EQUAL/8H ,8H ,8H ,8H ,8H , HEADING 267
+8H ,8H=====,8H===== ,8H=====,8H===== ,8H , HEADING 268
+8H ,8H ,8H ,8H=====,8H===== ,8H=====, HEADING 269
+8H===== ,8H ,8H ,8H ,8H , HEADING 270
+8H / HEADING 271
DATA PERIOD/8H ,8H ,8H ,8H ,8H HEADING 272
+8H ,8H ,8H ,8H ,8H HEADING 273
+8H ,8H ,8H ,8H ,8H HEADING 274
+8H ,8H ,8H ,8H ,8H HEADING 275
+8H / HEADING 276
DATA RPAREN/8H)) ,8H ,8H )) ,8H ,8H )) HEADING 277
+8H ,8H )) ,8H ,8H )) ,8H ,8H )) HEADING 278
+8H ,8H )) ,8H ,8H )) ,8H ,8H )) HEADING 279
+8H ,8H )) ,8H ,8H )) ,8H ,8H )) HEADING 280
+8H / HEADING 281
DATA ASTRK/8H ,8H ,8H ,8H ,8H HEADING 282
+8H ** ,8H ** ** ,8H** ,8H***** ,8H** HEADING 283
+8H*** ,8H***** ,8H*** ,8H ** ** ,8H** HEADING 284
+8H** ,8H ** ** ,8H** ,8H** HEADING 285

```

```

+,BH / HEADING 286
DATA COMMA/BH ,BH ,BH ,BH ,BH ,BH ,BH HEADING 287
+,BH ,BH ,BH ,BH ,BH ,BH ,BH HEADING 288
+,BH ,BH ,,,,BH, ,BH ,,,,BH, ,BH ,,,, HEADING 289
+,BH, ,BH ,,,,BH, ,BH ,,,,BH, ,BH ,,,, HEADING 290
+,BH / HEADING 291
DATA LPAREN/BH ,BH (( ,BH ,BH (( ,BH ,BH (( HEADING 292
+,BH (( ,BH ,BH(( ,BH ,BH(( ,BH ,BH(( HEADING 293
+,BH(( ,BH ,BH(( ,BH ,BH(( ,BH ,BH(( HEADING 294
+,BH(( ,BH ,BH(( ,BH ,BH(( ,BH ,BH(( HEADING 295
+,BH (( / HEADING 296
DATA APOS/BH ,BH ,BH ,BH ,BH ,BH ,BH HEADING 297
+,BH ,BH ,BH ,BH ,BH ,BH ,BH HEADING 298
+,BH ,BH ,BH ,BH ,BH ,BH ,BH HEADING 299
+,BH ,BH ,BH ,BH ,BH ,BH ,BH HEADING 300
+,BH / HEADING 301
DATA AMPSDN/BH ++++++,BH++++ ,BH+++++,BH++++ ,BH++++ ,BH++++ HEADING 302
+,BH ++ ,BH++ ,BH++ ,BH+++,BH++ ,BH+++,BH++ HEADING 303
+,BH+ ,BH+ ,BH+ ,BH+++,BH++ ,BH+++,BH++ HEADING 304
+,BH+++++ ,BH++ ,BH++ ,BH+++++,BH+++++ ,BH+++++ HEADING 305
+,BH+++++ / HEADING 306
DATA DOLLAR/BH $$,BH ,BH $$$$$$,BH$$$$ ,BH$$$$$$$$$ HEADING 307
+,BH$$$$$ ,BH$$$ $$,BH ,BH$$$ $$,BH ,BH$$$$$$$$$ HEADING 308
+,BH$$$$$ ,BH $$$$$$,BH$$$$$ ,BH $$$,BH $$$ ,BH $$$ HEADING 309
+,BH $$$ ,BH$$$$$$$$$,BH$$$$$ ,BH $$$$$$,BH$$$$$ ,BH $$$ HEADING 310
+,BH / HEADING 311
DATA ILLGAL/BH+++++,BH+++++ ,BH* ,BH* ,BH* ,BH* HEADING 312
+,BH * ,BH* ,BH* ,BH* ,BH* ,BH* HEADING 313
+,BHID * ,BH* CHARAC,BHTR * ,BH* ,BH* ,BH* ,BH* HEADING 314
+,BH * ,BH* ,BH* ,BH* ,BH* ,BH* HEADING 315
+,BH+++++ / HEADING 316
END HEADING 317

```

SUBROUTINE DATIM 74/74 OPT=1

FTN 4,6+452

15/02/78 17.25.31

SUBROUTINE DATIM (A,B)
CALL DATE (A)
CALL TIME (B)
A = SHIFT(A,6)
B = SHIFT(B,6)
RETURN
END

DATIM	2
DATIM	3
DATIM	4
DATIM	5
DATIM	6
DATIM	7
DATIM	8

EXFL
STORAGE ALLOCATION.

COMPASS 3.4=452.

15/02/78 17.28.53.

ADDRESS LENGTH

0 21
21

BINARY CONTROL CARDS.

IDENT EXFL
END

ENTRY POINTS.

EXFL 2+ CHFL 7+

EXTERNAL SYMBOLS.

SYS=

EXFL

COMPASS 3.4-452.

15/02/78 17.28.53.

PAGE

2

		IDENT EXFL		EXFL	2
		ENTRY EXFL,CHFL		EXFL	3
	*	ROUTINE TO EXTEND FIELD LENGTH OF COMMON		EXFL	4
	*	CALL EXFL (X,N) WHERE X IS THE COMMON ARRAY		EXFL	5
	*	AND N IS THE NUMBER OF WORDS		EXFL	6
	*			EXFL	7
0	00000000000000000000	STATUS DATA	0	EXFL	8
1	0530061400	VFD	30/0LEXFL,,30//	EXFL	9
2	00000000000000000000	EXFL DATA	0	EXFL	10
3	5120000002 +	SA2	EXFL	EXFL	11
	10622	BX6	X2	EXFL	12
4	5160000020 +	SA6	RTN	EXFL	13
	5021000001	SA2	A1+1	EXFL	14
5	53320	SA3	X2	EXFL	15
	36613	IX6	X1+X3	EXFL	16
	0400000012 +	EQ	MEM	EXFL	17
	*			EXFL	18
	*	CALL CHFL (N) N IS THE NEW TOTAL FIELD LENGTH		EXFL	19
	*			EXFL	20
6	031006140000000000001	VFD	30/0LCHFL,30/1	EXFL	21
7	00000000000000000000	CHFL DATA	0	EXFL	22
10	5120000007 +	SA2	CHFL	EXFL	23
	10622	BX6	X2	EXFL	24
11	5160000020 +	SA6	RTN	EXFL	25
	53210	SA2	X1	EXFL	26
	10622	BX6	X2	EXFL	27
12	20636	MEM LX6	30	EXFL	28
	5160000000 +	SA6	STATUS	EXFL	29
13	7160150515	MEMORY	CH,STATUS,1,,	EXFL	30
16	5120000000 +	CHK SA2	STATUS	EXFL	31
	20273	LX2	59	EXFL	32
17	0322000016 +	PL	X2,CHK	EXFL	33
20	00000000000000000000	RTN DATA	0	EXFL	34
21		END		EXFL	35

474008 CM STORAGE USED
MODEL 74 ASSEMBLY

62 STATEMENTS
0.204 SECONDS

7 SYMBOLS
18 REFERENCES

D-167

EXFL
SYMBOLIC REFERENCE TABLE.

COMPASS 3.4-452.

15/02/78 17.28.53.

CHFL	7	PROGRAM*	2/02 E	2/21 L	2/22	
CHK	16	PROGRAM*	2/30 L	2/32		
EXFL	2	PROGRAM*	2/02 E	2/09 L	2/10	
MEM	12	PROGRAM*	2/16	2/27 L		
RTN	20	PROGRAM*	2/12 S	2/24 S	2/33 L	
STATUS	0	PROGRAM*	2/07 L	2/28 S	2/30	2/30
SYS*	0	EXTERNAL*	2/30			

SAMPLE PROBLEM

* INPUT VALUES *

```

FUEL COMPOSITION
0.0000 WEIGHT FRACTION PU02          FRPU02
0.0000 WEIGHT FRACTION PU239        FRU39
0.0000 WEIGHT FRACTION PU240        FRU40
0.0000 WEIGHT FRACTION PU241        FRU41
1.0000 WEIGHT FRACTION UO2          FRUO2
.0300 WEIGHT FRACTION U235          FRU35
.9700 WEIGHT FRACTION U238          FRU38

FUEL DENSITY                      (FRACTION TD)  FRDEN      .950
DENSIFIED FUEL DENSITY            (FRACTION TD)  FRDEN2     .950
RESTRUCTURED FUEL DENSITY         (FRACTION TD)  FRDIN      .960
PELLET DIAMETER                   (INCHES)       DFS        .4200
INITIAL RESTRUCTURED FUEL DIAM.   (INCHES)       DSIN        0.0000
INITIAL CENTER HOLE DIAMETER      (INCHES)       DVGIDZ     0.0000
PELLET-TO-CLAD GAP                (INCHES)       GAP         .0090
CLAD INSIDE DIAMETER              (INCHES)       DCI         .4290
CLAD OUTSIDE DIAMETER             (INCHES)       DCO         .5035
FUEL LENGTH                       (INCHES)       LFUEL      144.00
SCHEED GAS CONTENT                (CC/GRAM)       S           0.0000
FRACTION OF SURGED GAS WHICH IS H2  XH            0.0000
PLENUM VOLUME                     (CU. IN.)       VPLENZ     2.000
COOLANT TEMPERATURE              (DEG F)         TIMLET(1)  533.0
AXIAL TEMPERATURE GRADIENT ACROSS CORE (DEG F)         DTEMP      20.0
FUEL SURFACE ROUGHNESS, ARITH. MEAN (INCH)          ROUF        .000039
CLAD ID SURFACE ROUGHNESS, ARITH. MEAN (INCH)          ROUC        .000020
FILL GAS PRESSURE                 (ATMOSPHERES)  ATMOS      1.0

FILL GAS COMPOSITION
1.00000 MOLE FRACTION HELIUM
0.00000 MOLE FRACTION ARGON
0.00000 MOLE FRACTION HYDROGEN
0.00000 MOLE FRACTION NITROGEN
0.00000 MOLE FRACTION KRYPTON
0.00000 MOLE FRACTION XENON

```

CLADDING IS ZIRCALOY-2

FUEL THERMAL CONDUCTIVITY VALUES DETERMINED FROM DATA OF LYONS, ET AL

FUEL THERMAL CONDUCTIVITY VALUES USED

TEMPERATURE (DEG F)	THERMAL CONDUCTIVITY (BTU/HR-FT-DEG F)
UNRESTRUCTURED FUEL	
77.	5.172
937.	2.466
1832.	1.649
2732.	1.359
3632.	1.336
4532.	1.517
5054.	1.710

FLUX DEPRESSION VALUES USED

DIAMETER (IN)	FLUX RATIO
0.0000	1.0000
.0382	1.0011
.0764	1.0046
.1145	1.0103
.1527	1.0164
.1909	1.0288
.2291	1.0416
.2673	1.0569
.3055	1.0746
.3436	1.0949
.3818	1.1176

**** INPUT OPTIONS ****

NPOW = 4	IFPEAK = 0	IGAS = 0
MINI = 0	NTIME = 4	
IOPT = 0	NPUEL = 0	NFLX = 0
ISTOR = 0	NOW = 0	IDENSF = 0
NCLAD = 0	INL = 0	IRELOC = 0
ZCLAD = 0	KSAVE = 0	ITNCUT = 10
JSTART = 0	NSTEPS = 0	IPLUT = 0

TIME EXPIRED IN PELLET = .912

```

ATMOS      = .1E+01.
RNEAK      = .2E+01. .1133E+01. .3407E+02. .2657E+02. .66E+12.
CP         = .1E+01.
CR         = .1E+01.
DCI        = .429E+00.
DCO        = .5035E+00.
DELTA      = .5E+00. .36E+02. .36E+02. .36E+02. .36E+02. .138364800194E+02. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
           0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
DF9        = .42E+00.
DSINZ      = 0.0.
DTEMP      = .2E+02.
DWDIDZ     = 0.0.
EXTP       = .105E+04.
FLXPAT     = .1E+01.
FR35       = .3E+01.
FR40       = 0.0.
FR41       = 0.0.
FRACAR     = 0.0.
FRACH      = 0.0.
FRACHE     = .1E+01.
FRACKR     = 0.0.
FRACN      = 0.0.
FRACXE     = 0.0.
FRDEN      = .95E+00.
FRDEN2     = .95E+00.
FRPUO2     = 0.0.
FRSIN      = .96E+00.
IDENSF     = 0.
IGAS       = 0.
IOPT       = 0.

```

```

IPEAK = 0.
IPLOT = 0.
IRELOC = 0.
IRL = 0.
ISTOR = 0.
ITMCUT = 10.
JSTART = 0.
ITAPE = 0.
KSAVE = 0.
LFJEL = .144E+03.
LVJIDZ = 0.0.
MINI = 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
MXPQWR = .1E+02.
NC = 4.
NCLAD = 0.
NF = 0.
VFJEL = 0.
VFLX = 0.
VQ4 = 0.
VQJ4 = 0.
VS4 = 0.
VSTEPS = 0.
VTIME = 4.
PRED4 = 0.0.
PSEUDO = .5E+00. .3E+01. .5E+01. .1E+02. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
PROFIL = .5E+00. .9E+00. .11E+01. .11E+01. .9E+00. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0. 0.0.
0.0. 0.0. 0.0.
RADS = 0.0.
ROJC = .2E-04.
ROJF = .39E-04.
S = 0.0.
TINLET = .533E+03.

```


STEP 1 BOUNDARY NUMBER 5 OF 6

SAMPLE PROBLEM

ISAVE = 0

LOCAL LINEAR HEAT RATING 1.6404E+03 W/M (.50 KW/FT)
 PEAK LINEAR HEAT RATING 1.6404E+03 W/M (.50 KW/FT)
 TIME IN-REACTOR 0. SEC. (0.000000 DAYS)
 BURNUP 0. J/KGM (0.0 MKD/MTM)

CLAD OD SURFACE HEAT FLUX 4.0812E+04 W/M2 (1.2946E+04 BTU/HR-FT2)
 FUEL SURFACE HEAT FLUX 4.8413E+04 W/M2 (1.5357E+04 BTU/HR-FT2)
 COOLANT TEMPERATURE 276.74 DEG.C (533.74 DEG.F)
 FILM COEFFICIENT 8.5174E+04 W/M2-DEG.C (1.5000E+04 BTU/HR-FT2-DEG.F)
 CLAD THERMAL CONDUCTIVITY 1.5396E+01 W/M-DEG.C (8.8957E+00 BTU/HR-FT-DEG.F)
 VOLUME AVERAGE FUEL TEMPERATURE 317.0 DEG.C (602.6 DEG.F)
 CHANGE IN FUEL RADIUS DUE TO THERMAL EXPANSION 1.2456E-05 METERS (.000490 INCH)
 CHANGE IN FUEL RADIUS DUE TO RELOCATION 0. METERS (0.000000 INCH)
 CHANGE IN FUEL RADIUS DUE TO DENSIFICATION 0. METERS (0.000000 INCH)
 CHANGE IN FUEL RADIUS DUE TO SWELLING 0. METERS (0.000000 INCH)
 TOTAL CHANGE IN FUEL RADIUS 1.2456E-05 METERS (.000490 INCH)
 ELASTIC CHANGE IN CLAD RADIUS 6.8609E-06 METERS (.000271 INCH)
 CREEP-PLASTIC CHANGE IN CLAD RADIUS -2.3919E-09 METERS (-.000000 INCH)
 TOTAL CHANGE IN CLAD RADIUS 6.8785E-06 METERS (.000271 INCH)

..... RADIAL TEMPERATURE

329. DEG.C (624. DEG.F)
 328. DEG.C (622. DEG.F)
 325. DEG.C (617. DEG.F)
 320. DEG.C (604. DEG.F)
 314. DEG.C (597. DEG.F)
 305. DEG.C (581. DEG.F)
 282. DEG.C (539. DEG.F)
 279. DEG.C (535. DEG.F)

..... RADIUS

.000025 METERS (.00100 INCH)
 .001090 METERS (.04290 INCH)
 .002150 METERS (.08480 INCH)
 .003218 METERS (.12659 INCH)
 .004282 METERS (.16859 INCH)
 .005346 METERS (.21049 INCH)
 .005455 METERS (.21477 INCH)
 .006403 METERS (.25210 INCH)

HOT GAP (RADIAL) 1.0872E-04 METERS (.004281 INCH)
 FUEL-CLAD INTERFACIAL PRESSURE 0. PA (0. PSI)

FUEL TO CLAD GAP CONDUCTANCE 2.0732E+03 W/M2-DEG.C (365.1 BTU/HR-FT2-DEG.F)
 CONDUCTANCE COMPONENTS 0. SOLID CONTACT, 99. GAS CONDUCTION, 1. RADIATION

THERMAL CONDUCTIVITY OF FILL GAS 2.3245E-01 W/M-DEG.C (1.3431E-01 BTU/HR-FT-DEG.F)
 TEMPERATURE JUMP DISTANCE 5.0670E-06 METERS (1.9949E-04 INCH)

GAS RELEASE FRACTION DURING CURRENT TIME STEP 0.0000 CUMULATIVE GAS RELEASE 0.0000
 INTERNAL GAS PRESSURE 1.9527E+05 PA (28. PSI)
 GAS CONTENT 1.9195E-06 KG-MOLE
 MOLE FRACTIONS 1.00 HE, 0.00 AR, 0.00 H, 0.00 N, 0.00 CO, 0.00 KR, 0.00 XE

***** MECHANICS RESULTS FOR AXIAL REGION 5 *****

FUEL STRAINS,PCT

FUEL STRESSES,KSI

CLAD STRAINS,PCT

CLAD STRESSES,KSI

AXIAL STRAIN/STRESS

RADIAL	HOOB	RADIAL	HOOB	EFFECTIVE	RADIAL	HOOB	RADIAL	HOOB	EFFECTIVE	FUEL	CLAD
0.0000	0.0000	0.00	0.00	.00	.2116	.1280	-.17	-7.47	6.32	.2267	.1693
0.0000	0.0000	0.00	0.00	.00	.2097	.1294	-.30	-7.31	6.07	-.0001	-3.7566
0.0000	0.0000	0.00	0.00	.00	.2059	.1323	-.55	-6.98	5.57		
0.0000	0.0000	0.00	0.00	.00	.2043	.1335	-.66	-6.84	5.36	PIN AXIAL DISPLACEMENT	
0.0000	0.0000	0.00	0.00	.00	.2010	.1360	-.87	-6.56	4.93	FUEL	CLAD
0.0000	0.0000	0.00	0.00	.00	.1997	.1369	-.96	-6.44	4.75	.3241	.2677
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							

TIME EXPIRED IN PELET = .929

ISAVE = 0

CLAD OD SURFACE HEAT FLUX	2.4454E+05 W/M2	(7.7569E+04 BTU/HR-FT2)
FUEL SURFACE HEAT FLUX	2.6995E+05 W/M2	(9.1977E+04 BTU/HR-FT2)
COOLANT TEMPERATURE	280.79 DEG.C	(537.42 DEG.F)
FILM COEFFICIENT	2.5174E+04 W/M2-DEG.C	(1.5000E+04 BTU/HR-FT2-DEG.F)
CLAD THERMAL CONDUCTIVITY	1.5520E+01 W/M-DEG.C	(5.9672E+00 BTU/HR-FT-DEG.F)
VOLUME AVERAGE FUEL TEMPERATURE	565.3 DEG.C	(941.5 DEG.F)
CHANGE IN FUEL RADIUS DUE TO THERMAL EXPANSION	2.3145E-05 METERS	(.000911 INCH)
CHANGE IN FUEL RADIUS DUE TO RELOCATION	0. METERS	(0.000000 INCH)
CHANGE IN FUEL RADIUS DUE TO DENSIFICATION	0. METERS	(0.000000 INCH)
CHANGE IN FUEL RADIUS DUE TO SWELLING	4.8482E-07 METERS	(.000019 INCH)
TOTAL CHANGE IN FUEL RADIUS	2.3630E-05 METERS	(.000930 INCH)
ELASTIC CHANGE IN CLAD RADIUS	7.2976E-06 METERS	(.000287 INCH)
CREEP-PLASTIC CHANGE IN CLAD RADIUS	-6.3182E-09 METERS	(-.000000 INCH)
TOTAL CHANGE IN CLAD RADIUS	7.2915E-06 METERS	(.000287 INCH)

```

..... RAOIAL TEMPERATURE .....
597. DEG.C      ( 1100. DEG.F)
589. DEG.C      ( 1092. DEG.F)
586. DEG.C      ( 1081. DEG.F)
530. DEG.C      ( 985. DEG.F)
469. DEG.C      ( 876. DEG.F)
418. DEG.C      ( 785. DEG.F)

300. DEG.C      ( 572. DEG.F)
264. DEG.C      ( 503. DEG.F)

```

```

***** RADIOS *****
.000025 METERS      ( .00100 INCH)
.001092 METERS      ( .04299 INCH)
.002158 METERS      ( .08497 INCH)
.003225 METERS      ( .12696 INCH)
.004291 METERS      ( .16894 INCH)
.005358 METERS      ( .21093 INCH)

.005456 METERS      ( .21479 INCH)
.006404 METERS      ( .25212 INCH)

```

HOT GAP (RADIAL)	9.7966E-05	METERS	(.003857 INCH)
FUEL-CLAD INTERFACIAL PRESSURE	0.	PA	(0. PSI)

FUEL TO CLAD GAP CONDUCTANCE 2.4495E+03 W/M2-DEG.C (431.4 BTU/HR-FT2-DEG.F)
CONDUCTANCE COMPONENTS 0. SOLID CONTACT, 96. GAS CONDUCTION, 2. RADIATION

THERMAL CONDUCTIVITY OF FILL GAS	2.4961E-01 W/M-DEG.C	(1.4422E-01 BTU/HR-FT-DEG.F)
TEMPERATURE JUMP DISTANCE	5.7400E-06 METERS	(2.2599E-04 INCH)

```

GAS RELEASE FRACTION DURING CURRENT TIME STEP          .0002      CUMULATIVE GAS RELEASE          .0001
INTERVAL GAS PRESSURE                                2.0631E+05 PA      ( 30. PSI)
      GAS CONTENT                                      1.9197E-06 KG-MOLE
MOLE FRACTIONS  1.00 HE, 0.00 AR, 0.00 H, 0.00 N, 0.00 CO, 0.00 KR, .00 XE

```

```
***** MECHANICS RESULTS FOR AXIAL REGION 5 *****
```

FUEL STRAINS,PCT		FUEL STRESSES,KSI			CLAD STRAINS,PCT		CLAD STRESSES,KSI		AXIAL STRAIN/STRESS	
RADIAL	HOOPE	RADIAL	HOOPE	EFFECTIVE	RADIAL	HOOPE	RADIAL	HOOPE	EFFECTIVE	FUEL CLAD
0.0000	0.0000	0.00	0.00	.00	.2271	.1363	-.19	-8.10	6.86	.3763 .1775
0.0000	0.0000	0.00	0.00	.00	.2232	.1370	-.32	-7.77	6.47	-.0002 -3.7524
0.0000	0.0000	0.00	0.00	.00	.2146	.1409	-.59	-7.02	5.57	
0.0000	0.0000	0.00	0.00	.00	.2114	.1417	-.66	-6.75	5.26	PIN AXIAL DISPLACEMENT
0.0000	0.0000	0.00	0.00	.00	.2037	.1444	-.90	-6.07	4.49	FUEL CLAD
0.0000	0.0000	0.00	0.00	.00	.2005	.1449	-.90	-5.84	4.24	.5351 .2785
0.0000	0.0000	0.00	0.00	.00						
0.0000	0.0000	0.00	0.00	.00						
0.0000	0.0000	0.00	0.00	.00						
0.0000	0.0000	0.00	0.00	.00						
0.0000	0.0000	0.00	0.00	.00						

TIME EXPIRED IN PELLET = .903

STEP 3 BOUNDARY NUMBER 5 OF 6

SAMPLE PROBLEM

ISAVE = 0

LOCAL LINEAR HEAT RATING 1.9685E+04 W/M (6.00 KW/FT)
 PEAK LINEAR HEAT RATING 1.9685E+04 W/M (6.00 KW/FT)
 TIME IN-REACTOR 2.5920E+06 SEC. (30.000000 DAYS)
 BURNUP 4.6607E+10 J/KGM (539.4 MWD/MTM)

CLAD CD SURFACE HEAT FLUX	4.8903E+05 W/M2	(1.5513E+05 BTU/HR-FT2)
FUEL SURFACE HEAT FLUX	5.7928E+05 W/M2	(1.8375E+05 BTU/HR-FT2)
COOLANT TEMPERATURE	283.25 DEG.C	(541.84 DEG.F)
FILM COEFFICIENT	6.5174E+04 W/M2-DEG.C	(1.5000E+04 BTU/HR-FT2-DEG.F)
CLAD THERMAL CONDUCTIVITY	1.5681E+01 W/M-DEG.C	(9.0605E+00 BTU/HR-FT-DEG.F)
VOLUME AVERAGE FUEL TEMPERATURE	716.7 DEG.C	(1325.7 DEG.F)
CHANGE IN FUEL RADIUS DUE TO THERMAL EXPANSION	3.8104E-05 METERS	(.001500 INCH)
CHANGE IN FUEL RADIUS DUE TO RELOCATION	0. METERS	(0.000000 INCH)
CHANGE IN FUEL RADIUS DUE TO DENSIFICATION	0. METERS	(0.000000 INCH)
CHANGE IN FUEL RADIUS DUE TO SWELLING	1.4514E-06 METERS	(.000057 INCH)
TOTAL CHANGE IN FUEL RADIUS	3.9556E-05 METERS	(.001557 INCH)
ELASTIC CHANGE IN CLAD RADIUS	7.7917E-06 METERS	(.000307 INCH)
CREEP-PLASTIC CHANGE IN CLAD RADIUS	-1.4636E-06 METERS	(-.000001 INCH)
TOTAL CHANGE IN CLAD RADIUS	7.7771E-06 METERS	(.000306 INCH)

..... RADIAL TEMPERATURE

947. DEG.C (1737. DEG.F)
 927. DEG.C (1701. DEG.F)
 869. DEG.C (1596. DEG.F)
 776. DEG.C (1428. DEG.F)
 654. DEG.C (1210. DEG.F)
 513. DEG.C (955. DEG.F)

321. DEG.C (610. DEG.F)
 289. DEG.C (552. DEG.F)

..... RADIUS

.000025 METERS (.00100 INCH)
 .001095 METERS (.04311 INCH)
 .002165 METERS (.08522 INCH)
 .003234 METERS (.12733 INCH)
 .004304 METERS (.16945 INCH)
 .005374 METERS (.21156 INCH)

.005456 METERS (.21481 INCH)
 .006404 METERS (.25215 INCH)

HOT GAP (RADIAL) 8.2530E-05 METERS (.003249 INCH)
 FUEL-CLAD INTERFACIAL PRESSURE 0. PA (0. PSI)

FUEL TO CLAD GAP CONDUCTANCE 3.0187E+03 W/M2-DEG.C (531.6 BTU/HR-FT2-DEG.F)
 CONDUCTANCE COMPONENTS 0. SOLID CONTACT. 98.GAS CONDUCTION. 2.RADIATION

THERMAL CONDUCTIVITY OF FILL GAS 2.6333E-01 W/M-DEG.C (1.5215E-01 BTU/HR-FT-DEG.F)
 TEMPERATURE JUMP DISTANCE 6.3501E-06 METERS (2.5000E-04 INCH)

GAS RELEASE FRACTION DURING CURRENT TIME STEP .0005 CUMULATIVE GAS RELEASE .0004
 INTERNAL GAS PRESSURE 2.2006E+05 PA (32. PSI)
 GAS CONTENT 1.9212E-06 KG-MOLE
 MOLE FRACTIONS 1.00 HE, 0.00 AR, 0.00 H, 0.00 N, 0.00 CO, .00 KR, .00 XE

***** MECHANICS RESULTS FOR AXIAL REGION 5 *****

FUEL STRAINS,PCT FUEL STRESSES,KSI CLAD STRAINS,PCT CLAD STRESSES,KSI AXIAL STRAIN/STRESS

RADIAL	HOOF	RADIAL	HOOF	EFFECTIVE	RADIAL	HOOF	RADIAL	HOOF	EFFECTIVE	FUEL	CLAD
0.0000	0.0000	0.00	0.00	.00	.2451	.1461	-.21	-8.78	7.47	.5743	.1871
0.0000	0.0000	0.00	0.00	.00	.2390	.1472	-.33	-8.29	6.92	-.0003	-3.7471
0.0000	0.0000	0.00	0.00	.00	.2250	.1510	-.64	-7.07	5.57		
0.0000	0.0000	0.00	0.00	.00	.2199	.1514	-.70	-6.66	5.16	PIN AXIAL DISPLACEMENT	
0.0000	0.0000	0.00	0.00	.00	.2071	.1544	-.93	-5.52	4.01	FUEL	CLAD
0.0000	0.0000	0.00	0.00	.00	.2024	.1544	-.96	-5.15	3.69	.8058	.2913
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							
0.0000	0.0000	0.00	0.00	.00							

TIME EXPIRED IN PELET = .907

ISAVE 0

REFERENCES

1. C. R. Hann, C. E. Beyer, and L. J. Parchen, GAPCON-THERMAL-1: A Computer Code for Calculating the Gap Conductance in Oxide Fuel Pins, BNWL-1778, 1973.
2. C. E. Beyer, C. R. Hann, D. D. Lanning, F. E. Panisko and L. J. Parchen, GAPCON-THERMAL-2: A Computer Program for Calculating the Thermal Behavior of an Oxide Fuel Rod, BNWL-1898, 1975.
3. C. L. Mohr, D. D. Lanning, F. E. Panisko, and K. B. Stewart, "GAPCON-3: A Computer Code to Analyze the Path Dependent Thermal and Mechanical Performance of Nuclear Fuel Rods," Proceedings of Specialists Meeting on the Behavior of Water Reactor Fuel Elements Under Accident Conditions. CSNI Report No: 13, Spatino, Norway, September 13-16, 1976.
4. E. A. Olsen and L. H. Boman, Three-Dimensional Thermal Effects in PWR Fuel Rods, 75-WA/HT-78 (ASME), 1976.
5. D. D. Lanning, F. E. Panisko and C. L. Mohr, Verification of the GAPCON-THERMAL-3 Computer Code, PNL-2435, to be published.
6. B. Finlayson, The Method of Weighted Residuals and Variational Principles, Academic Press, New York, NY. 1972.
7. C. S. Desai and J. F. Able, Introduction to the Finite Element Method, Van Nostrand and Reinhold Co., New York, NY. 1972.
8. R. Hill, "A Theory of the Yielding and Plastic Flow of Anisotropic Metals." Proceedings of the Royal Society of London 143A, PP. 281-297, 1948.
9. P. J. Pankaskie, Irradiation Effects on the Mechanical Properties of Zirconium and Dilute Zirconium Alloys. A Review. BN-SA-618, Battelle Pacific Northwest Laboratories, Richland, WA, July 15, 1976.
10. Liquid Metal Fast Breeder Reactor Materials Handbook, HEDL-TME-71-32, August 1971.
11. C. L. Mohr, P. J. Pankaskie, F. E. Panisko, A State-of-the-Art Report on the Anisotropic Deformation and Analysis of Zircaloy. RP-251-1, Battelle, Pacific Northwest Laboratories, Richland, WA, March 1976.
12. C. L. Wilson and E. L. Courtright, An Evaluation of Thermal Expansion Models for Calculating Pellet Expansions in the GAPCON-THERMAL Fuel Behavior Codes, FDD-77-1, 1977.
13. R. G. Bellamy and J. B. Rich, Grain-Boundary Gas Release and Swelling on High Burnup Uranium Dioxide, J. Nucl. Mat. 33, 64-78, 1969.

14. D. Geithoff, G. Karsten and K. Kummerer, "Irradiation Performance of Fast Reactor Fuels," Presented at International Symposium on Plutonium Fuels Technology, AIME Symposium 1967, Nuclear Metallurgy 13, 1968.
15. J. A. L. Robertson, Integral k_d in Fuel Irradiations, CRFD-835, 1959.
16. H. S. Carslaw and J. C. Jaeger, Conduction of Heat in Solids, Oxford University Press, Fairlawn, N.J., 1959.
17. S. Timoshenko, Strength of Materials, Part II, D. Van Nostrand, Princeton, N.J., 1958.
18. G. Kjaerheim and E. Rolstad, In Pile Determination of UO_2 Thermal Conductivity, Density Effects, and Gap Conductance, HPR-80, 1967.
19. G. Kjaerheim and E. Rolstad, In-Core Study of Fuel/Clad Interaction and Center Temperature, HPR-107, 1969.
20. N. Fuhrman, V. Pasupathi, et al., Evaluation of Fuel Rod Performance in Maine Yankee Core 1, EPRI-NP-218, 1976.
21. H. R. Freeburn, et al., Light Water Reactor Fuel Rod Modeling Code Evaluation, EPRI-NP-369, 1977.
22. C. R. Hann and R. K. Marshall, Comparative Analysis of Pellet-Clad Interaction from IFA-431 and IFA-432 Halden Reactor Tests, BNWL-2240, 1977.

DISTRIBUTION

No. of
Copies

OFFSITE

1	<u>DOE Chicago Patent Group</u> 9800 S. Cass Avenue Argonne, IL 60439 A. A. Churm
245	<u>DOE Technical Information Center for Basic Distribution Under NRC-1</u>
61	Supplementary NRC-3 Distribution List
3	<u>DOE Technical Information</u> M. Jinks Chief Mail & Files U.S. NRC Central Files Washington, D.C. 20555
10	<u>Nuclear Regulatory Commission, Division of Reactor Safety Research</u> W. V. Johnston, Chief Fuel Behavior Research Branch Washington, D.C. 20545

ONSITE

1	<u>DOE Richland Operations Office</u> Program Division H. E. Ransom
88	<u>Battelle-Northwest</u> C. R. Hann (4) Technical Information Files (5) Technical Publications

W. J. Bailey	C. L. Mohr (20)
J. O. Barner	F. E. Panisko (20)
S. Begej	P. J. Pankaskie
J. A. Christensen	K. B. Stewart (3)
E. L. Courtright	A. M. Sutey
M. E. Cunningham	D. S. Trent
M. D. Freshley	S. R. Wagoner
J. F. Garnier	R. E. Williford
D. D. Lanning (20)	C. L. Wilson
R. P. Marshall	