THESIS

FINITE ELEMENT SOLUTION OF THE NONLINEAR COUPLED NEUTRONIC-ENERGY EQUATIONS FOR A FAST REACTOR FUEL CELL

by

Roy Edward Kasdorf

December 1976

Thesis Advisor: D. Nguyen
Thesis Advisor: D. Salinas

Approved for public release; distribution unlimited.
# Finite Element Solution of the Non-linear Coupled Neutronic-Energy Equations for a Fast Reactor Fuel Cell

A transient overpower (TOP) accident in a Liquid Metal Fast Breeder Reactor (LMFBR) is considered. The analysis is formulated to model the dynamic response of the reactor fuel subassembly during the initial period of the postulated overpower transient. An equivalent cylindrical cell is used to model the fuel subassembly. The governing neutronic and heat transport equations for each region (fuel, clad, and...
coolant) of the equivalent cylindrical cell are developed. Nuclear Doppler broadening feedback is included in the dynamic model making the coupled equations non-linear. The resulting non-linear partial differential field equations are transformed into a system of ordinary differential equations by the finite element method. An isoparametric, quadratic, rectangular element is used for the discretization of the spatial domain. When using the finite element method, large system matrices may result. To facilitate solution of these large systems, an optimum compacting scheme is utilized. The implicit Gear's method is used for the solution of the system of ordinary differential equations. The results for a sample problem are presented.
Finite Element Solution of the Nonlinear Coupled Neutronic-Energy Equations for a Fast Reactor Fuel Cell

by

Roy Edward Kasdorf
Lieutenant, United States Navy
B.S.M.E., New Mexico State University, 1970

Submitted in partial fulfillment of the requirements for the degrees of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
and
MECHANICAL ENGINEER

from the
NAVAL POSTGRADUATE SCHOOL
December 1976

Author
Roy E. Kasdorf

Approved by:
Song of Nguyen
Thesis Advisor

David Salinas
Thesis Advisor

Richard Thrane
Second Reader

Allen E. Fuchs
Chairman, Department of Mechanical Engineering

Dean of Science and Engineering
ABSTRACT

A transient overpower (TOP) accident in a Liquid Metal Fast Breeder Reactor (LMFBR) is considered. The analysis is formulated to model the dynamic response of the reactor fuel subassembly during the initial period of the postulated overpower transient. An equivalent cylindrical cell is used to model the fuel subassembly. The governing neutronic and heat transport equations for each region (fuel, clad, and coolant) of the equivalent cylindrical cell are developed. Nuclear Doppler broadening feedback is included in the dynamic model making the coupled equations non-linear. The resulting non-linear partial differential field equations are transformed into a system of ordinary differential equations by the finite element method. An isoparametric, quadratic, rectangular element is used for the discretization of the spatial domain. When using the finite element method, large system matrices may result. To facilitate solution of these large systems, an optimum compacting scheme is utilized. The implicit Gear's method is used for the solution of the system of ordinary differential equations. The results for a sample problem are presented.
# TABLE OF CONTENTS

I. INTRODUCTION ................................................. 13

II. DESCRIPTION OF PROBLEM ..................................... 16
   A. PHYSICAL SYSTEM ........................................ 16
   B. SYSTEM MODEL ........................................... 18
   C. NUMERICAL SOLUTION ..................................... 20

III. MODEL DEVELOPMENT ......................................... 22
   A. NEUTRONIC ANALYSIS .................................... 22
      1. Fuel Region .......................................... 23
      2. Cladding Region ..................................... 25
      3. Coolant Region ....................................... 26
      4. Infinite Multiplication Factor ....................... 26
      5. Boundary Conditions ................................ 30
   B. HEAT TRANSFER ANALYSIS ................................ 32
      1. Fuel Region .......................................... 32
      2. Cladding Region ..................................... 33
      3. Coolant Region ....................................... 33
      4. Interface Conditions ................................ 34
      5. Boundary Conditions ................................ 36

IV. FINITE ELEMENT FORMULATION ............................... 39
   A. BASIC THEORY .......................................... 39
   B. SHAPE FUNCTIONS ........................................ 42
   C. COORDINATE TRANSFORMATIONS .......................... 43

V. APPLICATION OF FEM TO GOVERNING FIELD EQUATIONS .... 48
   A. GAUSSIAN QUADRATURE ................................... 48
LIST OF T E S

I. Physical Parameters - - - - - - - - - - - - - - 73
LIST OF FIGURES

1. Equivalent Cylindrical Cell - - - - - - - - - - - - 17
2. Doppler Broadening of a Resonance Peak - - - - - - - 28
3. Gap Heat Transfer Coefficient - - - - - - - - - - - - 35
4. Element Transformation - - - - - - - - - - - - - - - - 44
5. Normalized Shape Functions - - - - - - - - - - - - - - 45
6. Finite Element Discretization - - - - - - - - - - - - - - 62
7. Sample Data Deck - - - - - - - - - - - - - - - - - - - 74
8. Convergence of the Finite Element Method - - - - - 76
   Determination of the Critical Fission Cross Section - - - - - - - - - - - - 77
9. Flux Profile ($\rho = 10$) - - - - - - - - - - - - - - - - 79
10. Temperature Profile ($\rho = 10$) - - - - - - - - - - - - - 80
11. Radial Flux ($\rho = 10$) - - - - - - - - - - - - - - - - - 82
12. Axial Flux ($\rho = 10$) - - - - - - - - - - - - - - - - - 83
13. Radial Temperature Profile ($\rho = 10$) - - - - - - - - - - 84
14. Axial Temperature Profile ($\rho = 10$) - - - - - - - - - - 85
LIST OF SYMBOLS AND NOTATION

A. NOTATION

\(<\) \quad \text{Row vector}

\{\} \quad \text{Column vector}

[ ] \quad \text{Square matrix or indicates a reference}

[ ]^{-1} \quad \text{Inverse of a square matrix}

v \quad \text{Del operator}

\frac{\partial}{\partial x} \quad \text{Partial derivative with respect to } x

\Delta x \quad \text{Change in } x

\int_V \quad \text{Volume integral}

\int_x \quad \text{Integration with respect to } x

\text{det } [x] \quad \text{Determinant of } x

B. SYMBOLS

b \quad \text{Nuclear Doppler constant}

c \quad \text{Concentration of delayed neutron precursors}

C_p \quad \text{Specific heat} \quad [\text{cal/gm °C}]

D(r) \quad \text{Neutron diffusion coefficient} \quad [\text{cm}]

\epsilon \quad \text{Nuclear energy released per fission} \quad [\text{cal/fission}]

h \quad \text{Heat transfer coefficient} \quad [\text{cal/cm}^2\text{sec °C}]

J \quad \text{Jacobian matrix}

J(r,t) \quad \text{Neutron current} \quad [\text{neutrons/cm}^2\text{sec}]

k_\infty \quad \text{Infinite multiplication factor}

K_D \quad \text{Doppler constant}

k(r) \quad \text{Thermal conductivity} \quad [\text{cal/cm sec °C}]

LMFBR \quad \text{Liquid Metal Fast Breeder Reactor}
LOCA  Loss of Coolant Accident
N    Shape function
n    Number of delay neutron groups
n(r,t) Neutron density \( \frac{\text{neutrons}}{\text{cm}^3} \)
N,x Derivative of N with respect to x
\dot{q}(r,t) Nuclear generation \( \text{[cal/cm}^3\text{sec]} \)
R    Residual
r    Spatial coordinate \( \text{[cm]} \)
r,z Global coordinates
S(r,t) Neutron production \( \frac{\text{neutrons}}{\text{cm}^3 \text{sec}} \)
T    Temperature \( \text{[°C]} \)
t    Time \( \text{[sec]} \)
TOP Transient Overpower
v    Neutron velocity \( \text{[cm/sec]} \)
V_{co} Velocity of coolant flow \( \text{[cm/sec]} \)
W    Weighing function
\beta Fraction of fission neutrons which appear as delayed neutrons
\phi(r,t) Neutron flux \( \frac{\text{neutrons}}{\text{cm}^2 \text{ sec}} \)
\lambda Decay constant of the delayed neutron precursors \( \text{[1/sec]} \)
\eta,\xi Local coordinates
\nu Average number of neutrons released per fission
\rho Reactivity
\rho(r) Density \( \text{[g/cm}^2] \)
\Sigma Neutron cross section \( \text{[cm}^{-1}] \)
C. SUBSCRIPTS

a Absorption
c Clad
c0 Coolant
CR Critical
D Delayed, Doppler
f Fission
F Fuel
gap Fuel-clad interface
i,j Group, equation
F Prompt
g:o Clad-coolant interface

D. SUPERSCRIPTS

e Element
e* Adjacent element
0 At time zero
d Derivative with respect to time
ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. Dong Nguyen and to Dr. David Salinas, Professors of Mechanical Engineering, for their advice and guidance throughout the course of this work.

The author is obligated to Dr. Richard Franke, Professor of Mathematics, for his assistance in implementing the integration procedure and for his assistance as second reader.

The author wishes to thank Dr. Gilles Cantin, Professor of Mechanical Engineering, for his invaluable discussions concerning the finite element method.

Finally, the author wishes to thank his wife, Gail, for her encouragement and understanding throughout the course of this study.
I. **INTRODUCTION**

As the world's fossil fuel resources are depleted, more emphasis is being placed on the breeder reactor as a potential means of solving the coming energy crisis. While the development of new energy sources is being pushed, equal effort is being given to the maintenance of an environmentally clean world. To this end, the safety of breeder reactors is receiving a considerable amount of attention before assuming that the breeder reactor is the answer to the energy problem.

The Liquid Metal Fast Breeder Reactor (LMFBR) appears to be one of the most promising breeder reactors. Most engineers will concede there is little probability of a nuclear explosion occurring in the operation of a nuclear reactor. Of major concern to engineers is the loss of coolant accident (LOCA) and the transient overpower accident (TOP). The present analysis is concerned with a TOP accident in a LMFBR. The analysis is formulated to model the dynamic response of the reactor fuel subassembly during the initial period of the postulated overpower transient. The primary consideration is given to the early response of this fuel subassembly to various conditions of disturbances. The phenomenon which occurs after core disassembly (i.e., clad melting) is not the concern of this analysis. Only the time prior to clad melting is being considered.
No consideration is given here as to how the overpower transient occurs or to why the safety features of the reactor did not operate properly. It is postulated that the accident has occurred. In this analysis, the TOP accident is created by either a step increase in reactivity, a ramp increase in reactivity, or a combination of both.

An inherent safety feature of most reactors, nuclear Doppler broadening feedback, is included in the dynamic model of the fuel subassembly. The Doppler feedback acts to reduce the effect of the excursion. Consideration of this feedback creates a non-linear system model which is described by a non-linear, initial-boundary-value problem.

The conventional method of solution uses the standard point kinetics formulation. Recent studies have pointed out a non-negligible error in this model [1], particularly with asymmetric disturbances [2], or space-dependent feedback [3]. In Ref. [4], a somewhat novel approach of using the finite element method (FEM) for the space-time dependent solution of the reactor dynamics problem was demonstrated. The FEM is effective in handling these asymmetric disturbances and space-dependent feedbacks. Therefore, the finite element method was used so that the spatial effects on the postulated problem may be studied further.

The purpose of this work was to demonstrate further the applicability of the FEM to the non-linear reactor dynamics problem as well as to investigate the dynamic response of the reactor fuel subassembly. The analysis required a novel
approach to handle the gap conductances present at the interfaces of the equivalent cell model of the fuel subassembly; this will be clarified in the analysis.
II. DESCRIPTION OF PROBLEM

A. PHYSICAL SYSTEM

The typical Liquid Metal Fast Breeder Reactor (LMFBR) core consists of many hexagonal modules, each containing several hundred fuel pins. For this analysis, an equivalent cylindrical cell is used to model the fuel subassembly; see Figure 1. The use of equivalent cells as models for larger systems has been common practice in nuclear analysis (i.e., the well known Wigner-Sietz method). In using an equivalent cell, the actual shape of the reactor core is not important, and the analysis is applicable to any reactor which has the same equivalent cell.

The equivalent cell considered in this analysis, Figure 1, is fueled with enriched uranium dioxide, has a stainless steel cladding, and has liquid sodium for a coolant. The dimensions used are

\[
\begin{align*}
  a &= 0.254 \text{ cm}, \\
  b &= 0.292 \text{ cm}, \\
  c &= 0.365 \text{ cm}, \\
  H &= 33.0 \text{ cm}.
\end{align*}
\]

The gap between the fuel and cladding is very small and, in fact, may be nonexistent as in bonded fuels. The dimension of this gap has been assumed negligible. The height, \( H \), of the fuel rod is shorter than many proposed systems (Fast Flux Testing Facility and Clinch River Breeder Reactor). However,
Figure 1. Equivalent Cylindrical Cell
to facilitate the numerical solution a smaller rod was used. The dynamic behavior prior to fuel pin failure for this system should be similar to the behavior of larger systems.

The treatment of this problem in three dimensions would be prohibitive in computer usage. Therefore, azimuthal symmetry is assumed, and the problem becomes a two-dimensional cylindrical \((r,z)\) problem.

B. SYSTEM MODEL

The analysis considers the monoenergetic neutron diffusion approximation to model the transient neutron transport problem. A simple conduction-convection heat transfer model is used for the energy transport problem. The temperatures in the model are directly coupled to the neutron flux through the nuclear heat generation within the fuel. The neutron population is in turn coupled to the temperature through any of a number of reactivity feedback mechanisms. The nuclear Doppler effect is perhaps the most important of these mechanisms since it provides a negative temperature coefficient which increases the inherent stability of the reactor. Prior to core disassembly and fuel melting, the nuclear Doppler effect is the most dominant feedback and is, therefore, the only feedback mechanism considered in this analysis.

For irradiated, mixed-oxide fuels, a phenomenon of fuel restructuring has been commonly observed. This restructuring, essentially a change of phase of the fuel material, presents a unique heat transfer problem particularly during transient conditions. The problem has not been fully characterized and
is beyond the scope of this analysis. The fuel was, therefore, assumed to be a homogeneous mixture of enriched uranium dioxide.

At the fuel-cladding interface, there exists a gap which produces a thermal resistance. This thermal resistance is one of the most significant deterrents to the energy transfer to the coolant. The interface may be in physical contact or an actual gap may exist. The prediction of the thermal resistance is extremely complicated and must take into consideration many parameters: initial dimensions, type of bond, fill gas composition, fuel restructuring, fuel swelling, prior fuel life cycle, to mention a few. Reference [5] documents a computer program which attempts to predict the gap conductance, $\mathcal{H}_{\text{gap}}$. This treats the thermal resistance at the interface in the same manner as a convection heat transfer coefficient when considering convection heat transfer. Since it is not the objective of this analysis to predict the gap coefficient, a representative set of values for gap coefficient, as given in Ref [6], is used in this analysis. These values are assumed to remain static during the transient. The gap conductance profile actually varies with time and will have an effect upon the transient, as noted in Ref. [7]. The prediction of this variance was not considered important for this analysis; therefore, the static assumption was made.

An average convection heat transfer coefficient was used to determine the heat transfer from the cladding to the coolant. The value used was determined from an empirical formula given in Ref. [5] and repeated in Appendix C.
In this work, consideration is given to step and ramp increases in reactivity, although any reactivity transient may easily be considered. The step and ramp increases in reactivity probably represent the most realistic physical reactivity inputs in a reactor. Once the reactivity has been inserted, the transient overpower excursion begins. Unless the Doppler feedback can override the inserted reactivity, the excursion will continue until there is physical core disassembly.

C. NUMERICAL SOLUTION

The system of equations which models the proposed problem is a non-linear, initial-boundary-value problem. The conventional method of solution of the reactor dynamics is the point kinetics formulation. It was pointed out in Refs. [1], [2], [3], and [4], that there is a non-negligible error in this model, particularly under conditions of asymmetric disturbances or space-dependent feedback. Reference [4] demonstrates the somewhat novel approach of using the finite element method (FEM) to solve the space-time dependent reactor dynamics problem. As shown in Ref. [4], the FEM is quite effective in handling localized perturbations and space-dependent feedback. In this work, only uniform disturbances were considered; however, the feedback model was space-dependent. Therefore, the finite element method is used to solve the non-linear, coupled, space-time dependent neutronic and heat transport field equations. The solution technique results in a large computer storage requirement; therefore, an optimum compact storage scheme, Ref. [8], is utilized for storage of the discretized matrices.
Once the domain has been discretized by the FEM, the solution of the resulting ordinary differential equations was to be accomplished by the implicit Gear's method, Ref. [9].
III. MODEL DEVELOPMENT

A. NEUTRONIC ANALYSIS

In this section the governing field equations for the neutron population (flux) for each of the three regions (fuel, clad, and coolant) of the domain will be formulated. The monoenergetic, diffusion theory will be used.

Consider an arbitrary volume of material within a reactor. Applying the condition of conservation to the monoenergetic neutrons leads to the neutron equation of continuity [10]

$$\frac{\partial n(r,t)}{\partial t} = S(r,t) - \Sigma_a(r)\phi(r,t) - \text{div } J(r,t) \quad (1)$$

where

- $r$ - spatial point
- $t$ - time
- $n(r,t)$ - neutron density
- $S(r,t)$ - neutron production
- $\Sigma_a(r)$ - neutron absorption cross section
- $\phi(r,t)$ - neutron flux
- $J(r,t)$ - neutron current

The left-hand side of equation (1) represents the time rate of change of the neutron density which is related to flux, $\phi$, by

$$n(r,t) = \frac{1}{v} \phi(r,t) \quad (2)$$
where
\[ v \text{ - neutron velocity} \]

On the right-hand side of equation (1), the first term is neutron production, the second term is a neutron loss through absorption, and the third term is a neutron loss through leakage from the control volume.

Using equation (2) and applying Fick's Law to the equation of continuity results in the classical neutron diffusion equation

\[ \nabla \cdot (D(r) \nabla \phi(r,t)) - \Sigma_a(r) \phi(r,t) + S(r,t) = \frac{1}{v} \frac{\partial \phi(r,t)}{\partial t} \]  

(3)

where \( D(r) \text{ - neutron diffusion coefficient} \)

The vector notation used here is intended to include only two dimensions \((r,z)\) since azimuthal symmetry has been assumed.

Equation (3) is applicable to each of the three regions of the equivalent cylindrical cell. The subscripts \( F \) (fuel), \( c \) (cladding), and \( c_o \) (coolant), will be used to denote these regions.

1. Fuel Region

In applying equation (3) to the fuel region the material properties of the fuel must be used. Within the fuel the neutron source term is due to the nuclear fission process. During fission, neutrons are released as both prompt neutrons and delayed neutrons so that

\[ S(r,t) = s_p(r,t) - s_d(r,t) \]  

(4)
where \( S_p(r,t) \) - prompt neutron source  
\( S_D(r,t) \) - delayed neutron source

The neutron sources are commonly represented as [10]

\[
S_p = k_{\infty} \Sigma_a \phi_F (1 - \beta)
\]  
(5)

and

\[
S_D = \sum_{i=1}^{n} C_i \lambda_i
\]  
(6)

where

- \( k_{\infty} \) - infinite multiplication factor
- \( \beta \) - fraction of fission neutrons which appear as delayed neutrons
- \( n \) - number of delayed neutron groups
- \( C_i \) - concentration of delayed neutron precursors in the \( i^{th} \) group
- \( \lambda_i \) - decay constant of the delayed neutrons

The space and time variables, \( r \) and \( t \), will be dropped except where needed for clarification.

The concentration of delayed neutron precursors, \( C_i \), is given by the following first order partial differential equation [10]

\[
\frac{\partial C_i}{\partial t} = \beta_i k_{\infty} \Sigma_a \phi_F - \lambda_i C_i
\]  
(7)

where

- \( \beta_i \) - fraction of delayed neutrons which appear as delayed neutrons in the \( i^{th} \) group
The solution of equation (7) is

$$C_i = \beta_i \int_{0}^{t} e^{-\lambda_i (t-t')} k_\infty (r,t') \Sigma a_F \phi F dt' + C_i^0 e^{-\lambda_i t} \quad (8)$$

$C_i^0$ - concentration of delayed neutron precursors of the $i$th group at time zero

Reference [10] develops an expression for the initial concentration of delayed neutrons

$$C_i^0 = \beta_i k_\infty \Sigma a_F \phi^0 / \lambda_i \quad (9)$$

where

$k_\infty$ - initial finite multiplication factor

$\phi^0$ - initial neutron flux

Combining equations (3), (4), (5), (6), (8), and (9), yields the governing equation for the fuel region

$$\nabla \cdot (D_F \nabla \phi F) + \Sigma a_F \phi F [k_\infty (1-\beta) - 1]$$

$$+ \sum_{i=1}^{n} \lambda_i [\beta_i \int_{0}^{t} e^{-\lambda_i (t-t')} k_\infty (r,t') \phi F \Sigma a_F dt']$$

$$+ \beta_i k_\infty \Sigma a_F \phi F / \lambda_i e^{-\lambda_i t} = \frac{1}{v} \frac{\partial \phi_F}{\partial t} \quad (10)$$

2. Cladding Region

The cladding separates the fuel and coolant and contains the fuel and the fission by-products. Equation (3) governs the neutron flux in the clad. Since the clad contains
no fissile material, the neutron source term is zero. The field equation is, then,

\[ \nabla \cdot (D_c \nabla \phi_c) - \Sigma_a \phi_c = \frac{1}{v} \frac{\partial \phi_c}{\partial t} \]  

(11)

3. **Coolant Region**

The annular region around the cladding of the equivalent cell contains the coolant. As in the clad, there is no fissile material in the coolant and, therefore, no neutron source term. The equation governing the neutron flux in the coolant is

\[ \nabla \cdot (D_{co} \nabla \phi_{co}) - \Sigma_{aco} \phi_{co} = \frac{1}{v} \frac{\partial \phi_{co}}{\partial t} \]  

(12)

Equations (10), (11), and (12) are the one-velocity, diffusion approximation used to model the neutron transport problem.

4. **Infinite Multiplication Factor**

The infinite multiplication factor, \( k_\infty \), may be expressed as the infinite multiplication factor at time zero (start of transient), \( k_\infty^o \), plus the postulated reactivity insertion (such as a step or a ramp), \( \rho \), minus the change in the Doppler reactivity feedback, \( \Delta \rho_D \). Other feedback mechanisms are normally not as significant as the Doppler broadening feedback prior to fuel melting and have been neglected in this analysis. Therefore,

\[ k_\infty = k_\infty^o + \rho - \Delta \rho_D \]  

(13)
For a fast reactor, $k_\infty$ may be approximated as

$$k_\infty = \nu \frac{\Sigma_{fF}}{\Sigma_{aF}}$$

(14)

where

- $\nu$ - average number of neutrons released per fission
- $\Sigma_{fF}$ - fission cross section of the fuel
- $\Sigma_{aF}$ - absorption cross section of the fuel

The nuclear Doppler effect is a very important safety feature in a nuclear reactor. Nuclei in an atom are in continual motion due to their own thermal energy. As a result of this motion, even when monoenergetic neutrons interact with the atom, there appears to be a spread in the energy of the neutron - the Doppler effect. It can be shown that the cross section of a resonance becomes less in magnitude and wider as the motion of the nuclei increases [10]. As the temperature increases, the motion increases, and the shape of a resonance cross section broadens. This broadening increases the average cross section, thus, providing a negative temperature coefficient. This effect is shown in Figure 2. It is this nuclear Doppler broadening effect which provides one of the few inherent, reliable, negative reactivity feedbacks which slows an overpower transient and possibly stops a mild overpower transient.
Figure 2. Doppler Broadening of a Resonance Peak
The Doppler reactivity change with respect to fuel temperature changes should be written as [6]

\[
\frac{d\rho_D}{dT} = a T^{-3/2} + b T^{-1} + c T^m \tag{15}
\]

where \( a, b, \) and \( c \) are parameters determined from experimental work and \( m \) is an integer. However, as noted in Ref. [6], a substantial amount of work has shown that \( T \frac{d\rho_D}{dT} \) is very nearly constant over the temperature range under consideration. Therefore, the coefficients \( a \) and \( c \) have been set equal to zero, and \( b \) is defined as

\[
b = K_D = T \frac{d\rho_D}{dT} \tag{16}
\]

The constant, \( K_D \), is commonly called the Doppler constant. Solving equation (16) for \( \rho_D \) yields

\[
\rho_D = b \ln T_F + K \tag{17}
\]

where \( K \) is an integration which may be obtained from initial conditions.

\[
K = \rho_D^0 - b \ln T_F^0 \tag{18}
\]

where

\[
\rho_D^0 - \text{Doppler effect at time zero}
\]

\[
T_F^0 - \text{fuel temperature at time zero}
\]
Substituting for $K$ in equation (17) will give

$$
\rho_D - \rho_D^0 = \Delta \rho_D = b \ln(T_F/T_F^0)
$$

(19)

The infinite multiplication factor now becomes

$$
k_\infty = k_\infty^0 + \rho - b \ln(T_F/T_F^0)
$$

(20)

The effect of delayed neutrons is small compared to the prompt neutron effect; therefore, it may be assumed that the Doppler effect on delayed neutrons is insignificant to the overall problem. The $k_\infty$ of equation (8) is, then, assumed to be $k_\infty^0$. With this assumption and equation (20), the neutron diffusion equation in the fuel, equation (10), may be rewritten as

$$

V \cdot (D_F \nabla \phi_F) + \Sigma_{aF} \phi_F \left[ k_{\infty}^0 (1-\beta) - 1 + \rho (1-\beta) - b(1-\beta) \ln(T_F/T_F^0) \right]

$$

$$
+ \sum_{i=1}^{n} \lambda_i \beta_i \Sigma_{aF} \int_{0}^{t} e^{-\lambda_i (t-t')} \left[ k_{\infty}^0 + \rho \right] \phi_F dt' + C_i e^{-\lambda_i t} = \frac{1}{V} \frac{\partial \phi_F}{\partial t}

$$

(21)

To facilitate the present analysis, the number of delayed neutron groups is taken as one averaged group. So that, $\lambda_i$ and $\beta_i$ become $\lambda$ and $\beta$, respectively. This approximation should have little effect on the problem under consideration.

5. Boundary Conditions

The neutron diffusion problem involves solution of the partial differential equations (11), (12), and (21), with the following boundary, interface, and initial conditions:
1) \( \frac{\partial \phi_F}{\partial r} (0,z,t) = 0 \)

2) \( \phi_F (a,z,t) = \phi_C (a,z,t) \)

3) \( D_F \frac{\partial \phi_F}{\partial r} (a,z,t) = D_C \frac{\partial \phi_C}{\partial r} (a,z,t) \)

4) \( \phi_C (b,z,t) = \phi_{co}(b,z,t) \)

5) \( D_C \frac{\partial \phi_C}{\partial r} (b,z,c) = D_{co} \frac{\partial \phi_{co}}{\partial r} (b,z,t) \)

6) \( \frac{\partial \phi_{co}}{\partial r} (c,z,t) = 0 \)

7) \( \phi_F(r, \pm \frac{H}{2}, t) = \phi_C(r, \pm \frac{H}{2}, t) = \phi_{co}(r, \pm \frac{H}{2}, t) = 0 \)

8) \( \phi_F (r,0) = \phi_F^o (r) \)

9) \( \phi_C (r,0) = \phi_C^o (r) \)

10) \( \phi_{co}(r,0) = \phi_{co}^o (r) \)

Boundary condition 1) results from the assumed azimuthal symmetry. Interface conditions 2), 3), 4), and 5) are continuity conditions of the flux. Boundary condition 6) results from the use of an equivalent cell and basically indicates there is an equal number of neutrons transferred in and out of the cell at the outer boundary. This should be valid unless the cell is located near the outer edge of the reactor. Boundary condition 7) is an assumption that the flux is zero at the axial boundaries of the cell. Initial conditions 8), 9), and 10) are the assumed initial distributions of the neutron flux.
B. HEAT TRANSFER ANALYSIS

In this section, the principle of conservation of energy will be used to formulate the governing field equations for the heat transport in each of the three regions. A simple heat conduction model with convection heat transfer to the coolant is used to model the heat transport problem. A gap conductance model is used to describe the heat transport across the gap at the fuel-clad interface.

1. Fuel Region

Conservation of energy within the fuel region yields the unsteady heat conduction equation with a generation term

\[ \nabla \cdot (k_F(r) \nabla T_F(r,t)) + \dot{q}(r,t) = \rho_F(r) C_{PF}(r) \frac{\partial T_F}{\partial t}(r,t) \]  

(22)

where

- \( k_F(r) \) - thermal conductivity of the fuel
- \( T_F(r,t) \) - fuel temperature
- \( \dot{q}(r,t) \) - nuclear energy generation per unit volume
- \( \rho_F(r) \) - fuel density
- \( C_{PF}(r) \) - fuel specific heat

As in the neutronic analysis, the vector notation is intended to include only two dimensions, \( (r,z) \). The \( r \) and \( t \) will be dropped except where needed for clarification.

The nuclear generation term may be expressed as

\[ \dot{q} = \epsilon \Sigma_{FF} \phi_F \]  

(23)
where \( e \) - nuclear energy released per fission

As can be seen, it is through the nuclear generation term that the temperature is directly coupled to the neutron flux. This coupling and the temperature dependent Doppler reactivity feedback combine to make the coupled problem nonlinear.

Substituting equation (23) into equation (22) yields the governing thermal equation for the fuel

\[
\nabla \cdot (k_F \nabla T_F) + e \Sigma_{fF} \phi_F = \rho_F C_p F \frac{\partial T_F}{\partial t}
\]

(24)

2. Cladding Region

Conservation of energy within the clad will yield the heat conduction equation. With nuclear generation, a relatively small amount (~5%) of the energy will be released in the cladding and the coolant. However, in this analysis it is assumed that the total energy release is in the fuel region. This should not create any significant error. Using this assumption, the unsteady heat conduction equation for the cladding becomes

\[
\nabla \cdot (k_c \nabla T_c) = \rho_c C_{pc} \frac{\partial T_c}{\partial t}
\]

(25)

3. Coolant Region

Once again, conservation of energy will lead to the heat conduction equation plus an additional term which
takes into consideration the coolant flow. The governing equation is

\[ V_c (k_c \nabla T_c) - V_c \frac{\partial}{\partial z} \left( \rho_c C_{pc} T_c \right) = \rho_c C_{pc} \frac{\partial T_c}{\partial t} \]  

(26)

where \( V_c \) - coolant flow velocity

Equations (24), (25), and (26) are the governing equations used to model the energy transport problem.

4. **Interface Conditions**

The interface between the fuel and the cladding may be an actual gap with a finite distance, or the surfaces may be in intermittent contact on a microscopic scale. To model the heat transfer across this interface, a gap heat transfer coefficient is introduced. The gap coefficient must take into consideration many items (e.g., radiation heat transfer across the gap, heat transfer by solid-to-solid contact, heat conduction across a gas filled gap). The prediction of this gap coefficient is extremely complicated and beyond the scope of this work. In Ref. [6], a set of values for \( H_{\text{gap}} \) is given and the axial variation of \( H_{\text{gap}} \) in this analysis is approximately the same. A cosine curve has been fitted to the sample data to determine the gap coefficient, see Figure 3. The heat flux across the fuel-clad interface is, then,

\[ q = H_{\text{gap}} (T_F - T_C) \]  

(27)
Figure 3. Gap Heat Transfer Coefficient
The heat conducted out of the fuel is governed by Fourier's equation and is equal to the heat transferred across the gap

\[ q = -k_F \frac{\partial T_F}{\partial r} \]  

(28)

Equating equations (27) and (28) gives the fuel-clad interface condition

\[ T_F(a,z,t) + \frac{k_F(a,z)}{H_{\text{gap}}(z)} \frac{\partial T_F(a,z,t)}{\partial r} = T_C(a,z,t) \]  

(29)

and from continuity

\[ k_F(a,z) \frac{\partial T_F(a,z,t)}{\partial r} = k_C(a,z) \frac{\partial T_C(a,z,t)}{\partial r} \]  

(29a)

As is common practice in convection heat transfer analysis, a coolant surface heat transfer coefficient, \( h_{\text{surf}} \), is used to account for the thermal resistance at the clad-coolant interface. Similar to the fuel-clad interface analysis, the clad-coolant interface conditions may be determined

\[ T_C(b,z,t) + \frac{k_C(b,z)}{h_{\text{surf}}} \frac{\partial T_C(b,z,t)}{\partial r} = T_{\text{co}}(b,z,t) \]  

(30)

and

\[ k_C(b,z) \frac{\partial T_C(b,z,t)}{\partial r} = k_{\text{co}}(b,z) \frac{\partial T_{\text{co}}(b,z,t)}{\partial r} \]  

(30a)

5. **Boundary Conditions**

The boundary and initial conditions for the heat transport problem are:
1) \( \frac{\partial T_F}{\partial r} (0, z, t) = 0 \)

2) \( T_{CO}(r, -\frac{H}{2}, t) = T_{PLENUM} \)

3) \( \frac{\partial T_{CO}}{\partial z} (r, \frac{H}{2}, t) = 0 \)

4) \( \frac{\partial T_{CO}}{\partial r} (c, z, t) = 0 \)

5) \( \frac{\partial T_F}{\partial z} (r, \pm \frac{H}{2}, t) = \frac{\partial T_C}{\partial z} (r, \pm \frac{H}{2}, t) = 0 \)

6) \( T_F (r, 0) = T_F^0 (r) \)

7) \( T_C (r, 0) = T_C^0 (r) \)

8) \( T_{CO}(r, 0) = T_{CO}^0 (r) \)

Boundary condition 1) results from the assumed azimuthal symmetry. The coolant has been assumed to enter the flow channel at a constant temperature, \( T_{PLENUM} \). This results in condition 2). Boundary conditions 3) and 5) result from an assumption that no heat is transferred axially from the fuel rod. Boundary condition 4) is the result of the use of the equivalent cell. Conditions 6), 7), and 8) are the assumed initial conditions.

Solution of the nonlinear coupled neutronic and energy transport problem involves the solution of the partial differential equations (11), (12), (21), (24), (25), and (26), with the appropriate boundary, interface, and initial conditions.

Several works, Refs. [4], [11], [12], have demonstrated the feasibility and success of the finite element method in solving nuclear reactor dynamics problems. The FEM
is used to reduce the partial differential equations developed in this analysis to a system of ordinary differential equations. Integration of these ordinary differential equations (ODE) yields the solution.
IV. FINITE ELEMENT FORMULATION

In this section, the basic theory underlying the finite element method is formulated. Selection of the finite elements and the shape functions for the elements are given. Some simple transformations which facilitate the integration necessary in the FEM are also presented.

A. BASIC THEORY

To obtain a numerical solution, the governing partial differential field equations are transformed into a system of ODE in finite dimensional vector space. This may be accomplished in several manners such as the finite-difference method, the variational method, or the weighted residual method. In this work, the Galerkin method (a weighted residual method) is utilized for the discretization of the spatial domain. The Galerkin procedure will be applied to each of the governing field equations. Any of these equations may be considered to be in the following form

\[ \frac{\partial \psi}{\partial t}(r,t) - \ell \psi(r,t) = f(r,t) \quad (31) \]

where \( \psi \) represents the unknown function, e.g., in equation (21), \( \psi \) represents \( \phi_F \), \( \ell \) represents the operator for each individual equation, and \( f \) is a forcing function. In the finite element method the solution is approximated as

\[ \psi(r,t) \approx \bar{\psi}(r,t) = \sum_{i=1}^{N} N_i(r) \psi_i(t) = <N_i\{\psi_i}\rangle \quad (32) \]
where

\( N \) - the number of degrees of freedom

\( N_i \) - the element shape function

\( \psi_i \) - unknown coordinate function

\( \langle \cdot \rangle \) - matrix notation for a row vector

\( \{ \cdot \} \) - matrix notation for a column vector

\(<N_i> = \langle N_1 \, N_2 \ldots \, N_i \ldots \, N_N\rangle

\{N_i\} = \begin{bmatrix}
N_1 \\
N_2 \\
\vdots \\
N_i \\
\vdots \\
N_N
\end{bmatrix}

The residual, \( R(\mathbf{r}, t) \), is a measure of the error in this finite element approximation. The residual may be considered as

\[ R = \frac{\partial \psi}{\partial t} - \ell \psi - f \]  \hspace{1cm} (33)

The best solution for \( \psi \) is one which "minimizes" this residual. Various "minimums" are obtained by the weighted residual method by setting

\[ \int_V W_i(\mathbf{r}) R \, dV = 0 \quad i=1,2,\ldots,N \]  \hspace{1cm} (34)

With the Galerkin method, the weighting functions are the shape functions defining the approximation of equation (32) (i.e., \( W_i = N_i \)). A noteworthy attribute of the Galerkin method is the opportunity of using an integration-by-parts
of the terms involving the second order spatial derivatives. A lower order finite element may be used than would have been possible otherwise. Once the weighting functions have been chosen, the problem becomes
\[ \int_V N_i \left( \frac{\partial^2 \psi}{\partial x^2} - L \psi - f \right) dV = 0 \quad i=1,2,...,N \quad (35) \]

The integration involved in equation (35) is carried out on the element level, taking advantage of the use of a "local" coordinate system. Once the integration is accomplished, the results are merged into a system using "global" coordinates. On the element level
\[ \tilde{\psi}^e = \langle N_j \rangle \psi_j^e \quad j=1,2,...,N \quad (36) \]

where the superscript \( e \) indicates the element level. Substituting \( \tilde{\psi}^e \) into equation (35) and noting \( (\psi_j)^e \) is not a function of the spatial domain yields
\[ \int_V \langle N_j \rangle (N_j) \psi_j^e dV - \int_V \langle N_j \rangle L (N_j) \psi_j^e dV - \int_V \langle N_j \rangle f dV (\psi_j)^e = 0 \quad (37) \]

where \( i,j = 1,2,...,N^e \)

\( N^e \) - number of degrees of freedom for an element

The operator \( L \) will vary depending upon which governing equation is under consideration.
B. SHAPE FUNCTIONS

The shape functions, \( N_i \), are chosen to satisfy certain completeness and convergence criteria [13] and will depend upon the finite-element used for the spatial discretization.

Many previous works, Refs. [4], [11], and [12], utilized linear triangular shaped elements to discretize the spatial domain. This element was the first element considered. However, because the width of the cladding is very thin, elements in the cladding region would have extremely large aspect ratios (ratio of base to height) unless an extremely large number of elements in the axial direction were used. A large number of elements becomes numerically untractable. Previous experience with triangular elements had shown that large aspect ratios yield inaccurate results. Ziamai, Ref. [14], showed the error, \( e \), when using triangular elements, is proportional to the square of the longest side, \( h \), and inversely proportional to the sine of the smallest angle, \( \gamma \)

\[ e \propto \frac{h^2}{\sin \gamma} \]

A triangular element with a large aspect ratio necessarily must have a small related angle which adversely affects the error in the FEM. Hopefully to alleviate the problem, an isoparametric, quadratic, rectangular element was selected. The aspect ratio would still be large, but experience, Ref. [15], with the use of rectangular elements indicated that a large aspect ratio is not always a detrimental factor.
The shape functions for this element are well documented, Ref. [13]. Utilizing a "local" coordinate system (See Figure 4), the shape functions may be written as

Corner nodes  \( i = 1, 3, 5, 7 \)
\[
N_i = \frac{1}{2}(1+\xi_0)(1+\eta_0)(\xi_0+\eta_0-1)
\]

Midside nodes \( i = 2, 6 \)
\[
N_i = \frac{1}{2}(1-\xi^2)(1+\eta_0)
\]
\[
N_i = \frac{1}{2}(1+\xi_0)(1-\eta^2)
\]

where
\[
\xi_0 = \xi \xi_1
\]
\[
\eta_0 = \eta \eta_1
\]

These normalized shape functions are shown in Figure 5.

The local and global coordinates are related by the following

\[
r = <N_i>^e (r_i)^e
\]
\[
z = <N_i>^e (z_i)^e
\]

C. COORDINATE TRANSFORMATIONS

When using a local coordinate system, some simple transformations facilitate the integrations required by equation (37). In cylindrical coordinates with azimuthal symmetry

\[
dV = 2\pi rdrdz
\]

The derivative terms may be transformed by the following

\[
\begin{bmatrix}
\frac{\partial N_i}{\partial r} \\
\frac{\partial N_i}{\partial z}
\end{bmatrix} = [J]^{-1}
\begin{bmatrix}
\frac{\partial N_i}{\partial \xi} \\
\frac{\partial N_i}{\partial \eta}
\end{bmatrix}
\]
Figure 4. Element Transformation
Figure 5. Normalized Shape Functions
where $[J]^{-1}$ is the inverse of the $2 \times 2$ Jacobian matrix defined in Appendix A. As shown in Appendix A, this inverse can be easily shown to be (for this problem)

$$[J]^{-1} = \begin{bmatrix} J_{11}^* & J_{12}^* \\ J_{21}^* & J_{22}^* \end{bmatrix}$$ (42)

where

$$J_{11}^* = 2/\Delta r, \quad J_{12}^* = 0$$
$$J_{21}^* = 0, \quad J_{22}^* = 2/\Delta \zeta$$

$\Delta r$ - radial length of the element
$\Delta \zeta$ - axial length of the element

Elements of area transform as

$$drdz = \det[J] \, d\xi d\eta$$ (43)

For this particular problem, $\det[J]$ may be shown to be (Appendix A)

$$\det[J] = \frac{A^e}{4}$$ (44)

where $A^e$ - area of the element

Elements of axial length become

$$dz = \frac{L^e}{2}$$ (45)

where $L^e$ - axial length of the element
Utilization of these transformations makes the integrations required in equation (37) amenable to integration by numerical Gaussian quadrature.
V. APPLICATION OF FEM TO GOVERNING FIELD EQUATIONS

In this chapter, equation (37) is applied to the governing field equations previously derived. The element matrices for each of the operators are developed so that the discretization of the spatial domain may be accomplished. The integrations required by the application of equation (37) are performed numerically using Gaussian quadrature.

A. GAUSSIAN QUADRATURE

Prior to the application of equation (37), it is appropriate to discuss briefly the procedure used for the numerical integration. The product Gaussian quadrature formula is [16]

\[
I_A = \int_{-1}^{1} \int_{-1}^{1} g(\xi, \eta) d\eta d\xi = \sum_{i=1}^{m} \sum_{j=1}^{m} W_{i,j} g(\xi_i, \eta_j) \tag{46}
\]

where

- \( I_A \) - area integration
- \( g(\xi, \eta) \) - any function of \( \xi \) and \( \eta \)
- \( W_{i,j} \) - weight associated with location \( i \) or \( j \)
- \( m \) - number of Gauss sampling points in one-dimension

The values of the weights associated with each Gauss point are given in Ref. [16]. Equation (42) may be simplified somewhat by combining the summations and weights

\[
I_A = \sum_{k=1}^{m^2} W_k g(\xi_k, \eta_k) \tag{47}
\]
where \( k = \imath x_j \)
\[ W_k = W_i \times W_j \]

For line integrations, the Gaussian quadrature formula involves only one summation
\[
I_L = \int_{-1}^{1} f(n)dn = \sum_{i=1}^{m} W_i f(n_i) \quad (48)
\]

B. NEUTRONIC FIELD EQUATIONS

The discretization of the spatial domain by the finite element method is accomplished by applying equation (37) to the governing field equations, using

\[
\phi_k = \langle N_j \rangle \langle \psi_{kj} \rangle \quad j=1,2,...,8 \quad (49)
\]
\[ k = F, c, c_0 \]

1. Fuel Region

The governing equation for the fuel region, equation (20) after applying equation (37) becomes
\[
2\pi \int \int \frac{N_1}{V} \frac{\partial \phi_F}{\partial z} r dr dz - 2\pi \int \int \frac{N_1}{V} \left( \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial \phi_F}{\partial r} \right) \right) r dr dz
\][
\[ + \frac{\partial}{\partial z} \left( D_F \frac{\partial \phi_F}{\partial z} \right) + \Sigma_a \phi_F \left( 1 - \beta \right) \left( k_e + \rho - b \ln(T_e/T_F) \right) \]
\[ + \chi \Sigma_a \int_0^t e^{-\lambda_1 (t-t')} (k_e + \rho) \phi_F dt' + \chi C_e e^{-\lambda t} \right) r dr dz = 0 \quad (50)
\]
The second order terms in equation (50) may be reduced to a first order term by application of Green's Theorem or equivalently integration-by-parts (See Appendix B). Dividing through by $2\pi$ and reducing the second order terms yields

$$
\int \frac{rN_1DF}{\partial r} \frac{\partial \Phi}{\partial r} dz + \int \frac{rN_1DF}{\partial z} \frac{\partial \Phi}{\partial z} dr + \int \frac{N_1}{v} \frac{\partial \Phi}{\partial t} rdrdz
$$

$$
+ \int \int \left\{ \frac{\partial N_1}{\partial r} \frac{\partial \Phi}{\partial r} + \frac{\partial N_1}{\partial z} \frac{\partial \Phi}{\partial z} - N_1C\Phi(1-\beta)[k^\omega + \rho - \beta \ln(T_F/T_F^0)]
\right\} rdrdz = 0
$$

From continuity and boundary conditions the line integrals are zero. Now using the approximate functions of equation (49) and noting that $\psi_F^e (\Phi)$ is not a function of space, equation (51) may be written as

$$
\frac{1}{v} \int \int (N_i)^e (N_j)^e rdrdz(\psi_F^e) + \int \int D_F[(N_i,r)^e (N_j,r)^e + (N_i,z)^e (N_j,z)^e]rdrdz(\psi_F^e)
$$

$$
- \int \int (1-\beta)[k^\omega + \beta - \beta \ln(T_F/T_F^0)](N_i)^e (N_j)^e rdrdz(\psi_F^e) - \int \int e^{-\lambda(t-t')} (k^\omega + \rho) dt 
$$

$$
\int \int (N_i)^e (N_j)^e rdrdz(\psi_F^e) - \int \int e^{-\lambda(t-t')} (k^\omega + \rho) dt = 0
$$

(52)
The integrations may not be easily carried out with a shift to local coordinates and with use of the previously derived transformations. Rearranging equation (52) and assuming the properties are constant for each time step gives

\[ D_F \int_{-1}^{1} \int_{-1}^{1} \left[ (N_{1, \xi} J_{11})^* (J_{11})^* N_{j, \xi} + (N_{1, \eta} J_{22})^* (J_{22})^* N_{j, \eta} \right] \text{rdet}[J] d\xi d\eta (\psi_F) \]

\[ -\sigma_{Ff}(1-\beta)(k^* + \rho) \int_{-1}^{1} \int_{-1}^{1} (N_{i}) (N_{j}) r\text{det}[J] d\xi d\eta (\psi_F)^e \]

\[ -\sigma_{Ff}(1-\beta)b \int_{-1}^{1} \int_{-1}^{1} f(t) (N_{i}) (N_{j}) r\text{det}[J] d\xi d\eta (\psi_F)^e \]

\[ -\lambda \beta \sigma_{Ff} f(t) \int_{-1}^{1} \int_{-1}^{1} (N_{i}) (N_{j}) r\text{det}[J] d\xi d\eta (\psi_F)^e -\lambda C_{0} e^{-\lambda t} \int_{-1}^{1} \int_{-1}^{1} (N_{i}) r\text{det}[J] d\xi d\eta \]

\[ + \frac{1}{V} \int_{-1}^{1} \int_{-1}^{1} (N_{i}) (N_{j}) r\text{det}[J] d\xi d\eta (\psi_F)^e = 0 \] (53)

Since the element chosen has eight degrees of freedom (nodal points), the discretized matrices which result from the integration of equation (53) will be 8x8 matrices and the forcing function will be an 8x1 vector at the element level. Defining the matrices as

\[ [H_{12}]_{8 \times 8} = \frac{A_e}{4} \sum_{k=1}^{m^2} \left[ (N_{1, \xi}) (N_{j, \xi}) \right] k J_{11}^* \]

\[ + (N_{1, \eta}) (N_{j, \eta}) k J_{22}^* r_k w_k \] (54)
where \( W_k = W_i W_j \) and \( r_k = \sum_{i=1}^{6} r_i (N_i)_k \).

\[
\int_{-1}^{1} \langle N_i, N_j \rangle r \det[J] d\xi d\eta = [H_{31}]_{8x6} = \frac{4}{\pi} \sum_{k=1}^{m^2} (N_i)_k (N_j)_k r_k w_k \quad (55)
\]

\[
\int_{-1}^{1} \langle N_i \rangle r \det[J] d\xi d\eta = [F_1]_{8x1} = \frac{4}{\pi} \sum_{k=1}^{m^2} (N_i)_k r_k w_k \quad (56)
\]

\[
\int_{-1}^{1} \ln \left( \frac{T_F}{T_D} \right) \langle N_i, N_j \rangle r \det[J] d\xi d\eta = [H_{41}]_{8x8}
\]

\[
= \frac{4}{\pi} \sum_{k=1}^{m^2} \ln \left( \frac{T_F}{T_D} \right)_k (N_i)_k (N_j)_k r_k w_k \quad (57)
\]

To carry out the summation of equation (57), the temperature \( T_F \) must be known; however, this temperature is exactly what is being sought. To alleviate this problem, a linearization is used.

In the solution technique, the temperature is predicted for the next time step. It is this temperature which is used for the determination of matrix \( H_4 \).

Equation (53) simplifies to

\[
\{D_F [H_{12}] - [E_{af}(1-\beta)(k_{\omega} + \rho) + \lambda \delta E_{af}(t)] [H_{31}] + E_{af}(1-\beta)b[H_{41}] \} \{\psi_F \} e^{-\lambda c c_{\xi}^{\psi} (F_1)_e} + \frac{1}{\nu} [H_{31}] \{\psi_F \} = 0 \quad (58)
\]
The function \( f(t) \) is evaluated by summing the values at each time step using the trapezoid rule for numerical integration.

\[
f(t) = e^{-\lambda t} \int_0^t e^{-\lambda t'} [k_o + \rho] dt'
\]  

(59)

Defining

\[
I_i[g(t)] = \frac{1}{2} h_i \{g(t_i) + g(z_{i-1})\}  
\]

(60)

\[
g(t) = e^{\lambda t}(k_o + \rho)  
\]

\[
h_i = \text{time step taken}  
\]

\[
g_o = 0  
\]

The function may be expressed as

\[
f(t) = e^{-\lambda t} \sum_{i=1}^{S} I_i [g(t)]  
\]

(61)

\( S \) - number of time steps

2. **Clad Region**

Following the same procedure as with the fuel equation, the discretized form of equation (11) becomes

\[
[D_c[H_{12}^{ij}]] + E_{ac[H_3^{ij}]}(\psi_c)^e + \frac{1}{v[H_3^{ij}]}(\dot{\psi}_c)^e = 0  
\]

(62)

3. **Coolant Region**

The governing equation for the coolant region, equation (12), may be discretized into the form

\[
[D_{co[H12^{ij}]} + E_{aco[H3^{ij}]}](\psi_{co})^e + \frac{1}{v[H3^{ij}]}(\dot{\psi}_{co})^e = 0  
\]

(63)
Once the governing equations have been discretized at the element level, they are combined into a system of equations at the global level. On the global level the governing equation for neutron transport takes the form of

$$\left[H\right]_{n\times n} \{\psi\}_{n\times 1} + \left[P\right]_{n\times n} \{\psi\}_{n\times 1} + \{F\}_{n\times 1} = 0 \quad (64)$$

where $[H]$, $[P]$, and $[F]$ represent the system matrices and $n$ is the number of nodal points used in the discretization. There are, then, $n$ simultaneous ordinary differential equations used to describe the neutron transport problem.

C. HEAT TRANSPORT FIELD EQUATIONS

The spatial domain for the heat transport problem is discretized in the same manner as the domain for the neutronics problem. The same element matrices previously defined are valid. Let

$$T_k = \langle N_j \rangle \tau_{kj} \quad j = 1, 2, \ldots, 8$$  
$$k = F, C, CO \quad (65)$$

1. Fuel Region

The governing field equation for the fuel, equation (24), is discretized by applying equation (37). Using an integration by part to lower the order of the second order terms allows equation (24) to be written as

$$\int r \left[ N_1 k_F \frac{\partial T_F}{\partial r} \right]_0^z r dr dz + \int r \left[ N_1 k_F \frac{\partial T_F}{\partial r} \right]_0^z r dr dz + \int r \int r \times \{ T_k \left[ \frac{\partial N_1}{\partial t} \frac{\partial T_F}{\partial r} + \frac{\partial N_1}{\partial z} \frac{\partial T_F}{\partial z} \right] \}
\text{rdrdz} = 0 \quad (66)$$
From continuity considerations the line integrals are zero except along the boundaries of a region. It is assumed that no heat is transferred from the cell in the axial direction (boundary condition 5); therefore, the first line integral of equation (66) is zero. In the neutron diffusion problem there was continuity of flux at the interfaces so that the line integrals were zero; however, the heat transfer at the interfaces is affected by the gap and film conductances. The fuel-clad interface condition, equation (29), may be rewritten as

$$-k_F \frac{\partial T_F}{\partial r} = H_{\text{gap}}(T_F - T_C) = -k_C \frac{\partial T_C}{\partial r}$$  (67)

Substituting equation (67) into (66), dividing by minus one and utilizing equation (65) yields

$$\int_{r_0}^{r_F} \int_{z_0}^{z_F} [r H_{\text{gap}}(N_i) e^{N_j} \text{e} | \text{adz}(\tau_F) - \int_{r_0}^{r_F} \int_{z_0}^{z_F} [r H_{\text{gap}}(N_i) e^{N_j} \text{e} | \text{adz}(\tau_C) \text{e}^*$$

$$+ \int_{r_F}^{r} \int_{z_0}^{z_F} (N_i, r) e^{N_j} + (N_i, r) e^{N_j} \text{e} \text{rdrd}z(\tau_F) \text{e}$$

$$- \int_{r_F}^{r} \int_{z_0}^{z_F} e \Sigma_{\text{eff}}(N_i) e^{N_j} \text{e} \text{rdrd}z(\psi_F) \text{e}$$

$$+ \int_{r}^{r_F} \int_{z_0}^{z_F} \rho_F c_F (N_i) e^{N_j} \text{e} \text{rdrd}z \{\delta F\} \text{e} = 0$$  (68)

$$\text{e}^* - \text{element across the interface}$$
Transforming to local coordinates and integrating by Gaussian quadrature yields the same element matrices as given for the neutron flux, equations (54) and (55), except for the line integrals between regions. It should be noted that the line integrals exist only on the interfaces, along which there is a discontinuity of temperature. For the fuel equation the interface corresponds to the local coordinate $\xi=1$. Define

$$\int_{-1}^{1} (N_i, N_j) d\eta = \left[ k_{i,j} \right]_{8 \times 8} = \frac{1}{2} \sum_{k=1}^{m^2} (N_i)_k (N_j)_k h_k$$

(69)

where the $N$'s are evaluated at $\xi=1$. Many of the terms of $K_1$ will be zero since only the nodes on the $\xi=1$ boundary will have shape functions which are non-zero. It is through $K_1$ that the temperatures for each region are coupled together. Equation (68) may now be written as

$$r_a H_{\text{gap}} ([K_1] \{ \tau_F \}) + [K_1] \{ \tau_c \} = -\frac{1}{2} k_{HF} [H_{12}] \{ \tau_F \}$$

(70)

In obtaining this equation, it was assumed that material properties for each nodal point were constant at each time step. Perhaps a better assumption would have been to assume an average value for the properties of each element. The difference should not be significant, and the assumed constant nodal properties were numerically more tractable.
2. Clad Region

Applying equation (37) to the governing field equation for the clad, equation (25) gives the discretized form of the equation. Assuming no heat transfer in the axial direction on the boundaries (Boundary condition 5), equation (25) becomes

\[- \int [N_i r k_c \frac{\partial T_c}{\partial r}]_a^b \, dz + \int \int k_c [\frac{\partial N_i}{\partial r} \frac{\partial T_c}{\partial r} + \frac{\partial N_i}{\partial z} \frac{\partial T_c}{\partial z}] \, r \, dr \, dz\]

\[+ \int \int \rho_c C_{pc} N_i \frac{\partial T_c}{\partial t} \, r \, dr \, dz = 0 \quad (71)\]

In the cladding, there are two interfaces along which the line integral of equation (71) is not zero, along the fuel-clad interface and along the clad-coolant interface. For the fuel-clad interface, equation (61) applies. For the clad-coolant interface, the interface condition, equation (30), may be rewritten as

\[k_c \frac{\partial T_c}{\partial r} = h_{surf} (T_c - T_{co}) = -k_{co} \frac{\partial T_{co}}{\partial r} \quad (72)\]

Along the fuel-clad interface, the local coordinate corresponds to \(\xi = -1\). Define the new element matrix

\[1 \int_{-1}^{1} (N_i)^e (N_j)^e \, d\xi = [K_{1j}]_{8 \times 8} = \frac{1}{2} \sum_{k=1}^{m^2} (N_i)_k (N_j)_k W_k\]

where the \(N\)'s are evaluated at \(\xi = -1\). As with \(K1\), \(K2\) will have many zero values because the shape functions are evaluated at \(\xi = -1\).
Along the clad-coolant interface, the local coordinate corresponds to $\xi=1$ and the $K_I$ matrix is appropriate.

Substituting equations (67) and (72), equation (71) becomes

$$
\int_{z}^{z_a} r_h \text{surf}_{Ni}(T_c-T_{co}) \, dz - \int_{z}^{z_b} r_h \text{gap}_{Ni}(T_F-T_c) \, dz
$$

$$
+ \int \int_{r, z} k_c \frac{\partial T_c}{\partial r} + \frac{\partial T_c}{\partial z} \, r \, dr \, dz
$$

$$
+ \int \int_{r, z} \rho_c \ C_{pc} \ N_i \frac{\partial T_c}{\partial t} \, r \, dr \, dz = 0
$$

The governing equation for the clad region may now be written as

$$
r_a \text{gap} \{[K_2](\tau_c)^e-[K_2](\tau_f)^e\} + r_b \text{surf} \{[K_1](\tau_c)^e-[K_1](\tau_{co})^e\}
$$

$$
+ k_c \{H_{12}(\tau_f)^e + \rho_c \ C_{pc} [H_3](\dot{\tau}_c)\} = 0
$$

The line integrals of equation (73) affect only nodes which are on one of the boundaries; therefore, the nodal inputs into $K_1$ and $K_2$ are zero unless the node is on one of the boundaries.

3. Coolant Region

The field equation governing the coolant may be discretized in the same manner as above. After applying the Galerkin method and performing an integration by parts.
on the second order terms, equation (25) becomes

\[- \int [N_1 k_{co} \frac{\partial T}{\partial r}] \, dz + \int \int k_{co} C_{pco} \left( N_1 \frac{\partial T_{co}}{\partial r} + \frac{\partial N_1}{\partial z} \frac{\partial T_{co}}{\partial z} \right) \, r dz dr \]

\[+ \int \int v_{co} \rho_{co} C_{pco} N_1 \frac{\partial T_{co}}{\partial z} \, r dz dr = 0 \quad (75)\]

The line integral, when evaluated at \( c \), is zero (boundary condition 4). When evaluated at \( b \), or correspondingly at \( \xi = -1 \), equation (72) is valid and \( K_2 \) matrix is appropriate.

All the terms of equation (75) have been defined except the flow term. Define

\[
\int \int (N_1)^e <N_1, n>^e \, r det[J] \, d\xi d\eta = [H5_{ij}]_{8 \times 8}
\]

\[
= A^e m^2 \sum_{k=1}^{N_1, n} (N_1)_k (N_1, n)_k r_k W_k \quad (76)
\]

Transforming to local coordinates and integrating reduces equation (75) to

\[r_{b} h_{surf} [K2][\tau_{co}]^e - [K2][\tau_{c}]^e + k_{co}[H12][\tau_{co}]^e + v_{co} \rho_{co} C_{pco}[H5][\tau_{co}]^e + \rho_{co} C_{pco}[H3][\tau_{co}]^e = 0 \quad (77)\]
Now that the governing equations have been defined for each region on the element level, equations (70), (74), and (77), they may be assembled into a system equation on the global level. The equation will be in the general form of

\[
[K]_{nxn} \{\tau\}_{nx1} + [M]_{nxn} \{\psi\}_{nx1} + [G]_{nxn} \{\xi\}_{nx1} = 0
\]

(78)

D. DISCRETIZATION OF THE SPATIAL DOMAIN

Prior to the numerical solution of the governing equations, equations (64) and (78), the spatial domain must be divided into a number of elements. For this work the domain was subdivided as shown in figure 5.

Since there is a discontinuity of temperatures at the interfaces, as described by equations (29) and (30), a novel application of the FEM method was necessary. The common practice for handling these "flux" type boundary conditions is to define a constant reference temperature, \( T_r \), as when working with a convection heat transfer problem [17], or to define a known function, as when working with a fracture mechanics problem [18]. In either case the reference condition was known. The novel application here lies in the use of a different field equation to describe the reference temperature, e.g., the clad equation (74) is the reference condition for heat transfer from the fuel across the gap interface.
The discontinuity of temperature at the interface necessitated another novel application of the FEM. Since there is a temperature drop along each interface, a single node there is not adequate. In the discretization of the domain, two nodes were used for each interface point (for example, points 62 and 63 in figure 6). This allows the temperature drop due to the gap and film conductances to be taken into consideration. Since two nodes are used, the governing equations for each region are not directly coupled together. The coupling of the regions is accomplished by the "flux" boundary or interface conditions since it is assumed that any heat flux leaving a region enters the adjacent region. Consider a typical set of elements on an interface.
Figure 6. Finite Element Discretization
The coupling terms $K_1$ and $K_2$ may be combined into a system $K$ matrix which shows the coupling. The $K$ matrix for the simple set shown is

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & \cdots & 13 & 14 & 15 & 16 \\
1 & -a & c & -c & c & -c & a & 16/15 \\
2 & -a & c & -c & c & -c & b & 4/15 \\
3 & -c & c & -b & b & -d & d & e & 2/15 \\
4 & -c & c & -b & b & -d & d & f & 1/15 \\
5 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
13 & -c & c & -d & d & b & b & & & & & & & \\
14 & -c & c & -d & d & b & b & & & & & & & \\
15 & & & & & & & & & & & & & \\
16 & & & & & & & & & & & & & \\
\end{bmatrix}
\]

As can be seen, the nodes on the interface are coupled to the adjacent element interface nodes. For example, node 2 in element I is coupled to nodes 3, 8, and 14 in element II.

E. OPTIMUM COMPACTING SCHEME

The system matrices ($K$, $P$, etc.) are $n \times n$ matrices, where $n$ is the number of nodal points used in the discretization of the domain. In terms of computer storage, these matrices may become excessively large if they are stored as $n \times n$. There are several techniques available to reduce this storage requirement. The most common method is the banded storage scheme, whereby only the banded portion of the matrices are stored. With judicious numbering of the nodes,
considerable savings may be realized. However, it is not
the optimum storage scheme [8].

Since the shape functions, \( N_i \), for the \( k^{th} \) nodal equa-
tion are nonzero over only the element containing \( k \), the
system matrices are not only banded but sparse as well.
The sparseness is due to the non-consecutive numbering of
the nodes surrounding the \( k^{th} \) node. The optimum compact
storage (OCS) scheme compacts the matrices by storing only
the non-zero elements of the matrices. The implementation
of the OCS scheme requires two additional integer arrays,
say JA and NAME. The NAME array identifies the nodal points
which contribute to each nodal equation. The JA array acts
as a pointer to indicate where the nodal equation starts in
NAME. Consider the following simple 2x2 system with nodes
as indicated

\[
\begin{array}{ccc}
7 & 8 & 9 \\
11 & 4 & 5 \\
1 & 2 & 3 \\
\end{array}
\]

The NAME array starts with node 1 and identifies the nodes
which contribute to node 1 (i.e., 5, 4, and 2). The NAME
array would, then, give the nodes contributing to node 2
and so forth, so that

\[
\text{NAME} = \{1234 : 1578910 : 111314 : \ldots : J\} \\
\text{JA} = \{1511 \ldots\}
\]
The algorithm to assemble the element matrices into a compact storage vector is straightforward and represents a significant savings in computer storage [8]. The system matrices are stored as a vector rather than a two-dimensional array. For example, the value which would be stored in position (1,5) of the nxn array is stored in position 2 of the system vector.
VI. NUMERICAL SOLUTION

This section contains a brief description of possible solution techniques in addition to the solution technique chosen. Computer subroutines necessary to implement the technique are also described.

A. SELECTION OF METHOD

The numerical solution of the system of implicit ordinary differential equations, equations (64) and (78), may be accomplished by any of a number of different techniques such as Houbolt's method, Crank-Nicolson's method, Gear's method, or implicit Gear's method. It was not the objective of this analysis to determine which of the numerical solution schemes is the most efficient. Each method has its advantages and disadvantages. The Crank-Nicolson method is a single-step, implicit equation solver and, therefore, does not require storage of previous time solutions. When analyzing neutronic problems, the system of equations which arises is commonly very stiff (i.e., a rapid change in flux over a short period of time). The Crank-Nicolson method has, in a past work [8], demonstrated difficulty in tracking these stiff systems. Gear's method was specifically developed for stiff systems and can handle the problem very well. However, Gear's method is a multi-step, predictor-corrector method requiring storage of previous time solutions. In addition to this disadvantage, Gear's method requires the
transformation of the developed implicit O.D.E.'s into an explicit system of O.D.E.'s. After this transformation is done, the system matrices are no longer sparse or banded, thus eliminating the use of the optimum compacting scheme. In an effort to overcome these difficulties, Gear's method was modified, Ref. [9], to treat the implicit system of equations as well as to allow use of the optimum compacting scheme. A previous work, Ref. [8], has shown that the implicit Gear's method is particularly attractive in solving the type problem developed in this analysis. Therefore, the implicit Gear's method is used for the solution of the system of O.D.E.'s arising in this analysis.

No attempt will be made here to give the mathematics involved in developing the implicit Gear's method. Reference [9] may be consulted if details are desired. A listing of the computer program developed will be given in the Computer Program section. In order to utilize the implicit Gear's method, several user supplied subroutines must be developed: 1) DIFFUN, 2) JACMAT, and 3) HUITSL.

B. USER SUPPLIED SUBROUTINES TO IMPLEMENT THE IMPLICIT GEAR'S METHOD

1. DIFFUN

Subroutine DIFFUN evaluates equations (64) and (78) for a given time and for given values of \( \psi, \dot{\psi}, \tau \) and \( \ddot{\tau} \). Since at each nodal point, \( i \), there is a solution for the flux and for the temperature, the solution was set equal to \( D_{YI} \) and \( D_{YII} \), respectively. In addition to having flux and
temperature at each nodal point, there are also three different regions in the domain which have different governing equations. An integer array, ITYPE, was developed to indicate for each nodal point whether it was: 0) a fuel node not in an interface element, 1) a fuel node in an interface element, 2) a cladding node or, 3) a coolant node. Using ITYPE, the computer program is directed to a different section depending upon the type of node being considered. After all the nodes have been considered, boundary conditions are established by changing DYI and DYII for the appropriate boundary nodes. Since in this analysis there is continuity of flux at the interfaces, special considerations must be given to these nodes. At the fuel-clad interface, the value of DYI for the clad node was set to the value of the flux at that node minus the value of the flux at the adjacent node (i.e., \( \text{DYI}_1 = \psi_1 - \psi_{1-1} \)). Similarly, at the clad-coolant interface, the value of DYI for the coolant node was set to the value of the flux at that node minus the value of the flux at the adjacent node. During the solution of the problem, DYI is driven toward zero, which in the limit forces \( \psi_1 \) to equal \( \psi_{1-1} \). This is the desired continuity result.

2. JACMAT

Subroutine JACMAT, evaluates the Jacobian matrix (for Gear's method) at the given time and for the current values of the dependent variables. The Jacobian for an equation of the type,

\[
F(y, \dot{y}, t) = 0
\]  

(79)
may be represented as [19],

\[ J = \left[ \frac{\partial F}{\partial y} - \frac{\alpha_o}{\beta_0 h} \frac{\partial F}{\partial y} \right] \]  \hspace{1cm} (80)

where \( \alpha_o \) and \( \beta_o \) are coefficients from Gear's method and \( h \) is the time step. Using the notation of DIFFUN, let DYI and DYII represent equations (64) and (78), respectively. The Jacobian matrix may, then, be written as (\( J \) is called PW in JACMAT.)

\[ PW = \left[ \frac{\partial DYI}{\partial \psi} - \frac{\alpha_o}{\beta_0 h} \frac{\partial DYI}{\partial \psi} , \frac{\partial DYII}{\partial \psi} - \frac{\alpha_o}{\beta_0 h} \frac{\partial DYII}{\partial \psi} \right] \]  \hspace{1cm} (81)

It is the form of equation (81) which is programmed in JACMAT. As in DIFFUN, the integer array ITYPE is used to indicate the appropriate section of the program to be utilized. The problem boundary conditions must also be accounted for in JACMAT. In DIFFUN, the value of DYI or DYII was set to zero for constant boundary conditions (i.e., zero). This cannot be done in JACMAT since a division by zero would occur. For a constant boundary condition at the \( i \)th node, the value of PW is set to one for the diagonal term and zero for all other terms of the \( i \)th equation.

3. NUITSL

Subroutine NUITSL solves the system of equations for the quasi-Newton iterates. In this analysis the system is solved using a successive over-relaxation (SOR) method. In this work, the optimum amount of over-relaxation was not determined. Since no effort was made to find the optimum,
it was felt a small over-relaxation would be best. The over-relaxation factor of 0.02 was used. For small values of this factor, the SOR method approaches the Gauss-Siedel iteration technique.
VII. PROCEDURE

In this section, the method utilized to obtain a solution is described. The input data necessary to run the developed computer program will be documented.

Prior to initiating a transient overpower excursion, the steady-state conditions for the fuel cell must be known. Since the system of equations which were developed are not specifically designed to obtain a steady-state solution, the initial steady-state conditions must be part of the input data. The initial temperature distribution was obtained from the steady-state conditions given in Ref. [7].

The axial temperature distribution for the fuel center-line, fuel surface, clad, and coolant have been determined for several different fuel life cycles [7]. For this analysis, the beginning of life cycle for channel 10 was used. Although this distribution is somewhat artificial, it should be adequate for this analysis. It is the trends of the results which are considered important. The distribution within the fuel radially is taken to vary as the square of the radial distance; then

\[ T_F(r,z,0) = T_F(0,z,0)(1 - \frac{r^2}{a^2}) + T_F(a,z,0)(\frac{r^2}{a^2}) \]

Within the cladding and the coolant, the initial radial temperature distribution is assumed to be constant.
The initial flux distribution is assumed to be radially constant, a flat flux assumption. In the axial direction the flux is assumed to vary as the shape of the sine function. The maximum flux, the flux at the axial center, is an input parameter. For this analysis, the maximum initial flux was taken to be $10^{14}$ neutron/cm$^2$sec.

To obtain a steady state flux distribution, the value of fission cross section for the fuel is varied. A trial-and-error method is used until a critical fission cross section, $\Sigma_{fF}^c$, which gives a steady flux is obtained.

Once the steady-state conditions have been determined, the excess reactivity may be inserted. This starts the transient overpower excursion.

A. INPUT DATA

The first data card contains: the order of Gauss quadrature, the number of radial elements in the fuel, the number of axial elements, number of nodal points in the radial direction, and the height of the fuel rod. The next cards, one for each radial nodal point, contain the nodal radial distances. The next cards contain the fuel centerline, fuel surface, clad and coolant temperatures. There is one card for each axial node. The next card contains the maximum flux. The next four cards contain the physical parameters listed in Table I. The last input cards contain the time, end time, estimated initial time step, minimum time step, and maximum time step. A sample data deck is shown in figure 7.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel Diffusion Coefficient (DCF)</td>
<td>0.93 [cm]</td>
</tr>
<tr>
<td>Doppler constant (B)</td>
<td>0.006</td>
</tr>
<tr>
<td>Energy release per fission (E)</td>
<td>7.652x10^{-12} [cal/fissions]</td>
</tr>
<tr>
<td>Fraction of delayed neutrons (BETA)</td>
<td>0.0064</td>
</tr>
<tr>
<td>Fraction of delayed neutrons for the ith group (BETAI)</td>
<td>0.0064</td>
</tr>
<tr>
<td>Decay constant for ith delayed neutron group (DCLAMI)</td>
<td>0.0784</td>
</tr>
<tr>
<td>Initial flux for delayed neutrons</td>
<td>1x10^{10} [neutron/cm^2 sec]</td>
</tr>
<tr>
<td>Average number of neutrons released per fission (ANU)</td>
<td>2.44</td>
</tr>
<tr>
<td>Neutron velocity (VEL)</td>
<td>4.8 x10^8 [cm/sec]</td>
</tr>
<tr>
<td>Fuel absorption cross section (SIGAF)</td>
<td>0.088 [cm^{-1}]</td>
</tr>
<tr>
<td>Critical fuel fission cross section (SIGFF)</td>
<td>0.0586875 [cm^{-1}]</td>
</tr>
<tr>
<td>Blanket absorption cross section (SIGAB)</td>
<td>0.0 [cm^{-1}]</td>
</tr>
<tr>
<td>Blanket fission cross section (SIGFB)</td>
<td>0.0 [cm^{-1}]</td>
</tr>
<tr>
<td>Step reactivity input (RHOA)</td>
<td>Variable</td>
</tr>
<tr>
<td>Ramp reactivity input (RHOB)</td>
<td>Variable</td>
</tr>
<tr>
<td>Fuel density (DENF)</td>
<td>10.9 [gm/cm^3]</td>
</tr>
<tr>
<td>Clad diffusion coefficient (DCC)</td>
<td>1.1 [cm]</td>
</tr>
<tr>
<td>Clad specific heat (CPC)</td>
<td>0.12 [cal/gm °C]</td>
</tr>
<tr>
<td>Clad density (DENC)</td>
<td>8.0 [gm/cm^3]</td>
</tr>
<tr>
<td>Clad thermal conductivity (TKC)</td>
<td>0.0526 [cal/cm sec °C]</td>
</tr>
<tr>
<td>Clad absorption cross section (SIGAC)</td>
<td>0.0015 [cm^{-1}]</td>
</tr>
<tr>
<td>Coolant diffusion coefficient (DCCO)</td>
<td>1.55 [cm]</td>
</tr>
<tr>
<td>Coolant absorption cross section (SIGACO)</td>
<td>0.00004 [cm^{-1}]</td>
</tr>
<tr>
<td>Coolant flow velocity (VCO)</td>
<td>396.0 [cm/sec]</td>
</tr>
<tr>
<td>Surface heat transfer coefficient (HSURF)</td>
<td>0.7 [cal/cm^2 sec °C]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>0.6</td>
<td>2.0</td>
</tr>
<tr>
<td>1.5</td>
<td>4.0E-05</td>
</tr>
<tr>
<td>1.1</td>
<td>0.12</td>
</tr>
<tr>
<td>0.98</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ONE DATA CARD PER AXIAL NODE**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1580</td>
<td>755</td>
<td>413</td>
<td>307</td>
<td></td>
</tr>
<tr>
<td>1540</td>
<td>743</td>
<td>407</td>
<td>380</td>
<td></td>
</tr>
</tbody>
</table>

**SAMPLE DATA DECK**

*Figure 7. Sample Data Deck*
VIII. RESULTS

When using the finite element method, one of the first considerations must be given to the convergence of the method. To determine convergence, the results for a given point are compared for different finite element discretizations. As shown in figure 8, the results are comparable but no definite claim of convergence can be made. However, for this work, it was felt that these results were adequate. It was not the object of this analysis to arrive at the "final" result; it was the trends and methods that were of interest. Since the 66-element mesh appears to give a fair approximation of the results, the 66-element mesh was used as the discretized domain.

The next item of consideration was the determination of a neutronic steady-state condition. This proved to be a very time consuming task. The fission cross section for the fuel was varied by a trial-and-error method in an attempt to find the critical cross section which would give a steady state. As may be seen in figure 9, a change in cross section of less than one percent significantly affected the state of the problem. It was felt that the critical value was between the values of 0.05875 and 0.058625. Time did not permit investigation for critical value; therefore, it was assumed that the value for the critical fission cross section was halfway between the values (i.e., \( \Sigma_f^{cr} = 0.0586875 \)).
Figure 6. Convergence of Finite Element Method
Figure 9. Determination of Critical Fission Cross Section
Even if this value is in error, which it most probably is, the net effect would only be a small decrease or increase in the proposed reactivity insertion. The reactivity insertion would, then, be an approximation of the actual reactivity of the problem.

The first test problem considered was a step increase in reactivity of approximately ten dollars. For the uranium dioxide fuel, one dollar of reactivity was taken to be 0.0064. Figure 10 shows the time history of the flux at the center of the fuel rod. Figure 11 gives the corresponding temperature profile. The temperatures were taken at the hottest point of each region at the axial center (i.e., the fuel centerline, clad inside surface, and coolant inside surface). As seen on the fuel temperature time history, the fuel rapidly reaches the fuel melting point. The model developed does not take into consideration melting of the fuel. This melting would tend to decrease the effect of the transient. The problem was allowed to continue despite this inconsistency in the mathematical model. A short time after fuel melting, the inside surface of the clad reaches its melting point. The temperature in the coolant experienced what is felt to be a numerical phenomenon. The coolant temperature decreased prior to the small rise at the end of the transient. Intuitively, this decrease does not seem to be realistic. A similar occurrence was observed while conducting sample tests on the developed computer program. Reference[20] reported the same phenomenon. It is felt that this
Figure 11. Temperature Profile
phenomenon is a quirk of the finite element method. The radial and axial distributions of the neutron flux are presented in figures 12 and 13 for time equal to $7.39 \times 10^{-3}$ seconds. The distributions are, basically, as anticipated. The neutron flux peaks slightly before the axial center. It was expected to peak at the axial center. The radial and axial temperature distributions for the same time are presented in figures 14 and 15. As with the flux, the temperature profiles were, basically, as expected. The fuel temperature peaks slightly below the expected location, most likely in response to the peak in axial fluxes. The coolant unexpectedly drops near the outlet of the fuel rod. The finite element method characteristically has some problems on the boundaries of the domain; this may account for the drop in coolant temperature.

The temperature of the fuel and cladding do not appear to be as closely coupled as anticipated. As seen in figure 14, a significant increase in fuel temperature has resulted in a relatively small clad temperature increase. As noted in figure 11, there appears to be a time lag in temperature response for each region which may account for part of the apparent temperature disagreement. It is felt that the temperatures should be more closely related, which indicates a higher gap heat transfer coefficient should be utilized. Since values of the gap heat transfer coefficient were assumed, it is not unreasonable to believe the values used are too low.
FIGURE 12

RADIAL FLUX

Time = 7.39 x 10^-3
AXIAL CENTER
r = $10$

Figure 12. Radial Flux
Figure 13. Axial Flux Profile

Radial Center

Y_{max} = 739 \times 10^3

\varepsilon = 10
Figure 14. Radial Temperature Profile
Time did not permit investigation of other reactivity insertions. Other reactivity inputs may be investigated by students in the future.

This work does not represent a solution to the very complicated nuclear reactor problem. It does represent an application of a numerical technique which is relatively new to nuclear applications. Methods for implementing the finite element method have been discussed, and a computer code has been developed for the simplistic model considered.
IX. RECOMMENDATIONS

For the model developed, perhaps the most important item to pursue is the critical fission cross section. A better determination of this value is necessary so that the reactivity insertion is more accurately known. Different test cases for the prompt critical and prompt subcritical reactor could then be conducted.

In further developing the model, more consideration should be given the gap heat transfer coefficient. As noted in the results, the value used appears to be too small. Sample problems for different gap heat transfer coefficients would give a better indication of the values to use.

Melting of the fuel during the transient would probably be the next major improvement on the model. With relatively few changes, the model could be adapted to allow melting element by element. This, too, would be an approximation but, still, an improvement to the model. Perhaps at the same time, a simplified model to take into consideration the fuel restructuring could be implemented.

Another improvement would be to consider reactivity feedbacks in addition to the Doppler feedback. Sodium voiding and fuel rod expansion are two of the more important feedback effects to consider.

On the numerical side, probably the most important thing to do would be to run the computer program on the
"H-compiler", which optimizes the program. However, on several runs using the H-compiler, erroneous results were obtained. With sufficient time, this could be corrected to allow use of the H-compiler. The use of the H-compiler results in a savings in computer time. The present program runs on a "G-compiler" and takes excessive amounts of computer time (two to four hours per run).

In addition to this, the optimum over-relaxation factor in the implicit Gear's method could be determined by trial-and-error.

Implementation of these recommendations should enhance the analysis and lead to a more efficient computer code.
APPENDIX A
DEVELOPMENT OF TRANSFORMATIONS

The Jacobian matrix \([J]\) may be written (8) for two dimensions as

\[
[J] = \begin{bmatrix}
\sum_{i=1}^{N} N_i, \xi x_i & \sum_{i=1}^{N} N_i, \xi y_i \\
\sum_{i=1}^{N} N_i, \xi z_i & \sum_{i=1}^{N} N_i, \xi y_i
\end{bmatrix}
\]  

(A1)

For a simple 2x2 matrix \([A]\) the inverse is

\[
[A]^{-1} = \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}
\]

Applying this fact to equation (A1) gives

\[
[J]^{-1} = \frac{1}{\det[J]} \begin{bmatrix}
\sum_{i=1}^{N} N_i, \xi x_i - \sum_{i=1}^{N} N_i, \xi y_i \\
\sum_{i=1}^{N} N_i, \xi z_i & \sum_{i=1}^{N} N_i, \xi y_i
\end{bmatrix}
\]  

(A2)

\[
= \begin{bmatrix}
J_{11} & J_{12} \\
J_{21} & J_{22}
\end{bmatrix}
\]
From matrix algebra

$$\text{det}[J] = \sum_{i=1}^{N} N_1, \xi_i \varepsilon_1 \sum_{l=1}^{N} N_1, \eta_l z_l - \sum_{i=1}^{N} N_1, \eta_i r_l \sum_{l=1}^{N} N_1, \xi z_l \quad (A3)$$

The derivatives of the shape functions may be found from equations (38).

\[
\begin{align*}
N_1, \xi &= \frac{1}{4}(1+n)(2 \xi + n) \\
N_2, \xi &= - \xi (1-n) \\
N_3, \xi &= \frac{1}{4}(1+n)(2 \xi - n) \\
N_4, \xi &= - \frac{1}{2}(1-n^2) \\
N_5, \xi &= \frac{1}{4}(1-n)(2 \xi + n) \\
N_6, \xi &= - \xi (1-n) \\
N_7, \xi &= \frac{1}{4}(1-n)(2 \xi - n) \\
N_8, \xi &= \frac{1}{2}(1-n^2)
\end{align*}
\]

(44)

If one now considers an arbitrary element

\[
\begin{align*}
N_1, \eta &= \frac{1}{4}(1+n)(2 \eta + n) \\
N_2, \eta &= \frac{1}{4}(1+n)(2 \eta + n) \\
N_3, \eta &= \frac{1}{4}(1-n)(2 \eta + n) \\
N_4, \eta &= - \eta (1-n)
\end{align*}
\]

(44)

with the midside nodes exactly at the midpoint (not a necessary criteria for the FEH) and substitutes into equation (A3),
the result is

\[ \text{det}[J] = \frac{(z_3-z_1)(r_3-r_1)}{4} = \frac{A e}{4} \quad (A5) \]

Substituting into equation (A2) and using (A5) will yield

\[ J_{11}^* = \frac{4}{Ae} \left[ \frac{1}{2} (z_3-z_1) \right] = 2/r_3-r_1, \quad (A6) \]
\[ J_{12}^* = 0, \quad (A7) \]
\[ J_{21}^* = 0, \quad (A8) \]

and

\[ J_{22}^* = \frac{4}{Ae} \left[ \frac{1}{2} (r_3-r_1) \right] = 2/z_3-z_1. \quad (A9) \]

The inverse of the Jacobian matrix now becomes

\[
\begin{bmatrix}
2/r_3-r_1 & 0 \\
0 & 2/z_3-z_1
\end{bmatrix}
\quad (A10)
\]

For integration along a line, the transformation used is

\[ dz = \text{det}[J'] d\eta \quad (A11) \]

in this case

\[ \text{det}[J'] = \sum_{i=1}^{N} N_i z_i \quad (A12) \]

Again considering the arbitrary element and substituting (A4) into (A12) will result in

\[ \text{det}[J'] = \frac{z_3-z_1}{2} = \frac{L e}{2} \quad (A13) \]
APPENDIX B
REDUCTION OF SECOND ORDER TERM

The second order term of the governing field equations may be reduced to first order by integration by parts.

Consider, for example,

\[ \int \int N_1 \left[ \frac{1}{r^2} \partial \frac{\partial \psi}{\partial r} + \frac{3}{3} \left( \partial \frac{\partial \psi}{\partial z} \right) \right] \, r \, dr \, dz \]  

which may be expanded as

\[ \int \int \left[ N_1 \partial \frac{\partial \psi}{\partial z} + N_1 \partial \frac{\partial \psi}{\partial r} + N_1 \partial \frac{\partial \psi}{\partial z} + N_1 \partial \frac{\partial \psi}{\partial r} + N_1 \partial \frac{\partial \psi}{\partial z} + N_1 \partial \frac{\partial \psi}{\partial r} \right] \, r \, dr \, dz \]  

Integrating just the second order terms by parts will yield

\[ \int \int N_1 \partial \frac{\partial \psi}{\partial z} \, r \, dr \, dz = \int \int \left[ N_1 \partial \frac{\partial \psi}{\partial r} \right] r \, dr \, dz \]

and

\[ \int \int N_1 \partial \frac{\partial \psi}{\partial z} \, r \, dr \, dz = \int \int \left[ N_1 \partial \frac{\partial \psi}{\partial z} \right] r \, dr \, dz \]

Substituting the results of (B3) and (B4) into equation (B2) will give

\[ \int \int \left[ N_1 \partial \frac{\partial \psi}{\partial z} \right] r \, dr \, dz + \int \int \left[ N_1 \partial \frac{\partial \psi}{\partial z} \right] r \, dr \, dz - \int \int \partial \frac{\partial \psi}{\partial z} \, r \, dr \, dz \]

92
APPENDIX C
LIST OF RELATIONS FOR MATERIAL THERMAL PROPERTIES

A. FUEL (UO₂)

1. Specific Heat, Ref. [19]

\[ C_{pF} = \left[18.45 + 2.431 \times 10^{-3} T - 2.272 \times 10^{-5} T^2 \right]/270.07 \]

[cal/gm °C]

\[ T - °C \]

2. Thermal Conductivity, Ref. [5]

\[ T_{kF} = \left[1 - 2.5(1 - \rho_{TD})\right] \left[\frac{45.1}{T + 35} + 4.79 \times 10^{-13} T^3\right] \times 0.239 \]

[cal/cm sec °C]

\[ \rho_{TD} - \text{percent theoretical density} \]

\[ T - °K \]

B. CLAD (Stainless Steel)

Properties are assumed to be temperature independent, and average values from Ref. [5] were used for the clad properties.

C. COOLANT (Liquid Sodium)

1. Specific Heat, Ref. [5]

\[ C_{pCO} = 0.34574 - 0.79226 \times 10^{-4} T + 0.34085 \times 10^{-7} T^2 \]

[cal/gm °C]

\[ T - °F \]

2. Density, Ref. [5]

\[ \rho_{CO} = [59.566 - 7.9504 \times 10^{-3} T - 0.2872 \times 10^{-6} T^2 \]

\[ + 0.06035 \times 10^{-9} T^3] \times 0.01601 \]

[gm/cm³]

\[ T - °F \]
3. Thermal Conductivity, Ref. [5]

\[ T_{kco} = [54.306 - 1.878 \times 10^{-2} T + 2.0914 \times 10^{-6} T^2] \times 4.134 \times 10^{-3} \text{ [cal/cm sec °C]} \]

\( T = ^\circ F \)

D. SURFACE HEAT-TRANSFER COEFFICIENT

\[ h_{surf} = \frac{T_{kco}}{D_e} [7.0 + 0.025 \left( \frac{D_e V_{co} \rho_{co} C_{pco}}{T_{kco}} \right)^{0.8}] \text{ [BTU/hr ft}^2 \text{ °F]} \]

\( T_{kco} \) - [BTU/hr ft °F]
\( \rho_{co} \) - [lbm/ft³]
\( V_{co} \) - [ft/sec]
\( C_{pco} \) - [BTU/lbm °F]

\( D_e \) - equivalent diameter [ft]
*MAIN PROGRAM CALLS THE VARIOUS SUBROUTINES*

1IFLC17 INTEGER*2 (1-N)
INTEGER*4 KY,NL,N,JSKF
REAL*8 SN,DESN,DXSN,WT,GP,STJ11,STJ22,DETJ,XW,XSN

THE FOLLOWING DIMENSION STATEMENTS MUST BE
CHANGED FOR EACH DIFFERENT PROBLEM CONSIDERED

DIMENSION BIGH1(5525),BIGH3(5525),BIGH4(5525),BIGH5(5525),
1 BIGK(5525),NAME(5525)
DIMENSION BIGF(350),R(350),Z(350),VIT(350),JA(350),JB(350),
1 ITYPE(350),NAME(26,350)
DIMENSION Y(1,700),X(20000),NELGCA(11,66),NSTART(66),IBP(66)

APPROPRIATE DIMENSIONS ARE
BIGH1,BIGH3,BIGH4,BIGH5,BIGK,NAME----------JC
BIGF,R,Z,VIT,JA,JB,ITYPE----------NP
NAME(26,66)----------NP
NELGCA(11,66)----------NEL
NSTART(66)----------NEL

YW AC REQUIRED BY FRANK'S SUBROUTINES
JC IS THE TOTAL NUMBER OF NODES SEEN BY THE ACES
NP IS THE NUMBER OF NODAL POINTS
NEL IS THE NUMBER OF ELEMENTS

ALL THE REST OF THE SUBROUTINES WILL HAVE THEIR DIMENSIONS
CHANGED THROUGH THE CALLING STATEMENTS SO THAT THE ABOVE
DIMENSIONS ARE THE ONLY ONES WHICH REQUIRE CHANGING

DIMENSION DR(13)
DIMENSION H1(8,8),H3(8,8),F(8)
DIMENSION H4(8,8)
DIMENSION GP(5),SN(8,25),DESN(8-25),DXSN(8,25),STJ11(25),XW(5)
DIMENSION STJ22(25),XSN(5,5),WT(25)
DIMENSION NOUN(3)
DIMENSION X1(8,8),XK(8,8)
CCPCNC/CLAC/ CCC,CP/CPC,DENS,TKC,SIGAC,HGAP
CCPCNC/CENNC/ NP,HEC,IC
CCPCNC/COLD/CCO,SIGAC,VO,HSURF
CCPCNC/FUEL/DCF,B,E,BETA,BETAI,CCLAMI,FUXC,AKINF,VEL,
1 SIGAF,SIGFF,SIGAB,SIGFA,DENF
CCPCNC/GP/ GP,MORE
CCPCNC/INP/DR,HEIGHT,NPR,NEZ,NFE
CCPCNC/JACINV/ STJ11,STJ22,DETJ
CCPCNC/KMAT/XK1,XK2
CCPCNC/0MAT/H12, H3, F
CCPCNC/PMAT/H4
CCPCNC/REACT/RHGA,RFOB
CCMCMN/SHAFTON/1, DESN, DXSN
CCMCH/1P3F/EP, 1RHC
CCMCH/WEIGHT/WT
10 FCRMAT (15,F10.5)
11 FCRMAT (*9X,T4KF = 15)
20 FCRMAT (*9X,T4F15.6/
21 FCRMAT (*9X,ORDER OF GAUSS CLAUGHTRE IS 1); 15,1.10X, NUMBER C
1F RACIAL FUEL ELEMENTS IS 15,1/10X, NUMBER OF AXIAL FUEL ELEMENT
2S 15,1/10X, NUMBER OF RADIAL NODE POINTS IS 15,1/1CX,
3 1FUEL ROD HEAT (/ CM) IS F10.5
3C FCRMAT (F10.5)
35 FCRMAT (*9X,RADIAL NODAL CCCPCINTNAT E IS 15,*9X,13FS:4)
4C FCRMAT (*9X,CELLX, 10X,*8X,11X,*E*9X,*BETA*,8X,*BETA*,6X,
1 *CCCA,08X,*NU*,9X,*KINF*)
41 FCRMAT (9X,9SH9:5)
42 FCRMAT (*9X,12X,0FLUX*,8X,*VEL*,8X,*SIGAF*,7X,*SIGFF*,7X,
1 *SIGB*,7X,*SIGFB*,7X,*SIGAC*,6X,*SIGACC*)
43 FCRMAT (*9X,14X,0DCC*,9X,CCPC,8X,CCPC,9X,*TKC*,3X,*CCC*,
1 10X,*VCO*,8X,*HSURF*,8X,*DENF*)
44 FCRMAT (*9X,T4R4 = RHCA + RHBE*T,1/10X,FHCA = T,F10.3,
1 1CX,4-RDB = F10.3)

INPUT THE ORDER OF THE GAUSS QUADRATURE ---- 3,4, CR 5
INPUT THE NUMBER OF RACIAL ELEMENTS IN THE FUEL ---- 1 CR 2
INPUT THE NUMBER OF AXIAL ELEMENTS
INPUT THE NUMBER OF NODE POINTS IN THE RACIAL DIRECTION ---- 2*NFE+7
INPUT THE HEIGHT OF THE FUEL ROD ---- CM
READ (5,10) NORD,NFE,NEZ,NPR,HEIGHT
WRITE (6,25) NORD,NFE,NEZ,NPR,HEIGHT
READ (5,30) (IR(I),I=1,NPR)
WRITE (6,35) (DRI(I),I=1,NPR)
CALL GENERAL (IPT,E,NELCN,NSTART,R,Z)
CALL CFCOMP (IBP,JA,JE,NAME,NAME,NELCN,NSTART)
CALL INIT (IBP,NELCN,R,O,Y)
CALL SHAPE
CC 45 I = 1,JC
BIG=12(I) = 0.0
BIG=3(I) = 0.0
BIG=5(I) = 0.0
BIG=6(I) = 0.0
45 CC CONTINUE
DC = 0.0
BIGF(I) = C.0

SICRE YIT -- THE INITIAL TEMPERATURE DISTRIBUTION
YIT(I) = Y(I,1+NP)
5C CCATINLE
CALL MATH1 (NELCON)
LC 100 MH = 1, NEL
CALL JACOB (MH, NELCON, R, Z)
CALL NATH12 (MM, NELCON, R, Z)
CALL SYPH12 (MM, BIG12, BIGH3, BIGH5, BIGF, JA, JB, NAME, NELCON)
IF(NELCON(N1, NN) .EQ. C) GO TO 10C
CALL SYPH12 (MM, JA, JB, NAME, NELCON, R, Z, BIGK)
10C CCATINUE
C *** INPUT FUEL PHYSICAL CONSTANTS
  REAC(5, 15) DCF, B, E, BETA, BETAI, CCLAMI, FFLUXO, ANU, VEL, SIGAF, SIGFF,
  IFIP = ANU*SIGFF/SIGAF
C *** INPUT CLAD PHYSICAL CONSTANTS
  REAC(5, 15) DCC, CPC, DENC, TKC, SIGAC
C *** INPUT COOLANT PHYSICAL CONSTANTS
  REAC(5, 15) DCCO, SIGAC, VCO, HSRUF
  WRITE (6, 40)
  WRITE (6, 41) DCF, B, E, BETA, BETAI, DCLAMI, ANU, AKINF
  WRITE (6, 42)
  WRITE (6, 41) FFLUXO, VEL, SIGAF, SIGFF, SIGAB, SIGFB, SIGAC, SIGACC
  WRITE (6, 43)
  WRITE (6, 44) DCC, CPC, DENC, TKC, CCCC, VCO, HSRF, GENF
  WRITE (6, 44) RFQA, RFOB
  JSKF = C
  KY = NP*2
  NL = 0
  F = KY
  EFS = 0.01
  REAC(2, 15) T, TEND, L, HMIN, HMAX
  200 CALL SDEG1 (T, TEND, NY, AL, F, JSKF, 6, 1, L, HMIN, HMAX, EFS, N,
  1 BIGH2, BIGH3, BIGH5, BIGF, IBF, IYTP, JA, JB, NAME, NELCON, R, Z, YIT,
  2 BIGH4)
  WRITE (6, 11) JSKF
  SICF
  EAC
SLEROUTINE GINNY (ITYPE,NELCN,ASTART,R,Z)

SLEROUTINE GINNY DEVELOPS THE SYSTEM MESEX IT ALLOWS FOR ONE OR TWO RADIAL ELEMENTS IN THE FUEL AND ONE ELEMENT IN THE CLAD AND COOLANT NFE SETS THE NUMBER OF RADIAL FUEL ELEMENTS THE GENERATOR ALLOWS FOR ANY NUMBER OF AXIAL ELEMENTS DESIRED NUMBER OF AXIAL ELEMENTS SET BY NEZ THE GENERATOR WILL GIVE THE R AND Z DIMENSIONS FOR EACH NCCE IT ALSO DEVELOPS THE CONNECTIVE MATRIX

ITYPE -- AN ARRAY USED TO INDICATE THE TYPE OF NCCE 0 -- FUEL NODE NOT IN AN INTERFACE ELEMENT 1 -- FUEL NODE IN AN INTERFACE ELEMENT 2 -- CLADDING NODE 3 -- COOLANT NODE

NELCN -- THE CONNECTIVE MATRIX R -- RADIAL DIMENSION OF THE IT NODE Z -- AXIAL DIMENSION OF THE IT NODE

IFLIIT INTEGER*2(1-K)
DIMENSION DR(13)
DIMENSION ITYPE(1)
DIMENSION NELCN(11,1)
DIMENSION NSTART(1)
DIMENSION R(11), Z(1)
C(MMCN,CANH, NP, NEL, JC
CC(MCN/INP/DR,HEIGHT,NPR,NEZ,NFE
LL = 0
NF = 0
KK = 1
M = 2*AFE + 1
M* = M + 4
NEL = AFE + 2
AFZ = NEZ*2
ACCE = 3*AFE + 14
EZ = NEZ
Z1 = HEIGHT/(2,0*EZ)
CC 100 I = 1, N$P$Z
T = 1
IF (KK = EQ. 1) GO TO 80
LL = LL + 1
NSTART(1L) = NP + 1
60 CC 70 J = 1, NPR
AF = NP + 1
Z(NP) = Z1*T
R(NP) = CR(J)
JH = 1
GOTO 100
CC 50 JJ = 1,NPR,2
IF (JJ .EQ. MMH) GOTO 50
NF = NF + 1
Z(NP) = Z1*T
R(NP) = CR(JJ)
IF ((JJ .EQ. M) .CR. (JJ .EQ. MM)) GOTO 285
CC TO 9C
5E AF = AP + 1
Z(AP) = Z1*T
R(NP) = CR(JJ)
KK = 0
SC CONTINUE
1C CC CONTINUE
CC 125 J = 1,NPR
AF = NP + 1
Z(AP) = 0.0
R(AP) = DRIJ
1Z5 CC CONTINUE
NAEZ = N82 - 1
IF (NAEZ .EQ. 0) GOTO 151
CC 150 LL = 1,NNEZ
LST = NSTART(LL)
NCX = NST + M + 1
NCF = NOUT - 2
NEDC = NST + NPR - 3
NCLNT = 0
CC 150 NN = NST,NNEZ,2
IF (NN .EQ. NCFA) GOTO 150
NN = NN
NEL = NEL + 1
NCLNT = NACCOUNT + 1
IF (NN .EQ. NOUT) NN = NN - 1
NELCON (1,NEL) = NNN + NADD
NELCON (2,NEL) = NELCON (1,NEL) - 1
NELCON (3,NEL) = NELCON (2,NEL) - 1
NELCON (4,NEL) = NNN + NPR + 1 - ACCOUNT
NELCON (5,NEL) = NNN
NELCON (6,NEL) = NNN + 1
NELCON (7,NEL) = NNN + 2
NELCON (8,NEL) = NELCON (4,NEL) + 1
LCC CONTINUE
L51 LST = NSTART(LL)
NCF = NST + NPR - 3
NCLT = NCLT + M + 1
NCFA = NCFA - 2
II = 0
IF(I = EQ. NOUT) II = 1
IF (II = EQ. NCFA) GC TO 153
A = N + 1
NELCON(I, N) = 1 + 2 - II
NELCON(2, N) = 1 + 1 - II
NELCON(3, N) = 1 - II
NELCON(4, N) = 1 - NFE - 4 - N - II
IF (II = EQ. NENO) II = 2
NELCON(5, N) = NP - NPR + 2*A - 1 + II
NELCON(6, N) = NELCON(5, N) + 1
NELCON(7, N) = NELCON(5, N) + 2
NELCON(8, N) = NELCON(4, N) + 1
153 CONTINUE
II = NFE + 2
DC 155 J = 1, NFE
IF (II = EQ. NFE) GO TO 156
DC 155 J = 1, NEL, N
NELCON(I1, J) = 0
155 CONTINUE
GC TO 159
156 DC 158 J = 1, NEL, N
NELCON(I1, J) = 1
NELCON(I1, J+1) = 2
NELCON(I1, J+2) = 3
156 CONTINUE
155 CONTINUE
GC 160 J = 1, NEL
IF (II = EQ. NEL) GC TO 168
I1 = NELCON(I1, J)
II = NELCON(I1, J)
I1YPE(I1) = JJ
160 CONTINUE
WRITE (6, 15)
DC 160 I = 1, NAP, 3
II = I + 1
IF (II = EQ. NP) GC TO 161
II = I + 2
IF (II = EQ. NP) GO TO 162
161 WRITE (6, 20) I, R(I1), Z(I1), I1, R(I1), Z(I1), I2, F(I2), Z(I2)
GC TO 165
162 WRITE (6, 20) I, R(I1), Z(I1)
GC TO 165
165 WRITE (6, 25)
DC 170 J = 1, AEL
17C WRITE (6,30) (J, (NELCCN(I,J), I = 1,8))
2C FORMAT (10X, 14, 2(3X, F10.4), 9X, 14, 2(3X, F10.4), 9X, 14, 2(3X, F10.4))
2C RETURN
ENC
SUBROUTINE QPCCOMP (IBP, JA, JB, NAME, NAME, NELCCN, NSTART)

QPCCOMP calculates the NAME array, the JA array, and the JB array which are used in the optimum compacting scheme.

IBP -- an array used to store the FUEL-CALC INTERFACE nodes.
JA -- an array which indicates the number of nodes contributing to the ith node.
JB -- the pointer array which indicates where the ith equation begins in NAME.
NAME -- two-dimensional array used to develop NAME.
NELCCN -- connective matrix.
NSTART -- an array used to store centerline nodal points.

IFLICT INTEGER*2(I-N)
DIMENSION DR(13)
DIMENSION IBP(1)
DIMENSION JA(I), JB(1), NAME(1), NAME(26,1)
DIMENSION NELCCN(11,1)
DIMENSION NSTART(11)
COMMON/BCURD/NCCUR
COMMON/POOL/NP, NEL, JC
COMMON/INPDR, HEIGHT, NPR, NEZ, NFE
NELDOF = 0
WRITE (*,10) NP,NEL,NELDOF

10 FORMAT(2X,NUMNP=,15,5X,NUMEL=,15,5X,NELDOF=,15)
DC 42 I = 1, NP
JA(I)=1
4C CONTINUE
DC 6C I = 1, NP
CC 50 J = 1,26
NAME(J,I)=0
5C CONTINUE
6G CONTINUE
DC 65 I=1,NP
NAME(I,I)=1
65 CONTINUE

CC 100 I = 1,NEL
CC 50 J=1,NELDOF
J subscript K = NELCCA(K,1)
CC 80 K=1,NELDOF
IF(K.EQ.1) GC TO 80
K subscript K = NELCCA(K,1)
JA(JJ)=JA(JJ)+1
JA(KK)=JA(KK)+1
CC 7C L=2, JAA
JJA=LNAME(L, JJ)
IF(JJL=EQ.KK) JA(JJ) = JA(JJ) - 1
IF(JJL=EQ.KK) GO TO 80
CC CONTINUE
1CC CONTINUE
5CC CONTINUE
1CC CONTINUE

CC *** TC ACCCOUNT FOR THE INTERFACE CONDITIONS
ACCLNT = 3
IEP(1) = NPE - 6
N= NPE + 1
NLW = 2*NPE
WRITE (A,1000) (NSTART(i), I=1,NEZ)
DC 300 I = 1,NEZ
IEP(NCOUNT) = NSTART(I) + NUM
IEP(NCOUNT) + 1 = IEP(NCOUNT) + N
ACCLNT = NCOUNT + 2
3CC CONTINUE
WRITE (A,1000) (IBP(I), I=1,ACCOUNT)
KK = 1
CC 400 I = 1,ACCOUNT
JJ = IBP(1)
IF (KK - EQ. 0) GO TO 350
IF (I = EQ. ACCOUNT) GO TO 350
IF (I = EQ. 1) GO TO 350
CC 325 J = 1,13
NAME (J+13, JJ) = NAME (J, JJ+1)
NAME (J+13, JJ+3) = NAME (J, JJ+4)
NAME (J+4, JJ+3) = NAME (J, JJ+4)
325 CC CONTINUE
J(JJ) = JA(JJ) + 13
35C CC TO 400
I U * a
W++LU
2:7  .4I
*--C
0. cc
CLCI
u! Li li++I
17 a-)-
**)
U
-.. P
192i
-P1
NM
-r
44--+
14
)VA%
-tn
o-
A
wr, f
266i
m
W :4*
a
J ~-wfC
70Wwa
wc4 -
353x
NIx
412x
% 
000 -,
r
wcm
wm
a
V
*0U
Q'~ZZ
wuJ
w
104
SLERCUINE .INIT (IBP,AELCON,R,Z,Y)

INITIAL SETS THE STARTING VALUES FOR THE TEMPERATURE AND THE NEUTRON FLUX

C
C*** FRACIT INTEGER*2(1-N)
C DIMENSION CR(13)
C DIMENSION IBP(1)
C DIMENSION AELCON(11,1)
C DIMENSION R(1,2(1)
C DIMENSION Y(7,1)
C CC/CON/BOUND/ NCOLNT
C CCP/CELL/CPLAN/ NP, NEL, JC
C CC/CEL/INP/DR, HEIGHT, KPR, NEZ, NFE
C K = 0
C CC 100 N = 1, NCOUNT
C IB = IBP(N)
C IF(K.EQ. 1) GO TO 50
C IEF = IB - 2*NFE + NP
C IECC = IB + 4 + NP
C GC TO 75
C 5C IEC = IB - NFE + NP
C IBCC = IB + 3 + NP
C K = 0
C 75 IEC = IB + 1 + NP
C
C*** INPUT INITIAL TEMPERATURES FOR THE FUEL CENTERLINE, FUEL SLIPPEACE, CLADDING, AND THE COOLANT ALONG THE AXIAL DIRECTION
C
C REAC (5,1000) Y(1,IBF), Y(1,IB+NP), Y(1,IBC), Y(1,IBCC)
C 1CC CONTINUE
C
C*** CALCULATE THE RADIAL TEMPERATURES USING INPLT, TEMPERATURES AT CLADING TEMPERATURES ARE INITIALLY ASSUMED CONSTANT
C*** IN THE RADIAL DIRECTION.
C CC 600 N = 1 NEL
C IT = NELCON(1,R) + 1
C GC TO (200,300,400,400), IT
C 2CC CC 250 K = 1,8
C NCDE = NELCON(K,R)
C IF (K.GT. 3) GO TO 230
C NCDES = NELCON(K,K+1)
C NCDEM = NELCON(3,K)
A2 = R(NODE)**2
R3 = R(NODE)**2
Y(1, NODE+NP) = Y(1, ACCEH+NP)*(1.0-R2/A2) + Y(1, NODES+NP)*R2/A2
23C IF(K.EQ. 4) OR (K.EQ. 8)) GC TC 240
ACCES = NELCON(7, M+1)
ACCEH = NELCON(5, M)
A2 = R(NODE)**2
R3 = R(NODE)**2
Y(1, NODE+NP) = Y(1, NODEH+NP)*(1.0-R2/A2) + Y(1, NODES+NP)*R2/A2
24C ACDES = NELCON(8, M+1)
ACCEH = NELCON(4, M)
A2 = R(NODE)**2
R3 = R(NODE)**2
Y(1, NODE+NP) = Y(1, ACDES+NP)*(1.0-R2/A2) + Y(1, NODES+NP)*R2/A2
25C CONTINUE
GC TO 600
33C KK = 1
IF(NF8.EQ. 1) KK = 0
ACDES = NELCON(2, M)
ACCE = NELCON(1, M)
ACCEH = NELCON(3, M-KK)
R3 = R(NODE)**2
Y(1, NODE+NP) = Y(1, ACDES+NP)*(1.0-R2/A2) + Y(1, NODES+NP)*R2/A2
ACCE = NELCON(6, M)
ACCEH = NELCON(1, M)
Y(1, NODE+NP) = Y(1, NODEH+NP)*(1.0-R2/A2) + Y(1, NODES+NP)*R2/A2
44C NODE2 = NELCON(1, M) + NP
NODE3 = NELCON(2, M) + NP
NODE4 = NELCON(3, M) + NP
NODE5 = NELCON(4, M) + NP
NODE6 = NELCON(5, M) + NP
NODE7 = NELCON(6, M) + NP
NODE8 = NELCON(8, M) + NP
Y(1, NODE2) = Y(1, NODE3)
Y(1, NODE3) = Y(1, NODE4)
Y(1, NODE5) = Y(1, NODE6)
Y(1, NODE7) = Y(1, NODE8)
66C CONTINUE
C** INGL THE MAXIMUM FLUX (I.E. THE FLUX
C** AT THE AXIAL CENTER) RADIAL FLUX WILL BE ASSUMED
C** TC REMAIN CONSTANT (FLAT FLUX ASSUMPTION)
C** AXIALLY THE FLUX ASSUMED TC VARY AS TFE
C** SHAPE OF THE SINE FUNCTION
READ(5,1001) FMAX
   10C1 FFORMAT (F10.5)
       CC 700 P = 1,AP
       X = Z(M)*3.1415926/HEIGHT
       S = SIN(X)
       Y(1,M) = FMAX*S
       7CC CONTINUE
   WRITE (6,1050)
   1050 FFORMAT ('1X,'9X,'NODAL',5X,'INITIAL',3X,'INITIAL',/,'16X,'POINT',5X,
     1 'FLUX',6X,'TEMP',//)
       CC 800 I = 1,AP
       WRITE (6,1100) I, Y(I,I),Y(I,I+NP)
   8CC CONTINUE
   1100 FFORMAT ('9X,'I4,'E13.3,F10.3)
   RETURN
   END
SLERCUTINE SHAPE

SHAPE EVALUATES THE SHAPE FUNCTION AND
ITS DERIVATIVE WITH RESPECT TO XI AND
WITH RESPECT TO ETA AT EACH OF THE GAUSS POINTS.
SLERCUTINE SHAPE ALSO SETS THE WEIGHTS AND LOCATIONS
OF THE GAUSS POINTS DEPENDING UPON THE ORDER
OF GAUSSIAN QUADRATURE USED

INTEGER*4(I-N)
REAL*8 GP, ETA, SN, DESN, DXSN, XI, ETA1, ETA2, XI1, XI2, ETASC, XISC
REAL*8 XW, WT
DIMENSION SN(8,25), DESN(8,25), CXSN(8,25), WT(25), GF(5), XW(5)
COMMON/GP/ GP, NORD
COMMON/SHAUN/SN, CESN, DXSN
COMMON/HEIGHT/ WT
NC = NORD-2
N = 0
CC TO (4,6,8), NO
4 CF(1) = 0.1745566652414834
GF(2) = 0.0
GF(3) = -GF(1)
XW(1) = 0.5/9.0
XW(2) = 0.0/9.0
XW(3) = XW(1)
CC TO 10
6 GF(1) = 0.86136311594653
GF(2) = 0.339981043592856
GF(3) = -GF(2)
GF(4) = -GF(1)
XW(1) = 0.347854845137454
XW(2) = 0.5214515462546
XW(3) = XW(2)
XW(4) = XW(1)
CC TO 1C
8 GF(1) = 0.56179455938664
GF(2) = 0.384665310105683
GF(3) = -GF(2)
GF(4) = -GF(1)
XW(1) = 0.236926885056189
XW(2) = 0.47628676655366
XW(3) = 0.568888888888888
XW(4) = XW(2)
XW(5) = XW(1)
CC 13 I = 1, NORD
CC 13 J = 1, NORD
N = N + 1
I1(N) = XW(I)*XW(J)
N = 0
cc 15 i = 1, naid
eta = gp(i)
cc 15 j = 1, naid
xj = gp(j)
N = N + 1
eta1 = 1.0 + eta
eta2 = 1.0 - eta
xi = 1.0 + xj
eta sq = eta*eta
xisc = xi*xj
SN(1, N) = 0.25*eta1*xi*(xi+eta-1.0)
CSN(1, N) = 0.25*xi*(2.0*eta*xi)
CSA(1, N) = 0.25*eta1*(2.0*xi*eta)
SA(1, N) = 0.5*(1.0-xi*eta)*eta1
CSN(2, N) = SN(2, N)/eta1
CSN(2, N) = -xi*eta1
SN(2, N) = 0.25*xi*eta1*(-xi+eta-1.0)
CSN(3, N) = 0.25*xi2*eta1*(2.0*xi-eta)
SA(4, N) = 0.25*xi2*(1.0-etasq)
CSN(4, N) = -eta*xi2
CSN(4, N) = -SN(4, N)/xi2
SA(5, N) = -0.25*xi2*eta2*(xi+eta+1.0)
CSN(5, N) = -0.25*xi2*(2.0*eta*xi)
CSN(5, N) = -0.25*xi2*eta2*(2.0*xi+eta)
SA(6, N) = 0.5*(1.0-xi*eta)*eta2
CSN(6, N) = -SN(6, N)/eta2
CSN(6, N) = -xi*eta2
SA(7, N) = -0.25*xi1*eta2*(xi-eta-1.0)
CSN(7, N) = -0.25*xi1*(2.0*eta-xi)
CSN(7, N) = -0.25*xi1*eta2*(2.0*xi-eta)
SA(8, N) = 0.5*xi1*(1.0-etasq)
CSN(8, N) = -eta*xi1
CSN(8, N) = -SN(8, N)/xi1
RETURN
EAC
SUBROUTINE MATK1 (NELCCN)

SUBROUTINE MATK1 calculates the 8x8 element k1 and k2 matrices for the elements' scattering of the two interfaces.

INTEGER*2(I-N)
REAL*8 W*, GP, ETA, XSN, K, Z, XW, Y
DIMENSION GP(5), XSN(5,5), WT(25)
DIMENSION NELCCN(11,1)
DIMENSION XK1(8,8), XK2(8,8)
COMMON/GCNS/ NP, RNL, JC
COMMON/GP/ GP, NORD,
COMMON/KMAT/XK1, XK2
COMMON/WEIGHT/ WT
XW = WT(1)**0.5
EC 15 I = 1, NORD
EC 16 J = 1, NORD
XK1(I, J) = 0.0
XK2(I, J) = 0.0
15 CCUTINUE
EC 30 K = 1, NORD
ETA = GP(K)
XSA(I, K) = 0.5*(ETA+1.0)*ETA
XSA(2, K) = 0.5*(ETA-1.0)*ETA
XSN(3, K) = 1.0 - ETA*ETA
20 CCUTINUE
EC 100 I = 1, 3
EC 101 J = 1, 3
Z = 0.0
EC 60 K = 1, NORD
X = XSN(I, K)
Y = XSN(J, K)
Z = Z + X*Y*WT(K)/XW
60 CCUTINUE
XK1(I, J) = Z
10 CCUTINUE
XK1 (1, 7) = XK1 (1, 2)
XK1 (1, 8) = XK1 (1, 2)
XK1 (1, 9) = XK1 (1, 2)
XK1 (1, 10) = XK1 (1, 2)
XK1 (1, 11) = XK1 (1, 2)
XK1 (1, 12) = XK1 (1, 2)
XK1 (1, 13) = XK1 (1, 2)
XK1 (1, 14) = XK1 (1, 2)
120 CCUTINUE
XK2(3, 3) = XK1(3, 3)
XK2(3, 4) = XK1(3, 3)
XK2(3, 5) = XK1(3, 3)
XK2(3, 6) = XK1(3, 3)
XK2(3, 7) = XK1(3, 3)
XK2(3, 8) = XK1(3, 3)
XK2(3, 9) = XK1(3, 3)
XK2(3, 10) = XK1(3, 3)
SLBRUTINE JACOB (M, NELCN, R, Z)

JACOB EVALUATES THE JACOBIAN MATRIX,
ITS INVERSE, AND ITS DETERMINATE
FOR EACH OF THE GAUSS POINTS

* IS NUMBER OF THE ELEMENT UNDER CONSIDERATION

IMPLICIT INTEGER*2(I-N)
REAL*8 CR*12, DETJ, XJ11, XJ22, STJ11, STJ22
REAL*8 GP
DIMENSION GP(5)
DIMENSION NELCN(11, 1)
DIMENSION Ri(1, 251)
DIMENSION STJ(251, STJ22(25))
COMMON/JP, GP, NORD
COMMON/JACINV, STJ11, STJ22, DETJ
COMMON/CCM/ NP, NEL, JC
C CALCULATE THE DETERMINATE OF THE JACOBIAN
L = NELCN(1, M)
K = NELCN(5, M)
CF = R(L, L), R(K, K)
L = NELCN(5, M)
P2 = Z(K) - Z(L)
DETJ = C.25*DR*OZ
XJ11 = 2.0/DR
XJ22 = 2.0/0Z
AC = NORD*NORD
CF 100 I = 1, NO
C CALCULATE THE INVERSE CF THE JACOBIAN MATRIX
STJ11(I) = XJ11
STJ22(I) = XJ22
1CC
RETURN
END
SLEFCUTINE MATH12 (W, NELCON, R, Z)

SLEFCUTINE MATH12 CALCULATES THE 8X8 ELEMENT H12, F3, AND H5 MATRICES AND CALCULATES THE 8X1 VECTOR F
F IS STORED IN THE H4 ARRAY TO SAVE CORE STORAGE

IMPLICIT INTEGER*2(I-N)
REAL*8 SN, DESN, DXSN, STJ11, STJ22, CETJ, WT, X, 2Y, 2Z, 22Z, X, Y, XX, YY, RR
REAL*8 XXX, YYY, FF, RC, GP
DIMENSION H(8,8), H3(8,8), F(8)
DIMENSION F(8,8)
DIMENSION NELCON(11,1)
DIMENSION R(1), Z(1)
DIMENSION SN(8,25), CESN(8,25), CXSN(8,25), H(25)
DIMENSION STJ11(25), STJ22(25)
CHC=MCS/NN, GP, NULC, JC
CHC=MCS/GP, GP, NORD
CHC=MCS/JACIND, STJ11, STJ22, CETJ
CHC=MCS/MATRIX, H12, F3, F
CHC=MCS/WEIGHT, WT
RAC = NORD*NORD
LE = NELCON(11, M)
CC 100 I = 1,8
FF = 0.0
CC 80 J = 1,8
ZT = 0.0
ZY = 0.0
Z = 0.0
Z2 = 0.0
CC 60 K = 1,10
X = DXSN(I,K)*STJ11(K)
Y = DXSN(J,K)*STJ11(K)
XX = CESN(I,K)*STJ22(K)
YY = CESN(J,K)*STJ22(K)
XY = SN(I,K)
YY = SN(J,K)
RR = 0.0
CC 40 L = 1,8
LL = NELCON(L, M)
RC = R(L)
RF = RR + SN(L,K)*RD
40 CERTHUE
ZY = XX + XXX*YY*RR*WT(K)
Z2 = Z2 + XX*YY*RR*WT(K)
Z3Z = Z3Z + XXX*YYY*RR*WT(K)
IF(J .NE. 1) GO TO 60
IF((LE +EQ. 2) .OR. (LE +EQ. 3)) GO TO 60
FF = FF + XXX*RR*WY(K)
CC CONTINUE
*12(I,J) = (ZX + ZZ)*CETJ
*34(I,J) = ZZZ*DETJ
*44(I,J) = ZZPDETJ
F(I) = FF*CETJ
CC CONTINUE
1CC CONTINUE
RETURN
END
SINGLETINE MATH4 (P, W, DECON, A, Z, YIT, Y)
SINGLETINE MATH4 CALCULATES THE 8X8 TEMPERATURE
DEPENDENT ELEMENT H4 MATRIX

IMPLICIT INTEGER*2(I-N)
REAL*8 SN, DESK, DX5K, WT, XXX, YYY, RR, A, TEMF, TAU0
REAL*8 XX, YY, TT, RD, GP
DIMENSION H4(8, 8)
DIMENSION DECON(11, 1)
DIMENSION R(11, 25)
DIMENSION SN(8, 25), DESK(8, 25), CXSN(8, 25), WT(25), CF(5)
DIMENSION YIT(11)
COMMON /CMAN/ NP, NFL, JC
COMMON /GP/ GP, N0259
COMMON /PTRXH4/ H4
COMMON /SNPUN/SN, DESK, DXSA
COMMON /HEIGHT/ WT
N = N0RE*ACRD
I = NELCON(1, M)
J = NELCON(3, M)
K = NELCON(5, M)
CETJ = C-25*(R(I)-R(J))*Z(J)-Z(K)
CC 100 I = 1, 8
CC 40 J = 1, 8
PP = 0. C
CC 60 K = 1, A
YY = SN1(I, K)
YY = SN1(J, K)
PP = 0. C
TEMP = C-0
T15C = C-0
CC 40 L = 1, 8
CC 40 L = 1, 8
LC = NELCON(L, M)
RE = R(LL)
YY = Y11+L+NP
YY = Y11(LL)
RF = RR + SN1(L, K)*RE
TEMP = TEMP + SN1(L, K)*XX
TAUC = TAU0 + SN1(L, K)*YY
40 CONTINUE
IF (TEMP .LE. 1.0) TEMP = 1.0
IF (TAUC .LE. 1.0) TAUC = 1.0
IF (L1 = TEMP/TAU0
IF (TAU0 .GE. 0.0) GC TC 50
WRITE(*, 955) 1, J, M, TEMP, TAU0, RR
955 FCX=AT(5X, 3110, 5X, 3G18.4)
SLEROUTIN: SYSH4 (M, JA, JB, NAME, AECCCN, BIGH4)
SYSH4 DEPENDS ON THE SYSTEM H4 MATRIX
FROM THE ELEMENT H4 MATRICES

BIGH4 -- SYSTEM MATRIX FOR THE TEMPERATURE DEPENDENT MATRIX
NATURAL LCG(1/10) X NI X NJ

IMPLICIT INTEGER*2(I-N)
DIMENSION BIGH4(1)
DIMENSION H4(8,8)
DIMENSION JA(1), JB(1), NAME(1)
DIMENSION AECCCN(11,1)
DIMENSION NCPMC/CCM#/ NP, NEL, JC
DIMENSION PTRXF4/ H4

5C CC 120 K = 1, 8
KK = AECCCN(K, M)
KK = JA(KK)
LL = JD(KK) - 1
CC 100 I = 1, 8
II = AECCCN(I, M)
CC 60 L = I, KKK
LL = LLL + 1
KKM = NAME(LL)
IF (!II, EQ. KKM) GO TO 80
6C CONTINUE
6C BIGH4(LL) = BIGH4(LL) + H4(K, I)
12C CONTINUE
RCRNM
RETURN
END
SUBROUTINE SYSK12 (M,JA,JB,NAME,NELCON,R,Z,BIGK)

SYSK DEVELOPS THE SYSTEM K MATRIX FROM THE ELEMENT MATRICES

BIGK -- SYSTEM MATRIX FOR THE THERMAL INTERFACE CONDITIONS

INTEGER*2(NI*N)
DIMENSION BIGK(1)
DIMENSION JA(1), JB(1), NAME(1)
DIMENSION NDAU(1)
DIMENSION NELCON(1,1)
DIMENSION RX(1), Z(1)
DIMENSION X1(8,8), X2(8,8)
CALL CCAN/MM/ MCAN/ NP, NEL, JC
CALL KL/M1/KMAT/X1, X2
NELC(1) = 1
NELC(2) = 7
NELC(3) = 8
M = NELCON(11,1)
IG TO (1,1,2) 2N
1 CC 120 K = 1,3
K = NDAU(K)
KK = NELCON(N,1)
KKK = JA(KK)/2
LLL = JB(KK) - 1
DC 100 I = 1,3
DI = NDAU(I)
II = NELCON(N,N)
JJ = II + 1
J = 0
II = C
CC 80 L " 1, KKK
LL = LLL + L
LS = LL + KKK
KKK = NAME(L)
IF(JJ .NE. KKK) GC TO 60
66 GC BIGK(LL) = BIGK(LL) + X1(N,NN)
J = 1
CC 6C IF (JJ .NE. KKS) GC TO 3C
BIGK(LS) = BIGK(LS) - X1(N,NN)
N2 = 1
70 CC IF (N1 .EQ. 1) .AND. (N2 .EQ. 1)) GO TO 100
80 CC CONTINUE
100 CC CONTINUE
120 CC IF (MM .EQ. 1) GO TO 4
2 IF \( E = \text{NELCCN}(3, M) \)
   CC 220 \( K = 3.5 \)
   KK = \( \text{NELCCN}(K, M) \)
   KKK = \( \text{JA}(KK)/2 \)
   LLL = \( \text{JB}(KK) - 1 \)
   CC 200 \( L = 3.5 \)
   II = \( \text{NELCCN}(I, M) \)
   JJ = II + 1
   N2 = 0
   CC 180 \( L = 1, KKK \)
   LL = LLL + L
   LS = LL + KKK
   KK' = \( \text{KAME}(LL) \)
   KKS = \( \text{KAME}(LS) \)
   IF(II .NE. KKM) GC TO 160
   BIGK(LL) = BIGK(LL) + XKZ(K, I)
   N1 = 1
160 IF(II .NE. KKS) GC TO 170
   BIGK(LL) = BIGK(LL) + XKZ(K, I)
   N2 = 1
170 IF((N1 .EQ. 1) .AND. (N2 .EQ. 1)) GC TO 200
180 CC CONTINUE
200 CC CONTINUE
220 CC CONTINUE
4 RETURN
SINGLETON DIFFUSIONS YL,T,HI,A,DY,IGH12,BIG+3,BIG+4,BIG+5,BIGK,
1 EIGF,IPE,ITYPE,J,JA,JB,NAME,NELCC,N,L,Z,YI)

SINGLETON DIFFUSIONS FORMS THE NODAL EQUATIONS:
IT SETS THE ith EQUATION FOR FLUX EQUAL TO XI
AND FOR TEMPERATURE TO DYII.

IMPLICIT INTEGER*2(I-N)
REAL*8 CS,DS
DIMENSION BIGH12(1), BIGH3(1), BIGH5(1)
DIMENSION BIGH4(1)
DIMENSION BIGF(1)
DIMENSION YII(1)
DIMENSION DR(I3)
DIMENSION IBP(1), ITYPE(1)
DIMENSION JAV(1), JB(1), NAME(1)
DIMENSION NELCON(NL,N)
DIMENSION R(1,1)
DIMENSION Y(1,1)
COMMON/CLAD/ DCC,CPC,DEAC,TKC,SIGAC,HGAP
COMMON/CONN/ NP, NEL, NC
COMMON/CCELL/ECC,CAC,AC,ACF,SURF
COMMON/FULL/DCF, E, BETA, BETAI, DCLAM, FFLUXC, AKINF, VEL,
1 SIGAF, SIGSF, SIGAB, SIGFB, DENF
COMMON/INP/DA,HEIGHT,APR,NE2,NFE
COMMON/TIME/ F, G, RHO
DATA T100/-1.0/7
C++ CALCALE THE TIME DEPENDENT TERMS
IF(T.EQ. T100) GO TO 75
CALL FNC(T,T100)
C++ CALCULATE THE SYSTEM H4 MATRIX
C++ H4F IS TEMPERATURE DEPENDENT
CC 30 I = 1,JC
3C BIGH4(1) = 0.0
3C BIGH4(I) = 0.0, I.EQ. NEL
IF(NELCC111) I 1 GT 1) GO TO 50
CALL MAT4 (1, NELCC,R, Z, YII)
CALL SYSEX (1, J, JA, JE, NAME, NELCC, BIGH4)
5C CONTINUE
7C continue
77 T1 = 1/0/VEL 
Z1 = Z(IBP(I)) - Z(IBP(1))
E1 = 1.0 - BETA
SIGAF = SIGSF
SIGSF = SIGAF
CCAST1 = AKINF*B1 - 1.0
CCAST2 = SIGAF*BL1*RH
TC = 0.98
FPCC = 1.0/(0.55*EIGHT)
CC = 510 I = 1, NP
JEP = JAE(J)
JEB = JBI1)
JAP = JAA + JB - 1
IT = ITYPE(J) + 1
II = AP + 1
DI = 0.0
DI = 0.0
GC TO (100, 200, 300, 400), IT
1CC CC = 150 J = JBB, JAB
AP = AAPE(J)
AAA = NA + AP
DY1 = DY1 + (DCF*DBLE(BIGH2(J))-((CONST+F)*SIGAF*CCAST2)*
1CELE(BIGH3(J))+B*SIGAF*BL1*DBLE(BIGH4(J)))*Y(1,NN)-G*DBLE(BIGF(NN))
2 + VINV*HINV*DBLE(BIGH3(J))*Y(2,NN)
TCK = Y(1,NN) + 273.
TKF=((1.0-2.5*(1.0-TC))*(45.1/(135.*TDK) + 4.79E-13*TDK**3))*.235
CCFF = (18.45 + 2.431E-3*TDK - 2.272E5/TDK**2)/270.07
CY11 = CY11 +TKF*DBLE(BIGH2(J))*Y(1,NN) + FINV*DEAF*CPF*
1 CBLE(BIGH3(J))*Y(2,NN)-G*SIGF*CELE(BIGH3(J))*Y(1,NN)
15C CC = 500
C DETERMINE THE TYPE CF NCDE BEING CONSIDERED
C K = 0 NCDE NOT ON THE FUEL-CLAC INTERFACE
C K = 1 NCDE ON THE FUEL-CLAC INTERFACE
2CC K = 0
DC 210 IB = 1, NCCTAM
IF (IBP(I) .EQ. I) GO TO 215
21C CC = 220
GC TC = 220
21E K = 1
22C CC = 250 J = JBB, JAB
AA = AAPE(J)
AAA = NA + AP
DY1 = CY1 + (DCF*DBLE(BIGH2(J))-((CONST+F)*SIGAF*CCAST2)*
1CELE(BIGH3(J))+B*SIGAF*BL1*DBLE(BIGH4(J)))*Y(1,NN)-G*DBLE(BIGF(NN))
2 + VINV*HINV*DBLE(BIGH3(J))*Y(2,NN)
RA = R(NN)
TCK = Y(1,NN) + 273.
TKF=((1.0-2.5*(1.0-TC))*(45.1/(135.*TDK) + 4.79E-13*TDK**3))*.235
CCFF = (18.45 + 2.431E-3*TDK - 2.272E5/TDK**2)/270.07
F CAP = (1000.0 + 247.0*COS(3.14155*(ZNNN) + FPCD-0.81818)) + 1.356E-4
CCAST3 = F CAP*21
24C CYII = CYII + (TKC*DBLE(BIGH12(JJ)) + RA*CONST3*DBLE(BICK(JJ)))*
1 Y(1,NNN) + DENF*CPF + IANV*DBLE(BIGH3(JJ))*Y(2,NNN)
2 * E55*DBLE(BIGH3(JJ))*Y(1,NN)
IF(K - EC = 0) GO TO 250
J = J + JAA/2
AP = NAME(JJ)
DVI = CYI + (DCC*DBLE(BIGH12(JJ)) + SIGAC*DBLE(BIGH3(JJ)))*Y(1,NN)
1 + WINV*HINV*DBLE(BIGH3(JJ))*Y(2,AP)
25C CONTINUE
EC TC 5CO

C
DETERMINE THE TYPE OF NODE BEING CONSIDERED
K = 1 NODE NOT ON AN INTERFACE
K = 2 NODE ON THE FUEL-CLAD INTERFACE
K = 3 NODE ON THE CLAD-COOLANT INTERFACE

3CC K = 1
ACC = C
DC 320 IB = 1, NCCLAT
LEF = IFP(IB) + 1
IF(LDP - EQ. 1) GO TO 325
APF = LDP + 2
IF(NOCC = EC. C) GO TO 305
LEF = LDP - 1
ACC = C
EC TO 310
3CS ACC = 1
31C IF(AFP - EQ. 1) GG TC 33C
32C CONTINUE
EC TC 340
32S K = 2
EC TC 340
33C K = 3
34C DC 380 J = JBB, JAB
AT = NAPE(J)
AAA = AA + AP
IF(K - EC = 2) GO TC 360
CYI = DVI + (DCC*DBLE(BIGH12(JJ)) + SIGAC*DBLE(BIGH3(JJ)))*Y(2,NN)
1 + WINV*HINV*DBLE(BIGH3(JJ))*Y(2,AP)
36C *AP = (1000.0 + 247.0*COS(3.1415*2(2)(NNN) + FNCDC-0.3187))*1.356E-4
F = FGAF
IF(K - EC = 3)h = +SURF
RA = R(AM)
CYII = CYII + (TKC*DBLE(BIGH12(JJ)) + RA + 2)*CDBLE(BICK(JJ)))*
1 Y(1,NNN) + DENF*CPF + IANV*DBLE(BIGH3(JJ)) + HINV*Y(2,NNN)
IF(K - NC = 3) GO TO 380
J = J + JAA/2
AP = NAME(JJ)
C**
AFF = AF + 1
AP = NPR*2
AP = NPF - NN
CC 600 I = NN,NP
CY(I) = 0.0
6CO CONTINUE
C**
TC ACCOUNT FOR THE TEMP BOUNDARY CONDITIONS
C (CONSTANT COOLANT INLET TEMPERATURES)
C**
NAP = AP - 2
CC 700 I = NN,NP
CY(I+NP) = 0.0
7CC CONTINUE
RETURN
END
SLE ROUTINE FUNCT(+IAV)

C*** CALCULATES TIME DEPENDENT FUNCTIONAL
C*** FOR THIS ANALYSIS DELAYED NEUTRON GROUPS
C*** TAKEN TO BE ONE AVERAGED GROUP
C
CC CMN/FUEL/DCF,B,E,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BETA,BET
SIBROULTINE JACMAT (Y,VL,T,HINV,A2,NX,EPS,EY,F1,PW,BIGH2,BIGH3,
1 EIGH,BIGH5,BIGK,IBP,ITYPE,JA,JE,NAME,NELCNC,R,Z)

SIBROULTINE JACMAT CALCULATES THE JACOBIAN MATRIX.
FLUX JACOBIAN STORED IN THE FIRST JC ELEMENTS OF THE
PW ARRAY AND THE TEMPERATURE JACOBIAN STORED IN
THE SECOND JC ELEMENTS OF THE PW ARRAY

IMPLIED INTEGER*2(I-N)
INTEGER*4 N,NY
DIMENSION BIGH12(1), BIGH3(1), BIGH5(1)
DIMENSION BIGH(1)
DIMENSION DR(13)
DIMENSION IBP(1), ITYPE(1)
DIMENSION JA(1), JB(1), NAME(1)
DIMENSION NELCON(11,1)
DIMENSION R(1), Z(1)
DIMENSION Y(7,1), DY(1), PW(1)
CCMCH/BCUNA/MCGUA1
CCMCH/CLAD/DCC,CPDENC,TKC,SIGAC,HGAP
CCMCH/CEAN/NP,NELCNC
CCMCH/CCOL/DCCO,SIGACC,VCO,SURF
CCMCH/FUEL/CCF,EBETA,BETAI,DCLAMI,FLFUS
1 SIGAF, SIGFFI, SIGAB, SIGFII, DEAF
CCMCH/INP/DR,HEIGHT,NPR,NRE,NFE
CCMCH/TIME/ F, G,RHC
AF = -A2*HINV
Z1 = Z(1BP(2)) - Z(1BP(1))
VIN = 1.0/VEL
B1 = 1.0 - BETA
SIGFF = SIGFFI
SIGAF = SIGAF1
Casts1 = #KINF*B1 - 1.0
Casts2 = SIGAF*B1*RHC
TC = 0.58
FPCO = 1.0*(0.55*EIGHT)
CC 500 1 = 1.0
JAA = JA(1)
JBB = JAA + JBB - 1
- A2 = JAA/2
JAE = JA2 + JBB
IT = ITYPE(1) + 1
10C CC 15G J = JBB + JAB
AAA = NAME(J) + NP
PW(J) = DCF*BIGH2(J) - (CONST1+F)*SIGAF*CONST2 - AH*VIN1 + BIGH3(J)
1. $B*\text{SIGAF}\*B1*\text{BIGH}4(4J)$

$TCK = Y(1*\text{NNN}) + 273$.  
$TKF = ((1.0-2.5*(1.0-TD))\*(45.1/(135.*TDK) + 4.79E-13*TCK**3))\*C.235$  
$CFF = (1.45 + 2.431E-3*TDK - 2.2725/TKD**2)/270.07$  
$PM(JC) = TKF*BIGH12(J) + DENF*CFF*AH*BIGH3(J) + RA*CCNST3*BIGK(J)$

15C \text{\textbf{CONTINUE}}  
GC TC 500

3C \text{\textbf{CONTINUE}}  
GC TC 500

2C \text{\textbf{CONTINUE}}  
GC TC 210 1B = 1, NCCOUNT

21C \text{\textbf{CONTINUE}}  
GC TC 220

21C \text{\textbf{CONTINUE}}  
GC TC 250 1B = JBA,JAB

1. $B*\text{SIGAF}\*B1*\text{BIGH}4(4J)$

$N = M(4J)$  
$N = M(4J) + NF$

19C \text{\textbf{CONTINUE}}  
GC TO 500

3CC \text{\textbf{CONTINUE}}  
GC TO 500

3CC \text{\textbf{CONTINUE}}  
GC TO 500

3CC \text{\textbf{CONTINUE}}  
GC TO 500

3CC \text{\textbf{CONTINUE}}  
GC TO 500
IF (LEP .EQ. 1) GC TC 325
LEP = LEP + 2
IF (NCED .EQ. 0) GC TO 305
NCED = NCED - 1
NCED = C
GC TO 310
3C IF (NCED .EQ. 1) GO TO 330
3C CC Continue
GC TO 340
32C K = 2
GC TO 340
32C K = 3
34C CC 380 J = JBE,JAB
JAN = JANE(J) + NP
RA = R(JAN)
PB(J) = 0.0
IF (K .EQ. 2) GC TO 360
PB(J) = DCC*BIGH12(J) + (SIGAC + AH*VINV)*BIGH3(J)
36C LEP = (1000.0 + 247.0*GOS(3.14159*(ZNN) + MCC*C.8181))*.1356E-4
IF (K .EQ. 3) H = HSURF
PB(J+JC) = TKC*BIGH12(J)+RA*H*Z1*BIGK(J)+AH*CEIC*CPE*BIGH3(J)
IF (K .EQ. 2) GO TO 380
IF J .LT. JAB + J - JAB + 1
PB(J) = DCC*BIGH12(JJ) + (SIGACCC + AH*VINV)*BIGH3(JJ)
1F (J .NE. JAB) GC TO 380
PB(JBB) = PW(JBB) + PW(J)
PB(J) = 0.0
3C Continue
3C IF (K .NE. 2) GC TO 500
PB(JBB) = 1.0
PB(JAB3) = -1.0
GC TO 500
C
Determine the type of NCDE being considered
K = 1 Node not on clad-coolant interface
K = 2 NCDE on the clad-coolant interface
4C K = 1
NCED = C
DC 42C IB = 1,NCGLMT
LEP = LEP(1B) + 4
IF (NCED .EQ. 0) GC TO 410
LEP = LEP = 1
NCED = C
CC TO 415
410 NCCE = 1
41E IF(LBP = "EC. I") GG TO 425
42C CCNTINUE
GG TO 430
42E K = 2
42C CC 48C J = JBB,JAB
AA = NAME(J)
AAA = NAME(J) + N +
PM(J) = 0.
1F(K > EC. 2) GO TO 450
1F(J) = CCC*BIGH12(J) + (SIGACG + AS*VIAV)*BIGH3(J)
45C BB = R(AH)
TF = 1.8*Y1+NNN + 32.0
TKCO = (54.306 - 1.878E-2*TDF + 2.0914E-6*TDF**2)*4.134E-3
CFC = C.34574 - 0.7922E-4*TDF + C.34086E-7*TDF**2
CEKCC = (56.566 - 7.9564E-3*TDF - 0.2872E-6*TDF**2 +
1 C.GGC35E-9*TDF**3)*0.01661
A = DECC0*CPCO
1F(J+JC) = TKCO*BIGH12(J) + VCO*A*BIGH5(J) + A*AP3*BIGH3(J)
1 + RR*HSURF*Z1*BIGK(J)
46E CONTINUE
1F(K > H. 2) GO TO 500
1F(JBB) = 1.0
1F(JAB) = -1.0
50C CONTINUE

TC ACCOUNT FOR THE FLUX AXIAL BOUNDARY CONDITIONS

AFF = NP + 1
AA = NPS*C
AP = NPS*N
CC (6CC I = NN,NP
PA(JB(J)) = 1.0
JS = JB(J) + 1
JE = JBJ(J) + JAI(J) - 1
CC 600 J = JS*JE
FB(J) = 0.0
60C CONTINUE

TC ACCOUNT FOR THE TEMP BOUNDARY CONDITIONS
(CCASTART COOLANT INLET TEMPERATURES)

AA = KP - 2
CC 70C 1 = NN, NP
FB(JB(I)+JC) = 1.0
JS(J) = JB(I) + 1 + JC
EC 70C J = JS, JE
FL(J) = 0.0
700 CONTINUE
RETURN
END
SLEROUTINE MUITSL (PW, CY, F1, A, NY, EPS, YMAX, NEWPW, KRET, BIGH3,
1 IYPE, JA, JB, NAME, NELCCN)

SLEROUTINE MUITSL CALCULATES THE NEWTON-ITERATES
USING A *SOR* TECHNIQUE WITH AN CV-RRELAXATION
FACTOR EQUAL TO OMEGA1

**EXTERNAL**
C

**DIMENSION**
C

**DIMENSION**
C

**DIMENSION**
C

**DIMENSION**
C

CCPC/CCCN/ NP, NEL, JC
CCPC/FC/CCCN, E, BETA, BETA1, CCAM, FFLUXC, AKINF, VEL,
1 SIGA, SIGFF, SIGAB, SIGB, OENF
(CAT CMEO, OMEGAM1/1.02, 0.02)
TEST = 1.0E5
KRE1 = C
EPSS = EPS**2
EPSA2 = EPSS*0.0001
MCIT = I
EI = EPS**2
CC 100 I = 1
F1(I) = DY(I)/PW(JB(I))
100 F1(I)+NP = CV(I+AP)/PW(JB(I)+JC)
CC 300 I = 1, NGIT
CRF = 0.
CF = 0.
CRF-1 = C.
CRF-2 = C.
CRF-3 = C.
CC 200 I = 1, NP
JS = JB(I) + 1
JE = JB(I+1) - 1
IT = IYPE(I)
FA = CV(I)
TA = DY(I+AP)
CC 150 J = JS-1, JE
FA = PA - DBLE(PW(J))*F1(NAME(J))
TA = TN - DBLE(PW(J+JC))*F1(NAME(J+NP)
1F11T .GT. 1) GO TO 150
TA = TN + EI*DBLE(BIGH3(J))*F1(NAME(J))
150 CCNITINUE
FA = FN/PW(JS-1)
TA = TN/PI(JS+JC-1)
FA = FN*MEG - F1(I)*OMEGM1
TA = TN*MEG - F1(I+1)*OMEGM1
PCH = F1(I) - PN
TCH = F1(I+NP) - TA
IF((ABS(PCH).GT.TEST).OR.(ABS(TCH).GT.TEST)) GC TC 18C
CF1 = CF1 + (PCH/YMAX(I))**2
CF2 = CH2 + (TCH/YMAX(I+NP))**2
RCF1 = RCH1 + (PCH/AMAX1(ABS(SNL(CN))); EPS))**2
RCF2 = RCH2 + (TCH/AMAX1(ABS(SNL(TN)); EPS))**2
GC TO 150
18C CF1 = 1.0
CF2 = 1.0
RCF1 = 1.0
RCF2 = 1.0
15C F1(I) = PK
2CC F1(I+NP) = TN
CF = CH1 + CH2
RCH = RCH1 + RCH2
IF(RCH = LT EPS1) RETURN
IF(CH = LT EPSA2) RETURN
2CC CONTINUE
KRET = 1
RETURN
END
SLEQTUINE LDASUB (Y, YL, T, TEND, N, HY, M, JSTART, KFLAG, MAXOR, IFRT, I, P)

1) PIR, HMAX, RMSEPS, SAVE, YLSV, YMAX, ER, ESV, P
2) BIGP, BIGS, BIG, IFP, IITYPE, J, JB, NAME, NELCC, JRZ, VII

SLEQTUINE LDASUB IS A MODIFICATION OF SLEQTUINE CFASU
WHICH IS DUE TO R. L. BROWN AND C. W. GEAR. CFASU IS DOCUMENTED
IN THE REPORT

DOCUMENTATION FOR CFASU--
BY R. L. BROWN AND C. W. GEAR
REPORT UIUCDCS-R-73-575, JULY 1973
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801


THE MODIFICATION HERE IS DOCUMENTED IN THE REPORT
A PROGRAM FOR THE NUMERICAL SOLUTION OF LARGE SPARSE SYSTEMS OF ALGEBRAIC AND IMPILCITY DEFINED STIFF DIFFERENTIAL EQUATIONS
BY RICHARD FRANKE
REPORT NPS3FE76051, MAY 1976
NAVAL POSTGRADUATE SCHOOL
MCATREY, CALIFORNIA 93940

CALL LDASUB(Y, YL, T, TEND, N, HY, M, JSTART, KFLAG, MAXOR, IFRT, I, P)
1) PIR, HMAX, RMSEPS, SAVE, YLSV, YMAX, ER, ESV, P1, DY, PHI

WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS:

Y = ARRAY DIMENSIONED (I, NY); IF I IS ARRAY CONTAINS THE DEPENDENT VARIABLES AND THEIR SCALED DERIVATIVES.
Y(I,J+1) CONTAINS THE J-TH DERIVATIVE OF THE I-TH VARIABLE.
TABLE TIMES H*(I+1)-FACOTORIAL, WHERE F IS THE CURRENT STEP SIZE. ON FIRST ENTRY THE CALLER SUPPLIES THE INITIAL VALUES OF EACH VARIABLE IN Y(I,1) AND AN ESTIMATE OF THE INITIAL VALUES OF THE DERIVATIVES IN Y(2,1). ON SUBSEQUENT ENTRIES IT IS ASSUMED THAT THE ARRAY HAS NOT BEEN CHANGED. TO INTERPOLATE TO NON-MESH POINTS, THESE VALUES CAN BE USED AS FOLLOWS: IF H IS THE CURRENT STEP SIZE AND VALUES AT TIME T*E NEEDED, LET S = E/T AND THEN

I-TH VARIABLE AT T*E IS SUM Y(J+1, I)*S**J
J=0

NC  LEAD 460  LCA 470
THE VALUE OF NO IS OBTAINED IN THE CALLING PROGRAM BY NC = JSTART.

YL  - ARRAY OF NL = A - NY VARIABLES WHICH APPEAR LINEARLY. LCA 520
THE USER SUPPLIES INITIAL VALUES FOR THESE VARIABLES. LDA 530
W  - CURRENT VALUE OF THE INDEPENDENT VARIABLE (TIME) LDA 540
NC  - END TIME LDA 550
NY  - TOTAL NUMBER OF VARIABLES LDA 560
W  - NUMBER OF EQUATIONS AND ACALINEAR VARIABLES. LDA 570
W  - NUMBER OF VARIABLES INCLUDED IN THE ERROR TEST. LDA 580
THIS NUMBER CAN BE NC GREATER THAN NY. IF IT IS GREATER THAN NY, NY VARIABLES ARE USE IN THE ERROR TEST. LDA 590
JSTART - INPUT AND OUTPUT INDICATOR. LDA 600
ON INPUT JSTART HAS THE FOLLOWING MEANINGS.
<0  THIS INDICATES A RE-START FROM A PREVIOUS POINT FOLLOWING TERMINATION OF THE RUN OR SOLUTION OF ANOTHER PROBLEM USING THE SAME PARAMETERS IN THE CALLING SEQUENCE. LDA 610
0   MUS HAVE BEEN PRESERVED FROM THE PREVIOUS USE, PARTICULARLY THE ARRAYS LDA 620
SAVE, TENV, ESL, AND PNS. LDA 630
THES ARRAYS MUST BE SAVED AFTER A CALL TO SUBROUTINE LDASV, WHICH ALSO SAVES LDA 640
NECESSARY PARAMETERS INTERNAL TO LEASTSQ. LDA 650
=0  INDICATES AN INITIAL CALL TO LEASUQ. THE JSTART Initializes ITSELF, SCALES THE DERIVATIVES IN Y(2)41 AND THEN PERFORMS THE INTEGRATION UNTIL 1 > TEND. LDA 660
>0  INDICATES THE SOLUTION IS TO BE CONTINUED. LDA 670
AFTER THE INITIAL ENTRY IT IS NEITHER DESIRABLE OR NECESSARY TO RE-ENTER WITH JSTART = 0, SINCE THIS RE-INITIALIZES THE CODE, BEGINNING WITH A FIRST ORDER METHOD AGAIN. LDA 680
ON OUTPUT JSTART IS SET TO THE VALUE OF NC, THE ORDER OF THE FORMULA CURRENTLY BEING USED. LDA 690
KFLAG - THE COMPLETION CODE INDICATOR, WITH THE FOLLOWING MEANINGS.
+1  THE INTEGRATION WAS SUCCESSFUL. LDA 700
-1  ERROR TEST FAILURE FCR > HMIN. LDA 710
-3  CORRECTOR FAILED TO CONVERGE FCR > HMIN. LDA 720
-4  CORRECTOR FAILED TO CONVERGE FCR FIRST. LDA 730
-5  ERROR RETURN FROM SUBROUTINE NL1TS. LDA 740
MAXAL - MAXIMUM ORDER DERIVATIVE THAT SHOULD BE USED IN THE LDA 750
METHOD. IT MUST BE AT LEAST SIX. IF IT IS LESS THAN SIX, THE MAXIMUM ORDER USED WILL BE SIX.. LDA 960
GREATERTHAN SIX, THE MAXIMUM ORDER USED WILL BE SIX.. LDA 970
IPRT = INTERNAL PRINT CONTROL INDICATOR
LDA 980
0 = NO PRINT
LDA 950
> 0 = PRINT COUNTERS, STEPSIZE, CURRENT TIME
LDA 1000
AND VALUES OF DEPENDENT VARIABLES AT EACH STEP.
LDA 1020
IPRT
CURRENT STEPSIZE. AN INITIAL VALUE MUST BE SUPPLIED BUT NEED NOT BE THE ONE WHICH WILL BE USED, SINCE THE LDA 1040
SUBROUTINE WILL CHOOSE A SMALLER ONE IF NECESSARY. TELCA 1050
KEEP THE ERROR PER STEP SMALLER THAN THE SPECIFIED VALUE. IT IS BETTER TO UNDERESTIMATE THE INITIAL
STEP SIZE THAN TO OVERESTIMATE IT. THE STEPSIZE IS LDA 1070
NORMALLY NOT CHANGED BY THE USER.
LDA 1080
I-MIN = MINIMUM STEPSIZE ALLOWED
LDA 1100
I-MAX = MAXIMUM STEPSIZE ALLOWED
LDA 1120
IMSEFS = THE ERROR TEST CONSTANT. THE ROOT-MEAN-SQUARE
LDA 1140
OF THE SINGLE STEP ERROR ESTIMATES, ER(1), DIVIDED BY
LDA 1160
YMAX(2) = MAXIMUM VALUE OF CURRENT TIME IF Y(2) MIGHT BE
LDA 1180
LESS THAN RMSEFS. THE STEPSIZE AND/OR ORDER ARE
LDA 1200
VARIED TO ACHIEVE THIS.
LDA 1220
SAVE = AN ARRAY OF LENGTH AT LEAST M+NY
LDA 1220
YLSV = AN ARRAY OF LENGTH AT LEAST NL
LDA 1240
YMAX = A VECTOR OF LENGTH NY WHICH CONTAINS THE MAXIMUM
LDA 1260
BECAUSE SEEN SO FAR. ON THE FIRST CALL THESE WILL BE
LDA 1280
INITIALIZED AS YMAX(1) = MAX(I,N,Y(I,1)).
LDA 1300
ER = A VECTOR OF LENGTH NY
LDA 1220
ESV = A VECTOR OF LENGTH NY
LDA 1240
FL = A VECTOR OF LENGTH NL = NY + NL
LDA 1260
CY = A VECTOR OF LENGTH NL = NY + NL
LDA 1280
PH = AN ARRAY IN WHICH THE J-MATRIX COMPUTED
LDA 1300
IN SUBROUTINE JACMAT WILL BE STORED. SIZE WHICH
LDA 1320
MUST BE ALLOWED IS DETERMINED BY THE STORAGE TECH-
LDA 1340
QUE USED FOR IT. IT NORMALLY CAN'T BE MORE THAN
LDA 1360
N*2 + 6*N LOCATIONS, THE LATTER 2*N BEING REQUIRED
LDA 1380
BY THE LINEAR EQUATION SOLVER.
LDA 1400
INTEGRATOR: IBP, IYPE, JA, JB, NAME, NELCON
LDA 1420
CIPENSION BIGH1(1), BIGH3(1), BIGH5(1)
LDA 1440
CIPENSION BIGH4(1)
LDA 1460
CIPENSION BIGH6(1)
LDA 1480
CIPENSION IBP(1)
LDA 1500
CIPENSION IYPE(1)
LDA 1520
CIPENSION JA(1), JB(1), NAME(1)
LDA 1540
CIPENSION NELCON(1,1)
LDA 1560
CIPENSION RH(1), Z(1)
LDA 1580
CIPENSION VIT(1)
DIMENSION Y(7,1),YL(1),SAVE(7,1),YMAX(1),ER(1),YLSV(1),F1(1)
1 FERT(6,3),COF(21),ESV(1),DV(1),PWL(1),SAVI(1),A(25)
1/ 12G(L)END, (A(9),ER), (A(10),L), (A(11),EDN),
1/ 15FN), (A(15),EFS), (A(16),EUP),
1/ (A(17),AN), (A(18),PEPSH), (A(19),IOGPU), (A(21),K EAV)
3/ 4(21),R), (A(22),LCPYL), (A(23),LCPYV), (A(24),M ATER),
4/ (A(25),V1), (A(26),L), (A(27),AC), (A(28),AS), (A(29),KN)

THE COEFFICIENTS IN THE FERT ARRAY ARE USED FOR ERRCR TESTING AND
CHANGING STEP SIZE AND NEED TO BE accurate TO ONLY A FEW DIGITS.

DATA PER T/4, 9, 16, 25, 36, 49, 5, 6, 16, 25, 36, 49, 5, 1, 1, 25,
12.76056E-2,1.70559E-3,6.83925E-5/.

THE ENTRIES IN THE COF ARRAY ARE THE COEFFICIENTS FOR THE STIFFLY
STABLE METHODS USED IN THIS PROGRAM AND ARE TO BE THE MACHINE
PRECISION EQUIVALENTS OF THE FOLLOWING CONSTANTS.

-11/2, -1/2
-11/6 - -13/24 - -5/12 - -1/24
-25/120 - -15/72 - -17/24 - -17/8 - -1/120
-147/60, -203/90, -49/48, -35/144, -7/24, -1/12

DATA COF/-1, -1, -5, -1.8330333, -1, -1666667, -2, 4693333, -1.4583333,
-1, -4166667, -4166667, -2, 2063333, -1.815, -7083333, -1.25, -4083333333,
2, -2.45, -2.55556, -1.020833, -2.430556, -0.2516667, -.6033333

IF (START) 100, 110, 150

IF THIS IS A RESTART ENTRY, RESTORE Y AND YL FROM THE SAVE AND
YLSV ARRAYS WHERE THEY WERE SAVED BY A PREVIOUS CALL TO LCASAV.

CALL COPYZ (YL,SAVE,LCPYV)
CALL CCOPYZ (YL,YLSV,LCPYV)
GCC 1C 150

IF THIS IS THE FIRST CALL, INITIALIZE MAX, SCALE, AND OTHERS.
INITIALIZE INDICATORS AND SET ERROR TO ONE.

FOR DOUBLE PRECISION SET LCPYPY = 14*NY AND LCPYV = 2*AL IN
SLEROUTINE COPYZ IS IN SINGLE PRECISION.

A1 C = A - A
LCCPYY = 7*NY
LCCPYL = NL
P1 = P110NY
EPS = SCR1(FLOAT(N1))*RMSEPS
SACER = MINO(MAXOR,6)
IF (IPRT=LE.0) GC TC 120
PRINT 3; N,NL,RMSEPS,TEAK,H
PRINT 4
12C
AS = C
AW = 0
CC 130 J=1,NY
YMAX(J) = AMAX1(1.,ABS(Y(1,J)))
13C
Y(2,J) = Y(2,J)*H
AC = 1
AR = 1.
ASSIGN 150 TO IRET
SET COEFFICIENTS FOR THE CRCEC CURRENTLY BEING USED.
F IS A TEST FOR ERRORS OF THE CURRENT ORDER AC
EUF IS TO TEST FOR INCREASING THE ORDER, ECLN FOR DECREASING THE
CRCEC.
14C
K = NC*(NC-1)/2
CALL COPYZ (A(2),COF(K+1),NC)
K = AC+1
ICCBU = AC
EAC1 = .5/NC
EAC2 = .5/K
EAC3 = .5/(NU+2)
PEPS= EPS**2
E = FERT(AC,1)*PEPS
ELP = FERT(NU,2)*PEPS
ECL = FERT(NU,3)*PEPS
BAC = (EPS*ENQ)**2
ISEVAL = 1
GC TO IRET, (190,200,400,570)
15C
IF CALLER HAS CHANGED H, RESCALE DERIVATIVES TO REFLECT THAT NEW
HAS BEEN ON THE LAST CALL.
R = H/HAEN
ASSIGN 190 TO IRET
GC TO 610
C
SET JSTART TO NQ, THE CURRENT ORDER OF THE FETHEC, EEFCRE EXIT,
LCA 1620
LDA 1630
LDA 1640
LDA 1650
LDA 1660
LDA 1670
LDA 1680
LDA 1690
LDA 1700
LDA 1710
LDA 1720
LDA 1730
LDA 1740
LDA 1750
LDA 1760
LDA 1770
LDA 1780
LDA 1790
LDA 1800
LDA 1810
LDA 1820
LDA 1830
LDA 1840
LDA 1850
LDA 1860
LDA 1870
LDA 1880
LDA 1890
LDA 1900
LDA 1910
LDA 1920
LDA 1930
LDA 1940
LDA 1950
LDA 1960
LDA 1970
LDA 1980
LDA 1990
LDA 2000
LDA 2010
LDA 2020
LDA 2030
LDA 2040
LDA 2050
LDA 2060
LDA 2070
LDA 2080
LDA 2090
LDA 2100
LDA 2110
LDA 2120
LDA 2130
LDA 2140
LDA 2150
LDA 2160
LDA 2170
LDA 2180
LDA 2190
LDA 2200
LDA 2210
LDA 2220
LDA 2230
LDA 2240
LDA 2250
LDA 2260
LDA 2270
LDA 2280
LDA 2290
C AAD SAVE THE CURRENT STEPSIZE IN FNEW.
C
160 JSTART = AC
FNEW = T
RETURN
17C AS = NS+1
IF (IPRINT.LE.0) GO TO 180
C PRINT DATA IF DESIRED BY USER
PRINT 1, NS,NW,NQ,T,T,(Y(I,1),I=1,NY)
1EC CONTINUE
IF (KFLAG.GT.0) GO TO 160
IF (T.GE.TEND) GO TO 160
C TAKE ANOTHER STEP IF T < TEND
C
JSTART = 1
C
SAVE DATA FOR TRIAL WITH A SMALLER TIMESTEP IF THIS STEP FAILS
C
150 CALL CCPY2 (SAVE,Y,LCCPY2)
CALL CCPY2 (VLSV,YL,LCCPYL)
RA = N + 1
KFLAG = 1
TLC = T
QEC = NO
TECLC = T
2CC T = T + H
F = INV = 1./T
C
C CCFUTURE PREDICTED VALUES BY EFFECTIVELY MULTIPLYING DERIVATIVE VECTOR BY PASCAL TRIANGLE MATRIX
C
CC
210 J = 2,K
J3 = K+J-1
CC
210 J1 = J,K
J2 = J3-J1
CC
210 I = 1,NY
21C Y(J2,1) = Y(J2,1)+Y(J2+1,1)
CC
220 I = 1,NY
22C ER(I) = 0.
C 36C KFLAG = 1
   INCUB = lGCUB-1
   IF (1GCUB) 410,37C,510
37C CALL CCFY2 (ESV,EPF1)
   GC TO 510
C THE ERROR TEST FAILED. IF JSTART = 0, THE DERIVATIVES IN THE
C SAVE ARRAY ARE UPDATED. TESTS ARE THEN MADE TO FIX THE STEPSIZE
C AND PERHAPS REDUCE THE ORDER. AFTER RESTORING AND SCALING THE
C VARIABLES, THE STEP IS RETRIED.
C 38C IF (JSTART.GT.0) GC TO 400
C
D C 250 I=1,NY
C 35C SAVE(I2) = Y(I2)
C 44C KFLAG = KFLAG-2
   IF (K-LE.*PIN) GO TO 550
   T = TOLC
   IF (KFLAG.LE.-5) GO TO 530
41C FP2 = (C/E)**ENQ2*1.2
   IF (NP.LT.1) GO TO 430
   D = 0.
C
C 42C J=1,M1
   VY = AMAX1(ABS(Y(1,J)),YMAX(J))
C 43C C = C+(Y(J)/VM)**2
C 44C PR1 = (C/EDOWN)**ENR1*1.3
   IF (PR1.GE.PR2) GO TO 430
   PR2 = PR1
   IF (KFLAG.LT.0.OR.NC.GE.MAXCER) GC TO 45C
   B = 0.
C
C 45C J=1,M1
   VY = AMAX1(ABS(Y(1,J)),YMAX(J))
C 46C C = C+(ER(J)-ESV(J))/VM**2
C 47C PR1 = (C/EPL)**ENG3*1.4
   IF (PR1.GE.PR2) GO TO 450
   PR2 = PR1
   L = 1
48C R = 1./AMAX1(PR1-1,E-5)
   IF (KFLAG.LT.0.OR.R.GE.1.1) GO TO 460
   GCUB = 9
LCMA 4200
LCMA 4210
LCMA 4220
LCMA 4230
LCMA 4240
LCMA 4250
LCMA 4260
LCMA 4270
LCMA 4280
LCMA 4290
LCMA 4300
LCMA 4310
LCMA 4320
LCMA 4330
LCMA 4340
LCMA 4350
LCMA 4360
LCMA 4370
LCMA 4380
LCMA 4390
LCMA 4400
LCMA 4410
LCMA 4420
LCMA 4430
LCMA 4440
LCMA 4450
LCMA 4460
LCMA 4470
LCMA 4480
LCMA 4490
LCMA 4500
LCMA 4510
LCMA 4520
LCMA 4530
LCMA 4540
LCMA 4550
LCMA 4560
LCMA 4570
LCMA 4580
LCMA 4590
LCMA 4600
LCMA 4610
LCMA 4620
LCMA 4630
LCMA 4640
LCMA 4650
LCMA 4660
LCMA 4670

IJC TC 510

46C NEWC = AQ+L
       K = NEWC+L
       IF (NEWC-LE-NQ) GC TO 480
       R1 = A(NEWQ) FLOAT(NEWQ)
C
       DC 470 J=1,NY
       47C Y(K,J) = ER(J)*R1
C
6C CONTINUE

| IF THE STEP WAS OKAY, SCALE THE Y VARIABLES IN ACCORDANCE |
| WITH THE NEW VALUE OF P. IF KFLAG < 0, FERLHTER, USE THE |
| SAVED VALUES (IN SAVE AND YLSV). IN EITHER CASE, IF THE ORDER |
| HAS CHANGED IT IS NECESSARY TO FIX CERTAIN PARAMETERS BY CALLING |
| THE PROGRAM SEGMENT AT STATEMENT NUMBER 140. |

6C ICCLB = NQ
       IF (NEWC-EC-NQ) GC TO 500
       NC = NEWC
       ASSIGN 450 TO IRET
       GC TO 140
6C IF (KFLAG-EC-0) GC TO 500
       RACUM = RACUM*R
       GC TO 560
       5C R = AMAX1(AAMIN,HMAX/H,R)/HMIN/H
       P = H*R
       JREVAL = 1
       ASSIGN 510 TO IRET
       GC TO 610
C
51C DC E20 I=1,N1
52C YMAX(I) = MAX1(AABS(Y(I,1)),YMAX(I))

C
6C GC TO 170

| THE ERROR TEST HAS ACN FAILED THREE TIMES, SO THE DERIVATIVES ARE |
| IN BAD SHAPE; RETURN TO FIRST ORDER METHOD AND TRY AGAIN. CF |
| CLOSE, IF NO = 1 ALREADY, THEN THERE IS AN RC PCPE AND WE EXIT WITH |
| KFLAG = -4. |

53C IF (NO-EC-1) GC TO 540
       NC = 1
       ITCCUB = 1
       ASSIGN 570 TO IRET
       GC TO 140
6C RECLD = 1
6C KFLAG = -4

54C
TC RESTART THE USER FIRST CALLS LCAST TO RESTORE THE VALUES SAVED.
BY LCASAV, THEN RE-ENTERS LDA_SUB WITH JSTART < 0, ARE 
WITH THE 
LCA_SUB, PARTICULARLY THOSE ARRAYS PERSISTED ACROSS.

ENTRY LCA Sav(SAV)
LCEP Y = 25
CALL COPYZ (SAV, A, LCEP YZ)
CAL LGPYZ (SAV, Y, LGPYZ)
RETURN

ENTRY LCAST(SAV)
LCEP Y = 25
CALL COPYZ (A, SAV, LCEP YR)
RETURN

1 FERMAT (215, 12, 1P2E10-2, 7S14.6/132X, 7E14.6)
2 FERMAT (32X, 1P7E14.6)
3 FERMAT (i) N = '13,8' RMSEPS = 'DES.2,' TENC =
4 FERMAT (2,13,8) H = '18,8X,' T '8X,' Y(1,*) AND YL(*)//
END
SLBROUTINE SDESOL (Y, YL, T, TEND, NY, NL, M, JSKF, MAXDER, IPRT, T, HMIN), SDE 10
1 PMAX, RMSEPS, M, BIGH1, BIGH2, BIGH3, MICH, BIGF, TBP, ITYPE, JA, JB, RAPE,
2 KELCON, R, Z, YI1, BIGH4

SLBROUTINE SDESOL IS A DRIVER ROUTINE FOR SLBROUTINE LLOADSUB.
ITS PURPOSE IS TO SET UP THE NECESSARY REFERENCES TO A LARGE
BLOCK OF AUXILIARY STORAGE, AND OBTAIN INITIAL VALUES OF
DERIVATIVES.
THE CALLING SEQUENCE FOR SDESOL IS
CALL SDESOL(Y, YL, T, TEND, NY, NL, M, JSKF, MAXDER, IPRT, T, HMIN, PMAX, RMSEPS, M)
WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS.

Y — ARRAY DIMENSIONED (7, NY). THIS ARRAY CONTAINS THE JTH
DEPENDENT VARIABLES AND THEIR SCALED DERIVATIVES. Y(J+1, I)
CONTAINS THE JTH DERIVATIVE OF THE ITH VARIABLE, WHERE J IS THE
CURRENT STEP SIZE. ON FIRST ENTRY THE CALLER SUPPLIES T+1
INITIAL VALUES OF EACH VARIABLE IN Y(J+1, I). THE SLB-
SEQUENT ENTRIES IT IS ASSUMED THE ARRAY HAS NOT
BEEN CHANGED. TO INTERPOLATE TO THE NEAREST DISCRETIZED
VALUES THESE CAN BE USED AS FLOWCMS. IF H IS THE
CURRENT STEP SIZE AND VALUES AT TIME T+1 ARE NECESSARY,
LET S = E/H AND THEN

J = 0
J = 1, 2, 3, . . .

I -TH VARIABLE AT T+1 IS JLM Y(J+1, I)*S**J

THE VALUE OF JS IS OBTAINED IN THE CALLING PROGRAM
BY JS = IABS(JSKF/10).

YL — ARRAY OF NL VARIABLES WHICH ARE IDENTIFIED.

T — CURRENT VALUE OF THE INDEPENDENT VARIABLE (TIME).

TEND — END TIME.

NY — NUMBER OF DIFFERENTIAL EQUATIONS AND CANALINEAR

NL — NUMBER OF LINEAR VARIABLES.

M — NUMBER OF VARIABLES INCLUDED IN THE ERRCR TEST.

JSKF — AN INDICATOR USED BOTH ON INPUT AND ULPILP.

ON INPUT, JSKF = -1 INDICATES A RESTART CALL TO
SDESOL. JSKF = 0 INDICATES AN INITIAL CALL TO
SDESOL. JSKF > 0 INDICATES A CONTINUATION OF THE
PREVIOUS CALL TO SDESOL. JSKF < -1 MAY HAVE RESULTED FROM
FROM THE USER NEGLIGING TO TEST FOR ERROR RETURNS.
FROM SDESOL. BECAUSE OF THIS VAGIBILITY, JSKF < -1 SDE 470.
RESULTS IN TERMINATION OF THE RUN WITH THE
APPROPRIATE COMMENT.
ON OUTPUT, JKF CONSISTS OF TWO DIGITS AND SIGMA,
+ OR - CF. Q IS THE ORDER OF THE FORMULA CURRENTLY
BEING USED. P INDICATES THE TYPE OF RETURN, AS
FOLLOWS:
JKF > 0, P = 1 IS THE NORMAL RETURN
JKF = 0 IS AN ERROR RETURN, WITH THE FOLLOWING
MEANING.
 P = 1 ERROR TEST FAILURE FOR F > FMIN
 P = 2 CORRECTOR FAILED TO CONVERGE FOR F > FMIN
 P = 3 CORRECTOR FAILED TO CONVERGE FOR FIRST ORDER METHOD
 P = 4 ERROR RETURN FROM SUBROUTINE KLITSL
 P = 5 ERROR RETURN FROM SUBROUTINE DERVAL
 P = 6 ERROR RETURN FROM SUBROUTINE EDEL
MAXCER - MAXIMUM ORDER DERIVATIVE THAT SHOULD BE USED IN
METHOD. IT MUST BE NO GREATER THAN SIX.
IPRT - INTERNAL PRINT CONTROL INDICATOR FOR LEASE.
 IPRT = 0 NO PRINT
 IPRT > 0 PRINT COUNTERS, STEPSIZE, CURRENT TIMES
 AND VALUES OF DEPENDENT VARIABLES AT EACH STEP.
 F - CURRENT STEPSIZE. AN INITIAL VALUE MUST BE SUPPLIED
 BUT NEED NOT BE THE ONE WHICH MUST BE USED. SINCE THIS
 SUBROUTINE WILL CHOOSE A SMALLER ONE IF NECESSARY TO
 KEEP THE ERROR PER STEP SMALLER THAN THE SPECIFIED
 VALUE, IT IS BETTER TO UNDERESTIMATE THE INITIAL
 STEPSIZE THAN TO OVERESTIMATE IT. THE STEPSIZE IS
 USUALLY NOT CHANGED BY THE USER.
 FMIN - MINIMUM STEPSIZE ALLOWED
 HMAX - MAXIMUM STEPSIZE ALLOWED
 RMSEPS - THE ERROR TEST CONSTANT. THE ROOT-MEAN-SQUARE CF
 THE SINGLE STEP ERROR ESTIMATES, ER(i,j), DIVIDED BY
 YMAX(i,j) = (MAXIMUM TO CURRENT TIME CF Y(I,J)) MUST BE
 LESS THAN EPS. THE STEPSIZE AND/OR THE CRITER
 ARE VARY TO ACHIEVE THIS.
 W - STORAGE ARRAY. MUST BE AT LEAST 13*N*Y + 5*N.
 LOCATIONS. PLUS THOSE REQUIRED FOR STORAGE OF THE
 MATRIX PM (SEE DESCRIPTION OF SUBROUTINE JACMAT).
 THE STORAGE OF PM WILL NORMALLY REQUIRE 16*P*Y (PM
*2, 3N LOCATIONS, AND IT IS NOT IMPACT STORAGE TECHNIQUES ARE USED, CAN BE MUCH FEWER.

INTEGRAL, ITYPE, JA, JB, NAPE, NAPECN
DIMENSION Y(I,J), YLL(I,J)
DIMENSION BIGH2(I,J), BIGH3(I,J), BIGH5(I)
DIMENSION BIGH4(I,11)
C DIMENSION BKIC(1)
C DIMENSION BIC(1)
C DIMENSION IBP(1), ITYPE(1)
C DIMENSION JA(1), JBP(1), NAME(1)
C DIMENSION NCON(11,1)
C DIMENSION P(1), Z(1)
C DIMENSION YIT(1)
C
C IF (JSKF.GT.0) GO TO 12C
C IF (JSKF.LT.-1) GO TO 140
C A = NY+AL
C IF (JSKF.LT.0) GO TO 110

C IF THIS IS THE (.RST ENTRY, GET ALL VALUES OF THE DIAGONALS.
C CALL DERRAL Y VL T N NY K RETR BIGH1 Z11 BIGH3 BIGH4 BIGH5 BIGE
C IF (RETTER NE.0) GO TO 130

C NCb SET UP STORAGE BLOCKS IN THE W ARRAY. THIS NEEDS TO BE DONE
C ONLY INITIALLY AND ON RESTARTS.
C THE ARRAY SAVE STARTS AT LOCATION 1 IN THE W ARRAY
C THE ARRAY YLV STARTS AT LOCATION NSV IN THE W ARRAY
C THE ARRAY ER STARTS AT LOCATION KER IN THE W ARRAY
C THE ARRAY ESV STARTS AT LOCATION NESV IN THE W ARRAY
C THE ARRAY IF STARTS AT LOCATION NF1 IN THE W ARRAY
C THE ARRAY DY STARTS AT LOCATION NDFW IN THE W ARRAY
C THE MATRIX PW STARTS AT LOCATION NFW IN THE W ARRAY

11C NSVL = 7*ky+1
C RMAX = NSVL+NL
C RER = NRMAX+NY
C RSVV = NER+NY
C API = RESV+NY
C AFW = AFY*N

12C JS = JSKF
C CALL LCASUB (Y VL T TEND N NY M JS KF MAXDEF IPR H PMAX IPR
C SESS WINSVL W NMAX W I M WINES W MFV W F1 W NDW W MFV
C 1B1H2 B1H3 B1H4 B1H5 BIGB IBP ITP N APE NKECNN A Z YIT)
C
C CGCE JSKF ON RETURN FROM LCASUB
C JSKF = I SIGN JSK*10+LA BS (KF,KF)
C RETURN

13C JSKF = -6

RETURN
C 14C PRINT 1, JSKF
STOP

C 1 FORMAT (*IT IS AN ERROR TO ENTER SOESOL WITH JSKF = 1, 10/)
END
RUN HAS BEEN TERMINATED. *
SUBROUTINE CPYZ(S,Y,L)
DIMENSION S(I),Y(I)

THIS SUBROUTINE COPIES THE ARRAY Y, OF LENGTH L, INTO THE ARRAY S

IF(L.LE.0)RETURN
DC 100 J=1,L
100 S(J) = Y(J)
RETURN
END
SUBROUTINE DERVAL (Y,YL,T,N,YW,KERET,BIGH12,BIGH3,BIGH4,BIGH5,
  I EIGK,BIGF,IBP,ITYPE,JA,JB,NAME,ELCCN,R,Z,VIT)
  INTEGER I BP, IYPE, JA, JB, NAME,AE LCCN
  D IMENSION BIGH12(1), BIGH3(1), BIGH5(1)
  D IMENSION BIGH4(1)
  D IMENSION BIGH5(1)
  D IMENSION IBP(1), IYPE(1)
  D IMENSION JA(1), JE(1), NAME(1)
  D IMENSION ELCCN(1,1)
  D IMENSION R(1,2)
  D IMENSION VIT(1)

THIS SUBROUTINE CALCULATES THE INITIAL VALUES OF THE DERIVATIVES.
IN THE GENERAL CASE IT IS WRITTEN SO THAT IT SHOULD WORK IF THE
FIRST N2 EQUATIONS ALL INVOLVE DERIVATIVES. IT ATTEMPTS TO SOLVE
THE FIRST N2 EQUATIONS USING NEWTON'S METHOD, BUT SINCE IT TRIES
to evaluate CF/DV by calling JACPAK in such a way as to make the DER
CF/DT TERM INSIGNIFICANT, IT IS POSSIBLE THAT IT MAY FAIL FOR THAT
REASON. IT MAY FAIL FOR OTHER REASONS AS WELL. IF IT DOES FAIL
THE USER CAN SUPPLY HIS OWN VERSION OF DERVAL OR MODIFY THIS
SUBROUTINE IN SUITABLE FASHION. THIS ROUTINE ASSUMES THAT VALUES OF
THE LINEAR VARIABLES HAVE BEEN SUPPLIED PREVIOUSLY. IF THESE
MUST BE SOLVED FOR SIMULTANEOUSLY WITH THE DERIVATIVES, THE USER
MUST SUPPLY HIS OWN VERSION OF DERVAL.

THE CALLING SEQUENCE FOR THIS SUBROUTINE IS
CALL CERVAL(Y,YL,T,N,YW,KERET)

WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS:

Y - SAME AS IN LOADSUB AND SODESOL. Y(1,1) CONTAINS THE
  INITIAL VALUES OF THE DEPENDENT VARIABLES. THE
  VALUES OF THE DERIVATIVES ARE RETURNED IN Y(2,1).
  DER 20

YL - SAME AS IN LOADSUB AND SODESOL. THE INITIAL VALUES OF
  THE LINEAR VARIABLES MUST BE SUPPLIED TO THIS VERSION
  DER 10

T - INITIAL TIME
  DER 50

N - SAME AS IN LOADSUB. TOTAL NUMBER OF VARIABLES
  DER 60

NY - SAME AS IN LOADSUB. NUMBER OF DIFFERENTIAL EQUATIONS
  AND NONLINEAR VARIABLES
  DER 70

W - SCRATCH ARRAY W FROM THE CALLING SEQUENCE OF SODESOL.
  THIS CAN BE USED AS NEEDED IN THIS SUBROUTINE.
  DER 80

KERET - RETURN INCORPORATED
  =0 NORMAL RETURN
  =1 ERROR RETURN
  DER 90
C DIMENSION Y(7,1), YL(1), W(1)

C

OC 1CO I=1, NY
W(2*N+1) = AMAX1(ABS(Y(I,1)), 1.)

CC Y(3,1) = 0.

CC INV = 16.*20
KERET = 0
EPS2 = KY/2.65
EPS = SCRT(EPS2)

CC 140 IT=1,10

CC 110 I=1, NY

CC Y(I2,1) = Y(3,1)/INV

CC CALL DIFFUN (Y,YL,INV, W,BIGH12,BIGH3,BIGH4,BIGH5,BIGK,BIGF,
1 IEP, IYPE, JA,JB,NAME, NELCOM,RZ, Y(I))

CC CALL JACMAT (Y,YL,INV, NY,AY,EPS,W,W(N+1),W(3*N+1),BIGH12,
1 BIGH3,BIGH4,BIGH5,BIGK,IBP,IYPE,JA,JB,NAME, NELCC,RZ)

CC AEPW = I

CC 120 I=1, NY

CC W(I) = W(I)*INV

CC CALL NUISLE (W(3*N+1), W(W(N+1)), NY,AY,EPS, W(2*N+1), AEPW, KRET,
1 BIGH3,IYPE,JA,JB,NAME, NELCC)

CC IF (KRE,NE,0) GC TC 170

CC EF = 0.

CC 130 I=1, NY

CC Y(3,1) = Y(3,1)=W(N+1)

CC W(2*N+1) = AMAX1(ABS(Y(3,1)), 1.)

CC ER = ER+W(N+1)/AMAX1(ABS(Y(3,1)), 1.)*2

CC IF (ER,LT, EPS2) GC TO 50

CC CONTINUE

CC TO 170

CC 160 I=1, NY

CC Y(I2,1) = Y(3,1)

C RETLNR
KERET = 1
RETURN
EAD
LIST OF REFERENCES


<table>
<thead>
<tr>
<th>No.</th>
<th>Distribution List</th>
<th>No. Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Defense Documentation Center</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Cameron Station</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, Virginia 22314</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Library, Code 0142</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Department Chairman, Code 69</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Mechanical Engineering</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Dr. D. H. Nguyen (thesis advisor)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4625 Larchmont NE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Albuquerque, New Mexico 87115</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Assoc. Professor D. Salinas (thesis advisor)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Mechanical Engineering</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Assoc. Professor R. Franke (second reader)</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Mathematics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Professor Gilles Cantin</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Department of Mechanical Engineering</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>LT R. E. Kasdorf, USN (student)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>129 Redondo Ct.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marina, California 93933</td>
<td></td>
</tr>
</tbody>
</table>