Simulation of a nuclear power plant, with a two-time-scale matrix linear decoupling algorithm

by

Michael Francis Nollet

A Thesis Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE

Major: Nuclear Engineering

:y

Ames, Iowa

1984

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A	coefficient matrix; also control system gain
B	decoupled state variable
Bu	driving function
C	delayed-neutron precursor; also Celsius
D	correction matrix; also diameter of fuel assemblies
DENTC	change in energy transfer from primary to secondary side of steam generator from one tim step to the next
D_{hfg}	enthalpy
E_t	total transferred from primary to secondary side of steam generator
I	identity matrix
J	Jordan canonical diagonal eigenvalue matrix
K	K matrix; also thermal conductivity; also Kelvin
K*	shorthand symbol for Lyapunov equation
L	L matrix
LMTD	logarithmic mean temperature difference
M	fundamental eigenvector matrix
M_f	mass of reactor fuel
M_m	mass of reactor moderator
Nu	Nusselt number
PWR	steam generator power
PWRCH	change in steam generator power from one time step to the next
Pr	Prandtl number
Q	inverse fundamental eigenvector matrix

R residual matirx

R* shorthand form of algebraic

Re Reynolds number

T transformation matrix

T* transformation matrix where K=0

TD temperature difference between primary and secondary sides of steam generator

TDENTC total change in energy transfer from primary to secondary side of steam generator

T_f fuel temperature

T_i reactor coolant inlet temperature

T_m reactor moderator temperature

T_o reactor coolant output temperature

T_{pii} reactor inlet temperature

T_{poi} reactor outlet temperature

P_{sg} average steam gnerator temperature

T_{sii} inlet temperature on secondary side of steam generator

T_{sor} outlet temperature of secondary side of steam generator

U flow velocity of reactor coolant

W mass flow rate

X state variable

Y transformed state variable

c heat capacity

c_{ps} heat capacity of fuel

c_{pm}	heat capacity of moderator
h	time step
h_f	fast time step
h_p	heat transfer coefficient
h_s	slow time step
i	subscript of iteration or time step
j	subscript of delayed neutron fraction
k	neutron multiplication factor
kg	kilogram
m	meter
n	neutron power
s	second
t	time
α_f	Doppler coefficient of reactivity
α_m	moderator coefficient of reactivity
β	delayed-neutron fraction
γ	rampt-input rate
δ	differential quantity
ε	trial value for testing iteration cessation
Λ	neutron generation time
λ	delayed-neutron decay constant
μ	kinematic viscosity
ν	dynamic viscosity
ρ	reactivity; also density
ρ_f	fast decoupling ratio

ρ_s slow decoupling ratio

ρ_t total reactivity

τ constant of integral controller

τ_c time constant of differential controller

I. INTRODUCTION

Computer simulation of processes that occur in the "real world" is steadily becoming a more significant learning tool as computer costs continue to decline. One advantage of this method of learning is that students are able to see how natural or industrial processes work without having to engage in deductive reasoning or calculations themselves. In this way, the regions of the human brain that engage in inductive, or nonquantified thinking, can be reached. Students can quickly acquire a "feel" for how these processes will work in general and later on (or perhaps simultaneously) learn how to perform the calculations analytically.

The modern nuclear power plant is a system that can be simulated on a digital computer. In addition to the obvious desirability of being able to simulate a nuclear power plant, such a simulation will enable a user to graphically see such phenomena as the effects of feedback, the effects of control systems, the relationship between a reactor and its steam generator, to name just a few.

II. PROPOSED OBJECTIVE AND SCOPE

The objective of the research described in this thesis is to provide a simulation of a modern nuclear power plant (PWR) with steam generator. The program associated with this thesis has been designed such that its user will be able to specify the parameters around which the system will operate. The user will be able to control feedback, a control system, reactor power levels, reactivity insertions, power output, steam generator throttle valve position, which reactor kinetics model to use, load following, and which fuel isotope is used. Output can be either in the form of a table or in graphics.

In addition, an algorithm has been devised that decouples the system of equations that describes reactor kinetics. This system is divided into two parts: one that contains the slow-acting phenomena, and another that contains the fast phenomena. After separate solutions have been obtained for each, the solutions are transformed back into the original variables. In this fashion, more efficient use of computer time is made.

The reactor coolant loop operates in an 11 second cycle. Coolant takes 5 seconds to travel from the reactor to the steam generator, remains 2.8 seconds in the steam generator, takes 3 seconds to travel back to the reactor, and remains in the reactor 0.2 seconds.

All of the differential equations used in this program are solved for transient quantities. That is, a variable that is being solved for is made up of two components: a steady-state component and a transient component. The steady-state component represents initial value of the variable, and the transient component represents the difference between the current value of the variable and its steady-state value. Mathematically, this relationship is expressed, using a sample variable X, as

$$X = X_0 + \delta X,$$

where

X_0 is its steady-state component

δX is its transient component

Since X_0 represents an unchanging value,

$$\frac{dX_0}{dt} = 0, \text{ and } \frac{dX}{dt} = \frac{d\delta X}{dt}$$

A group of simple reactor models suitable for classroom use is the purpose of this work. Because this project is for instructional use, the SI system of measurement will be used except in referring to temperatures, where Celsius (C) units will be used instead of Kelvin (K) units, and in referring to reactor primary side and secondary side pressures, where the English system is used. All calculations employ SI.

III. LITERATURE REVIEW

The objective of this study is to develop a program that simulates the operation of a modern nuclear power plant in a simplified way.

The program associated with this thesis solves the point-kinetics equations, thermal-hydraulics equations, and steam generator equations. The point-kinetics equations are solved through various models based on Duderstadt and Hamilton [5] and Hetrick et al. [8]. Their solution is aided by an algorithm based on papers by Anderson [1] and Hetrick [8] that can solve separately for the slow components and the fast components of the point-kinetics equations.

Specifications for the reactor and the steam generator were taken from Babcock and Wilcox Company [2] and from the Preliminary Safety Analysis Report for the Greenwood reactor of the Detroit Edison Company [4]. The Greenwood reactor, construction of which has since been cancelled, was to have been supplied by Babcock and Wilcox Company, so naturally the specifications were similar for both. Correlations for physical parameters such as specific heat, kinematic viscosity, thermal conductivity, and density where obtained from El-Wakil [6] and Keenan and Keyes [10]. The most useful heat-transfer correlations such as the Dittius-Boelter correlation were obtained from Karlekar and Desmond

[9]. Reactor pressure data were obtained from the U.S. Atomic Energy Commission [12]. The reactor control system was based on Danofsky [3], while the feedback theory and some of the reactor models used were based on Schultz [11].

IV. THE TWO-TIME-SCALE MATRIX DECOUPLING ALGORITHM

The point-kinetics equations are a system of first-order nonlinear differential equations used in solving nuclear reactor kinetics problems. They are in general composed of one equation that solves for reactor power output (or neutron activity, to which power output is directly related) and a subsystem of equations that solves for the delayed-neutron precursors, whose existence is so important in reactor kinetics.

The delayed neutrons originate from the radioactive decay of fission fragments. Since it is radioactive decay from many different isotopes that produce them, they come in a wide range of energies and mean decay times. The mean decay times typically are on the order of several seconds, while the mean lifetime of prompt neutrons will be taken as 0.0001 seconds [7]. Since the time scales of the two types of neutrons are so different, the same time step that would be most useful in solving equations for one type of neutron would not be suitable for the other.

Small time steps are needed in applying numerical techniques to solve differential equations for the prompt response. However, when they are used for solving for the much slower delayed-neutron precursors, progressive arithmetic error can cause degradation of the quality of the solutions. Also, valuable Central Processing Unit (CPU) time in the digital computer solving the equations is wasted.

On the other hand, if time steps appropriate for the slow precursors are used in the prompt response, then meaningless answers are derived. The ideal case would be to use large time steps for the slow precursors, and small ones separately for the prompt response. A means will be developed in this section for doing just that.

Consider the system described by the vector equation

$$\dot{X} = AX. \qquad (4-1)$$

This represents the 7x7 coupled system of point-kinetics equations, which is to be decoupled into two independent systems of equations. One system will contain the slow mode variables, while the other will contain the fast mode variables.

Decompose the system represented by Equation (4-1) into the system

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \qquad (4-2)$$

where X_1 represents the slow mode variables and X_2 the fast mode variables. The slow mode corresponds to the delayed-neutron precursors, and the fast mode to the prompt neutron response.

Let

$$X = TY, \tag{4-3}$$

where T is a transformation matrix. Then

$$T = \begin{bmatrix} I_{n_1} & -K \\ -L & I_{n_2}+LK \end{bmatrix}, \tag{4-4}$$

where I_{n_1} is an $n_1 x n_1$ identity matrix, I_{n_2} is an $n_2 x n_2$ identity matrix, K is an $n_1 x n_2$ matrix, and L is an $n_2 x n_1$ matrix. K and L will be defined later. Also,

$$T^{-1} = \begin{bmatrix} I_{n_1}+KL & K \\ L & I_{n_2} \end{bmatrix}. \tag{4-5}$$

Now again,

$$X = TY. \tag{4-3}$$

This implies

$$Y = T^{-1}X, \tag{4-6}$$

which further implies

$$\dot{Y} = T^{-1}\dot{X} . \tag{4-7}$$

Equations (4-1) and (4-3) imply that

$$\dot{X} = AX = A[TY] = ATY, \tag{4-8}$$

and Equations (4-7) and (4-8) further imply that

$$\dot{Y} = T^{-1}\dot{X} = T^{-1}[ATy] = T^{-1}ATy. \qquad (4\text{-}9)$$

Expanding equation (4-9) yields

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} I_{n_1}+KL & K \\ L & I_{n_2} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I_{n_1} & -K \\ -L & I_{n_2}+LK \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$$

$$= \begin{bmatrix} I_{n_1}+KL & K \\ L & I_{n_2} \end{bmatrix} \begin{bmatrix} A_{11}-A_{12}L & -A_{11}K+A_{12}+A_{12}LK \\ A_{21}-A_{22}L & -A_{21}K+A_{22}+A_{22}LK \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$$

and

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} A_{11}+KLA_{11}-A_{12}L-KL\,A_{12}L+KA_{21}-KA_{22}L & -A_{11}K+A_{12}+A_{12}LK-KL\,A_{11}K+KLA_{12}+KL\,A_{12}LK-K\,A_{21}K+K\,A_{22}+K\,A_{22}LK \\ LA_{11}-LA_{12}L+A_{21}-A_{22}L & -LA_{11}K+LA_{12}+LA_{12}LK-A_{21}K+A_{22}+A_{22}LK \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}. \qquad (4\text{-}10)$$

Introduce the algebraic Riccati equation

$$LA_{11} + A_{21} - LA_{12}L - A_{22}L - 0. \qquad (4\text{-}11)$$

If the algebraic Riccati equation is satisfied, then Equation (4-10) reduces to

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} \begin{matrix} (A_{11}-A_{12}L) \\ +K(LA_{11}+A_{21} \\ \quad -LA_{12}L-A_{22}L) \end{matrix} & \begin{matrix} (-A_{11}K+A_{12} \\ +A_{12}LK+KLA_{12} \\ +KA_{22})+K(LA_{11} \\ +A_{21}-LA_{12}L-A_{22}L)K \end{matrix} \\ \begin{matrix} (LA_{11}-LA_{12}L+A_{21} \\ -A_{22}L) \end{matrix} & \begin{matrix} (LA_{11}+A_{21}-LA_{12}L \\ -A_{22}L)(-K) \\ +LA_{12}+A_{22} \end{matrix} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \tag{4-12}$$

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} A_{11}-A_{12}L & \begin{matrix} -A_{11}K+A_{12} \\ +A_{12}LK+KLA_{12} \\ +KA_{22} \end{matrix} \\ 0 & LA_{12}+A_{22} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}. \tag{4-13}$$

Introduce the vectors

$$[B_1] = [A_{11}-A_{12}L] \tag{4-14}$$

$$[B_2] = [LA_{12}+A_{22}] \tag{4-15}$$

and the Lyapunov equation

$$KB_2 - B_1K + A_{12} = 0. \tag{4-16}$$

Substitution of Equations (4-14) and (4-15) into Equation (4-13) will further reduce equation (4-13) to

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1 & K(LA_{12}+A_{22}) - (A_{11}+A_{12}L)K + A_{12} \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$$

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1 & KB_2 - B_1K + A_{12} \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}. \tag{4-17}$$

If the Lyapunov equation is satisfied, then Equation (4-17) is even further reduced to

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}. \tag{4-18}$$

Therefore, provided that an L matrix and a K matrix can be found that satisfy the algebraic Riccati equation and the Lyapunov equation, any system of first order coupled differential equations of the form

$$\begin{bmatrix}\dot{X}_1\\ \dot{X}_2\end{bmatrix} = \begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix} \begin{bmatrix}X_1\\ X_2\end{bmatrix}$$

can be transformed into a decoupled system of the form

$$\begin{bmatrix}\dot{Y}_1\\ \dot{Y}_2\end{bmatrix} = \begin{bmatrix}B_1 & 0\\ 0 & B_2\end{bmatrix} \begin{bmatrix}Y_1\\ Y_2\end{bmatrix}. \qquad (4\text{-}18)$$

Note that this is no longer one system of differential equations, but two entirely separate systems. For a numerical solution of these systems, each system 1 and 2 can utilize whatever time step is appropriate. Since these are separate systems, the size of the time step used in one system will have no effect on the other. One advantage to such an ordering will be that CPU time on a digital computer will be reduced to a minimum, because no time steps will be smaller than that needed for a given system.

Other advantages arise from the nature of numerical error. In numerical solutions of differential equations, if the time step used is too large, then transient phenomena between iterations can become so significant as to render any solution meaningless. On the other hand, if the time step is smaller than needed, then there can be so many iterations that simple arithmetic errors generated by the computer can become progressively larger, and degrade the quality of the solutions.

The actual choice of the size of the time steps is discussed in Section V.B.

In practice, the L matrix is calculated first, and the differential equations are solved for the decoupled variables. The K matrix is required only for the transformation of the solutions derived from the decoupled variables back into the coupled variables (in other words, K is needed only to transform Y back into X).

To compute the L matrix, let

$$A = MJQ, \tag{4-19}$$

where MJQ is the Jordan canonical form of the A matrix. J is the diagonal matrix made up of the eigenvalues of the A matrix, M is the fundamental matrix of eigenvectors corresponding to J, and Q is the inverse fundamental matrix

$$M^{-1} = Q$$

and

$$Q^{-1} = M.$$

Thus,

$$\begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} MJQ \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \qquad (4\text{-}20)$$

It is important to note that J is assembled in ascending order of the absolute values of the eigenvalues. Since

$$QM = I, \qquad (4\text{-}21)$$

$$\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} I_1 & 0 \\ 0 & I_2 \end{bmatrix}.$$

This implies that

$$Q_{11}M_{11} + Q_{12}M_{21} = Q_{21}M_{12} + Q_{22}M_{22} = I \qquad (4\text{-}22)$$

$$Q_{21}M_{11} + Q_{22}M_{21} = Q_{11}M_{12} + Q_{12}M_{22} = 0, \qquad (4\text{-}23)$$

which leads to the result that

$$Q_{21} = -Q_{22}M_{21}M_{11}^{-1}, \qquad (4\text{-}24)$$

or

$$-M_{21}M_{11}^{-1} = Q_{22}^{-1}Q_{21} \ . \qquad (4\text{-}25)$$

Equation (4-22) implies that

$$Q_{11} + Q_{12}M_{21}M_{11}^{-1} = M_{11}^{-1}$$

$$Q_{12}M_{21}M_{11}^{-1} = M_{11}^{-1} - Q_{11}$$

$$M_{21}M_{11}^{-1} = Q_{12}^{-1}M_{11}^{-1} - Q_{12}Q_{11}.$$

Substituting Equation (4-25) yields

$$-Q_{22}Q_{21} = Q_{12}^{-1}M_{11}^{-1} - Q_{12}Q_{11}$$

$$M_{11}^{-1}-Q_{11} = -Q_{12}Q_{22}^{-1}Q_{21},$$

which yields the identity

$$M_{11}^{-1} = Q_{11} - Q_{12}Q_{22}^{-1}Q_{21} . \qquad (4\text{-}26)$$

Similarly, from Equation (4-24),

$$0 = Q_{21}M_{11} + Q_{22}M_{21} \qquad (4\text{-}27)$$

$$= Q_{21}M_{11}M_{21}^{-1} + Q_{22}$$

$$= Q_{22}^{-1}Q_{21}M_{11}M_{21}^{-1} + I$$

$$= Q_{22}^{-1}Q_{21}M_{11} + M_{21}$$

$$0 = Q_{22}^{-1}Q_{21} + M_{21}M_{11}^{-1} . \qquad (4\text{-}28)$$

From Equation (4-22),

$$I = Q_{21}M_{12} + Q_{22}M_{22}$$

$$Q_{22}^{-1} = Q_{22}^{-1}Q_{21}M_{12} + M_{22}$$

$$Q_{22}^{-1}Q_{21}M_{12} = Q_{22}^{-1}-M_{22}$$

$$Q_{22}^{-1}Q_{21} = Q_{22}^{-1}M_{12}^{-1} - M_{22}M_{12}^{-1} .$$

Substitution of Equation (4-28) yields

$$Q_{22}^{-1}\,{}_{12}^{-1} - M_{22}M_{12}^{-1} = M_{21}M_{11}^{-1}$$

$$Q_{22} - M_{22} = -M_{21}M^{-11}M_{12},$$

which leads to the identity

$$Q_{22}^{-1} = M_{22} - M_{21}M_{11}^{-1}M_{12}. \qquad (4\text{-}29)$$

Both Equations (4-26) and (4-29) will be used in developing Theorem 1, in which the conditions under which the algebraic Riccati eqution is satisfied are developed.

Introduce the variables R* and K*. If R* is defined as a shorthand symbol for the algebraic Riccati equation, and K* is a shorthand symbol for the Lyapunov equation, then

$$R^* = LA_{11} + A_{21} - LA_{12}L - A_{22} \qquad (4\text{-}30)$$

and

$$K^* = KB_2 - B_1K + A_{12} \qquad (4\text{-}31)$$

In that case, Equation (4-13) can be restated

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1+KR^* & K^*+KR^*K \\ R^* & -R^*K+B_2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \qquad (4\text{-}32)$$

Of course, if the algebraic Riccati equation and Lyapunov equation are satisfied, then Equation (4-14) reduces to Equation (4-18).

It follows from Equations (4-3) and (4-6) that

$$X_1 = Y_1 - Y_2 K \tag{4-3a}$$

$$X_2 = -Y_1 L + (I_{n_1} + LK) Y_2 \tag{4-3b}$$

$$Y_1 = (I_{n_1} + LK) X_1 + KX_2 \tag{4-6a}$$

$$Y_2 = LX_1 + X_2 \ . \tag{4-6b}$$

From Equation (4-6), the following transformation holds:

$$\begin{bmatrix} X_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ L & I \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} . \tag{4-33}$$

Note that this is equivalent to a transformation of the X_2 variables only. Note, if $K = 0$, the transformation matrix used here is the same as the inverse transformation matrix T^{-1}.

By hypothesis,

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} . \tag{4-34a}$$

Therefore,

$$\dot{X}_1 = A_{11} X_1 + A_{12} X_2 . \tag{4-34b}$$

Substitution of Equations (4-3a), (4-3b) and (4-34a) result in

$$\dot{X}_1 = A_{11}X_1 + A_{12}X_2 \tag{4-34b}$$

$$= A_{11}Y_1 - KA_{11}Y_2 + A_{12}(-Y_1L + (I_{n_1}+LK)Y_2)$$

$$= A_{11}Y_1 - KA_{11}Y_2 - A_{12}LY_1 + A_{12}Y_2 + LKA_{12}Y_2$$

$$= (A_{11}-A_{12}L)(Y_1-KY_2) + A_{12}Y_2$$

$$= (A_{11}-A_{12}L)X_1 + A_{12}Y_2$$

$$\dot{X}_1 = B_1X_1 + A_{12}Y_2$$

It is already known that if the algebraic Riccati and Lyapunov equations are satisfied, then

$$\dot{Y}_2 = B_2Y_2 \tag{4-18a}$$

Therefore, the transformation (4-33) reduces to

$$\begin{bmatrix} \dot{X}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1 & A_{12} \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_2 \end{bmatrix} \tag{4-35}$$

where

$$\begin{bmatrix} B_1 & A_{12} \\ R^* & B_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ L & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ -L & I \end{bmatrix}. \tag{4-36}$$

A. Computing the L Matrix

A practical means of computing the L matrix must be found. Fortunately, a means for doing this is suggested by Anderson [1], which is repeated here.

Theorem 1: The L matrix satisfies the algebraic Riccati equation if and only if

1) M_{11} is full rank
2) $L = -M_{21}M_{11}^{-1} = Q_{22}^{-1}Q_{21}$. (4-37)

Proof: Assume first that Equation (4-11) is satisfied. Rewrite it as

$$L(A_{11}-A_{12}L) = -A_{21}+A_{21}L$$

and let $(A_{11}-A_{12}L)$ have the Jordan form XGX^{-1}. Setting $Y = -LX$, it follows that

$$A_{11} - A_{12}L = XGX^{-1}$$

$$A_{21} - A_{22}L = -LXGX^{-1} .$$

Postmultiply by X to obtain

$$A_{11}X + A_{12}Y = XG$$
$$A_{21}X + A_{22}Y = YG$$

or

$$A\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} X \\ Y \end{bmatrix} G$$

or

$$A = \begin{bmatrix} X \\ Y \end{bmatrix} G \begin{bmatrix} X \\ Y \end{bmatrix}^{-1}.$$

Thus, the diagonal elements of the Jordan form G are n_1 of the n eigenvalues of A, and

$$\begin{bmatrix} X \\ Y \end{bmatrix}$$

are n, corresponding eigenvectors of A. Also, X is full rank, completing the first half of the proof.

For the second half of the proof, recall Equation (4-9)

$$\dot{Y} = T^{-1}ATY$$

or

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1+KR^* & K^*+KR^*K \\ R^* & -R^*K+B_2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix},$$

where again,

$$R^* = LA_{11} + A_{21} - LA_{12}L - A_{22}L. \qquad (4\text{-}30)$$

It will be demonstrated that the Lyapunov equation need not be satisfied for this theorem to be true. In other words, K can assume any value, including

$$K = 0.$$

If

$$K = 0,$$

then the transformation matrices T and T^{-1}, respectively, reduce to

$$T^* = \begin{bmatrix} I_{n_1} & 0 \\ -L & I_{n_2} \end{bmatrix}$$

and

$$T^{*-1} = \begin{bmatrix} I_{n_1} & 0 \\ L & I \end{bmatrix} .$$

For the purposes of this theorem, K will equal 0, because this will make the proof easier without affecting its validity. Thus,

$$T^{*-1}AT^* = \begin{bmatrix} I & 0 \\ L & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ -L & I \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}-A_{12} & A_{12} \\ LA_{11}+A_{21} & A_{22}+LA_{12} \\ -LA_{12}L-A_{22}L & \end{bmatrix}$$

$$T^{*-1}AT^{*} = \begin{bmatrix} B_1 & A_{12} \\ LA_{11}+A_{21} & B_2 \\ \quad -LA_{12}L-A_{22}L & \end{bmatrix}.$$

Suppose $L = -M_{21}M_{11}^{-1} = Q_{22}^{-1}Q_{21}$. Then, we need to show that as a result,

$$LA_{11} + A_{21} - LA_{12}L - A_{22}L = 0. \tag{4-11}$$

If

$$L = -M_{21}M_{11}^{-1} = Q_{22}^{-1}Q_{21}$$

and

$$\dot{Y} = T^{-1}ATY$$

and

$$A = MJQ$$

and

$$K = 0,$$

then, write A in Jordan form and complete the product so that

$$T^{*-1}MJQT^{*}$$

$$= \begin{bmatrix} I_{n_1} & 0 \\ -M_{21}M_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} I_{n_1} & 0 \\ -Q_{22}Q_{21} & I_{n_2} \end{bmatrix}. \tag{4-38}$$

Substitution of Equations (4-26) and (4-29) yields

$$T^{*-1}MJQT^{*} =$$

$$\begin{bmatrix} I_{n_1} & 0 \\ -M_{21}M_{11}^{-1} & I_{n_2} \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} Q_{11}-Q_{12}Q_{22}^{-1}Q_{21} & Q_{12} \\ 0 & Q_{22} \end{bmatrix}$$

$$= \begin{bmatrix} M_{11} & M_{12} \\ 0 & -M_{21}M_{11}^{-1}M_{12} + M_{22} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} -Q_{12}Q_{22}^{-1}Q_{21} & Q_{12} \\ 0 & Q_{22} \end{bmatrix}$$

$$= \begin{bmatrix} M_{11} & M_{12} \\ 0 & Q_{22}^{-1} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} M_{11}^{-1} & Q_{12} \\ 0 & Q_{22} \end{bmatrix}$$

$$= \begin{bmatrix} M_{11}J_1 & M_{12}J_2 \\ 0 & Q_{22}^{-1}J_2 \end{bmatrix} \begin{bmatrix} M_{11}^{-1} & Q_{12} \\ 0 & Q_{22} \end{bmatrix}$$

$$= \begin{bmatrix} M_{11}JM_{11}^{-1} & M_{12}J_2Q_{12}+M_{12}J_2Q_{22} \\ 0 & Q_{22}^{-1}J_2Q_{22} \end{bmatrix} , \qquad (4\text{-}39)$$

completing the proof.

A few observations are in order here:

1) Inspection of Equation (4-38) will show that because the left column of the T matrix and the bottom row of the T^{-1} matrix have no K term in the first place, then the lower left entry in the product matrix (4-39) is independent of K. In fact, this entry, which is equal to zero, <u>is</u> the algebraic Riccati equation for

$$L = -M_{21}M_{11}^{-1} = Q_{22}^{-1}Q_{21}.$$

The same result would have been produced for any K; however, the other entries in the product matrix (4-39), which were not used in this proof, would have been much more complicated. Hence, the decision to set K equal to zero.

2) The M_{11} matrix is part of the M matrix, which in turn, is the fundamental matrix of eigenvectors corresponding to the diagonal J matrix of eigenvalues. In practice, when solving the point-kinetics equations, the eigenvalues of the J_1 matrix are always approximately equal to the negatives of the decay constants of each of the 6 delayed neutron groups. Since these eigenvalues are real and distinct, the M_{11} matrix is always full rank, which is one of the prerequisites for Theorem I.

3) The A matrix is precisely defined. The M, J, and Q matrices are generated from the A matrix by means of

subroutines in the PORT Library. Apparently, the matrices thus generated are not absolutely accurate, because the algebraic Riccati equation does not exactly equal zero when the matrix L from the equation

$$L = -M_{21}M_{11}^{-1} = Q_{22}^{-1}Q_{21}$$

is substituted into it. Rather, this value for L is only an excellent first approximation. However, there is an iterative technique, which will later be outlined, that generates an L matrix to an extremely high degree of accuracy. In practice, about 6 iterations are required. The K matrix is also computed by means of an iterative technique.

4) One corollary of Theorem 1 is that

$$B_2 = Q_{22}^{-1}J_2Q_{22}. \qquad (4\text{-}40)$$

This means that B_2 can be computed without the L matrix, even though it was originally defined in terms of L. This is a useful fact which facilitates the computation of L, as will be seen.

5) In practice, there are no complex eigenvalues (and therefore, no complex eigenvectors) under any circumstances encountered in this program. Also, for reasons that will be explained later, the A matrix is a 7x7 matrix which is decoupled into a system consisting of six slow variables and one fast variable. Consequently, the A_{11} matrix (as well as

the M_{11} and Q_{11} matrices) is a 6x6 matrix, the A_{22} matrix (and the M_{22} and the Q_{22} matrices) is 1x1, and the A_{12} and A_{21} matrices are, respectively, 6x1 and 1x6. Also, the L matrix is 1x6 and the K matrix is 6x1. The fact that there are no complex numbers here and that some of the matrices have only one row or column, greatly facilitates computation.

6) In another theorem which will be repeated here without proof, Anderson [1] shows that for a given two-time-scale system there is only one decoupling matrix L. In another theorem, which he cites, it is proven that provided B_1 and B_2 have no common eigenvalues, the situation which turns out always to be the case in this program, there is likewise only one K matrix.

7) In developing the background of the two-time-scale decoupling algorithm, it was both necessary to repeat many of the steps found in Anderson's [1] paper and to complete other steps he left incomplete or unstated. This is because in developing the proof of Theorem 1 and the background of it, Anderson utilized several equations and identities without proof. He seems to have written his paper more to explain how to use a decoupling algorithm than to explain why it is mathematically valid. This thesis is the only known work in which a full background is to be found.

To compute the L matrix, first set

$$L_0 = Q_{22}^{-1} Q_{21} \qquad (4\text{-}41)$$

$$i = 0.$$

The Q submatrices are used instead of the $-M_{21}$ and M_{11}^{-1} submatrices because the M_{11}^{-1} submatrix is a 6x6 matrix and the Q_{22}^{-1} submatrix is only 1x1. Use of the one instead of the other greatly facilitates computation.

Define the residual matrix

$$R_0 = L_0 A_{11} + A_{21} - L_0 A_{12} L_0 - A_{22} L_0,$$

and evaluate its Euclidean norm. (Since R_i is only 6x1, this is also particularly easy.) If

$$||R_i|| \leq \varepsilon,$$

then stop. The L matrix is already well-defined. In this program,

$\varepsilon = 1.0 \times 10^{-6}$ was used.

If more iterations are needed, define the correction matrix D_i as follows:

$$(A_{22} + L_i A_{12}) D_i = R_i,$$

or

$$B_2 D_i = R_i,$$

and

$$D_i = B_2^{-1} R_i .$$

Since B_2 and B_2^{-1} are both 1x1, this is also an easy operation. Additionally, since B_2 exactly equals $Q_{22}^{-1} J_2 Q_{22}$, it is not necessary to use first approximations of L in order to generate first approximations for B_2. If B_2 were only approximately defined, then this would induce errors in successive computations of L; which would have the effect of increasing the number of iterations necessary to achieve convergence.

In practice, since Q_{22}^{-1}, Q_{22}, B_2^{-1} and B_2 are all 1x1,

$$Q_{22}^{-1} = \frac{1}{Q_{22}}$$

$$B_2^{-1} = \frac{1}{B_2}$$

and

$$B_2 = Q_{22}^{-1} J_2 Q_{22}$$

$$= \frac{JQ_{22}}{Q_{22}}$$

and the elegant result that

$$B_2 = J_2 \tag{4-42}$$

is achieved. Of course, J_2, like B_2, is 1x1.

Also, because $B_2^{-1} = 1/B_2$,

$$D_i = B_2^{-1} R_i$$

$$= R_i / B_2. \tag{4-43}$$

To continue with the description of the iteration process, let

$$L_{i+1} = L_i + D_i.$$

Then recompute the residual matrix R_i,

$$R_i = L_i A_{11} + A_{21} - LA_{12} L_i - A_{22} Li \tag{4-44}$$

and evaluate its Euclidean norm. If it is sufficiently small, stop; if not, iterate further. Note that Equation (4-44) is the algebraic Riccati equation, which, it will be remembered, is supposed to equal zero; hence, the requirement that its Euclidean norm be very small.

To summarize, rewrite the above as:

Algorithm 1:

1) Obtain an initial approximation L_0 from Equation (4-41).

2) Evaluate

   $R_i = L_i A_{11} + A_{21} - LA_{12} L - A_{22} L$;

   and stop if $||R_i|| \leq \varepsilon$.

3) Evaluate

$$D_i = R_i/B_2$$

and let

$$L_{i+1} = L_i + D_i.$$

4) Let

$$i = i+1$$

and go to 2).

B. Computing the K Matrix

Restate the decoupled differential equations

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}. \quad (4\text{-}18)$$

Since B_1 and B_2 are defined only in terms of A and L, it has thus far been unnecessary to compute the K matrix. Furthermore, the decoupled systems of differential equations of Equation (4-18) can be solved, also without invoking K.

The only reason to compute the K matrix is to transform the solutions for the Y variables into solutions for the X-variables, via the T matrix (Equation (4-4)). In practice, only the Y_2 variable will be transformed into X_2, because the Y_2 and X_2 variables describe reactor power, and it is only reactor power that is of interest.

The Y_1 variables describe the behavior of the six delayed-neutron precursors. While they are very important for reactor power computations, once reactor power is computed, they are of no further interest and are not utilized in program output. Hence, it is not necessary to transform them into the X_1 variables.

To compute the K matrix, utilize Algorithm 2.

Algorithm 2:

1) Set $K_0 = 0$

$D_i = 0$

$R_0 = -A_{12}$

$i = 0.$

2) Solve

$D_i = R_i/B_2.$

Let

$K_{i+1} = Ki+D_i.$

3) Evaluate

$$R_{i+1} = -K_{i+1}B_2 + B_1K - A_{12}. \qquad (4\text{-}45)$$

Stop if

$||R_{i+1}|| \leq \varepsilon.$

4) Set $i = i+1$ and go to 2.

A few observations are also in order here:

1) Unlike Algorithm 1, the K, D, and R matrices in Algorithm 2 are 6x1 instead of 1x6.

2) As in Algorithm 1, $\varepsilon = 1.0 \times 10^{-6}$.

3) In computing the R matrix of Algorithm 2, first a preliminary estimate of K is made, then substituted into the Lyapunov equation. However, when using the standard form of the Lyapunov equation, Equation (4-16), it was found that in successive iterations, $||R_i||$ diverges instead of converging toward zero. Upon inspection, the reason becomes apparent. In this program,

$$B_2 = J_2, \tag{4-42}$$

where J_2 is the eigenvector of the fast mode. Typically

$$-70.0 \leq J_2 \leq -15.0$$

(again, J_2 is 1x1). The B_1 matrix is a 6x6 diagonal matrix made up approximately of the negatives of the 6 delayed-neutron precursors. Typical ranges are from -0.005 to -3.00.

Thus, all the entries in either the B_1 or the B_2 matrices are negative numbers. Furthermore, all the entries in the K matrix are also negative numbers. Restate the standard form of the Lyapunov equation

$$R_i = K_{i+1}B_2 - B_1K_{i+1} + A_{12}. \tag{4-45}$$

If K is too large, then the residual R_i tend to become a matrix made up of only positive entries. This is because if K is negative, the first term of the Lyapunov equation will be positive and the second term negative. The third term A_{12} is always positive. It is comprised of the six delayed-neutron constants, which are always positive. Since B_2 is much larger than B_1, the first term will dominate the second term and the residual as a whole will be positive.

The residual matrix R_i is then used to compute the correction matrix D_i via the equation

$$D_i = R_i/B_2.$$

Since B_2 is always a negative number, it follows that if R_i is a positive matrix, then D_i is a negative one.

Then a new K matrix, K_{i+1}, is computed via the equation

$$K_{i+1} = K_i + D_i.$$

Since the K_i matrix is already a negative matrix, addition of another negative matrix to it will result in a yet larger K matrix, all of whose elements are larger than previously. When the new residual matrix R_{i+1} is computed, it will be larger than the preceding residual matrix R_i. In other words,

$$||R_{i+1}|| > ||R_i||,$$

which indicates that divergence, not convergence, is taking place. The same process will occur, albeit in a different direction, if the initial estimate of K is too small.

The Lyapunov equation in standard form is written

$$0 = KB_2 - B_1K + A_{12} \cdot \qquad (4\text{-}16)$$

However, since it equals zero, it can also be expressed in the form

$$0 = -KB_2 + B_1K - A_{12}. \qquad (4\text{-}46)$$

Now the situation is much different. For if

$$R_{i+1} = -K_{i+1}B_2 + B_1K_{i+1} - A_{12}, \qquad (4\text{-}45)$$

then, for instance, if K is too large, R_{i+1} will now tend to be a matrix made up of negative entries. The correction matrix D_i will then tend to be a matrix made up of positive entries, (since the division of two negative numbers results in a positive number) and the new K_{i+1} matrix will tend to become smaller. This will result in R_{i+2} now being smaller than R_{i+1}, or

$$||R_{i+1}|| < ||R_i||,$$

which indicates that finally K is converging toward its true value.

In general, the following principle can be stated. Consider either form of the Lyapunov equation.

$$R_{i+1} = K_{i+1}B_2 - B_1K_{i+1} + A_{12} \qquad (4\text{-}47)$$

or

$$R_{i+1} = -K_{i+1}B_2 + B_1K_{i+1} - A_{12} \; . \qquad (4\text{-}45)$$

If it is known that K is predominantly a positive matrix, use Equation (4-47). When K is negative, use Equation (4-45). When in doubt, experimental runs will have to be made.

In this program, it is known that K is a negative matrix, since K approximately equals $-A_{12}$, and all of the entries of A_{12} are always positive. Hence, Equation (4-45) will always be the equation of choice in determining the K matrix.

C. The Choice of Slow and Fast Variable

The equations that are solved by decoupling are the point-kinetics equations. The A matrix is a 7x7 matrix featuring six equations for the six groups of delayed-neutron precursors, and one equation for the prompt response. Since the delayed precursors have half lives of up to several minutes, it would be expected that the slow mode should contain all the delayed precursors and the fast mode only the prompt response. This turns out to be the case, but there is a means of verifying this.

Assume that the columns of M are normalized so that each is of length 1, and define the slow mode decoupling ratio ρ_s and the fast mode decoupling ratio ρ_f

$$\rho_s = \frac{||M_{21}||}{||M_{11}||} \tag{4-48}$$

and

$$\rho_f = \frac{||M_{12}||}{||M_{22}||} \tag{4-49}$$

To generate a M matrix for test purposes, a run was made for U-235 fuel, with a step reactivity input of 10 cents. Table 1 shows the resulting M matrix, and Table 2 shows the same matrix with columns normalized to a length of 1. (In this case, it actually made very little difference whether the columns were normalized or not, because the seventh column of the M matrix - the one that contains M_{12} and M_{22} - was already of unit length.)

With an ordering of six slow variables and one fast variable, the M_{11} submatrix is a 6x6 matrix occupying the first six rows of the first six columns, the M_{21} submatrix is a 1x6 matrix in the bottom row of M, the M_{22} submatrix is the single entry in the lower right corner, and the M_{12} submatrix is a 6x6 latrix in the right-hand M matrix. Under this ordering,

Table 1. Inverse eigenvector matrix of coefficient matrix of point-kinetics equations, with U-235 fuel

-5.8428212E-Ø3	-2.2450656E-Ø2	1.138648	-6.1259083E-Ø2
5.5526220E-Ø3	3.1158078E-Ø2	-4.7610782E-2	-1.119222
5.5012584E-Ø2	-1.151598	-0.1298485	-0.1955647
0.6542489	0.6703086	0.6762639	0.6739961
0.1145651	0.1106563	0.1093184	0.1098212
-0.7772868	-0.3515190	-0.2937855	-0.3132723
-0.3808326	-0.3190138	-0.3014579	-0.3078748

-2.1475744E-ϕ3	-5.8924372E-ϕ4	-2.3943800E-ϕ4
1.9070993E-ϕ3	5.0920382E-ϕ4	2.0575027E-ϕ4
1.6006008E-ϕ2	-1.0369849E-ϕ3	1.6115000E-ϕ3
0.6197168	0.5076421	0.3722187
0.1248162	0.1994342	-0.9808449
0.4277675	5.7167120E-ϕ2	2.083645E-ϕ2
-0.6933042	0.2852145	7.4262023E-ϕ2

$$||M_{11}|| = 2.449288031$$

$$||M_{12}|| = 0.491622207$$

$$||M_{21}|| = 0.042813149$$

$$||M_{22}|| = 0.8712999$$

and

$$\rho_f = \frac{||M_{12}||}{||\ _{22}||}$$

$$= 0.564239945$$

$$\rho_s = \frac{||M_{21}||}{||M_{11}||}$$

$$= 0.017479834.$$

Since ρ_f is very large compared to ρ_s, the system is considered "strongly coupled".

When there are five variables in the slow mode and 2 in the fast, ($s = 5$, $f = 2$)

$$\rho_f = 0.580217398$$

$$\rho_s = 0.010980874.$$

For $s = 4$, $f = 3$,

$$\rho_f = 0.60220452$$

$$\rho_s = 0.01743454.$$

For increasingly larger fast groups, ρ_f will start to decrease, but on the other hand, ρ_s will start increasing rapidly, and the system will cease to be strongly coupled.

All of the three orderings above are strongly coupled due to the fact that $\rho_f >> \rho_s$, but when s = 6 and f = 1, ρ_f is minimized and ρ_s nearly so. According to Anderson [1], there are several criteria that provide an indication as to how to order the variables. One can choose to minimize $||M_{21}||$, or $||M_{12}||$, or ρ_f, or ρ_s. When f = 1, $||M_{12}||$ is definitively minimized, $||M_{21}||$ and ρ_s nearly so, and ρ_f slightly so. For these reasons, the seven variables of the point-kinetics equations are ordered such that the six delayed-neutron precursor groups are placed in the slow mode, and the one variable for the prompt response placed in the fast mode.

Another advantage to this ordering is that because some of the vectors used are 1x6 or 6x1 or even 1x1, computations are greatly facilitated, for reasons outlined earlier.

D. Solution of the Point-Kinetics Equations Utilizing Two-Time-Scale Decoupling Methods

From Hetrick et al. [8], the point-kinetics equations for six groups of delayed-neutron precursors without an external neutron source are expressed as

$$\frac{d}{dt}\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ n \end{bmatrix} = \begin{bmatrix} -\lambda_1 & 0 & 0 & 0 & 0 & 0 & \beta_1/\Lambda \\ 0 & -\lambda_2 & 0 & 0 & 0 & 0 & \beta_2/\Lambda \\ 0 & 0 & -\lambda_3 & 0 & 0 & 0 & \beta_3/\Lambda \\ 0 & 0 & 0 & -\lambda_4 & 0 & 0 & \beta_4/\Lambda \\ 0 & 0 & 0 & 0 & -\lambda_5 & 0 & \beta_5/\Lambda \\ 0 & 0 & 0 & 0 & 0 & -\lambda_6 & \beta_6/\Lambda \\ \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & \frac{\rho-\beta}{\Lambda} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ n \end{bmatrix}, \qquad (4\text{-}50)$$

where

C_j is the delayed-neutron precursor number for the jth group of precursors

n is reactor power

λ_j is the delayed-neutron decay constant for the jth group

β_j is the delayed-neutron fraction for the jth group

β is the total delayed-neutron fraction for all groups

Λ is neutron generation time

ρ is reactivity.

Note that the 7x7 matrix in Equation (4-50) is the A matrix.

Another way of expressing Equation (4-50) is

$$\frac{dC_j}{dt} = \frac{\beta_i}{\Lambda} n - \lambda_j C_j \qquad (4\text{-}51a)$$

$$\frac{dn}{dt} = \frac{\rho-\beta}{\Lambda} n + \Sigma_j \lambda_j C_j \ . \qquad (4\text{-}51b)$$

The numerical method used to solve the equations is the method of finite differences. As developed in this program, Equations (4-51a) and (4-51b) will be solved only for transient quantities. Steady-state quantities will be computed once, and added to transient quantities when total output is desired.

Accordingly, restating Equation (4-51a) as

$$\frac{d}{dt}(C_{io}+\delta C_i) = \frac{\beta_i}{\Lambda} (n_o+\delta n) - \lambda_i (C_{io}+\delta C_i), \qquad (4\text{-}52)$$

where

C_{jo} is steady-state precursor number

δC_j is transient precursor number

n_o is steady-state reactor power

δn is transient reactor power.

Equation (4-52) is further developed as follows.

The steady-state terms can be separated from Equation (4-52). Thus,

$$\frac{d}{dt}(C_{io}) = \frac{\beta_j}{\Lambda}(n_o) - \lambda_j (C_{jo}). \qquad (4\text{-}53)$$

Since C_{jo} is a steady-state variable, it does not change with respect to time, and

$$\frac{d}{dt}(C_{jo}) = 0.$$

Therefore,

$$0 = \frac{\beta_j n_o}{\Lambda} - \lambda_j C_{jo},$$

which leads to the important result that

$$\lambda_j C_{jo} = \frac{\beta_j n_o}{\Lambda}$$

$$C_{jo} = \frac{\beta_j n_o}{\Lambda\lambda_j} \quad . \qquad (4\text{-}54)$$

Solving for the transient variables yields

$$\frac{d}{dt}(\delta C_j) = \frac{\beta_j \delta n_i}{\Lambda} - \lambda_j \delta C_j,$$

or, using a finite difference numerical method,

$$(\delta C_j(i+1) - \delta C_{ij})/h = \frac{\beta_j \delta n_i}{\Lambda} - \lambda_j \delta C_{ij},$$

where C_{ij} is the precursor number from the previous iteration, $C_j(i+1)$ is the precursor number to be computed during the current iteration, and h is the time step used. This method is used in industry and will be used here too. Thus,

$$\delta C_j(i+1) = h(\frac{\beta_i \delta n_i}{\Lambda} - \lambda_i \delta C_{ij} + \delta C_{ij}. \qquad (4\text{-}55)$$

Similarly,

$$\delta_{n_{i+1}} = h(\frac{1}{\Lambda}(\rho_o \delta n + \delta\rho n_o + \delta\rho\delta n_i - \beta\delta n_i)$$
$$+ \Sigma_j \lambda_j \delta C_{ij}) + \delta n_i, \qquad (4\text{-}56)$$

where

ρ_o is the steady-state reactivity (usually zero)

$\delta\rho$ is the transient reactivity

j and i are subscripts referring, respectively, to the six delayed-neutron energy groups and to the current time step

Note that since these equations are not decoupled, the time step h is the same in each equation.

Recall Equation (4-3),

$$X = TY \tag{4-3}$$

In terms of the point-kinetics equations, this is equivalent to

$$\begin{bmatrix} C \\ n \end{bmatrix} = \begin{bmatrix} I_{n_1} & -K \\ -L & (I_{n_2}+LK) \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}. \tag{4-57}$$

Since K is a 6x1 matrix and L is 1x6, the lower right hand entry in the transformation matrix is 1x1 and

$$I_{n_2}+LK = 1.0 + \Sigma_j L_j K_j.$$

Also, the upper left hand entry in the transformation matrix is a 6x6 identity matrix.

Accordingly, restate Equation (4-57),

$$\begin{bmatrix} \delta C_1 \\ \delta C_2 \\ \delta C_3 \\ \delta C_4 \\ \delta C_5 \\ \delta C_6 \\ \delta n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -K_1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -K_2 \\ 0 & 0 & 1 & 0 & 0 & 0 & -K_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & -K_4 \\ 0 & 0 & 0 & 0 & 1 & 0 & -K_5 \\ 0 & 0 & 0 & 0 & 0 & 1 & -K_6 \\ -L_1 & -L_2 & -L_3 & -L_4 & -L_5 & -L_6 & (1.0+ \, {}_i L_i K_i) \end{bmatrix} \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \\ \delta Y_3 \\ \delta Y_4 \\ \delta Y_5 \\ \delta Y_6 \\ \delta Y_7 \end{bmatrix} . \tag{4-58}$$

The only term here that is of interest in program output is the reactor power term δn. The δC_i terms are of no interest. However, their analogues, the Y_i(1-6) terms, are of interest because they are multiplied by the $-L_i$ terms to obtain δn.

Since the C_j terms are of no interest, restate Equations (4-57) and (4-58) to obtain

$$\delta n = \begin{bmatrix} -L & (I_{n_2}+LK) \end{bmatrix} \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \end{bmatrix} \tag{4-59}$$

and

$$\delta n = \begin{bmatrix} -L_1 & -L_2 & -L_3 & -L_4 & -L_5 & -L_6 & (1.0+\Sigma_j L_j K_j) \end{bmatrix} \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \\ \delta Y_3 \\ \delta Y_4 \\ \delta Y_5 \\ \delta Y_6 \\ \delta Y_7 \end{bmatrix}. \quad (4\text{-}60)$$

The Y_i terms themselves are obtained by solving Equation (4-18), which is repeated here.

$$\begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \quad (4\text{-}18)$$

or

$$\begin{bmatrix} \dot{Y}_1 \end{bmatrix} = \begin{bmatrix} B_1 Y_1 \end{bmatrix} \quad (4\text{-}61)$$

$$\begin{bmatrix} \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} B_2 Y_2 \end{bmatrix}. \quad (4\text{-}62)$$

These equations will be solved by a finite-differences method. However, since they are decoupled, they can and will use different size time steps.

For Equation (4-61), use time step h_s; for Equation (4-62), use time step h_f. For reasons that will be explained in Section V.B, the time steps

$h_s = 0.2$ sec

$h_f = 0.001$ sec

were chosen.

Restating Equation (4-18) as

$$\frac{d}{dt}\begin{bmatrix} Y_{10} + \delta Y_1 \\ Y_{20} + \delta Y_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} Y_{10} + \delta Y_1 \\ Y_{20} + \delta Y_2 \end{bmatrix},$$

implies

$$\frac{\delta Y_1(i+1) - \delta Y_{1i}}{h_s} = B_1 \delta Y_{1i},$$

$$\delta Y_1(i+1) = h_s B_1 \delta Y_1 + \delta Y_{1i}.$$

Similarly,

$$\delta Y_2(i+1) = h_f B_2 \delta Y_{2i} + \delta Y_{2i}.$$

These solutions are then substituted back into Equation (4-59) to obtain reactor power.

The preceding analysis has discussed linear systems of differential equations; that is, systems of the form

$$\dot{X} = AX, \qquad (4\text{-}1)$$

where each element of the A matrix is a constant.

Such systems represent either steady-state systems with no transient phenomena or systems in which the only transient phenomena are caused by natural processes. Such a system cannot be purposefully driven, or controlled, to produce desired changes.

However, in the point-kinetics equations (Equation 4-50), the element in the (7,7)th position is

$$\frac{\rho-\beta}{\Lambda} .$$

Because the reactivity ρ is not constant but varies, Equation (4-50) represents a nonlinear system, not a linear one.

There are two ways of solving such a system. The first is to treat ρ as a constant. Doing this will mean that every time ρ changes, the A matrix changes, and will have changed eigenvalues and eigenvectors. Because of this, the L and K matrices also change. In fact, treating ρ as a constant will mean that every time ρ changes it will be necessary to run the entire two-time-scale matrix decoupling algorithm in order to recompute the L and K matrices.

For many reasons this is unsatisfactory. In the two-time-scale matrix decoupling algorithm, the original variables are transformed into their decoupled analogues via a transformation matrix, the L and K matrices are computed, the differential equations are solved, and then the decoupled variables are transformed back into the original variables. From here, feedback and control system effects are determined and a new ρ calculted.

Since so many steps are needed for the calculation of ρ, and from this calculation further values for ρ are computed, there is a significant possibility of progressive arithmetic error if the two-time-scale matrix decoupling algorithm is called every time the value of ρ changes. Further, the two-time-scale matrix decoupling algorithm is a lengthy algorithm which calls on two PORT library subroutines in order to work. This defeats one of the purposes of using the two-time-scale matrix decoupling algorithm, which is the reduction of CPU time by means of the larger time steps that can be used. Last, it is inelegant to frequently use such a long algorithm, if valid results can be obtained by not doing so.

The second way of solving a nonlinear system is to remove the nonconstant variable from the A matrix and rewrite Equation (4-1) as a system of differential equations with a driving function. By this means, the A matrix again contains only constant terms; the nonconstant variable is now located outside the A matrix and is the driving function.

Equation (4-1) is rewritten as

$$\dot{X} = AX + Bu, \qquad (4\text{-}63)$$

where Bu is the driving function.

Hetrick, Girijashankar et al. [8] state that the dynamics of nonlinear systems can be assumed to be well-represented by those of dynamic systems. As long as any changes do not vary greatly from the steady-state conditions, a nonlinear system can be approximated to a high degree of accuracy by a linear system with a driving function. In the program written to support this thesis, it is also possible to use a linear system with a driving function to approximate a nonlinear system in which there are small changes from initial conditions. The method by which this is done will be outlined later.

To develop the decoupled form of the point-kinetics equations with driving function, first the power equation of the point-kinetics equations in numerical form, Equation (4-56), is restated:

$$\delta n_{i+1} = h\left(\frac{1}{\Lambda}(\rho_0 \delta n + \delta\rho n_0 + \delta\rho n_i - \beta\delta n_i + \Sigma_j \lambda_j \delta C_{ij}\right) + \delta n_i . \qquad (4\text{-}56)$$

With little loss of accuracy, the $\delta\rho\delta n_i$ term can be dropped, because it is small compared to the others. This will be done, because the term $\delta\rho\delta n_i$ contains the variable δn_i, which is one of the two variables solved for in the point-kinetics equations. Since the purpose for having a driving function is to separate the variable $\delta\rho$ from the A matrix, the driving function will be a function of $\delta\rho$. It is important that the driving function not contain any of the variables that appear in the point-kinetics equations. The reason is that if it does, the system of coupled point-kinetics equations cannot be decoupled, even if the transformation into a decoupled system is attempted.

To develop the point-kinetics equations with a driving function, first the variable ρ is expressed as a function of its steady-state value and its transient value:

$$\rho = \rho_0 + \delta\rho \tag{4-64}$$

Restating Equations (4-51) and (4-52) using transient variables,

$$\frac{d}{dt}(\delta C_j) = \frac{\beta_j \delta n}{\Lambda} - \lambda_j \delta C_j \tag{4-65}$$

$$\frac{d}{dt}(\delta n) = \frac{1}{\Lambda}(\rho_0 \delta n + \delta\rho n_0 + \delta\rho\delta n - \beta\delta n) + \Sigma_j \lambda_j \delta C_j \tag{4-66}$$

The following system of differential equations with a driving function is obtained:

$$\frac{d}{dt}\begin{bmatrix} \delta C_1 \\ \delta C_2 \\ \delta C_3 \\ \delta C_4 \\ \delta C_5 \\ \delta C_6 \\ \delta n \end{bmatrix} = \begin{bmatrix} -\lambda_1 & 0 & 0 & 0 & 0 & 0 & \beta_1/\Lambda \\ 0 & -\lambda_2 & 0 & 0 & 0 & 0 & \beta_2/\Lambda \\ 0 & 0 & -\lambda_3 & 0 & 0 & 0 & \beta_3/\Lambda \\ 0 & 0 & 0 & -\lambda_4 & 0 & 0 & \beta_4/\Lambda \\ 0 & 0 & 0 & 0 & -\lambda_5 & 0 & \lambda_5/\Lambda \\ 0 & 0 & 0 & 0 & 0 & -\lambda_6 & \beta_6/\Lambda \\ \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & \frac{\rho_0-\beta}{\Lambda} \end{bmatrix} \begin{bmatrix} \delta C_1 \\ \delta C_2 \\ \delta C_3 \\ \delta C_4 \\ \delta C_5 \\ \delta C_6 \\ \delta n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{n_0+\delta n}{\Lambda} \end{bmatrix} \delta\rho \, . \tag{4-67}$$

Equating the δn term in the driving function to zero for reasons explained earlier, Equation (4-67) is expressed as

$$\frac{d}{dt}\begin{bmatrix}\delta C_1\\ \delta C_2\\ \delta C_3\\ \delta C_4\\ \delta C_5\\ \delta C_6\\ \delta n\end{bmatrix} = \begin{bmatrix}-\lambda_1 & 0 & 0 & 0 & 0 & 0 & \beta_1/\Lambda\\ 0 & -\lambda_2 & 0 & 0 & 0 & 0 & \beta_2/\Lambda\\ 0 & 0 & -\lambda_3 & 0 & 0 & 0 & \beta_3/\Lambda\\ 0 & 0 & 0 & -\lambda_4 & 0 & 0 & \beta_4/\Lambda\\ 0 & 0 & 0 & 0 & -\lambda_5 & 0 & \beta_5/\Lambda\\ 0 & 0 & 0 & 0 & 0 & -\lambda_6 & \beta_6/\Lambda\\ \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & \frac{\rho_0-\beta}{\Lambda}\end{bmatrix}\begin{bmatrix}\delta C_1\\ \delta C_2\\ \delta C_3\\ \delta C_4\\ \delta C_5\\ \delta C_6\\ \delta n\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ 0\\ 0\\ 0\\ n_0/\lambda\end{bmatrix}\delta\rho. \tag{4-68}$$

As before, restate the precursor variables δC_j as the slow mode variable δX_1 and the power variable δn as the fast mode variable δX_2. Then, Equation (4-68) can be restated as

$$\frac{d}{dt}\begin{bmatrix}\delta X_1\\ \delta X_2\end{bmatrix} = \begin{bmatrix}A_{11} & A_{12}\\ A_{21} & A_{22}\end{bmatrix}\begin{bmatrix}\delta X_1\\ \delta X_2\end{bmatrix} + \begin{bmatrix}0\\ n_0/\Lambda\end{bmatrix}\delta\rho. \tag{4-69}$$

As before,

$$X = TY \tag{4-3}$$

$$Y = T^{-1}X \tag{4-6}$$

$$T = \begin{bmatrix} I_{n_1} & -K \\ -L & I_{n_2}+LK \end{bmatrix} \tag{4-4}$$

$$T^{-1} = \begin{bmatrix} I_{n_1}+KL & K \\ L & I_{n_2} \end{bmatrix} . \tag{4-5}$$

To decouple the system of differential equations of Equation (4-69), apply the transformation matrix T and substitute Equation (4-3) as before. Equation (4-69) then becomes

$$\frac{d}{dt} T \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} T \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \end{bmatrix} + \begin{bmatrix} 0 \\ n_0/\Lambda \end{bmatrix} \delta\rho$$

$$\frac{d}{dt} \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \end{bmatrix} = T^{-1} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} T \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \end{bmatrix} + T^{-1} \begin{bmatrix} 0 \\ n_0/\Lambda \end{bmatrix} \delta\rho . \tag{4-70}$$

Provided that suitable L and K matrices can be derived, this system can be transformed into

$$\frac{d}{dt} \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \end{bmatrix} + \begin{bmatrix} I_{n_1}+KL & K \\ L & I_{n_2} \end{bmatrix} \begin{bmatrix} 0 \\ n_0/\Lambda \end{bmatrix} \delta\rho$$

$$\frac{d}{dt}\begin{bmatrix}\delta Y_1 \\ \delta Y_2\end{bmatrix} = \begin{bmatrix}B_1 & 0 \\ 0 & B_2\end{bmatrix}\begin{bmatrix}\delta Y_1 \\ \delta Y_2\end{bmatrix} + \begin{bmatrix}Kn_0/\Lambda \\ n_0/\Lambda\end{bmatrix}\delta\rho. \tag{4-71}$$

This is a decoupled system. It can be also be expressed as

$$\frac{d\delta Y_1}{dt} = B_1\delta Y_1 + \frac{Kn_0\delta\rho}{\Lambda} \tag{4-72}$$

$$\frac{d\delta Y_2}{dt} = B_2\delta Y_2 + \frac{n_0}{\Lambda}\delta\rho. \tag{4-73}$$

Once these equations are solved, they are transformed back into the original variables using the transformation matrix T. Equation (4-60) is used as before, without changes.

For whatever size reactivity perturbation, the two-time-scale matrix decoupling algorithm is computed once. The L and K matrices are computed once. Power transients are then computed from an initial point via the driving function.

For further details the reader is referred to the comments section in the subroutine GALBA and to Section V.B.

V. MAIN DEVELOPMENT OF THE PROGRAMS

A. Main Program NERO

The program NERO is the program which controls all the others. It operates by prompting a user to select options or parameters. The user must select an option when prompted to do so, or the program stops. When all the choices have been made, NERO summarizes them on the computer screen, whence they may be transcribed via graphics or printed.

When prompting the user, NERO frequently will provide brief explanations of what is being requested. In general, NERO will present the user with the choice to be made, and then direct him (or her) to make a choice by typing in a number (usually 1 or 2).

Should a user select a number that cannot be used to specify an option, NERO will reject that choice and direct the user to select again. Similarly, if a user selects a parameter (for instance, power level) whose value lies outside permissible limits, NERO will reject that choice and direct the user to try again.

Except for conversions of output data into forms that can be used in graphics or tabular displays, NERO performs no calculations, but rather only receives input parameters as data and then controls subprograms.

Options that the user can select include:

i) fuel isotope used

ii) reactor kinetics with no feedback

iii) reactor feedback with no control system

iv) reactor control system with no natural feedback

v) two-time-scale matrix decoupling algorithm

vi) ramp-input model

vii) prompt-jump approximation

viii) steam valve perturbation instead of reactivity perturbation

ix) output in graphics or a table

x) lengthening of time of run

xi) abbreviation of table output .

Parameters selected by the user include:

i) reactivity and reactivity perturbation

ii) coefficients of reactivity

iii) control system parameters

iv) initial power and power step (if the power step option is selected)

v) magnitude of ramp input and period over which it operates

vi) magnitude of valve perturbation (if that option is selected)

vii) extent to which table output is abbreviated

viii) length of run.

All dimensions used in NERO and its subroutines utilize the SI system of measurements.

B. Reactor Kinetics Subroutine GALBA

The subroutine GALBA solves the reactor kinetics equations. It simulates the operation of a 3000 MWt pressurized water reactor (PWR) operating at a pressure of 2250 psia (or 15,513,875.1 pascals, or 15.514 MPa).

As input GALBA receives the following:

i) step reactivity information

ii) steady-state power information

iii) control system and reactivity parameters

iv) directions on whether the two-time-scale matrix decoupling algorithm, or the ramp input model, or the prompt jump approximation, or none of these will be used to solve for the reactor kinetics equations.

As output, GALBA computes reactor power changes.

GALBA performs power computations using, in most cases, time steps of 0.001 seconds. It performs power computations alternatively with OTHO until 0.2 seconds of reactor time has passed. At this point, NERO causes the program to temporarily terminate GALBA and OTHO computations and pass on to the steam generator subroutine DMTN.

1. Point-kinetics equations

Which ever algorithm or model is chosen, GALBA computes reactivity power via the point-kinetics-equation (4-5) and (4-52). Accordingly, from Section IV:

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda}n - \lambda_j C_j \tag{4-51}$$

$$\frac{dn}{dt} = \frac{\rho-\beta}{\Lambda}n + \Sigma_j \lambda_j C_j, \tag{4-52}$$

where

C_j is the delayed-neutron precursor number for the jth group of precursors

n is reactor power

λ_j is the delayed-neutron decay constant for the jth group

β_j is the delayed-neutron fraction for the jth group

β is the total delayed-neutron fraction for all groups

Λ is neutron generation time

ρ is reactivity.

The numerical method used to solve these equations is the method of finite differences. As developed in this program, Equations (4-51) and (4-52) will be solved only for transient quantities. Steady-state quantities will be computed once, and added to transient quantities when total output is desired.

The development of the numerical form of the point-kinetics equations from Equations (4-51) and (4-52) was presented in Section IV and will not be repeated here. Neither will the development of the two-time scale matrix decoupling algorithm, which was outlined in Section IV.

However, the numerical form of the point-kinetics equations will be repeated here. They are:

$$\delta C_{j(i+1)} = h(\frac{\beta_i \delta n_i}{\Lambda} - \lambda_i C_{ij}) + C_{ij} \qquad (4\text{-}55)$$

and

$$\delta n_{i+1} = h(\frac{1}{\Lambda}(\rho_0 \delta n_i + \delta \rho n_0 + \delta \rho \delta n_i - \beta \delta n_i + \Sigma_j \lambda_j C_{ij}) + \delta n_i, \qquad (4\text{-}56)$$

where

ρ_o is the steady-state reactivity (usually zero)

$\delta\rho$ is the transient reactivity

h is the time step

j is a subscript referring to the delayed-neutron groups

i is a subscript referring to the current time step.

When solving the point-kinetics equations without matrix decoupling,

$h = 0.001$ sec

The basis for this and all other selections for h is developed in Section IV.B.4.

With decoupling,

$h_s = 0.2$ sec,

$h_f = 0.001$ sec,

where h_s is the time step for the slow mode, and h_f is the time step for the fast mode.

One other observation is in order. The power response to a step change in reactivity is characterized by a very rapid transient on the order of the prompt-neutron lifetime, followed by a much more slowly varying response governed by the delayed neutron behavior. If the prompt-neutron lifetime is taken to be essentially zero, then the power level jumps immediately to its slowly varying behavior level. This is the so-called prompt-jump approximation.

According to Hetrick [7], the prompt-jump approximation and the numerical methods of solving the point-kinetics equation are valid when

$$\frac{dn}{dt} \text{ and } \frac{dC_j}{dt}$$

do not vary greatly over a time step. However, during the prompt jump, $\frac{dn}{dt}$ and $\frac{dC_j}{dt}$ do vary greatly. Therefore, all of the subroutines used in this program utilize arbitrarily small time steps during the period of the prompt jump (on the order of 0.001 sec, although for some Λ, the prompt-jump will be even shorter than this).

In the case of the matrix decoupling algorithm, h_s is defined as being 200 times the current value of h_f, where h_f is initially equal to 1.0×10^{-6} sec, but quickly increases to 0.001 sec. Where matrix decoupling is not used, h equals 1.0×10^{-6} sec initially, and likewise quickly increases to 0.001 seconds. Note that when h_f equals 0.001 sec, h_s equals 0.2 sec.

2. Ramp-input model

The usual method of inducing a reactivity perturbation is by introducing a step input of reactivity. That is, a reactivity perturbation is introduced instantaneously.

However, in the "real world", reactivity changes are not instantaneous (although they are sometimes so fast as to be considered nearly so). Usually they are deliberately slow and last several minutes.

To reflect this reality, a ramp-input model can be employed. In it, reactivity is introduced at a certain rate per second (selected by the user), and at the end of a time period (also selected by the user), the ramp input ceases to contribute any more reactivity. The subroutine NERO has safeguards within it that prevent any combination of ramp-input rate multiplied by ramp-input time period to exceed 90% of prompt critical.

The ramp-input model uses the same point-kinetics Equations (4-55) and (4-56), that are used in solving reactor kinetics problems with step reactivity insertion. The only difference is that $\delta\rho$ in those equations varies as a function of time as well as is a function of feedback and the reactor control system. The ramp-input model is also compatible with the two-time-scale matrix decoupling algorithm.

The equations for the ramp-input model are

$$\rho_t = \gamma t \tag{5-1}$$

and

$$\gamma = \rho/\text{sec}, \tag{5-2}$$

where

ρ_t is total reactivity

γ is ramp-insertion rate

t is time

ρ is reactivity.

3. Prompt-jump approximation

The prompt-jump that occurs after a step insertion of reactivity was mentioned earlier, in Section V.B.1. It lasts typically less than 0.001 seconds, and then the temporary absence of delayed-neutron precursors corresponding to the prompt-jump in power acts to inhibit further rapid power changes. Further power changes proceed relatively slowly.

Since the prompt-jump occurs very rapidly, it is possible to approximate it by assuming it takes place instantaneously.

Restating Equations (4-51) and (4-52),

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda} n - \lambda_j C_j \tag{4-51}$$

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \Sigma_j \lambda_j C_j \ . \tag{4-52}$$

When Λ is small and $\rho < \beta$ (always the case in this program), the right-hand side of Equation (4-52) contains a large negative number $((\rho-\beta)/\Lambda)$ and a large positive number $(\Sigma_j \lambda_j C_j)$. Under these circumstances, $\frac{dn}{dt}$ can be set equal to zero, and Equation (4-52) can be restated as

$$\frac{dn}{dt} = 0 - \frac{\rho-\beta}{\Lambda}n + \Sigma_j \lambda_j C_j . \quad (5-3)$$

Therefore,

$$- \frac{(\rho-\beta)}{\Lambda}n = \Sigma_j \lambda_j C_j$$

$$\frac{\beta-\rho}{\Lambda}n = \Sigma_j \lambda_j C_j ,$$

and

$$n = \frac{\Lambda(\Sigma_j \lambda_j C_j)}{\beta-\rho} . \quad (5-4)$$

Feedback or a reactor control system is difficult to use with the prompt-jump approximation in this program. The reason is that in the prompt-jump approximation, the prompt-jump takes place instantaneously. Because it is instantaneous, feedback has no effect until it is over. Then, there have been so many environmental changes caused by the prompt-jump that the resulting feedback overcompensates for the perturbation caused in the prompt-jump, causing divergent power oscillations. This was observed in several trial runs.

One remedy is to estimate a feedback effect before the prompt-jump takes place, and use this effect to modify the prompt-jump itself. With a modified prompt-jump, the resulting feedback may not be as large as it would have been without feedback, and the divergent oscillations may thereby be avoided.

Note that Equation (5-4) contains variables that include both transient and steady-state quantities. Accordingly, Equation (5-4) can be rested as

$$(n_o+\delta n) = \frac{\Lambda(\Sigma_j\lambda_j(C_j+\delta C_j))}{\beta-(\rho_o+\delta\rho)}. \tag{5-5}$$

Equation (5-5) is the equation used in GALBA to solve the point-kinetics equations using the prompt-jump approximation.

4. Choice of time steps

The size of the time step used in numerical solutions of differential equations is crucial. If a time step is too large, then transient phenomena occurring in the system between iterations will lead to divergence away from correct solutions. Time steps that are too small can lead to error through progressive arithmetic error. Also, they can waste CPU time.

Analytical techniques exist whereby the time step that is optimal for a given system can be determined. One of them is given by Hetrick [7].

Consider the differential equations system

$$\frac{d}{dt}\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}\begin{bmatrix} X \\ Y \end{bmatrix}. \tag{5-6}$$

Using finite-difference methods, the solutions are

$$X_{i+1} = (1 + Ah)X_i + BhY_i \tag{5-7}$$

and

$$Y_{i+1} = ChX_i + (1+Dh)Y_i. \tag{5-8}$$

On the other hand, system (5-4) may be integrated in the form

$$X(t) = X(t_o)e^{A(t-t_o)} + B\int_{t_o}^{t} Y(t')e^{A(t-t')}dt'$$

$$Y(t) = Y(t_o)e^{D(t-t_o)} + C\int_{t_o}^{t} X(t')e^{D(t-t')}dt' .$$

During a time interval in which X and Y do not greatly change,

$$X(t') \cong X(t_o)$$

and

$$Y(t') \cong Y(t_o)$$

The integrals may then be evaluated:

$$X(t) \cong X(t_o)e^{A(t-t_o)} + \frac{BY(t_o)}{A}[e^{A(t-t_o)}-1]$$

and

$$Y(t) \cong Y(t_o)e^{D(t-t_o)} + \frac{CX(t_o)}{D}[e^{D(t-t_o)}-1].$$

Letting

$$h = t-t_o$$

$$X(t) = X_{i+1}$$

$$X(t_o) = X_i$$

$$Y(t) = Y_{i+1}$$

$$Y(t_o) = Y_i,$$

where h is the time step,

$$X_{i+1} \cong e^{Ah}X_i + \frac{B}{A}(e^{Ah}-1)Y_i$$

$$Y_{i+1} \cong \frac{C}{D}(e^{Dh}-1)X_i + e^{Dh}Y_i \ .$$

Rearranging terms, Equations (5-7) and (5-8) yields

$$\begin{bmatrix} X \\ \\ Y \end{bmatrix}_{i+1} = \begin{bmatrix} 1+Ah & Bh \\ \\ Ch & 1+Dh \end{bmatrix} \begin{bmatrix} X \\ \\ Y \end{bmatrix}_i = \begin{bmatrix} e^{Ah} & \frac{B}{A}(e^{Ah}-1) \\ \\ \frac{C}{D}(e^{Dh}-1) & e^{Dh} \end{bmatrix} \begin{bmatrix} X \\ \\ Y \end{bmatrix}_i. \tag{5-9}$$

By inspection, it can easily be seen that Equation (5-9) holds only if $|Ah|$ and $|Dh|$ are both small compared to unity. Thus, the larger of A or D will determine the size of the time step.

In order for Equation (5-9) to hold, it will be decided in advance that both ratios $\frac{1+Ah}{e^{Ah}}$ and $\frac{1+Dh}{e^{Dh}}$ will have to be greater than or equal to 0.99.

By trial and error, the criterion that

$|Ah| < 0.1$

$|Dh| < 0.1$

seems to work well. For if

$Ah = 0.1$, then

$1 + Ah = 1.1$,

$e^{Ah} = e^{0.1} = 1.105$,

and

$$\frac{1+Ah}{e^{Ah}} = \frac{1.1}{1.105} = 0.995.$$

For $|Ah| < 0.2$,

$$\frac{1+Ah}{e^{Ah}} = 0.982,$$

which may still be large enough. However, for

$|Ah| < 0.5$,

$$\frac{1+Ah}{e^{Ah}} = 0.910,$$

which is definitely too small.

Here, the criterion that

$|Ah| < 0.1$ will be used.

Repeating the point-kinetics equations,

$$\frac{dn}{dt} = \frac{\rho-\beta}{\Lambda}n + \Sigma_j\lambda_jC_j \qquad (4\text{-}51b)$$

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda}n - \lambda_jC_j. \qquad (4\text{-}51a)$$

Here,

$$A = \frac{\rho-\beta}{\Lambda}$$

$$D = \lambda_i.$$

For U-235,

$$\beta = 6.5 \times 10^{-3}.$$

For all cases,

Λ = 0.0001 seconds was used.

For all the delayed groups, λ_j ranges from about 0.01 to 3.0 sec^{-1}. Its weighted average is about 0.0767 sec^{-1} for U-235, somewhat less for the other isotopes used.

ρ, of course, is variable. However, its absolute value will never exceed $|\beta|$.

For any isotope, then,

$$A = \frac{\beta}{\Lambda}$$

$$D = \lambda_{max}.$$

For U-235,

$$A = \frac{6.5 \times 10^{-3}}{1.0 \times 10^{-4}} = 65.0.$$

For other isotopes, A equals about 20.0.

In all cases,

A>D, and therefore, A dominates.

Set

$|Ah| < 0.1.$

Then

$$h < \frac{0.1}{65.0}$$

$$h < 1.53 \times 10^{-3}.$$

Since 1.53×10^{-3} is not a round number, use

$$h = 1.0 \times 10^{-3} \text{ seconds}.$$

5. Reactivity, feedback, and the reactor control system

Modern reactors are designed such that reactivity changes are to be controlled, either to maintain a given power level, or to dampen any unwanted reactivity excursions.

Reactivity can be dampened by feedback. That is, a given reactivity insertion will cause a power change, which will in turn, cause fuel temperature changes, and moderator pressure and temperature changes. These in turn affect the environment in which neutrons are produced. A change in any or all of these environmental conditions can increase or decrease neutron production rates, thus affecting power levels.

To define reactivity, a few concepts are needed first.

The neutron multiplication factor k is defined as the ratio of the number of fissions in any one generation to the number of fissions in the immediately preceding generation. When $k = 1$, the number of fissions in each generation is constant, and a nuclear chain reaction will proceed at a constant rate. Such a system is said to be critical. Since each fission is caused by a neutron splitting a uranium or plutonium atom, this is equivalent to saying that neutron production equals neutron losses.

Reactivity is defined as

$$\rho = \frac{k-1}{k} .$$

Note that when $K > 1.0$, ρ is positive. This means that the reactor is beyond critical (is "supercritical"), and reactor power is increasing. Conversely, if $k < 1.0$, power will decrease.

The term β is the delayed-neutron fraction. Since it takes on the order of several minutes for delayed-neutron precursors to start producing their share of the neutrons needed to sustain a chain reaction, the presence of delayed-neutron precursors tends to inhibit power changes.

However, when $\rho > \beta$, the prompt neutrons contribute enough neutrons to sustain the chain reaction by themselves, and the delayed neutrons are no longer needed to keep the reactor supercritical. This condition is called "prompt critical", and is a condition that the program will not allow to occur. The point-kinetics equations would still be valid, but power would be changing so quickly that very small time steps would be needed to follow the transient.

Feedback is a phenomenon that refers to the stability of dynamic systems. In general, a perturbation in a system causes environmental changes that in turn affect the conditions under which the system is operating. These altered conditions can change the levels at the system is operating. In reactor kinetics, the two most common feedback mechanisms come from Doppler broadening and moderator temperature changes.

Doppler broadening is a phenomenon that affects neutron absorption, and hence reactor power. At higher energies than thermal, there is a "peak" where the microscopic absorption nuclear cross section is considerably greater than at other energies. As temperature increases, the greater thermal motion causes the energy band of the resonance to widen and the peak to decrease. The total cross section integrated over all energies remain the same. However, at low temperatures, most of the neutron absorption occurs in a small band of resonance energies. This causes neutron absorption to occur mostly in the surface of the fuel and not the interior, through a phenomenon known as "self-shielding". At higher temperatures, the existence of broader resonances mean that there is less self-shielding and greater neutron absorption in the interior. In other words, greater temperatures mean greater neutron absorption, causing reactivity decreases.

Most reactors are designed to operate with thermal neutrons. That is, a moderator (in the United States, usually light water) is placed between fuel assemblies and this moderator slows down neutrons to thermal energies by means of elastic collisions between neutrons and moderator molecules. Since neutrons on the average interact with fuel atoms at optimum energies, the temperature of the moderator can very much affect the energy of a neutron, since temperature is really a measure of moderator energy or how quickly moderator molecules are moving.

If the feedback is such that a perturbation on a dynamic system causes a change that tends to restore the system to a prior equilibrium level, the feedback is considered negative. Otherwise, it is positive; i.e., a negative coefficient of reactivity will cause negative feedback, while a positive coefficient of reactivity will cause positive feedback.

In terms of reactor kinetics,

$$\delta\rho_f = \alpha_f \delta T_f + \alpha_m \delta T_m, \qquad (5\text{-}10)$$

where

$\delta\rho_f$ is change in reactivity due to feedback

α_f is Doppler coefficient of reactivity

δT_f is change in fuel temperature

α_m is moderator temperature coefficient of reactivity

δT_m is change in moderator temperature.

Most reactor designs endeavor to keep feedback negative whenever possible. However, as reference [4] indicates, reactivity coefficients can vary depending on whether the fuel cycle is its beginning, middle, or end. In some cases, reactivity coefficients can be positive.

The computer program simulating this system can handle a range of reactivity coefficient reflecting all of these conditions. By default, median values for both coefficients are selected in case the user makes no changes. Both of these median values result in negative feedback.

The moderator temperature coefficient of reactivity is actually a combination pressure and moderator temperature coefficient. The pressure component comes from a pressurizer, which acts as a kind of surge tank or pressure relief mechanism to counteract any pressure perturbation, such as might be caused by moderator temperature changes. Parameters for the pressure coefficient of reactivity were taken from reference [10]. Since pressure can be directly related to temperature, the pressure and moderator temperature coefficient of reactivity can be reduced to the same dimensions, and therefore, combined into one coefficient, which was done.

While internal feedback may be regarded as a self-adjustment made by a system in reaction to a perturbation, a control system is a means by which a forced adjustment is made on a system in response to a perturbation.

In this program, a control system of the form

$$\frac{\rho_c}{dt} + \frac{1}{\tau_c}\rho_c = A[\delta T_{av} + \frac{1}{\tau}\int_0^t \delta T_{av} dt], \qquad (5\text{-}11)$$

where

ρ_c is reactivity due to the control system

$\frac{1}{\tau_c}$ is the time constant of the differential equation that describes the working of the mechanical actuator.

δT_{av} is change in moderator temperature

$\frac{1}{\tau}$ is a constant used to adjust the effect of the integral portion of the controller

A is control system gain.

The right side of Equation (5-11) represents a proportional controller plus an integral controller, and the left side represents the mechanical actuator.

Like all other equations in this program, Equation (5-11) is solved by finite-difference techniques. First, it assumes the form

$$\frac{\rho_{c(i+1)} - \rho_{ci}}{h} + \frac{1}{\tau_c}\rho_{ci} = A[\delta T_{avi} + \frac{1}{\tau}[h\Sigma T_{av_i}]],$$

which yields

$$\rho_{c(i+1)} = h[A[\delta T_{av_i} + \frac{1}{\tau}[h\delta T_{avi}]] - \frac{1}{\tau_c}\rho_{c_i}] + \rho_{c_i}, \quad (5\text{-}12)$$

where

i is a subscript referring to the current time step in use

h is the time step. In this program, h = 0.2, in other words, the control system insert corrections every 0.2 seconds.

C. Thermal Hydraulics Equations OTHO

The subroutine OTHO solves the thermal-hydraulics equations. The thermal-hydraulics equations are a system of two coupled equations that solve for reactor moderator temperature and reactor fuel temperature. From these data, reactor output temperature is computed and used as input in the steam generator subroutine DMTN. The moderator and fuel temperature changes are used to calculate feedback in GALBA. As input, OTHO utilizes reactor power data from the reactor kinetics subroutine GALBA and steam generator output temperature data from DMTN.

OTHO commences operations every 0.2 seconds of reactor time. Since its own time steps are much smaller than 0.2 seconds, it undergoes several iterations until 0.2 seconds passes, at which time control of the program passes to the steam generator subroutine DMTN.

The thermal-hydraulics equations are

$$C_{p_f} M_f \frac{dT_f}{dt} = n - h_p A(T_f - T_m) \quad (5\text{-}13)$$

$$C_{p_m} M_m \frac{dT_m}{dt} = h_p A(T_f - T_m) - C_{p_m} W(T_o - T_i), \quad (5\text{-}14)$$

where

C_{p_f} is reactor fuel heat capacity (J/kgC)

M_f is mass of reactor fuel (kg)

T_f is reactor fuel temperature (C)

n is reactor power (MW)

h_p is heat transfer coefficient (W/m^2C)

A is heat transfer area (m^2)

T_o is reactor coolant output temperature (C)

T_i is reactor coolant input temperature (C)

C_{p_m} is reactor coolant heat capacity (J/kgC)

M_m is reactor coolant mass (kg)

T_m is reactor coolant temperature (C)

W is reactor coolant mass flow rate (kg/s).

The finite differences method is used to solve these equations. As in GALBA, these equations will be solved for transient quantities only. Accordingly, Equations (5-13) and (5-14) are restated in transient quantity form:

$$C_{p_f} M_f \frac{d\delta T_f}{dt} = \delta n - h_p A(\delta T_f - \delta T_m) \quad (5\text{-}15)$$

$$C_{p_m} M_m \frac{d\delta T_m}{dt} = h_p A(\delta T_f - \delta T_m) - C_{p_m} W(\delta T_o - \delta T_i), \qquad (5\text{-}16)$$

where

δT_f is transient reactor fuel temperature

δT_m is transient reactor coolant temperature

δT_o is transient reactor coolant output temperature

δT_i is transient reactor coolant input temperature.

In a method similar to that used in Section IV for the development of the point-kinetics Equations (4-51) and (4-52), Equations (5-15) and (5-16) are transformed into the finite difference form

$$\delta T_{f(j+1)} = \frac{h}{C_{p_f} M_f} (\delta n_j - h_p A(\delta T_{fj} - \delta T_{mj})) + \delta T_{fj} \qquad (5\text{-}17)$$

and

$$\delta T_{m(j+1)} = \frac{h}{C_{p_m} M_m}((h_p A(\delta T_{fj} - \delta T_{mj})) - C_{p_m} W(\delta T_{oj} - \delta T_{ij})), \qquad (5\text{-}18)$$

where

j is a subscript referring to the current time step

h is the time step.

The reactor core modeled is based on Babcock and Wilcox designs (references [2] and [4]). In these designs, reactor coolant flow is held constant at all power levels. Instead, the coolant temperature at the outlet of the core is allowed to vary directly and linearly with reactor power. Also, reactor inlet coolant temperature varies oppositely but linearly with reactor power; however, since reactor inlet coolant temperature is actually the outlet temperature of the steam generator, there are delays built-in.

Although both reactor coolant inlet and outlet temperatures vary with reactor power, they vary in such a way that the average temperature of the coolant does not change at all. In other words, changes in outlet temperature are offset by temperature changes at the inlet. An increase in one is offset by a decrease in the other, and vice versa. Because of built-in delays, such offsets do not occur immediately; rather, a given change in one quantity will eventually be followed by a negative change in the other.

For this reason, average coolant temperature can change as a transient. Eventually, it converges back toward its steady-state value of 313.89 C. For this reason, the effect of moderator temperature on the reactor is not great.

In general, a linear average temperature was used.

$$T_m = \frac{T_o + T_i}{2.0}, \qquad (5\text{-}19)$$

where

T_m is moderator temperature

T_o is reactor outlet temperature

T_i is reactor inlet temperature

Since transient quantities are used in this program, Equation (5-19) is restated as

$$\delta T_{mi} = \frac{\delta T_{oi} + \delta T_{ii}}{2.0} , \qquad (5\text{-}20)$$

where

δT_{oi} is the transient reactor outlet temperature (that is, the difference between current temperature and initial temperature)

δT_{ii} is the transient reactor inlet temperature

Equation (5-20) leads to the important result that

$$\delta T_{oi} = 2.0 \times \delta T_{mi} - \delta T_{ii} . \qquad (5\text{-}21)$$

Since heat capacity of reactor coolant at constant pressure is an exponential function of temperature, the average heat capacity varies slightly (less than 1%) as a function of power, even though the average temperature itself remains constant. This is because the inlet and outlet temperatures of the coolant vary as a function of power, and at these extremes, heat capacity does not vary linearly. A correlation for average heat capacity was derived as a function of temperature, also by a least-squares fitting using an exponential model.

Because average heat capacity is not exactly constant, neither is reactor coolant flow rate. Depending on initial power level, coolant flow rate is fixed by NERO at the beginning of the program run. It remains constant thereafter.

Since average heat capacity of the reactor coolant is slightly dependent upon reactor power, so is the initial reactor coolant temperature. Initial reactor coolant temperature is computed at the beginning of the program run, and changes in coolant temperature are computed for powers different from this initial point.

The values of the coefficients of the expressions in Equations (5-17) and (5-18) need to be developed.

To develop the time step h, recall Equation (5-6),

$$\frac{d}{dt}\begin{bmatrix} X \\ X \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}. \qquad (5\text{-}6)$$

where the larger term A or D will determine the size of the time step. In the thermal-hydraulics Equations (5-17) and (5-18), the A-term must be compared to the D-term, where

$$A = \frac{h_p A}{C_{p_f} M_f} ,$$

$$D = \frac{h_p A}{C_{p_m} M_m} .$$

Typical values for these terms are:

$h_p = 34000\ W/m^2K$

$A = 5945.0\ m^2$

$C_{p_f} = 160$ J/kgC

$M_f = 95000$ kg

$C_{p_m} = 6000$ J/kgC

$M_m = 13300$ kg.

Thus,

$A = 13.3$

$D = 2.5,$

and

$A > D.$

Setting

$|Ah| < 0.1,$

$13.3\ h < 0.1$

$h < 7.52 \cdot 10^{-3}.$

h will be taken to be 0.005 seconds, except for arbitrarily small time steps at the beginning of the program run to account for the prompt jump in power that has taken place in GALBA.

From reference [10] (commonly referred to as the "Steam Tables"), converted into SI dimensions, formulae for the physical quantities of heat capacity of water, thermal conductivity of water, density of water, and kinematic viscosity of water were derived. Where possible, least-squares fittings using an exponential model were fitted to the data points from reference [9]. These correlations are functions of temperature, at constant pressure. Separate correlations were derived for reactor pressure (2250 psia) and steam generator secondary side pressure (900 psia).

From reference [2], a value of 191,000 kg for the mass of the fuel was obtained. Since differences in density between the three fuel isotopes that can be used are so slight, this value is fixed for all isotopes and reactor conditions.

A linear correlation for heat capacity of fuel was derived from data contained in reference [6].

A permanent value of 5945 m^2 for the heat transfer area of the fuel assemblies was obtained from reference [4]. A correlation was derived from the same reference relating average fuel temperature to reactor power. At the beginning of the program run, a starting fuel temperature is computed. Fuel temperature changes around this point are computed as the program progresses. These fuel temperature changes are the fuel temperature changes used in computing feedback caused by Doppler broadening.

Since heat transfer is actually dependent upon the surface temperature of the body from which heat is being transferred, the thermal-hydraulics equations are solved for changes in moderator temperature and fuel cladding temperature. After changes in cladding temperature are computed, changes in fuel temperature are computed as a linear function of changes in cladding temperature.

The heat transfer coefficient is based upon a correlation originally from Rohsenow (reference [9]). It is based upon three dimensionless numbers: the Reynolds number, the Prandtl number, and the Nusselt number. The Reynolds number is a quantity that describes the type of flow that a fluid is undergoing in a specific geometry; laminar or turbulent. The Prandtl number is a measure of how rapidly momentum is dissipated compared to the rate of diffusion in a fluid. For water under the conditions encountered in this program, its value is always approximately equal to one. The Nusselt number is a measure of the ratio of the thermal resistance of the fuel assemblies to the thermal resistance of the coolant.

The Reynolds number is expressed as Re.

$Re = UD/\nu$;

where

U is flow velocity of reactor coolant (m/sec)

D is diameter of fuel assemblies (by reference [4], D = 0.12 m

ν is kinematic viscosity (m^2/s).

$\nu = \mu/\rho$

where

μ is dynamic viscosity (kg/m s)

ρ is density of reactor coolant (Kg/m^3)

Correlations for μ and ρ were derived from data points obtained in reference [10].

The Prandtl number is expressed as Pr.

$$Pr = C_{pm} \nu \rho / K,$$

where

C_{pm} is heat capacity of reactor coolant (J/kgC)

ν is kinematic viscosity (m^2/sec)

ρ is density of reactor coolant (kg/m^3)

K is thermal conductivity of reactor coolant (W/mC).

As with the other variables, a correlation for K was derived from data points obtained from reference [9].

The Nusselt number is expressed as Nu. By reference [8],

$$Nu = 0.025 \, Re^{0.8} Pr^{0.6}.$$

Finally, the heat transfer coefficient h_p is given as

$$h_p = Nu \, k/D.$$

Table 2 lists the correlations for the physical quantities mentioned in this section, along with the type of fit and the correlation coefficient R^2 (which is a measure from 0 to 1 of how good the fit is), where available.

Table 2. Correlations for physical constants of H_2O as a function of temperature

Variable	Correlation	Type of fit	Pressure	R^2
Kinematic viscosity ν	$6.1777117 \cdot 10^{-12} T^2 - 3.20997 \cdot 10^{-9} T + 5.5038552 \cdot 10^{-7}$ m^2/s	Parabolic	15.514 MPa (2250 psia)	none (only 3 data points)
Thermal conductivity k	$0.7207553673 - 4.5873157 \cdot 10^{-3} \cdot EXP (0.012380238\ T)$ W/m C	Exponential	"	0.9995
Heat capacity C_p	$4992.4097749 + 2.49340775 \cdot 10^{-4} \cdot EXP (0.04825458\ T)$ J/kg C	Exponential	15.514 MPa (2250 psia)	0.9999
Density ρ	$881.6309649 - 2.86514041 \cdot EXP (0.0133034152\ T)$ kg/m^3	Exponential	"	0.9999
Heat capacity at 15.514 MPa and 3.3189 C as function of power	$5916.241929 + 16.32498553 \cdot EXP (6.4880554 \cdot 10^{-10} \cdot Power)$	Exponential	"	0.9995

D. Steam Generator Subroutine DMTN

The purpose of the steam generator is to convert the thermal energy contained in reactor coolant into steam which can be used by the turbo-generator. Although this program does not concern itself with electric energy, nevertheless, the reactor power must be eventually transferred to the steam generator, and then to the turbine generator. It is the transfer of thermal energy that this program concerns itself with.

The steam generator is made up of two sides - the primary side and the secondary side. The primary side is the side that contains reactor coolant water, which is to be cooled by transferring its energy to the secondary side. As part of the reactor coolant system, the primary side operates around an average temperature of 313.89 C, and a pressure of 2250 psia (15.51 MPa). The secondary side operates in a temperature range of from 235 (42.76 degrees C subcooled) to 311.1 C (33.3 degrees C of superheat) at a pressure of 900 psia (16.21 MPa). It operates at an average temperature of 277.76 C, which is boiling temperature at this pressure. More than 83% of all heat transfer to the secondary side takes place in transforming saturated liquid to saturated steam vapor, without altering the temperature.

Unlike the reactor, the secondary side of the steam generator does not conduct power changes by allowing inlet and outlet temperatures to vary. Rather, the coolant at the inlet is assumed to be at a constant temperature of 235 C (reflecting the fact that it is rejected water from the steam turbine), and coolant flow rate is varied to maintain a constant outlet temperature of 311.1 C (which corresponds to 33.3 C of superheat). This means that coolant flow rate can vary anywhere from 0 kg/sec to 1577 kg/sec, depending on whether the reactor is operating at zero power, full power, or anything in between.

The steam generator is based on designs obtained from references [2] and [4]. As modeled in this program, it has an inside diameter of 3.5 m, and a height of 20.0 m. Primary side coolant flows through 15,500 tubes with a diameter of 0.016 m each. Cross sectional area for secondary flow is 6.5 m^2. Heat transfer area is 22400 m^2. Mass, including water, outer walls and tubing, is 250,000 kg.

The steam generator is basically a gigantic one-pass counter-flow heat exchanger, labeled thus because the primary and secondary sides flow in opposite directions and pass by each other only once. As in any heat transfer system, power is exchanged between primary and secondary sides as a direct, linear function of the temperature difference between the two sides. Since this temperature difference is not the same for all areas of the steam generator, an average temperature difference is computed, based upon the inlet and outlet temperatures of both the primary and secondary sides. This temperature difference is called the Logarithmic Mean Temperature Difference (LMTD) and is defined as

$$\mathrm{LMTD} = [(T_{poi}-T_{soi})-(T_{pii}-T_{sii})]/[\ln[(T_{poi}-T_{soi}) - (T_{pii}-T_{sii})]], \quad (5\text{-}22)$$

where

T_{poi} is the inlet temperature on the primary side (outlet temperature, from the reactor)

T_{soi} is the outlet temperature on the secondary side (always equal to 311.1 C)

T_{pii} is the outlet temperature on the primary side (inlet temperature, to the reactor)

T_{sii} is the inlet temperature on the secondary side (always equal to 235.0 C).

In Section V.C, the heat transfer coefficient was found to be a function of the Reynolds number raised to the 0.8 power (reference [7]). Since the Reynolds number is a linear function of coolant velocity, which in turn is a linear function of mass flow rate, it follows that the heat transfer coefficient is a function of the mass flow rate raised to the 0.8 power.

Based upon the known quantities of heat transfer area, and mass flow rates and LMTDs at various steady-state power levels, heat transfer coefficients were computed for each of six different power levels from 0 MW to 3000 MW. Using these results as data points, a least-squares fitting was derived, assuming that h is a function of mass flow rate to about the 0.8 power. This least-squares fitting has an R^2 correlation coefficient of 0.98, with a much higher R^2 value for power levels above 1000 MW. Accordingly, the heat transfer coefficient h_s used in DMTN is given as

$$h_s = 9.726\ W^{0.806}, \qquad (5\text{-}23)$$

where

W is mass flow rate (kg/sec).

The quantity 9.726 is a constant of proportionality. The heat transfer coefficient h_s has dimensions of W/mK.

DMTN solves the steam generator equations to compute the thermal power output of the power plant. The steam generator equations are a system of two coupled differential equations that solve for the temperature of the primary side of the steam generator and the temperature of the secondary side of the steam generator.

The steam generator equations, given in transient quantities are

$$(M_{msg}Cpm)\frac{d}{dt}\ T_{sg} = C_{pm}W_p(T_{sgo}-T_{sgi}) - h_sA(T_{sg}-T_s) \qquad (5\text{-}24)$$

and

$$C_{ptm}\ \frac{d}{dt}\ T_s = h_sA(T_{sg}-T_s) - W_sD_{hfg}, \qquad (5\text{-}25)$$

where

M_{msg} is the mass of the coolant in the primary side of the steam generator

C_{pm} is the average heat capacity of the coolant in the primary side adjusted slightly for reactor power (J/kgC)

T_{sg} is temperature of the coolant of the primary side

W_p is mass flow rate of the primary side

T_{sgo} is inlet temperature of the primary side

T_{sgi} is outlet temperature of the primary side

h_s is the heat transfer coefficient between the primary side and the secondary side. It is a term that combines forced convection and boiling heat transfer (W/m^2C)

A is heat transfer area (always 22400.0 m^2)

T_s is temperature of the secondary side of the steam generator

C_{ptm} is a composite term consisting of the sum of (heat capacity of iron times mass of steam generator) plus (average heat capacity of secondary side water times quantity of secondary side water). It represents the total specific heat of the secondary side. It equals 2.54×10^8, and has the dimensions of (J/C)

W_s is the mass flow rate of the fluid on the secondary side (kg/sec)

D_{hfg} is quantity of energy required to raise one kg of water at 900 psia from 235.0 C to vapor at 311.1 C. It equals 1,901,744.17 J/kg.

It should be noted that the change of temperature variable on the secondary side reflects average conditions. More than 83% of all energy transfer takes place in the two-phase region, where there is no temperature change at all upon addition of energy. Any actual temperature changes take place in the subcooled and superheated regions.

Like all other systems of coupled equations in this program, Equations (5-24) and (5-25) are solved for transient quantities, using finite difference methods. Accordingly, they are restated in transient quantity form as

$$\delta T_{sg(i+1)} = \left(\frac{h}{M_{msg}C_{pm}}\right)[((C_{pm}W_p)(\delta T_{o(i-24)} + TD - \delta T_{ii})) - ((h_s A)(\delta T_{sgi} + LMTD - \delta T_{si}))] + \delta T_{sgi} \tag{5-26}$$

and

$$\delta T_{s(i+1)} = \left(\frac{h}{CPTM}\right)[((h_s A)(\delta T_{sgi} + LMTD - \delta T_{si})) - (W_s D_{hfg})], \tag{5-27}$$

where

TD is the initial temperature difference between inlet and outlet temperatures on the primary side.

A few remarks must be made concerning Equations (5-26) and (5-27). Although these equations are solved for transient quantities, transient temperature differences are not used in the right side. This is because the heat transfer coefficient h_s is based upon full power transfer, not transient power transfer, and the full temperature instead of transient differences must be used to produce accurate results. Since the derivative of the full variable equals the derivative of the transient quantity, no inaccuracy is introduced.

Also, Equation (5-27) is solved only for the change in secondary side temperature that has occurred during that time step. Cumulative, or total, transient temperature is not computed. This is reflected in the absence of an isolated δT_{si} term in the right side of Equation (5-27). The reason this is done is that it is considered that after each time step, secondary side flow rate is changed to the extent necessary to absorb any temperature change. This will ensure that for any power level, secondary side outlet temperature is always 311.1 C.

Analysis of the steam generator equations led to the result that the ratio of energy transferred to the change of temperature on the secondary side is 50,600.0. This leads to Equation (5-28).

$$E_t = 50{,}600.0 \ \frac{J}{C} \ \delta T_{si} \ C, \qquad (5\text{-}28)$$

where

E_t is energy transferred from the primary side to the secondary side per kilogram coolant (J/kg)

δT_{si} is the change of temperature on the secondary side.

Over an extended period, steam generator power could be computed by adding E_t to D_{hfg}, and multiplying this quantity by W_s. This would represent the old power level plus any power changes caused by transient temperature differences on the secondary side, which themselves are caused by changes in power transferrals from the primary side to the secondary side.

However, a modification to this procedure must be made. The reason is that without such a modification, calculations cannot be correctly performed.

Recall Equations (5-18) and (5-21).

$$\delta T_{oi} = 2.0 \times \delta T_{mi} - \delta T_{ii}, \tag{5-21}$$

where

δT_{oi} is the change in reactor outlet temperature in a given time step

δT_{mi} is change in reactor coolant temperature

δT_{ii} is the change in reactor inlet temperature,

and

$$\delta T_{m_{(i+1)}} = \frac{h}{C_{pm} M_m} [(h_p A(\delta T_{ei} - \delta T_{mi})) - (C_{pm} W_p (\delta T_{oi} - \delta T_{ii}))] . \tag{5-18}$$

Ideally, after a power step, reactor outlet and inlet temperatures should converge toward the values they would have had if the power step had been part of the original power. This would mean that δT_{oi} and δT_{ii} would change by the same amount, albeit with opposite signs. For this reason,

δT_{oi} and δT_{ii} would cancel each other out and δT_m would remain at zero.

During the discussion to follow, the concept of the temperature that would prevail under steady-state conditions at a new power level will be utilized. This refers to the equal and opposite changes in δT_{oi} and δT_{ii} that were discussed in the previous paragraph.

However, because this program has delays built it takes 11 seconds for reactor coolant to make a complete circulation. If, for instance, the reactor is operating at steady-state and then a reactivity step is inserted, reactor moderator and outlet temperatures will immediately change, because δT_{ii} still equals zero. Because δT_{ii} still equals zero, it does not now cancel δT_{oi} to produce an average moderator temperature change of zero (see Eq. 5-21). Because of this and the fact that δT_{ii} still equals zero, the δT_{oi} term in Equation (5-21) is twice as large as it would be if the δT_{ii} term had the value to which it ought to be converging at the new power level.

This causes no problems in OTHO. Application of the thermal-hydraulics equations (Equations (5-17) and (5-18)) will cause a certain amount of power to be transferred from fuel to coolant for a given sum of δT_{oi} plus the negative of δT_{ii}, keeping in mind that under steady-state conditions,

δT_{oi} is equal and opposite in sign to δT_{ii}. If δT_{ii} equals zero, the same level of power will be transferred if δT_{oi} equals twice the value that it would have for steady-state conditions at the new power level. This in fact is what happens in OTHO.

However, if uncorrected, this wreaks havoc in DMTN. For there too, energy transfer from the primary side is dependent only on the temperature difference between the steam generator inlet and outlet, since primary side flow rate is constant. If the reactor outlet temperature change δT_{oi} coming into the steam generator is twice as large as its steady-state value for the new power level, then the new δT_{ii} that it computes will equal zero, in a mirror image of the process that takes place in OTHO. Total power transferred remains the same, except for a small distortion caused by that fact that the temperature difference between primary and secondary sides is different from what it would be under steady-state conditions.

Since δT_{ii} is calculated by DMTN to be equal to zero, δT_{ii} remains zero when the coolant returns to the reactor. Thus, there is no δT_{ii} to reduce δT_{oi} back toward the value it would have under steady-state conditions at the new power level. Furthermore, δT_{mi} does not converge back toward zero, as it would if δT_{ii} had the value that it would have under steady-state conditions at the new power level, This distorts reactor performance, as there is feedback associated with any value for δT_{mi} other than zero.

The fact that δT_{oi} is twice as large as it "ought" to be, can be used by DMTN as a criterion for boosting W_s to compute a δT_{ii} that is equal to the value that it would have under steady-state conditions at the new power level. After the delay needed to transport the coolant back to the reactor, this value of δT_{ii} will cause the moderator temperature change to converge back toward zero and the magnitude of δT_{oi} to converge toward the value it would have under steady-state conditions at the new power level (see Equation 5-2).

This would mean that until δT_{oi} is reduced, the steam generator power change is 50% greater than the reactor power change. This may seem impossible, but it is important to remember that, over an extended period of time, not only must reactor power equal steam generator power, but total reactor energy output must equal total steam generator energy output. If, for instance, the reactor had boosted power by 100 MW, it would be producing this extra energy and increased δT_{oi} for five seconds until the hotter coolant arrived at the steam generator. Even if the steam generator then boosted its power immediately by 100 MW to match the reactor, the fact would remain that the reactor would have produced 500 MJ more energy than the steam generator over an extended period of time. Since this is impossible, it follows that in order to make up the energy deficiency, the steam generator must temporarily produce more power than the reactor. Since inlet and outlet temperatures on the secondary side are to be constant, this must be done by varying the flow.

Introduce the variable TS. TS equals the difference between actual δT_{oi} and the value that it would have under steady-state conditions at the new power level. It is the negative of what δT_{ii} would be under steady-state conditions at the new power level.

Introduce the variable W_{sa}, where W_{sa} is the mass flow rate change (kg/s) on the secondary side needed to provide the power boost needed to make up the energy deficit.

$$W_{sa} = 49.9035 \times TS. \qquad (5\text{-}29)$$

Analysis of the steam generator equations led to the result that the mass flow rate on the secondary side (under steady-state conditions) equals 49.90 times the total difference between steam generator inlet and outlet temperatures. This is the origin of the constant of proportionality 49.90 in Equation (5-29).

TS represents half of the total temperature change between steam generator outlet and inlet on the primary side that would occur if a given reactor power change caused the outlet and inlet temperatures of the steam generator to converge toward the values they would have under steady-state conditions at the new power level. In actuality, δT_{ii} at this point still equals zero. Since δT_{oi}, at this point, is twice as large as it would be if the steam generator were converging toward steady-state conditions at the new power level, the δT_{ii} level toward which the steam generator would ideally be converging under steady-state conditions at the new power level is equal to half of the negative of δT_{oi}, or exactly equal to half of the negative of TS. By changing the mass flow rate on the secondary side by an amount equal to 49.90 times TS, the steam generator heat transfer coefficient and steam generator power will change, causing δT_{ii} to converge toward the value it would have under steady-state conditions at the new power level.

The change in energy transfer from primary to secondary side in any given time step caused by the flow of W_{sa} is given by

$$DENTC_i = 50{,}600.0 \times h \times (W_{sai} - W_{sa(i-1)}) \times D_{hfg}/CPTM, \qquad (5\text{-}30)$$

where "i" and "i-1" subscripts on W_{sa} refer to the current and to the most recent time step, respectively.

To compute the change in power transfer from the primary to the secondary side caused by the flow of W_{sa}, the total sum of DENTC over all time steps is computed:

$$\text{TDENTC} = \sum_{i=1} \text{DENTC}_i . \tag{5-31}$$

The power change caused by the flow of W_{sa} is computed by Equation (5-32) as

$$\text{PWRCH} = W_{sa} \ (\text{TDENTH} + D_{hfg}). \tag{5-32}$$

Steam generator power is computed by Equation (5-33)

$$\text{PWR} = W_s (\text{DENTH}+D_{hfg}) + \text{PWR} . \tag{5-33}$$

After the new steam generator power is computed, a new W_s is computed by means of Equation (5-34).

$$W_s = (\text{PWR}-\text{PWRCH})/D_{hfg} . \tag{5-34}$$

The variable W in Equation (5-23) stands for the sum of W_s and W_{sa}. Because of the effect of W_{sa}, the heat transfer coefficient h_s has a different value from what it would have if it were computed on the basis of W_s alone. This results in an improved value for δT_{sg}.

δT_{ii} is now computed by means of Equation 5-35.

$$\delta T_{ii} = 2.0 \times \ T_{sgi} - T_{oi} . \tag{5-35}$$

Because the secondary side flow rate has been adjusted by means of the W_{sa} term, the δT_{ii} term that is now computed is the value for δT_{ii} that would prevail under steady-state conditions at the new reactor power level. This δT_{ii} term then travels to the reactor, where, via Equations (5-17) and

(5-18), δT_{mi} is brought back toward zero and δT_{oi} is brought back to the value it would have under steady-state conditions at the new power level. Coolant at the new temperature for δT_{oi} then travels back to the steam generator. There, W_{sa} and PWRCH are both caused to converge back to zero, ending the power boost needed to make up the previously mentioned energy deficit.

In actuality the process is not this simple, except for the special case of a step change in reactor power that does not change after insertion of the step. The reason the process is not this simple is that by the time the coolant at the temperature of δT_{ii} reaches the reactor, reactor power itself has changed, and the δT_{ii} signal is not powerful enough to quickly cause δT_{mi} to converge back to zero or δT_{oi} to converge toward its steady-state value at the new power level. Furthermore, any changes in δT_{mi} will cause additional power changes through feedback, masking the process further.

DMTN is also capable, within limits, of load following. Load following is the fixing of the steam generator at a given constant power level, different from current reactor power, and then allowing the reactor to converge toward the steam generator power through feedback.

Secondary side flow at full power (3000 MW) equals

1577.72 kg/sec. This can be throttled down through the use of a valve. Introduce the variable VO. VO is the fraction that represents the percentage (from 0% to 100%) of full power flow that the valve allows to pass.

If load following is desired by the user, it is selected in NERO. If it is selected, the user then decides how large the valve opening is to be. If the choice for VO results in a steam generator power level that is not within 10% of current reactor power, NERO will reject that choice and instruct the user to select again.

Once load following is selected, if the selected steam generator power is different from reactor power, steam generator outlet temperature will be immediately affected, but inlet temperature will not change at all. δT_{ii} will immediately assume some nonzero value, while δT_{oi} will still remain equal to zero. When the coolant at the new temperature of δT_{ii} arrives at the reactor inlet, it will immediately affect the value of the moderator temperature, which in turn, will affect reactor power through feedback. The new reactor power level will then determine what the reactor outlet temperature δT_{oi} will be. Eventually, reactor power and total energy output will equal that of the steam generator.

Except for arbitrarily small time steps during the first 0.1 seconds to take the prompt-jump into account, the time step in DMTN always equals 0.2 seconds.

An unsuccessful attempt was made to devise a steam generator subroutine that divided the steam generator into five heat-transfer regions: one node for superheat flow, three for two-phase flow, and one for subcooled flow. Heat transfer coefficients appropriate to each region were devised. The system broke down because of a lack of good correlations for heat transfer in the two-phase region and because the system was extremely complicated, requiring many calculations to achieve results similar to those that can be achieved by using a simple model with few calculations.

E. Graphics Subroutine VESPASIAN

Should the user wish it, program output can be displayed graphically instead of in a table. The graphics output consists of 3 displays, successively drawn. After one display is drawn, the program does not draw a successive display until the user signals that this is desired. All three displays project power plant phenonema such as power and temperature as functions of time in all cases.

All graphics are displayed inside an artificial window drawn on the computer terminal screen. This screen has viewing conveniences such as tic marks and labels. Because some phenomena such as reactor and steam generator power are so close to each other, at the user's option the graphics display can be expanded in order to show contrasts better. Also, when two or more phenomena are simultaneously drawn in the same display, different patterns of dashed lines for each phenomenon are used. If the user has previously chosen to allow the program to run longer than its default time period, the labeling in each display will accurately depict this. Depending on the type of computer terminal the user is using, hard copies of graphics output can be obtained.

The first graphics display shows reactor power and steam generator power. The second display shows average reactor fuel temperature. The third shows total changes in reactor outlet temperature, reactor inlet temperature, and average moderator temperature.

For further information, the reader is directed to the comments statements of the subroutine VESPASIAN.

VI. TESTS AND RESULTS

Many test runs were made to illustrate the effects of different options chosen. In order to more clearly depict the effects of a given option, the reactor had an initial power level of 2000 MW in all cases. Except where otherwise noted, pre-perturbation reactivity was always zero, any reactivity insertion was always 10 cents, U-235 was the fuel, and default values were used for system parameters.

All output to be discussed here is graphics output. Graphics output in any run consists of three displays, drawn successively. Display 1 plots both reactor power and steam generator power. A user can study the effects that a perturbation in one system has on the other. Display 2 plots fuel temperature only. Display 3 plots changes in reactor inlet temperature, reactor outlet temperature, and average moderator temperature.

All displays have the same general format. They consist of a window (referred to in graphics as a "viewport") enclosing the plots. As all plots are a function of time, numbers denoting point-in-time (in seconds) are written where appropriate in the viewport. Tic marks corresponding to these points in time are superimposed on all plots, as an aid in interpreting them. Tic marks on the vertical axis aid in interpreting response as a function of time. In cases where more than one plot appears in the same display, different dash patterns are used for the lines of each plot. A legend describing the type of dash pattern associated with each plot appears under the viewpoint.

The reactor coolant operates on an 11-second cycle. That is, a given perturbation in the reactor will require 5 seconds before its effects arrive at the steam generator. Steam generator changes then take 3 seconds to be completed, while these changes then take another 3 seconds to arrive back at the reactor, completing the loop.

Thus, a given reactor perturbation will take 5 seconds before it affects steam generator performance, and another 6 seconds before those effects cause further reactor perturbations through feedback. These new reactor perturbations will cause a third generation perturbation at time equals 22 seconds, and a fourth generation perturbation at 33 seconds. As can be seen in Figure 1, these perturbations dampen out rapidly. The sixth generation perturbation at 56 seconds is barely perceptible.

Figure 1 depicts the performance of the point-kinetics equations with no modifications. Initial power is 2000 MW. The perturbation in the reactor consists of a step insertion of 10 cents of reactivity.

There are many points of interest in Figure 1. First is

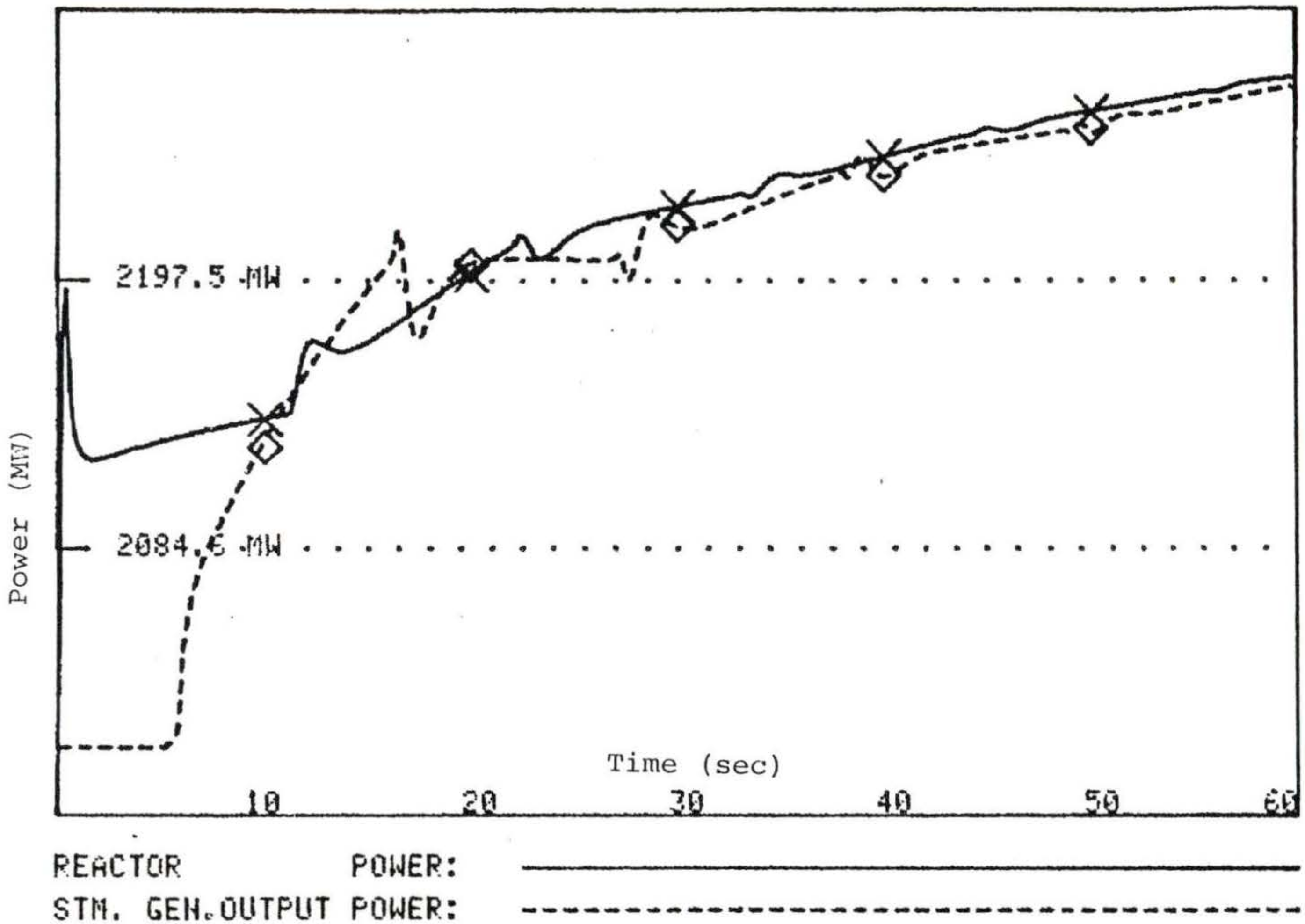

Figure 1. Reactor and steam generator power with feedback. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel

the existence of the prompt jump, at time of less than a second. Note that in less than half a second, reactor power jumps from 2000 MW to almost 2200 MW, then drops almost as rapidly to about 2110 MW.

The reason for this is feedback. While a given reactor power step can occur very rapidly, because they have heat capacities, fuel and moderator temperatures do not rise as quickly. Since feedback reactivity is directly dependent upon fuel and moderator temperatures, it is possible for reactor power to rise to a given level before the feedback that would normally be associated with that level can actually be generated. Because of this, reactor power can rise higher than it otherwise would.

However, while fuel and moderator temperatures do lag behind the prompt jump, they can reach their proper levels in less than a second. While this is much longer than the prompt jump, which takes place in less than a millisecond, it is still quite rapid. Since the prompt jump is finite, it ceases its rapid climb in a very short period, giving fuel and moderator temperatures the chance to reach their proper levels. Figures 11 and 12 show that this is mostly accomplished by one second after the reactivity insertion.

After fuel and moderator temperatures approach their proper levels, there is insufficient reactivity to support the high power levels. Thus, a rapid power decrease, or falloff, takes place as Figure 1 indicates. Actually, even though a power decrease takes place, reactivity is still positive, as is shown by the fact that the power change always remain positive. The reason reactor power decreases anyway is that the delayed-neutron precursor density is still not great enough to support the new power level. This is a dramatic illustration of the importance of delayed-neutron precursors in reactor control Hetrick [7] states that this is a common phenomenon in thermal reactors with large negative coefficients of reactivity.

In the absence of any further feedback, the slope of the power plot between 1 second and 11 seconds in Figure 1 indicates that reactor power would reach the level of the peak of the prompt jump again at about one minute. This agrees nicely with the fact that the delayed-neutron precursors have a half-life on the order of one minute, and takes about that long to build up.

During the first 11 seconds, the steam generator is increasing its power output in response to increased reactor power. This results in a decrease of steam generator outlet temperature. This is the same thing as a decrease in reactor inlet temperature, since coolant flows from steam generator to reactor.

When this coolant with lowered temperatures arrives at the reactor at 11 seconds, it immediately lowers the average moderator temperature, as is shown in Figure 16. This in turn quickly lowers fuel temperature, as is seen in Figure 15. The rapid lowering of fuel and moderator temperatures adds a large reactivity step because of feedback, resulting in another rapid power rise at 11 seconds. This shows a power falloff similar to the one that took place after the prompt jump. Note that the prompt jump and the power falloff that take place at 11 seconds are not nearly so large as those that take place in the first second, even though they are caused by the same phenomena. Note too that the pattern of successive generations of rapid power rises followed by rapid power falloffs every eleven seconds shows a tendency to dampen out. As mentioned previously, the sixth generation, at 56 seconds, is barely perceptible.

The steam generator responds to reactor perturbations five seconds after the reactor perturbation takes place. As shown in Figure 16, reactor outlet temperature shows a drop at 11 seconds followed by 11 more seconds of more or less constant temperatures. This results in a steam generator power drop 5 seconds later at 16 seconds, followed by a recovery and essentially constant power output until 28 seconds. This pattern of power falloff and recovery repeats itself every 11 seconds until it too is dampened out at about 60 seconds. Overall, the steam generator is capable of matching any reactor power change with only minor lagging.

Figures 2 and 3 also depict the performance of the unmodified point-kinetics equations with a reactivity step increase of 10 cents from an initial power level of 2000 MW. The difference is that in Figure 2, U-233 is used as fuel, and in Figure 3, Pu-239 is used. The main difference between Figures 2 and 3 on the one hand, and Figure 1 on the other, is that the power falloff after the prompt jump in Figures 2 and 3 is not as rapid as in Figure 1. This is because the Doppler coefficients of reactivity used for U-233 and Pu-239 in Figures 2 and 3 are smaller than the coefficient used for U-235 in Figure 1.

Figure 4 illustrates the workings of the prompt-jump approximation. Hetrick [7] shows the prompt-jump approximation will yield solutions of the point-kinetics equations that are slightly higher than those yielded by the unmodified point-kinetics equations. In this run, that was not the case. The reason is that the prompt jump generated by the prompt-jump approximation did indeed show a higher peak than that generated by the unmodified point-kinetics equations. Because of this, feedback caused a

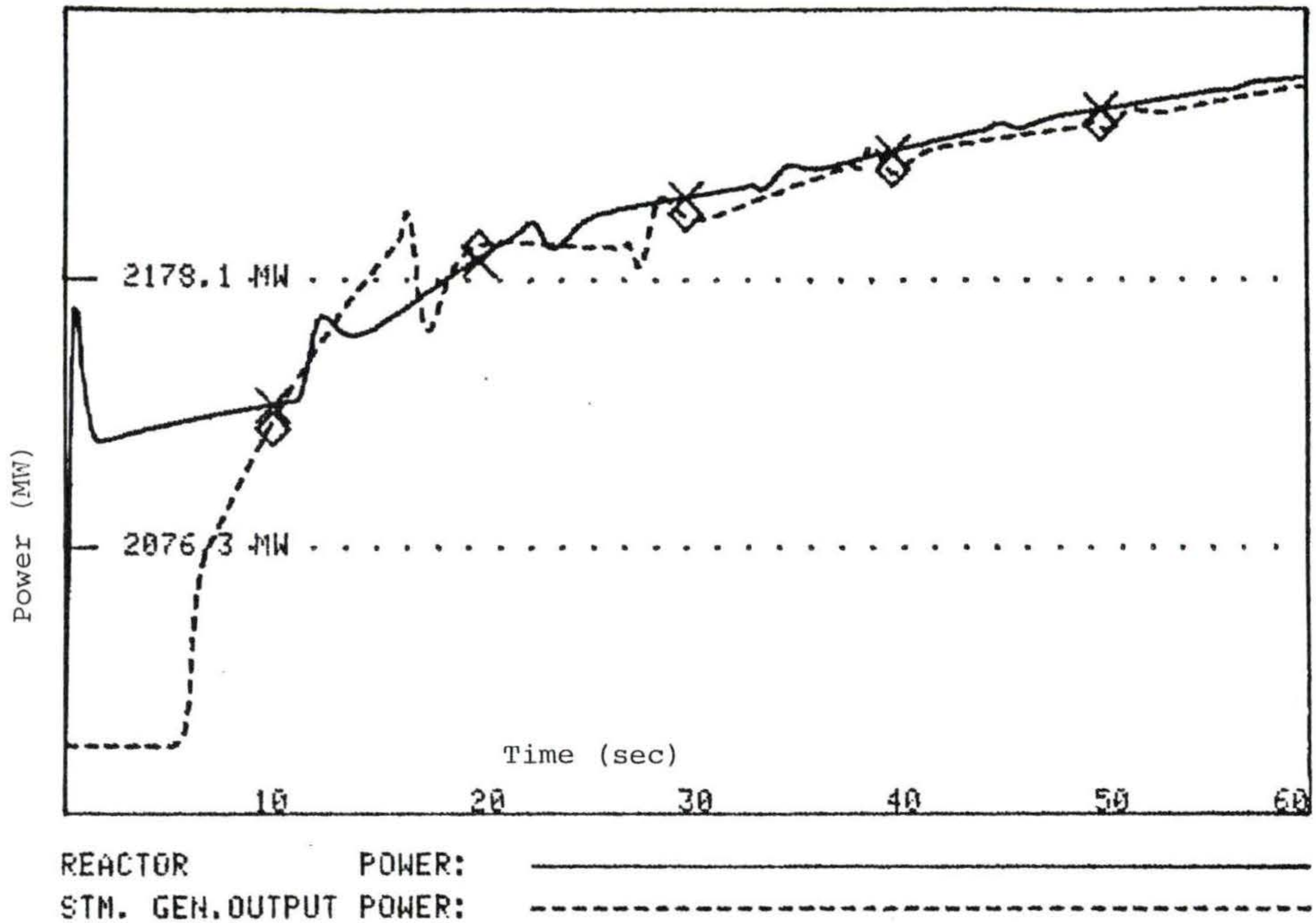

Figure 2. Reactor and steam generator power with initial power 2000 MW, 10 cents reactivity step insertion, U-233 fuel

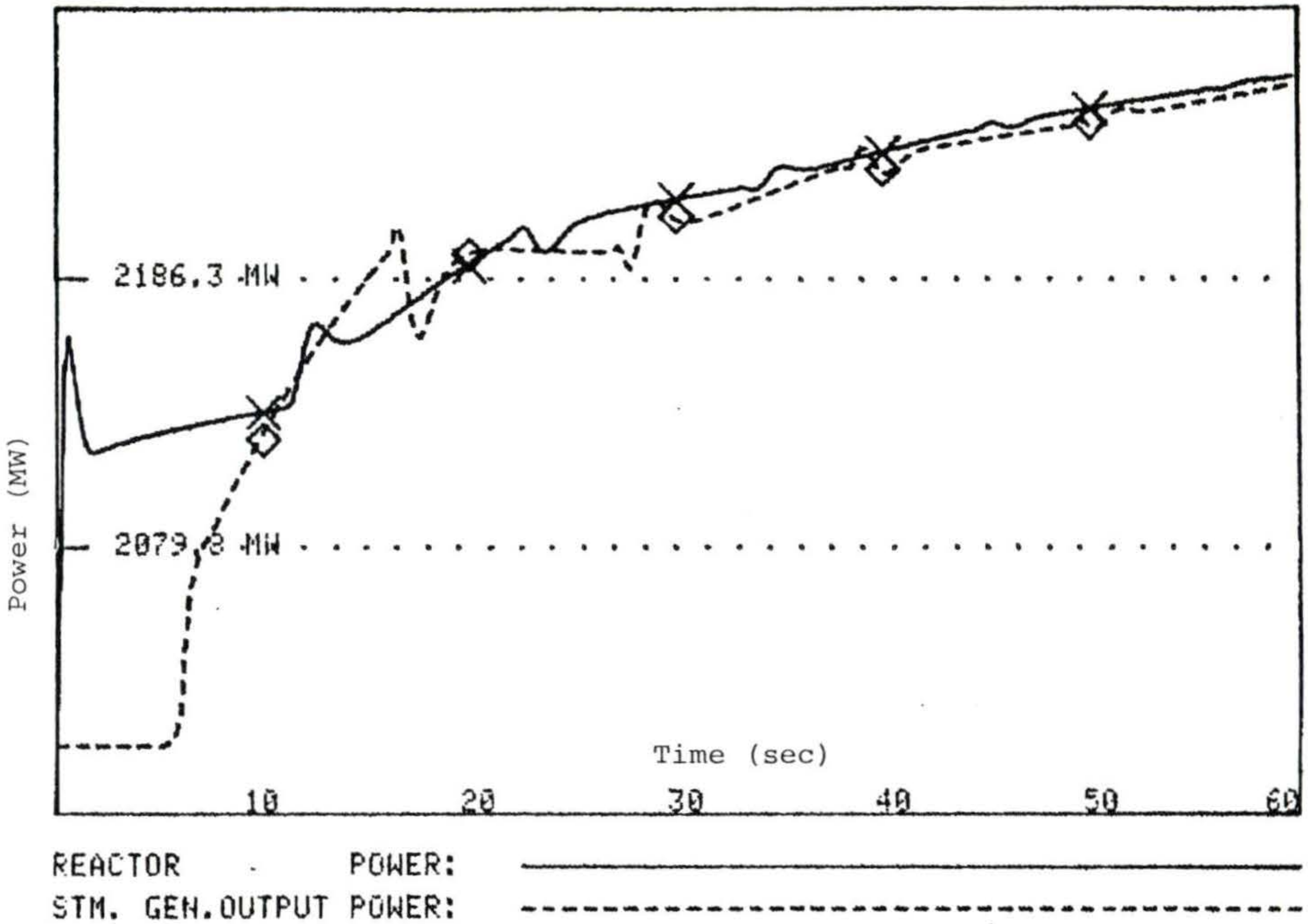

Figure 3. Reactor and steam generator power with initial power 2000 MW, 10 cents reactivity step insertion, Pu-239 fuel

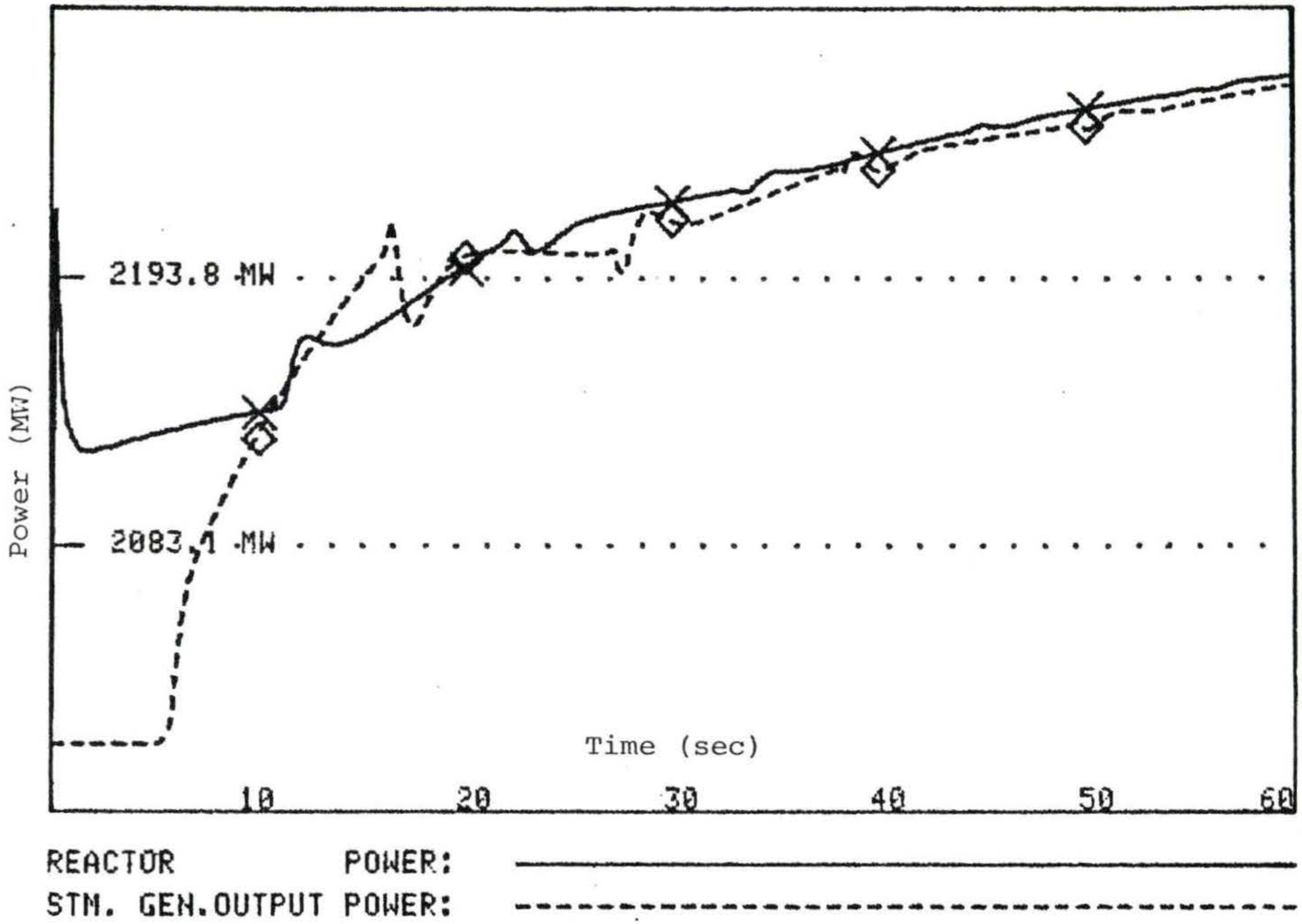

Figure 4. Reactor and steam generator power with feedback and prompt-jump approximation. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel

larger power falloff for the prompt-jump approximation than for the unmodified point-kinetics equations. This resulted in the power level in the prompt-jump approximation leveling off at a slightly lower level than in the unmodified point-kinetics equations, and remaining lower.

Figure 5 shows results of the two-time-scale matrix decoupling algorithm. The perturbation consisted of a step reactivity insertion from an initial power level of 2000 MW. Comparison with Figure 1 shows similarity in all respects except one - the decoupling algorithm shows power level changes 50% greater than those shown by the unmodified point-kinetics equations. Similar results are obtained for much smaller perturbations as well. It should also be noted that the reactor power falloff from the prompt jump peak level is not nearly so great with matrix decoupling as it is with unmodified reactor kinetics.

Although computer costs are reduced using matrix decoupling, this discrepancy is too great to recommend the use of the two-time-scale matrix decoupling algorithm as a general method. Since the prompt-jump approximation shows such good agreement with the unmodified point-kinetics equations, it follows that it is the matrix decoupling algorithm that is inaccurate, rather than the unmodified point-kinetics equations. The reasons for this are still unclear,

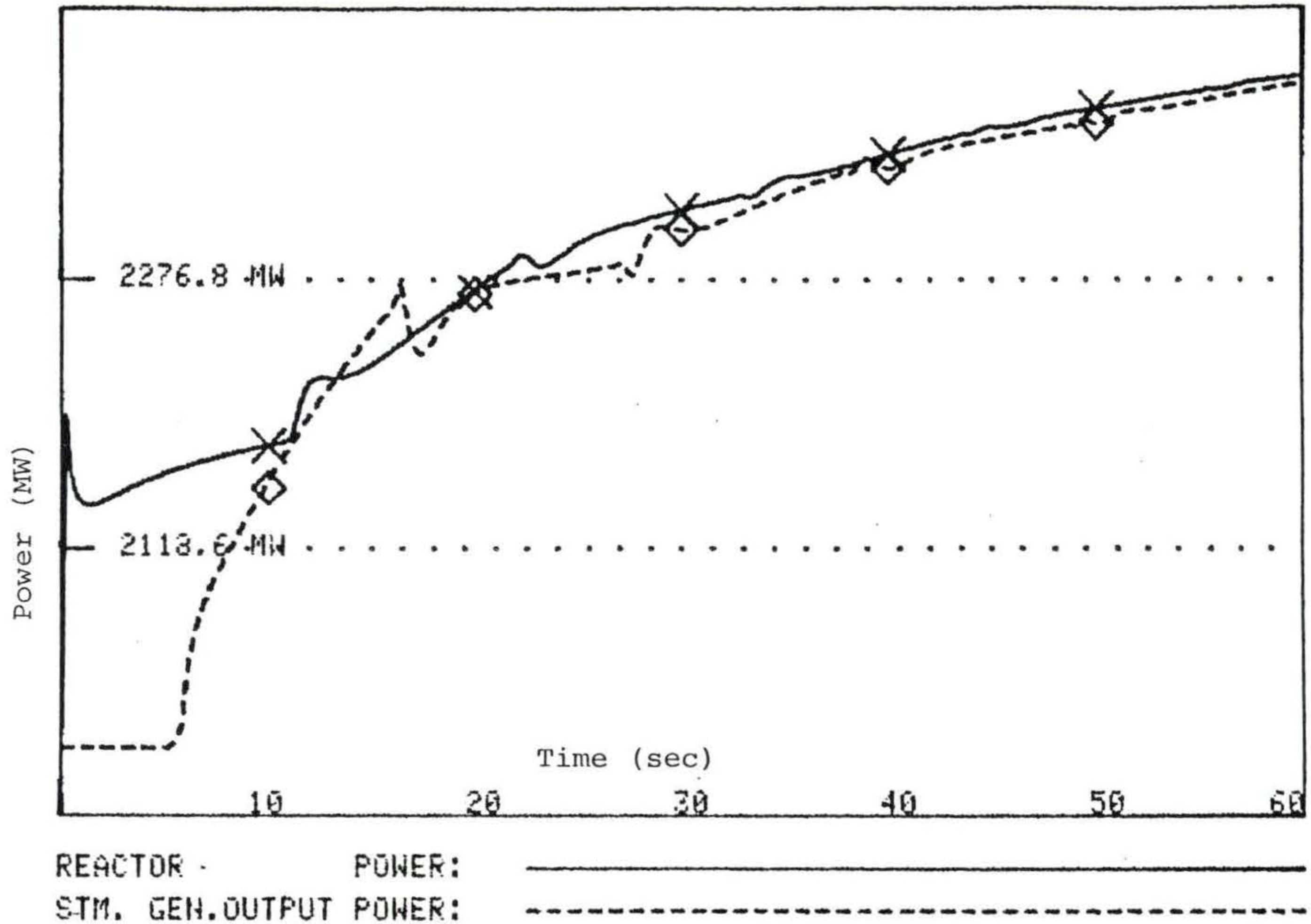

Figure 5. Reactor and steam generator power with two-time-scale matrix decoupling algorithm and feedback. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel

and are a possible subject for further research.

Figures 6 and 7 show a run of the ramp-input model. Initial power was 2000 MW. The ramp-input was 0.5 cents per second for 20 seconds, for a total reactivity insertion of 10 cents. The point-kinetics equations are used. Figure 7 runs for a total of 3 minutes, Figure 6 for 1 minute. The noteworthy aspect of both figures is that both reactor power and steam generator power show smooth responses, without the abrupt changes that are evident in Figure 1. Note too that reactor power changes in a given time period are not as great with the ramp-input as with a step insertion of the same quantity of reactivity. This is because the prompt jump gives a massive "head start" of power when a step insertion versus a ramp insertion is used. Eventually, the same power levels will be achieved by either method.

Figure 8 shows the solution of the point-kinetics equations for a step insertion of 10 cents of reactivity with no feedback and no reactor control system. Note the very smooth reactor power response even though the steam generator response is as abrupt as ever. Figure 9 shows the same situation, only with the prompt-jump approximation. Note that its reactor power response is almost identical to that in Figure 8; it is only slightly higher, as would be

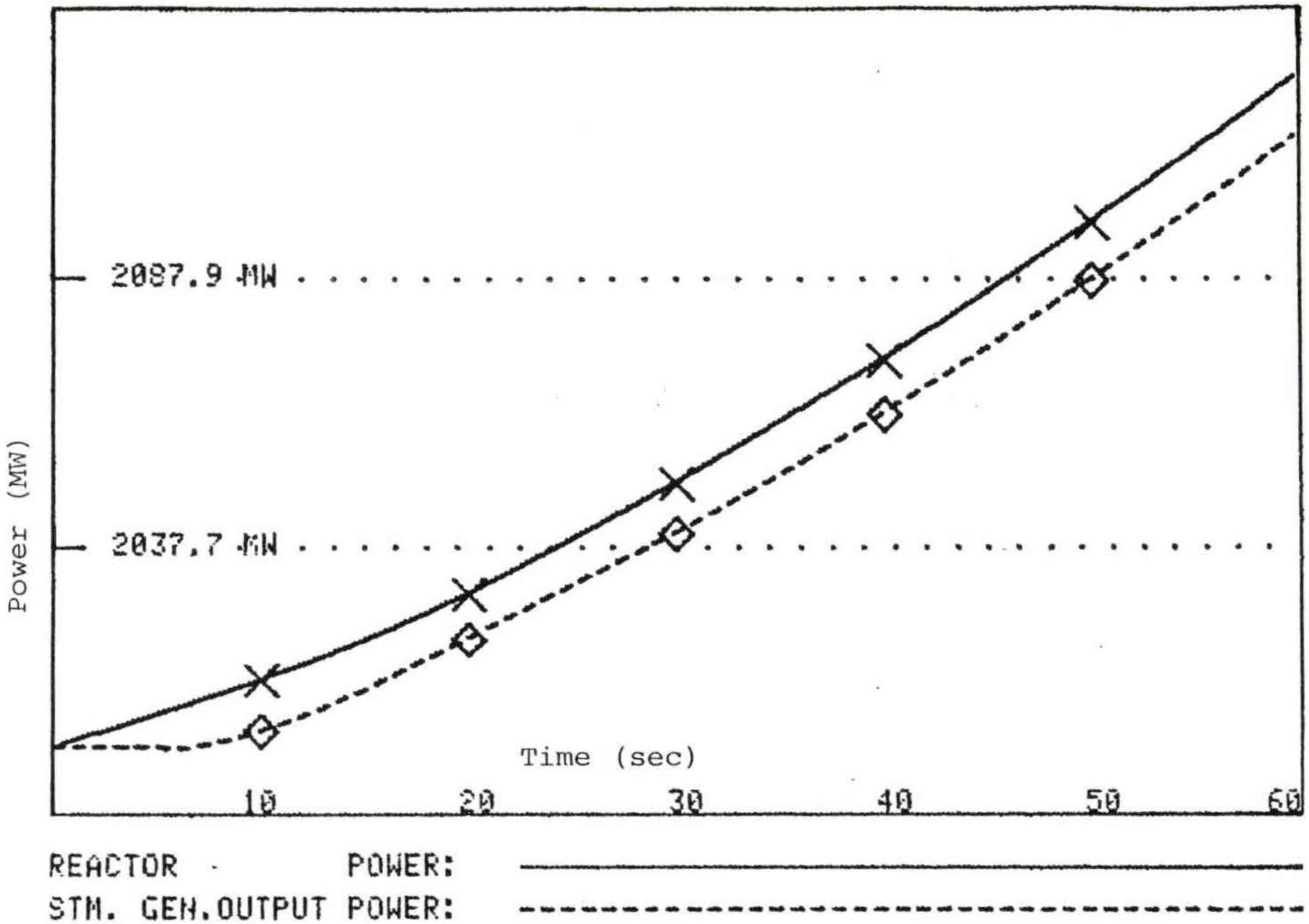

Figure 6. Reactor and steam generator power with feedback and ramp-input model. Initial power level 2000 MW, ramp-inlet model with ramp of 0.5 cents reactivity per second for 20 seconds, U-235 fuel

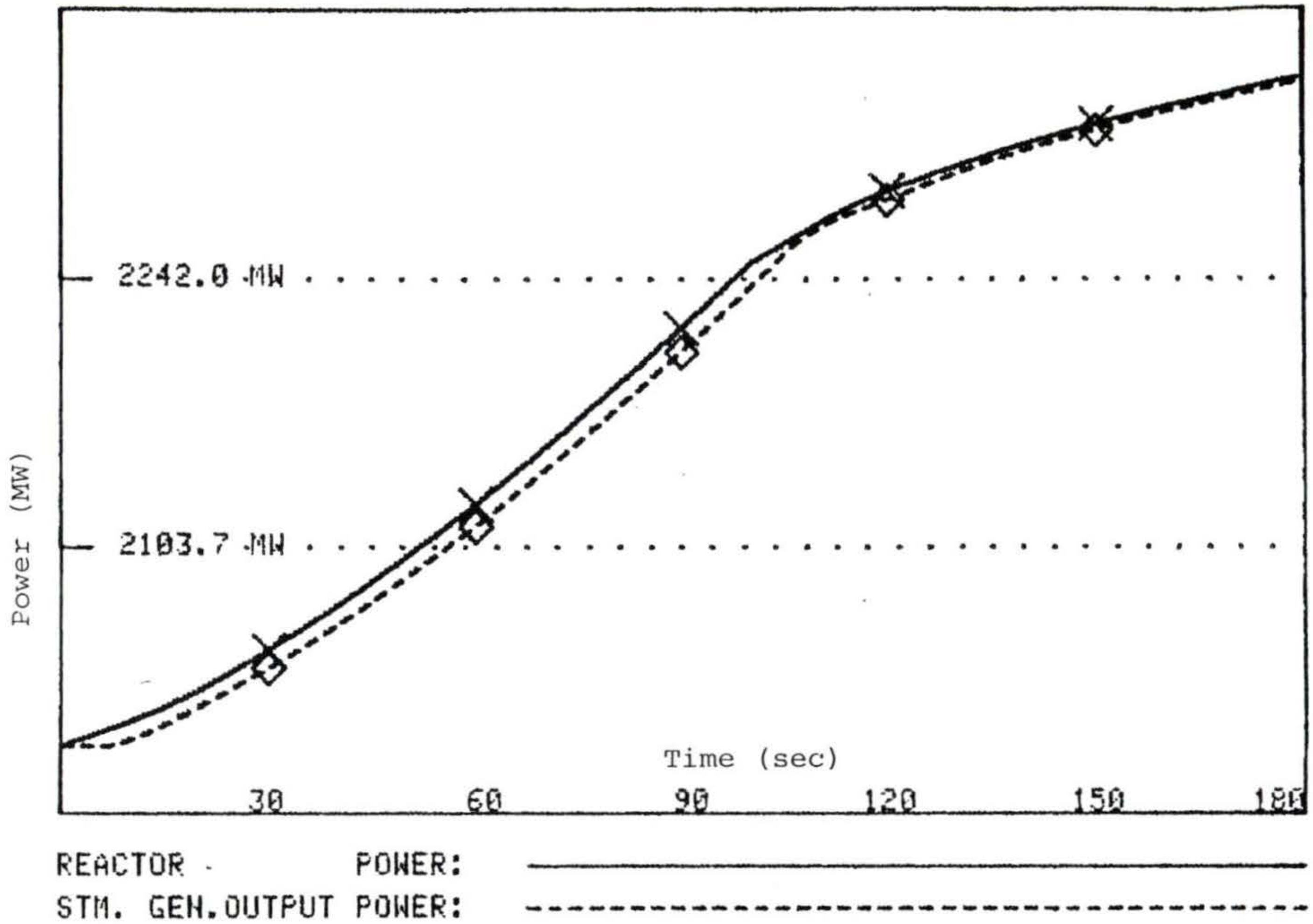

Figure 7. Reactor and steam generator power with ramp-input model.
Time of run 3 minutes

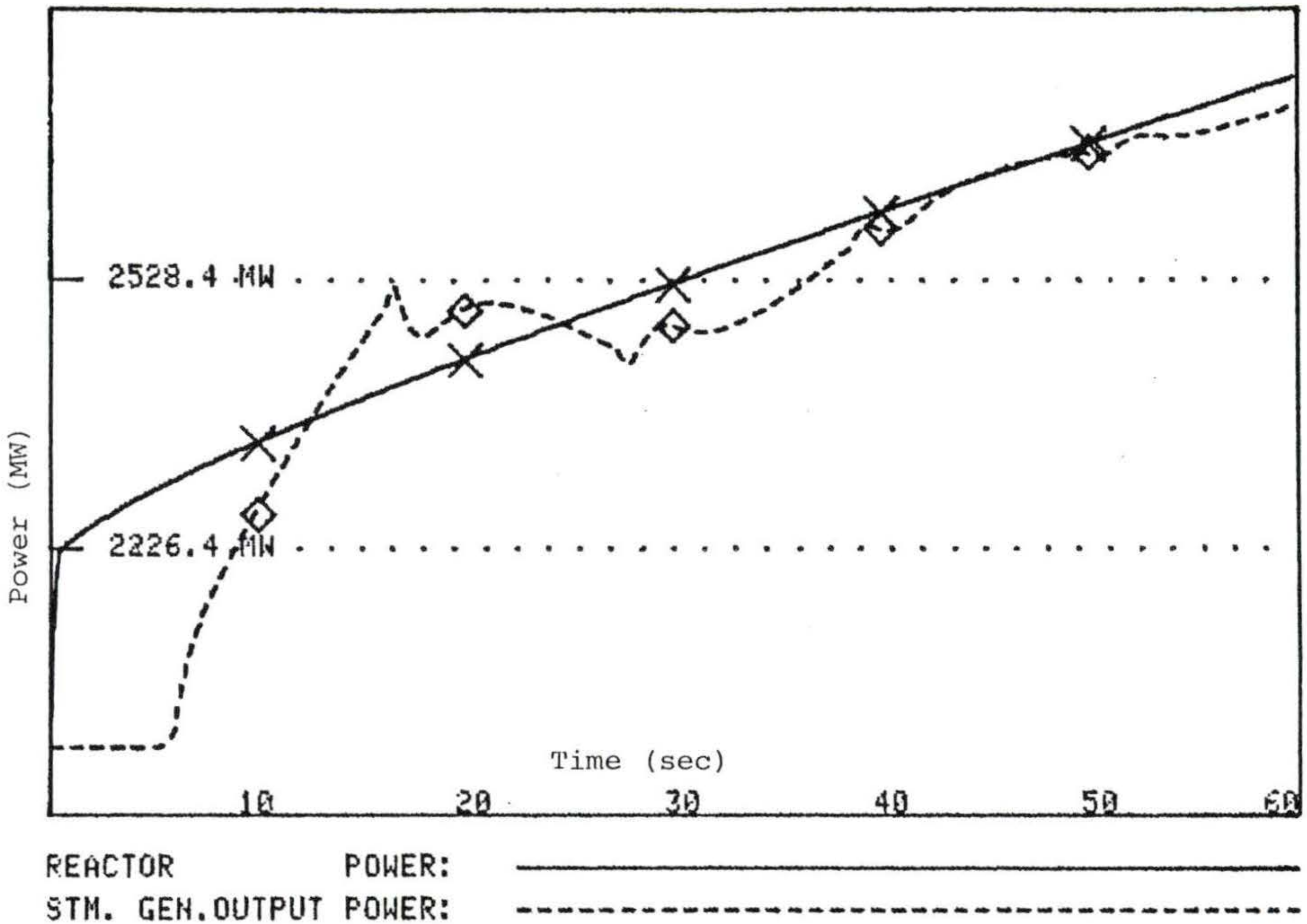

Figure 8. Reactor and steam generator power with no feedback and no control system. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel

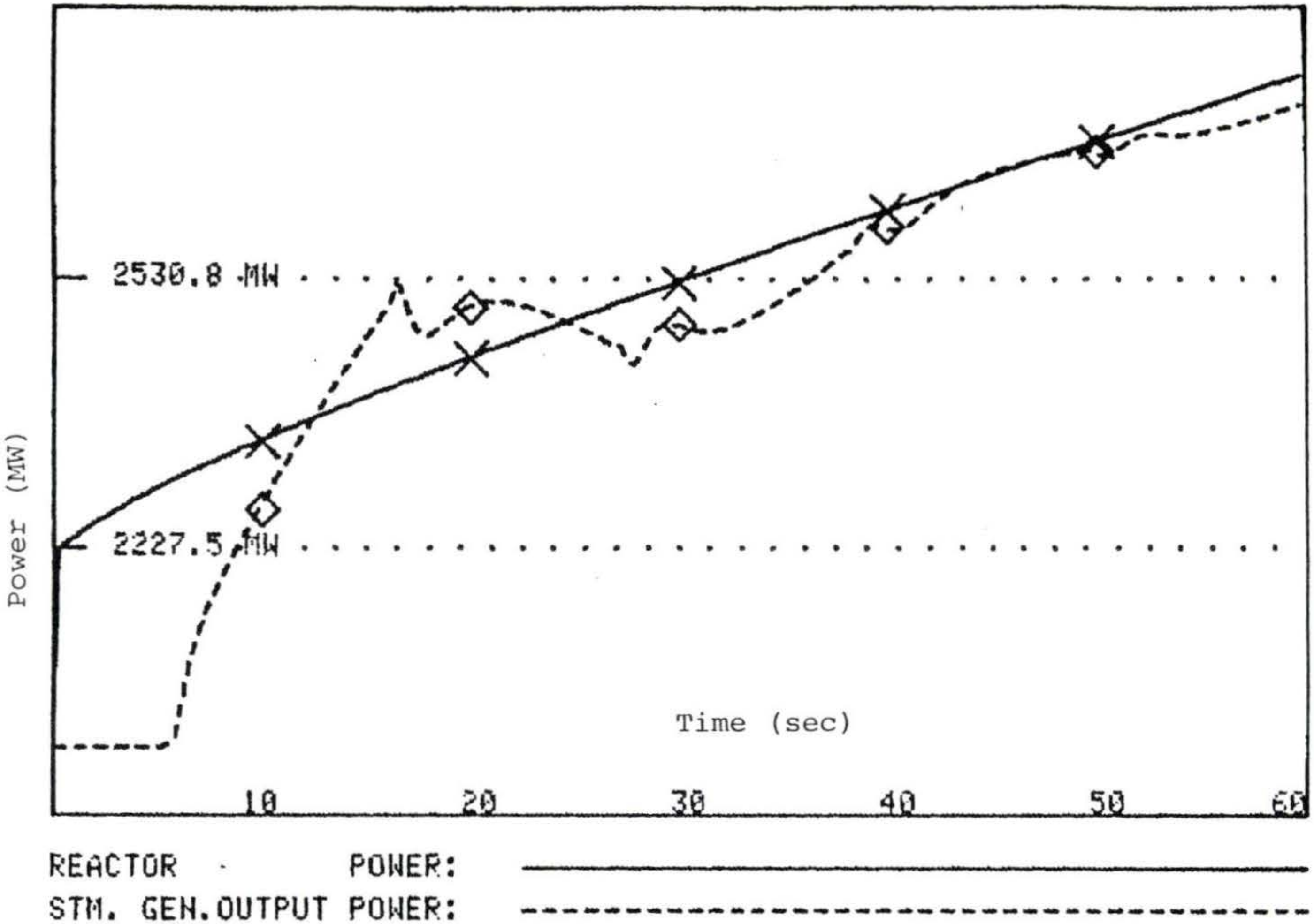

Figure 9. Reactor and steam generator power with no feedback and no control system and using the prompt-jump approximation

expected.

Figures 10 and 11 show the effects of a reactor control system. A step insertion of 2000 MW was used with a step insertion of 10 cents of reactivity. In Figure 10 a control system with a gain of -1.10^{-7} units of reactivity per degrees-second. In Figure 11, the gain is -1.10^{-6}. Comparison with Figure 1 shows that use of a reactor control system results in a lesser power rise for a given positive reactivity insertion than without one. It also shows that the greater the gain, the greater the power reduction. This is what would be expected with a reactor control system.

Figure 12 shows an example of load following. At time equals 5 seconds, the throttle valve on the secondary side of the steam generator is opened 10%, thus, allowing flow on the secondary side to rise 10%. Initial power is 2000 MW. This results in greater heat transfer, causing greater steam generator power output. At the same time, the lowered reactor inlet temperature causes lower average moderator and fuel temperatures, causing increased reactor power through feedback. As can be seen, reactor power rises dramatically, but never quite matches steam generator power. The reason is that as reactor power starts rising, feedback starts acting to keep it down. Further power rises after the prompt jump are very slow. Eventually, the steam generator power and reactor power

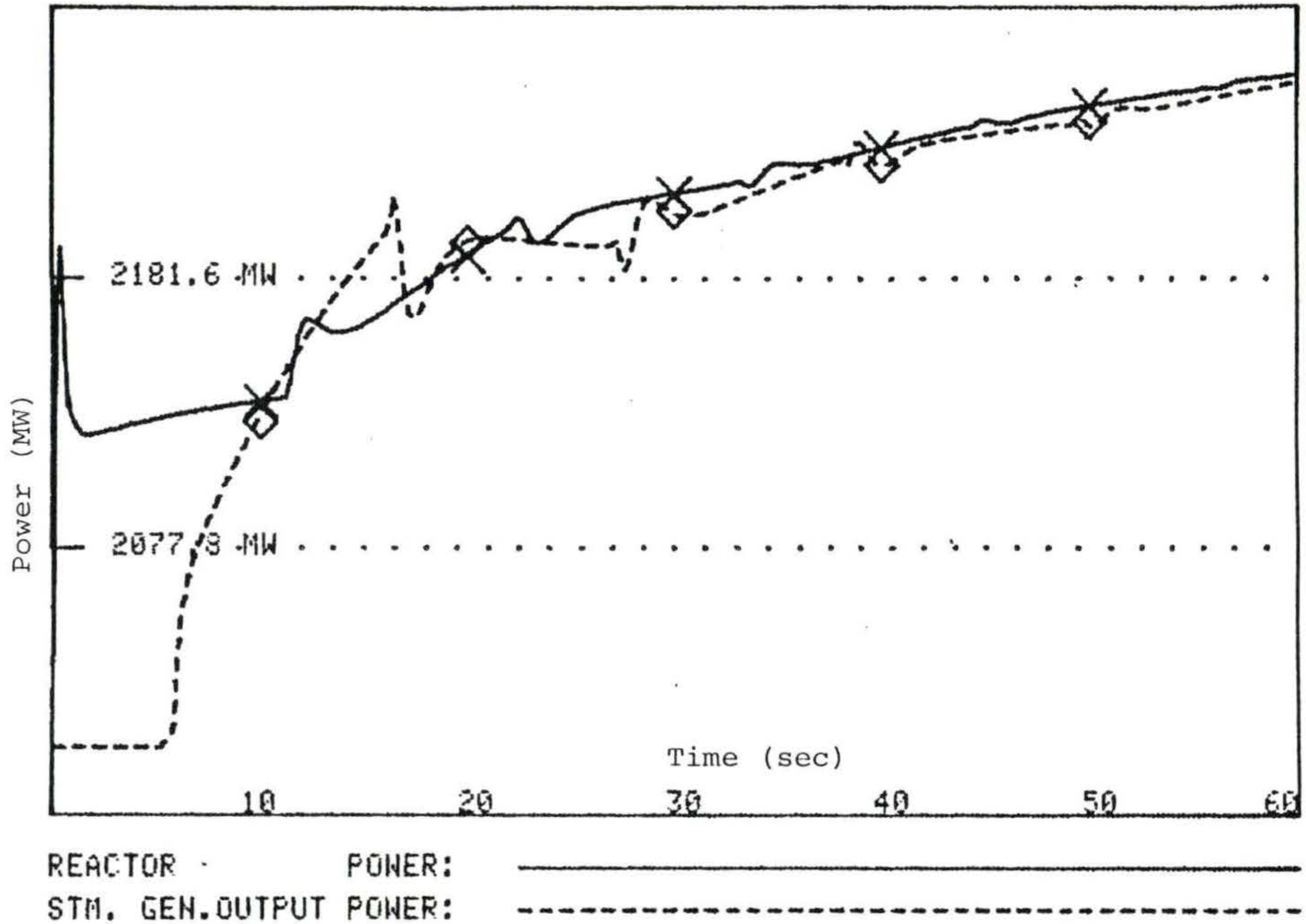

Figure 10. Reactor and steam generator power with feedback and reactor control system. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel, gain -1.0E-07

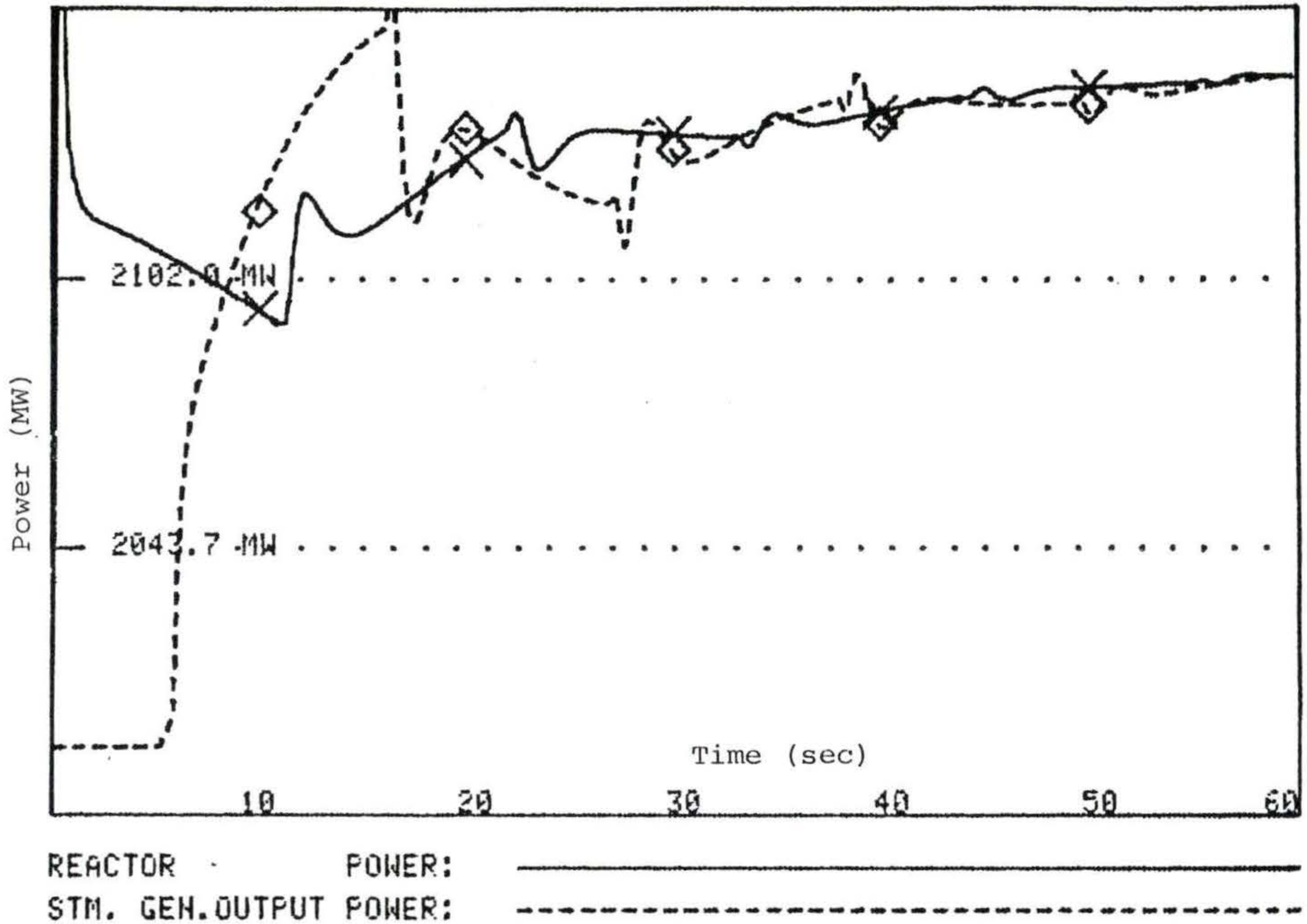

Figure 11. Reactor and steam generator power with feedback and reactor control system. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel, gain -1.0E-06

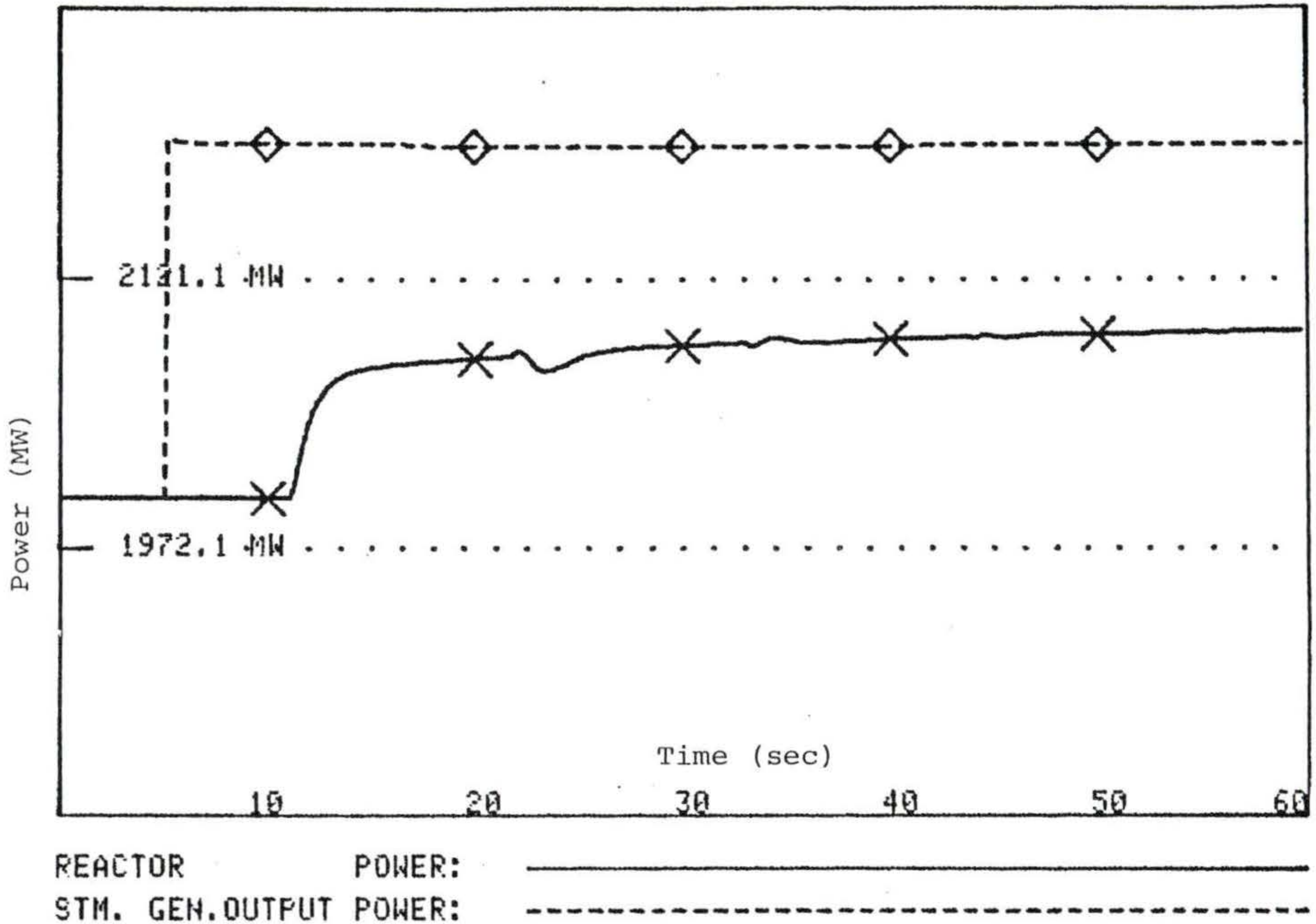

Figure 12. Reactor and steam generator power with load following and feedback. Initial power 2000 MW, throttle valve opened 10%, U-235 fuel

should approximately equal each other, but not until a long interval has passed.

Figures 13 and 14 feature steam generator response to a step change in reactor power. Note that in these figures, reactor power is constant and the steam generator responds to the new power level.

In Figure 13, reactor power jumps from 2000 MW to 2100 MW. After the five-second lag from reactor to steam generator, steam generator power rises to the reactor power level and exceeds it for a time. This reflects the fact that over a period of time, total steam generator energy output must equal reactor energy output. Since during the first 10 seconds reactor power is greater than steam generator power, steam generator power must exceed reactor power for a brief time in order to compensate for the energy deficit that occurred during the first 10 seconds. As can be seen in Figures 13 and 14, eventually steam generator power converges toward the new reactor power level.

Since total energy output over a period of time from both steam generator and reactor should be equal over a period of time, it follows that the integrals of their power functions - that is, the "areas under the curves" - should be equal. Inspection of Figure 13 shows that this is not quite so. The reason is probably that in increasing

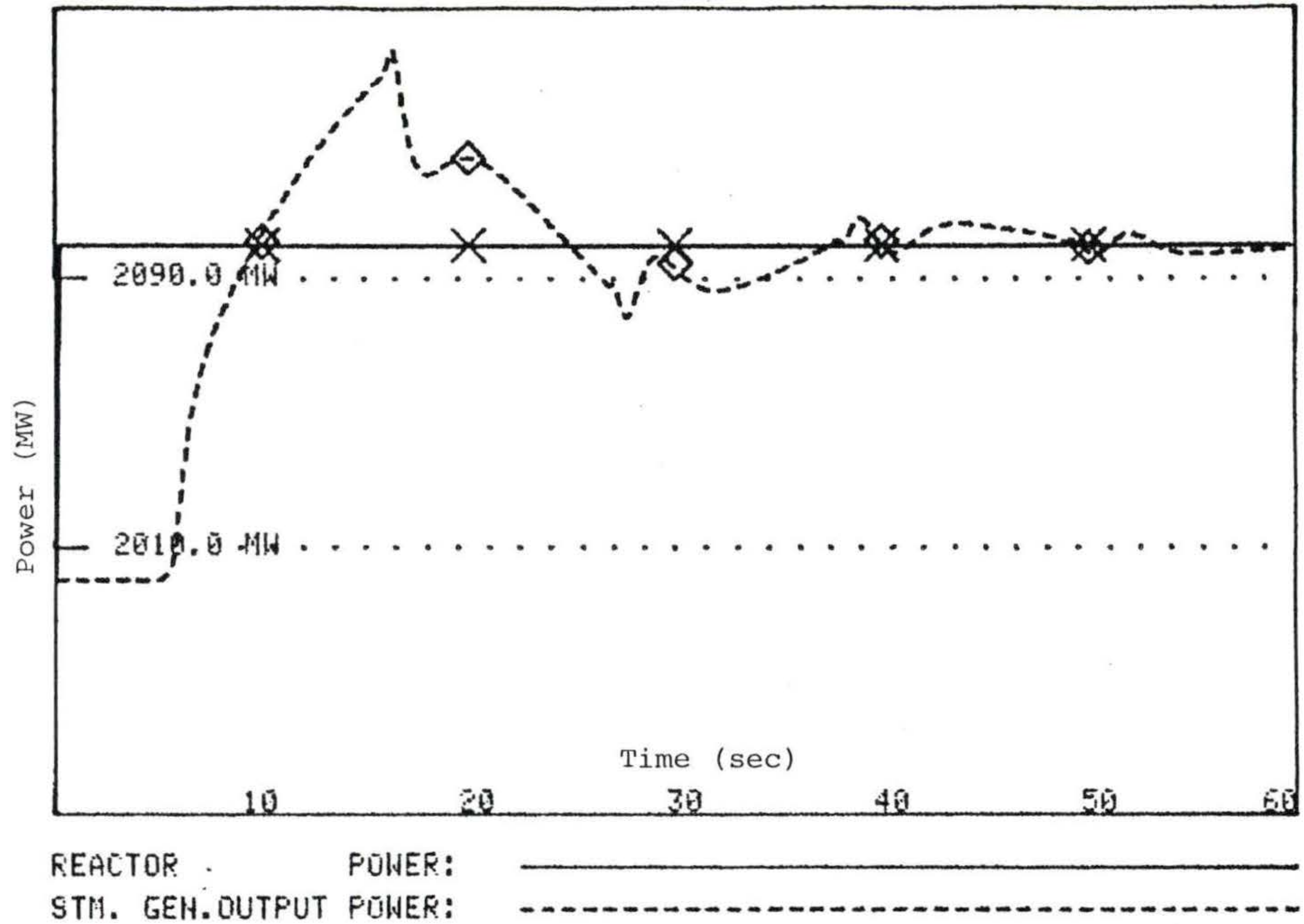

Figure 13. Reactor and steam generator with reactor power step change. Initial power 2000 MW with 100 MW step change

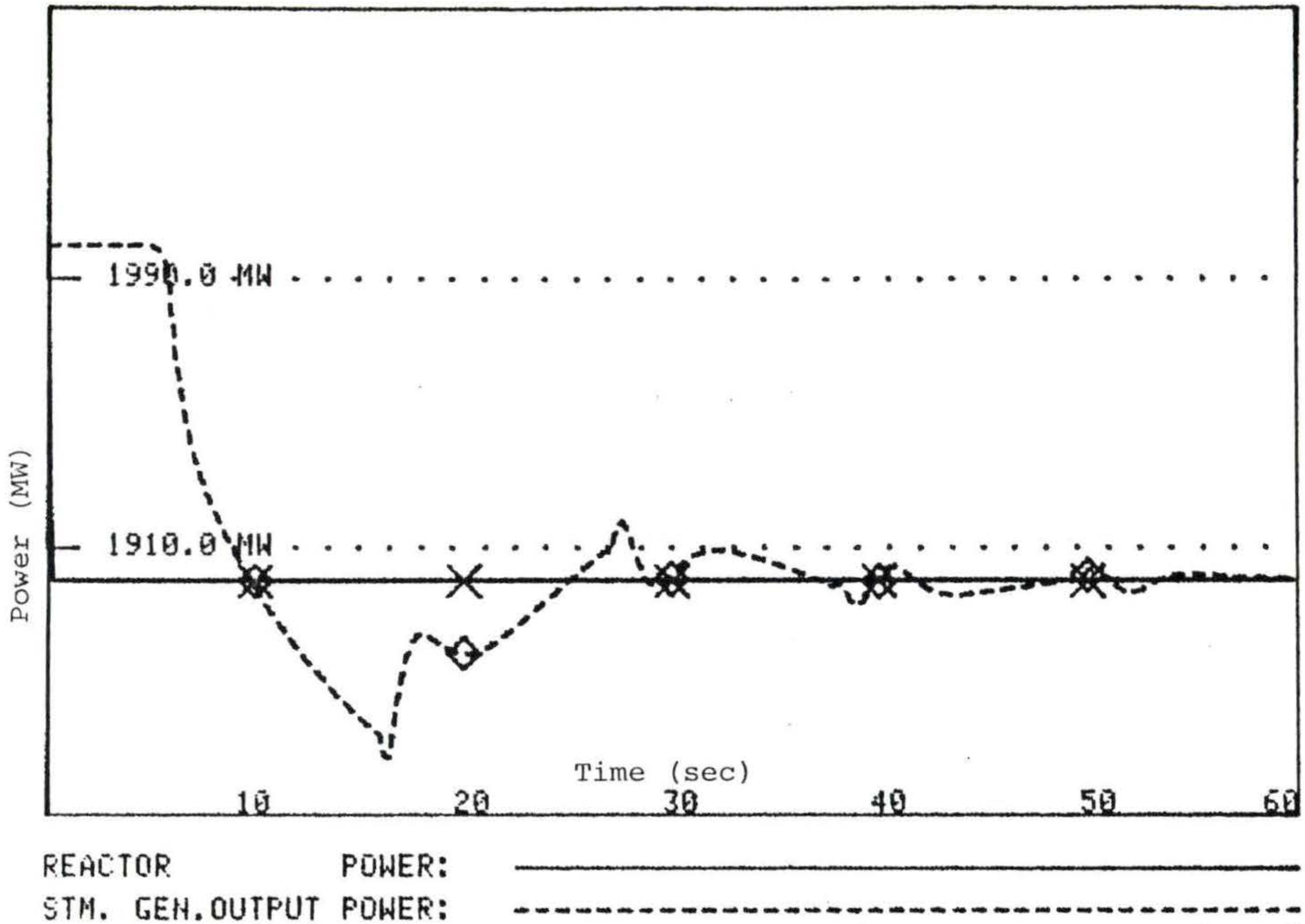

Figure 14. Reactor and steam generator power with reactor power step change. Initial power 2000 MW with -100 MW step change

steam generator power, primary side temperature is increased and this requires some of the energy that would otherwise have gone to increase steam generator output power.

Figure 15 depicts reactor fuel temperature. Initial power was 2000 MW, U-235 fuel was used, and a reactivity step of 10 cents was inserted. Several features are noteworthy. First, fuel temperature does show a "prompt jump" in temperature, but no falloff from a peak level. This is because fuel temperature does not rise as quickly as reactor power during the prompt jump. After the reactor inlet temperature starts declining at 11 seconds, average moderator temperature also declines, forcing a drop in fuel temperature. As can be seen by comparing Figures 15 and 16, fuel temperature responses are a function of average moderator temperature changes. Eventually, fuel temperature increases slowly as a result of slowly increasing reactor power even though the average moderator temperature change is decreasing very slowly.

Figure 16 shows changes in reactor inlet temperature, outlet temperature, and average moderator temperature from an initial condition. Initial power level was 2000 MW with a reactivity insertion of 10 cents, and U-235 fuel. Under steady-state conditions, outlet and inlet

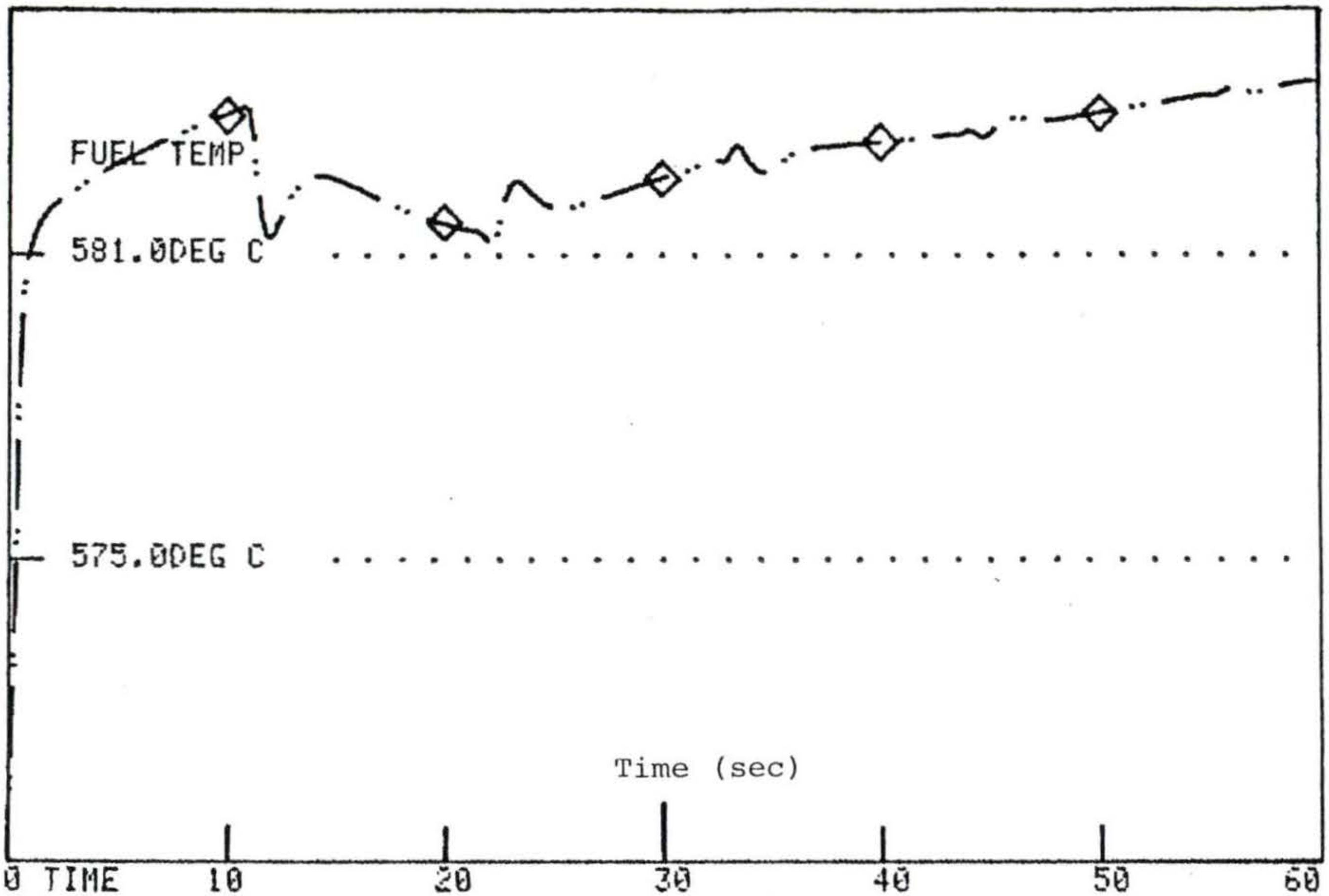

Figure 15. Fuel temperature change. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel

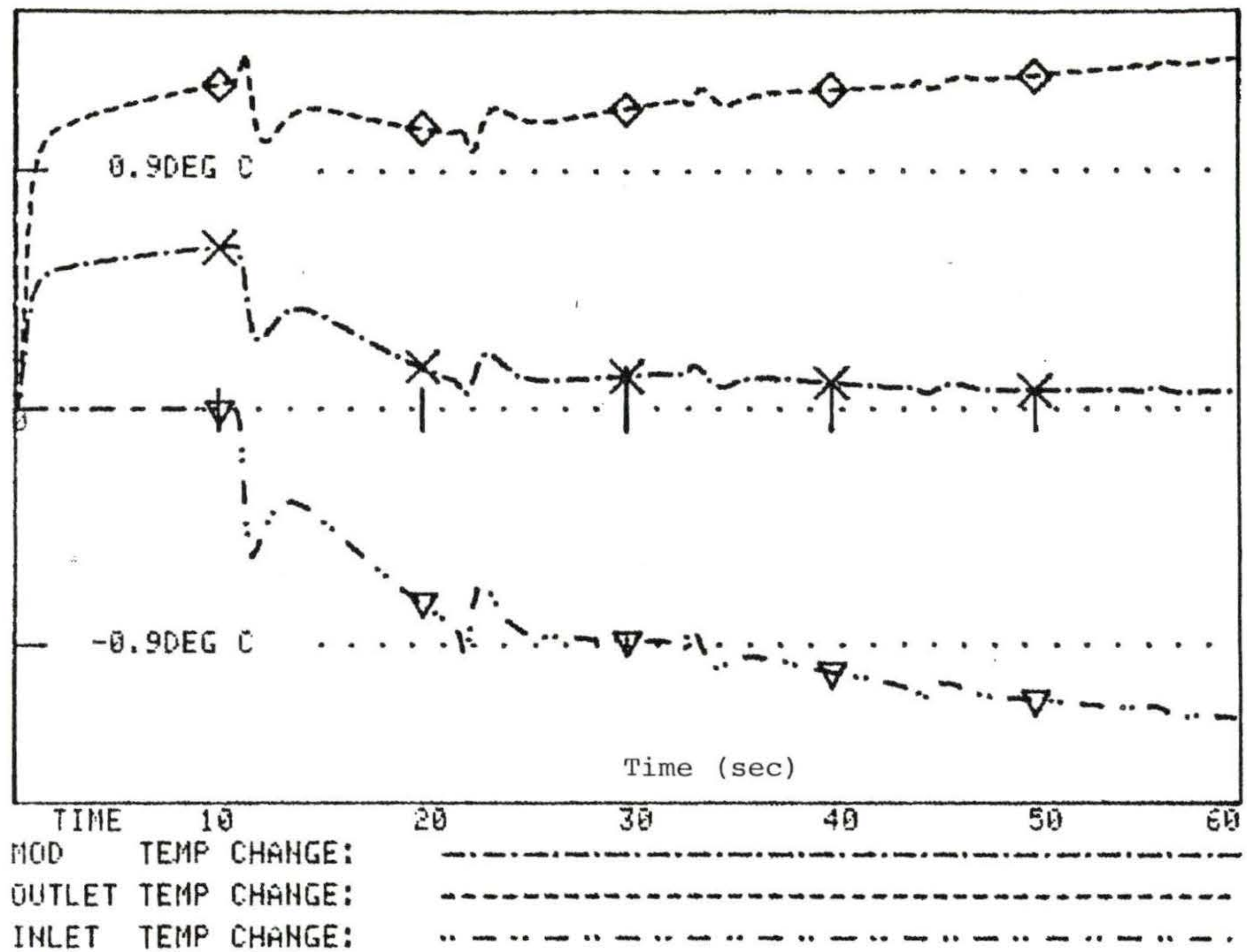

Figure 16. Reactor inlet, outlet, and average moderator temperature changes. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel

temperature changes should be equal and opposite in sign, and moderator temperature change should be zero. However, since the reactor inlet temperature change is the result of steam generator actions, any reactor perturbation has no effect on inlet temperature until 11 seconds later. At 11 seconds, reactor inlet temperature drops suddenly, as a result of the steam generator power increase that took place 6 seconds earlier.

Since there has been no change in reactor inlet temperature, reactor outlet temperature change is twice as large as it would have been in the presence of reactor inlet temperature change. Also, the moderator temperature change is not equal to zero. At 11 seconds, however, reactor inlet temperature does start to change. It affects moderator temperature relatively slowly, as the noncoolant sections of the core must also be cooled. Due to the construction of the thermal-hydraulic equations, it affects output temperature immediately, which accounts for the brief outlet temperature increase at 11 seconds. (This also leads to a brief steam generator power pulse at 16 seconds, as is seen in Figure 1). This is probably not an accurate reflection of the way a real reactor core works, but this error is induced by the fact that a simple model is used. In any event, this error dampens itself out.

Note that the reactor inlet temperature at 11 seconds drops suddenly, rises back somewhat, then declines again. This pattern repeats itself every 11 seconds, although it does dampen out. The reason for the rise after the drop is that the steam generator adjusted power based on the difference between actual outlet temperature and what it would be if the reactor inlet temperature change did not have the 11-second lag. After the prompt rise in outlet temperature, the downward pressure on reactor inlet temperature is no longer so great, and inlet temperature tends to rebound.

Overall, as Figure 16 shows, average moderator temperature change does tend to converge back toward zero after a perturbation, and reactor outlet and inlet temperatures do tend to become equal and opposite.

VII. SUMMARY AND CONCLUSIONS

The programs developed provide a good simulation of a nuclear power plant system that can be used in a classroom environment. Except for the two-time-scale matrix decoupling algorithm, all of the models used show consistent results. The programs can be used to simulate many different situations with output in whatever form desired.

The major disappointment was the poor performance of the two-time-scale matrix decoupling algorithm. Use of it yielded results that were in disagreement with the results of the other methods of solving for reactor kinetics by 50%. Perhaps further research would indicate the reasons for this.

One possible indication of error lies in the speed with which the steam generator can change its power level. While the reactor undoubtedly can change its power level quickly, it does seem unlikely that the steam generator can raise its power level 200 MW in 10 seconds, as Figure 1 indicates. Perhaps one reason why the program indicates this is that the steam generator subroutine DMTN assumes that fluid flow rates on the secondary side can change instantaneously every 0.2 seconds. This may not be a valid assumption, especially since the secondary side of the steam generator provides the feed steam for the turbine coupled to the electric generator, and the turbine certainly cannot change its power levels as quickly as can the reactor. This could be another area for further investigation.

One suggestion for changing the computer programs that form the basis of this thesis is to vary the delays between the reactor and the steam generator. This would be an easy change to make, and would add another element of variability for the user.

Another possible area of improvement would be the use of predictor-corrector methods for solving the point-kinetics equations. The point-kinetics equations constitute a system of stiff differential equations; that is, a system dominated by one large eigenvalue. Such systems undergo a prompt response, followed by a much slower response. After the prompt response, the variables all vary slowly, and comparatively large time steps can be used.

The concept of using short time steps during the prompt response and larger time steps after the prompt response is being utilized now; however, the use of predictor-corrector methods would provide a quantitative measure of how large the time steps can be before encountering intolerable error.

As satisfactory solutions are being obtained now, without the use predictor-corrector methods, use of them would not appreciably improve the quality of the solutions obtained and would quite possibly increase the size of time steps, and reduce the number of computations and any progressive error.

In general, the programs associated with this thesis are a useful learning tool for their users and are sufficiently versatile to allow extensive modifications and improvements to be made.

VIII. LITERATURE CITED

1. Anderson, Leonard. 1978. Decoupling of Two-Time-Scale Linear Systems. Aerospace and Mechanical Engineering Department, University of Arizona, Tucson, Arizona.

2. Babcock & Wilcox Company. 1975. Steam/Its Generation and Use. Babcock & Wilcox Company, New York.

3. Danofsky, Richard A. 1975. Reactor with a Control System and Reedback. Department of Nuclear Engineering, Iowa State University, Ames, Iowa.

4. Detroit Edison Company. 1973. Greenwood Preliminary Safety Analysis Report. Detroit Edison Company, Detroit, Michigan.

5. Duderstadt, James J., and Hamilton, Louis J. 1976. Nuclear Reactor Analysis. John Wiley and Sons, Inc., New York.

6. El-Wakil, M. M. 1971. Nuclear Heat Transport. American Nuclear Society, Lagrange Park, Illinois.

7. Hetrick, David L. 1971. Dynamics of Nuclear Reactors. University of Chicago Press, Chicago, Illinois.

8. Hetrick, D. L., Girijashankar, P. V., Keepin, W. N., Palusinski, O. A. 1981. A Technique for Efficient Simulation of Large Dynamic Systems. Paper presented at USNRC-EPRI Conference titled "Simulation Methods for Nuclear Power Systems". University of Arizona, Tucson, Arizona.

9. Karlekar, Bhalchandra V., and Desmond, Robert M. 1977. Engineering Heat Transfer. West Publishing Company, St. Paul, Minnesota.

10. Keenan, Joseph H., and Keyes, Frederick G. 1937. Thermodynamic Properties of Steam. John Wiley and Sons, New York.

11. Schultz, M. A. 1961. Control of Nuclear Power Plants, 2nd edition. McGraw-Hill, New York.

12. U.S. Atomic Energy Commission. 1964. Naval Reactors Physics Handbook. U.S. Government Printing Office, Washington, D.C.

IX. ACKNOWLEDGMENTS

The author greatly acknowledges the kind, patient assistance of Professor Richard A. Danofsky in the pursuit of this study. Not only were his suggestions usually invaluable, but his refusal to accept shortcuts made this a better work than it otherwise may have been. Special thanks are also in order for Professors G. W. Burnet, D. M. Roberts, and R. J. Lambert.

Finally, the author wishes to thank his family for their constant encouragement and help.

X. APPENDIX A: USER'S GUIDE

The simulation of a nuclear power plant accomplished by this program is a versatile system whose performance can be controlled by the user.

The user can select the kind of system he or she desires by responding to prompting from the computer. The computer will write messages across the screen of the terminal that the user is operating, describing the choice(s) the user is about to make, and then inviting him/her to make a choice. The choice is made by typing in an appropriate number, then pressing the carriage return key (<CR>). Choices made outside permissible parameters are rejected, and the user must select again.

Many of the parameters of the power plant, for instance those of the reactor control system, have default values; that is, values that the parameters automatically assume even if the user takes no action.

The user can select the following options and parameters:

1. Reactor fuel isotope used.

This includes uranium 233 (U-233), uranium 235 (U-235), and plutonium 239 (Pu-239).

2. Whether feedback is desired, and if so, the values of the reactivity coefficients.

3. Whether a reactor control system (RCS) is desired, and if so, the values of the parameters of gain, the differential controller, and the integral controller.

4. Whether the system is operated with free kinetics, no feedback and no automatic RCS.

5. The magnitude of any reactivity step and of any pre-perturbation reactivity. Total reactivity must be greater than -90 cents and less than +90 cents.

6. Whether the two-time-scale matrix decoupling algorithm is to be used.

7. Whether the prompt-jump approximation is to be used.

8. Whether the ramp-input model is to be used, and if so, the magnitude of the ramp and the duration of its run.

9. Whether a simple power step is to be inserted that uses no reactor kinetics at all, and if so, the magnitude of this step.

10. Whether a steam generator throttle valve change is to be made, and reactor power to be changed only by the feedback effects caused by the throttle valve change. Within limits, the magnitude of the throttle valve change is selected by the user.

11. Whether output is to be in a table or in a graph.

12. Whether output is to be printed on paper, or on a computer terminal.

13. If tabular output is selected, whether it is to be displayed in abbreviated format.

14. Whether the duration of the program run is to be lengthened anywhere from one to five minutes.

After the user makes all the selections, a summary of selections is written on the computer screen. A printed copy of this summary can be made.

Program output can be in one of four forms: output in a table on the computer screen, output in a table that is printed on paper, graphics output on a terminal, and printed graphics output.

To obtain any kind of output, first the program must be run. To do this, log on to any VAX terminal and type in the phrase RUN NERO. All computer promptings and user responses are made on the computer terminal.

Table output on the screen is the simplest to obtain. The user responds to all computer promptings. After this is done, the computer program will simply run its course, on the screen. No additional user action is necessary.

Printed table output is almost as easily obtained. Any printed table output must utilize one of the Computer

Science Center printer queues, for instance, queue BC0131U in Coover Hall. The print option is selected by the user when the program prompts him to choose between terminal output and printed output. If printed output is selected, the output will initially be stored in a data file. To print the contents of this data file, type in PRINT FOR.008.DAT/Q = (name of queue). For instance, if queue BC0131U is the printer of choice, type in PRINT FOR008. DAT/Q = BC0131U. (Do not include the period at the end of the last sentence!)

For graphics output that appears on screen only, any Tektronix 4051 terminal or any of the light blue terminals labeled "GRAPHICS" can be used. As with table output that appears on screens only, graphics output is selected as a result of user responses to computer promptings. Once this is done, the computer program will run its course, on the screen. No additional user action is necessary.

For printed graphics output, the user must be logged on a Tekronics 4051 unit. The 4051 unit is actually a microcomputer that can be turned into a VAX terminal. This must be done to obtain a graphics display.

The following steps are needed to turn the 4051 unit into a VAX terminal.

1. Turn the power on.

2. After power comes on, press the HOME PAGE key to clear the screen.

3. Insert the casette tape labeled "MARK's EASY LOGON".

4. Press the AUTO LOAD key.

5. Eject the tape after the I/0 light goes off.

The user should now be able to log into the VAX system.

There are two ways to get printed output. The easier way is to use the Tektronix hard copy printer. This is basically a photoreproduction machine that exactly duplicates whatever is on the screen of the 4051 unit at any given time. It can reproduce other things than graphics, too; for instance, any table output.

To use, simply tie this machine into a 4051 terminal and press the lighted button whenever you see something on the screen of the 4051 unit you would like duplicated.

NOTE WELL! Before any graphics is displayed, the program will ask you whether you want a "4662 copy" or not. To use this machine, write in 2 to indicate you do <u>not</u> want a "4662 copy". Here, the term "4662 copy" refers to the output of the 4662 plotter. Since the Tektronix hard copy printer is basically an extension of the 4051 unit, it does not come under the category of "4662 copy", as defined by this program.

To use the 4662 plotter, one must log on with a 4051 unit, as before. Then the 4662 unit must be activated, which is done in the following manner:

1. Turn the machine on.

2. Place paper down.

3. Press the LOAD button to down position.

4. Smooth the paper.

5. Press LOAD button to high position.

6. Set the lower left limit of the paper. Use the joystick to position the pen to the right place, then push the set button until it beeps.

7. Do the same with the upper right position.

In order to plot anything, the LOAD button must be in the down position.

It is possible to get printed copies of both the summary of user's selections, and graphics. One can simply use the hard copy machine, or one can create and print a data file. Since graphics is selected, no table output will appear, but the summary normally preceding it will.

The following are general hints that may contribute toward more efficient use of this program:

1. The 4051 unit has no scrolling capability. That is, once the screen is filled up with characters, nothing more will happen until the user erases everything on the

screen by pressing the HOME PAGE button.

2. In graphics, sometimes the "output" will stop by itself. It can be resumed by pressing carriage return (<CR>).

3. The user can stop further output from appearing on a screen by pressing the CNTRL key and the S key simultaneously. Pressing CNTRL Q will start the output once again. Pressing CNTRL Y will kill the entire computer run.

4. When prompting the user to input parameters, the program will direct the format to be used. Be sure to follow the format rules exactly.

Table 3 lists the parameters that can be varied by the user. Their default values plus the lower and upper limits inside which the user may vary them are also listed.

The following format changes can be inserted by the user. First, table output by default occurs at an interval of 0.2 seconds. This interval can be changed to 1 second, 2 seconds, 4 seconds, 5 seconds, and 10 seconds. Second, the computer run by default lasts for 1 minute of world time. This can be lengthened to 2 minutes, 3 minutes, 4 minutes, or 5 seconds.

Some of the parameters have additional restrictions. Initial power plus power step must not exceed 3000 MW.

Table 3. User-selected parameters, their default valves and lower and upper limits

Parameter	Default value	Lower limit	Upper limit
Doppler coefficient of reactivity for U-233 (α_f)	-1.07E-05 (δk/k)/C	-0.001 (δk/k)/C	+0.001 (δk/k)/C
Doppler coefficient of reactivity for U-235 (α_f)	-2.61E-05 (δk/k)/C	-0.001 (δk/k)/C	+0.001 (δk/k)/C
Doppler coefficient of reactivity for Pu-239 (α_f)	-0.85E-05 (δk/k)/C	-0.001 (δk/k)/C	+0.001 (δk/k)/C
Moderator coefficient of reactivity (α_m)	-8.6E-06 (δk/k)/C	-0.0001 (δk/k)/C	+0.0001 (δk/k)/C
Pre-perturbation reactivity (ρ_0)	0	-90 cents	+90 cents
Reactivity perturbation ($\delta\rho$)	0	-90 cents	+90 cents
Control system gain (A)	-1.0E-06 reactivity/(unit error -second)	0	4.0E-06 reactivity/(unit error -second)

Control system parameter tau (τ)	5 seconds	2 seconds	20 seconds
Control system time constant (τ_c)	5 seconds	2 seconds	20 seconds
Ramp insertion rate (reactivity per second)	0	-10 cents/s	+10 cents/s
Duration of ramp insertion	0	none	none
Initial power level (no)	-	0 MW	3000 MW
Power step change	-	0 MW	3000 MW
Throttle valve change	0	-10% of initial setting	+10% of initial setting

Pre-perturbation reactivity plus reactivity perturbation must not exceed plus or minus 90 cents. Ramp insertion rate times duration of ramp must not result in a total reactivity greater than plus or minus 90 cents.

Any attempt to vary the parameters outside the limits listed in Table 3 will result in the choice being rejected. The user will then have the opportunity to select again.

XI. APPENDIX B: PROGRAM LISTING

```
C         MAIN PROGRAM NERO
C
C         THIS IS THE MAIN PROGRAM NERO.  IN THIS PROGRAM THE USER SE-
C    LECTS THE PARAMETERS AND OPTIONS TO FIT THE SYSTEM WHOSE BEHAVIOR
C    HE OR SHE WISHES TO EXAMINE.  SOME OF THESE CHOICES ARE INCLUDED IN
C    THE COMMON BLOCK AND CONTROL THE BEHAVIOR OF THE SUBROUTINES.
C    OTHERS CONTROL THE FORMAT OF THE OUTPUT.
C
C         THE SUBROUTINE GALBA     CONTROLS REACTOR KINETICS.
C         THE SUBROUTINE OTHO      CONTROLS HEAT TRANSFER IN THE REACTOR
C                                  CORE.
C         THE SUBROUTINE VESPASIAN CONTROLS OUTPUT GRAPHICS.
C         THE SUBROUTINE DMTN      CONTROLS THE STEAM GENERATOR MODEL.
C         THE SUBROUTINE EIGEN AND THE SUBROUTINE CLINQ ARE PORTLIBRARY
C                                  SUBROUTINES USED IN GALBA TO COMPUTE
C                                  EIGENVALUES, AND FUNDAMENTAL EIGEN-
C                                  VECTOR MATRICES AND THEIR INVERSES.
C
C         ALL DIMENSIONS ARE DEFINED IN THE SI SYSTEM OF MEASUREMENTS.
C
C
C         THE FOLLOWING VARIABLES ARE USED IN THE COMMON BLOCK:
C
C         AA       IS THE GAIN OF THE REACTOR CONTROL SYSTEM.
C         ALPHF    IS THE DOPPLER   COEFFICIENT OF REACTIVITY.
C         ALPHM    IS THE MODERATOR COEFFICIENT OF REACTIVITY.
C         B        IS THE 1X6 ARRAY FOR THE 6 GROUPS OF DELAYED-NEUTRON
C                         PRECURSORS.
C         CPPAV    IS THE AVERAGE HEAT CAPACITY OF THE REACTOR COOLANT.
C                         IT IS SLIGHTLY A FUNCTION OF REACTOR POWER.
C         CS       IS THE VARIABLE WHOSE VALUE, SELECTED BY THE USER,
C                         DETERMINES A REACTOR CONTROL SYSTEM IS USED.
C         DN1      IS THE DIFFERENCE BETWEEN CURRENT REACTOR POWER AND
C                     THE INITIAL (STEADY-STATE) POWER.
C         DRO      IS THE DIFFERENCE BETWEEN CURRENT REACTIVITY    AND
C                     THE INITIAL (STEADY-STATE) REACTIVITY.
C         DTF      IS THE DIFFERENCE BETWEEN CURENT FUEL TEMPERATURE AND
C                     THE INITIAL (STEADY-STATE) TEMPERATURE.
C         DTI      IS THE DIFFERENCE BETWEEN CURRENT REACTOR INLET TEMP.
C                         AND INITIAL (STEADY-STATE) TEMPERATURE.
C         DTM      IS THE DIFFERENCE BETWEEN CURRENT MODERATOR TEMPERATURE
C                         AND THE INITIAL (STEADY-STATE) TEMPERATURE.
C         DTO      IS THE DIFFERENCE BETWEEN CURRENT REACTOR OUTLET TEMP.
C                         AND INITIAL (STEADY-STATE) TEMPERATURE.
C         DTS      IS THE DIFFERENCE BETWEEN CURRENT TEMPERATURE OF THE
C                         SECONDARY SIDE OF THE STEAM GENERATOR AND THE
C                         INITIAL (STEADY-STATE) TEMPERATURE.
C         DTSG     IS THE DIFFERENCE BETWEEN CURRENT TEMPERATURE OF THE
C                         PRIMARY   SIDE OF THE STEAM GENERATOR AND THE
C                         INITIAL (STEADY-STATE) TEMPERATURE.
C         EIG      IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                         DETERMINES WHETHER THE TWO-TIME-SCALE MATRIX
C                         DECOUPLING ALGORITHM IS USED IN GALBA TO
C                         COMPUTE REACTOR KINETICS.
C         H        IS THE TIME STEP USED IN THE SUBROUTINES.  IS ALLOWED
C                         TO VARY FROM SUBROUTINE TO SUBROUTINE.
C         IS       IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
```

```
C                         DETERMINES WHICH ISOTOPE OF FISSILE MATERIAL
C                         WILL BE USED.  ISOTOPES USED CAN INCLUDE U-233,
C                         U-235, PU-239.
C      K         IS THE SUBSCRIPT OF THE ARRAYS USED IN DEPICTING
C                         OUTPUT, WHETHER IN A TABLE OR IN GRAPHICS.  IT
C                         REPRESENTS TIME INCREMENTS OF 0.2 SECONDS.  IT
C                         IS ALSO USED IN SOME SUBROUTINES AS A "TIMER",
C                         WHOSE VALUE CAN TRIGGER CERTAIN EVENTS.
C      KF        IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                         DETERMINES WHETHER NO REACTOR CONTROL SYSTEM
C                         AND NO REACTOR FEEDBACK IS USED.
C      LMB       IS THE 1X6 ARRAY FOR THE 6 GROUPS OF DELAYED-NEUTRON
C                         DECAY COEFFICIENTS.
C      MDOTP     IS THE MASS FLOW RATE OF REACTOR COOLANT.  IT IS
C                         SLIGHTLY A FUNCTION OF INITIAL REACTOR POWER.
C      NK        IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                         DETERMINES WHETHER REACTOR KINETICS IS USED,
C                         OR WHETHER A SIMPLE POWER STEP IS POSTULATED.
C      NN        IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                         DETERMINES WHETHER THE OUTPUT IS IN A TABLE
C                         OR IN GRAPHICS.
C      N10       IS THE INITIAL POWER LEVEL OF THE REACTOR.
C      PJ        IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                         DETERMINES WHETHER THE PROMPT-JUMP APPROXIMATION
C                         IS USED OR NOT.
C      RI        IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                         DETERMINES WHETHER THE RAMP-INPUT MODEL IS USED
C                         OR NOT.
C      RIR       IS THE RAMP-INPUT RATE.
C      RHO       IS THE INITIAL (STEADY-STATE) AMOUNT OF REACTIVITY.
C      RP        IS THE TOTAL REACTOR POWER AT ANY GIVEN TIME.  THIS IS
C                         USED IN DMTN AS A CRITERION FOR DECIDING HOW
C                         MUCH TO ALTER SECONDARY SIDE FLUID FLOW IN ORDER
C                         COUNTERACT ANY PERTURBATION CAUSED BY ALTERED
C                         REACTOR POWER.
C      STMGEN    IS THE STEAM GENERATOR OUTPUT POWER.
C      TAU       IS ONE OF THE REACTOR CONTROL SYSTEM (RCS) PARAMETERS.
C                         THIS ONE IS USED TO ADJUST THE INTEGRAL
C                         CONTROLLER.
C      TAUC      IS ONE OF THE RCS PARAMETERS.  THIS ONE IS USED TO
C                         ADJUST THE DIFFERENTIAL CONTROLLER.
C      TF1       IS THE INITIAL FUEL      TEMPERATURE.
C      TM1       IS THE INITIAL MODERATOR TEMPERATURE.  ALWAYS EQUALS
C                         313.89 DEG. C.
C      TT        IS THE TOTAL TIME.  IT IS AN ARRAY WHOSE VALUES FORM
C                   THE INDEPENDENT VARIABLE (THE "X-AXIS") IN GRAPHICS
C                         AND IS PRINTED IN THE TABULAR OUTPUT AS WELL.
C      TTOT      IS THE TOTAL AMOUNT OF TIME THAT THE RAMP INPUT IS IN
C                         EFFECT.
C      VO        IS THE FRACTIONAL AMOUNT OF OPENING OF THE THROTTLE
C                         VALVE ON THE SECONDARY SIDE OF THE STEAM
C                         GENERATOR.
C      VOSS      IS THE FRACTIONAL AMOUNT OF THE TOTAL OPENING OF THE
C                         THROTTLE VALVE ON THE SECONDARY SIDE OF THE
C                         STEAM GENERATOR, BEFORE ANY CHANGES ARE APPLIED.
C      Y         IS THE NUMBER THAT CHANGES BY ONE EVERY 0.2 SECONDS.
C                         ITS FUNCTION IS THAT IT ALLOWS INITIALIZATIONS
```

```
C                        TO BE MADE IN GALBA DURING THE FIRST 0.2 SECONDS
C                        ONLY.
C      ZZ     IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                        DETERMINES WHETHER THROTTLE VALVE POSITION IS TO
C                        BE PERMANENTLY FIXED, AFTER IUT IS PERTURBED
C                        FROM SOME STEADY-STATE VALUE.
C
C      THE FOLLOWING REAL VARIABLES ARE USED IN NERO:
C
C      A      IS THE SUMMATION OF THE 6 GROUPS OF DELAYED-NERUTRON
C                        PRECURSORS.
C      ABN    IS THE TIME INTERVAL AT WHICH TABULAR OUTPUT IS
C                        PRINTED.
C      ALPHFS IS THE VARIABLE THAT RETAINS THE PREVIOUS VALUE FOR
C                        ALPHF, IN CASE THE USER LATER DECIDES NOT TO
C                        CHANGE IT AFTER ALL.
C      ALPHMS IS THE VARIABLE THAT RETAINS THE PREVIOUS VALUE FOR
C                        ALPHM, IN CASE THE USER LATER DECIDES NOT TO
C                        CHANGE IT AFTER ALL.
C      CPM    IS THE HEAT CAPACITY OF THE REACTOR COOLANT, USED IN
C                        COMPUTING THE HEAT TRANSFER COEFFICIENT.
C      DRON   IS THE REACTIVITY PERTURBATION, IN CENTS.
C      DVO    IS THE VALVE PERTURBATION ON THE SECONDARY SIDE OF THE
C                        STEAM GENERATOR, IN PERCENT.  AFTER CONVERSION
C                        TO A DECIMAL, IT IS ADDED TO THE STEADY-STATE
C                        QUANTITY.
C      FRAC   IS THE VARIABLE USED IN DETERMINING THE REACTOR COOLANT
C                        TEMPERATURE SO THAT THE HEAT TRANSFER COEFFI-
C                        CIENT CAN BE COMPUTED.  IT ABOUT EQUALS 0.5, BUT
C                        VARIES SLIGHTLY BECAUSE HEAT CAPACITY DOES NOT
C                        VARY LINEARLY WITH TEMPERATURE.
C      HP     IS THE HEAT TRANSFER COEFFICIENT FOR THE REACTOR
C                        COOLANT.
C      KP     IS THE THERMAL CONDUCTIVITY OF THE REACTOR COOLANT.
C      LNT    IS THE TOTAL TIME THAT THE PROGRAM WILL RUN, IN MINUTES
C      MUP    IS THE DYNAMIC VISCOSITY    OF THE REACTOR COOLANT
C      NU     IS THE NUSSELT NUMBER       OF THE REACTOR COOLANT.
C      NUP    IS THE KINEMATIC VISCOSITY  OF THE REACTOR COOLANT.
C      N110   IS THE TOTAL REACTOR POWER, IN MW, AND IS PRINTED IN
C                    THE TABULAR OUTPUT.
C      PROD   IS THE TOTAL AMOUNT OF REACTIVITY IN THE RAMP-INPUT
C                        MODEL, IN CENTS.
C      PRP    IS THE PRANDTL  NUMBER      OF THE REACTOR COOLANT.
C      REP    IS THE REYNOLDS NUMBER      OF THE REACTOR COOLANT.
C      RHON   IS THE PRE-PERTURBATION REACTIVITY (USUALLY ZERO),
C                        IN CENTS.
C      ROP    IS THE DENSITY              OF THE REACTOR COOLANT.
C      STMG   IS THE CURRENT STEAM GENERATOR OUPUT, IN MW.  IT IS
C                        WHAT IS PRINTED IN THE TABULAR OUTPUT.
C      TC     IS THE AVERAGE CLADDING SURFACE TEMPERATURE.
C      TF     IS THE AVERAGE FUEL TEMPERATURE.
C      TI1    IS THE INITIAL REACTOR INLET TEMPERATURE.
C      TO1    IS THE INITIAL REACTOR OUTLET TEMPERATURE.
C      TOTHO  IS THE TIME MEASURE DETERMINING WHETHER NERO WILL
C                        PASS CONTROL FROM OTHO TO GALBA.  AT THE END
C                        OF EVERY 0.2 SECOND INTERVAL, IT WILL NOT; AT
C                        ALL OTHER TIMES IT WILL.  THIS IS SO ACCURATE
```

```
C                                REACTIVITY CALCULATIONS AFFECTING FEEDBACK CAN
C                                BE MADE.
C          TM      IS THE AVERAGE FUEL MODERATOR TEMPERATURE.  IT IS
C                                PRINTED IN TABULAR OUTPUT.
C          TW      IS THE INTERVAL AT WHICH TABULAR OUTPUT IS PRINTED.
C          VMOD    IS THE VELOCITY OF FLUID FLOW WHILE IN THE CORE.
C
C          THE FOLLOWING ARE INTEGER VARIABLES:
C
C          AB      IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                                DETERMINES WHETHER THE OUTPUT ABBREVIATION
C                                OPTION IS TO BE USED.
C          ALA     IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                                DETERMINES WHETHER ANY OR ALL OF THE REACTIVITY
C                                COEFFICIENTS WILL BE VARIED.
C          CSC     IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                                DETERMINES WHETHER THERE IS TO BE A CHANGE IN
C                                ANY OF THE REACTOR CONTROL SYSTEM PARAMETERS.
C          CSCG    IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                                DETERMINES WHETHER THE GAIN PARAMETER OF THE
C                                RCS WILL BE ALTERED.
C          CSCG    IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                                DETERMINES WHETHER THE GAIN PARAMETER WILL BE
C                                ALTERED.
C          CSCG    IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
C                                DETERMINES WHETHER THE GAIN PARAMETER WILL BE
C                                ALTERED.
C

      REAL AA, ALPHF, ALPHM, ALPHP, CPPAV, DN1, DRO, H, MDOTP, N10,
     +      RHO, RIR, STMGEN, TAU, TAUC, TF1, TM1, TTOT, VO, VOSS, Y
      REAL ADTI(1510), ADTO(1510), DTF(1510), DTI(1510), DTM(1510),
     +      DTO (1510), DTSG(1510), DTS(1510), RP(1510),  TT (1510),
     +      PT  (1510), STMG(1510), B(6)
      INTEGER CS, EIG, F, IS, J, K, KF, LND, NK, NN, PJ, Q, RI, ZZ
      COMMON  AA, ADTI, ADTO, ALPHF, ALPHM, CPPAV, CS, DN1, DRO,
     +        DTF, DTI, DTM, DTO, DTS, DTSG, EIG, F, H, IS, K, KF,
     +        LND, MDOTP, NK, NN, N10, Q, PJ, PT, RHO, RI, RIR,
     +        RP, STMG, STMGEN, TAU, TAUC, TF1, TM1, TT, TTOT, VO,
     +        VOSS, Y, ZZ
      REAL    A, ABN, ALPHFS, ALPHMS, CPM, DN2, DRON, DVO, FRAC, HP,
     +        KP, MUP, NU, NUP, N110, PROD, PRP, REP, RHON, ROP,
     +        STMGN, TC, TF, TI1, TO1, TOTHO, TM, TW, VMOD
      INTEGER AB, ABC, ALA, CSC, CSCG, CSCTC, CSCT, I, LN, LNT, P,
     +        YY

          A       =         0.0

C
C         GENERAL COMMENTS
C
C         THE PURPOSE OF THE MAIN PROGRAM NERO IS TO CONTROL THE OPERA-
C     TION OF THE OTHER PROGRAMS, CONTROL THE FORMAT OF THE OUTPUT, AND
C     ESTABLISH SOME OF THE INITIAL VALUES OF THE OPERATING PARAMETERS,
C     AT THE USER'S OPTION.
C         NERO TYPICALLY OPERATES IN THE FOLLOWING MANNER:  INSTRUCTIONS
C     TO CHOOSE A PARAMETER ARE ISSUED BY NERO.  IF THE SIGNIFICANCE OF
```

```
C     THE PARAMETERS IS NOT IMMEDIATELY OBVIOUS. AN EXPLANATION  S PRO-
C     VIDED.  THEN THE USER IS INSTRUCTED TO CHOOSE WHETHER HE OR SHE
C     WANTS THE OPTION PRESENTED OR NOT.  HE OR SHE MUST MAKE A CHOICE
C     OR NERO WILL CHOOSE AN OPTION OR A VALUE BY DEFAULT.  THEN A TEST
C     IS APPLIED.  IF THE USER'S CHOICE MEETS CERTAIN ELIGIBILITY CRI-
C     TERIA, (THAT IS, IF THE CHOICE IS WITHIN PERMISSIBLE LIMITS)
C     NERO JUMPS TO THE NEXT SET OF INSTRUCTIONS.  IF THE CRITERIA ARE
C     NOT MET, NERO PRINTS A MESSAGE TO THAT EFFECT, AND JUMPS BACK-
C     WARD AND MAKES THE USER CHOOSE AGAIN.
C           IN CHOOSING AN OPTION, WHEN THE USER TYPES IN 1, THE OPTION
C     IS ELECTED; WHEN 2 IS TYPED IN, THE OPTION IS NOT SELECTED.
C
C
C           HERE,THE CHOICE OF ISOTOPE IS MADE.  THE USER MAY CHOOSE
C     EITHER U-235, PU-239, OR U-233.
C
 10     WRITE  (6, 20)
 20     FORMAT ('0', '               WHICH ISOTOPE DO YOU WISH TO UTILIZE?')
        WRITE  (6, 30)
 30     FORMAT (X, '               TYPE  1  FOR U-233,  2  FOR U-235,  3
      +  FOR PU-239.')
        READ    40, IS
 40     FORMAT (I1)

        IF ((IS .EQ. 1) .OR. (IS .EQ. 2) .OR. (IS .EQ. 3)) GO TO 70

        WRITE  (6, 50)
 50     FORMAT (X, '               YOU HAVE TYPED IN A NUMBER THAT CANNOT
      + BE UTILIZED.  YOU WILL HAVE')
        WRITE  (6, 60)
 60     FORMAT (X, '               TO TRY AGAIN.')

        GO TO 10

C
C           HERE THE DEFAULT PARAMETERS OF THE REACTOR CONTROL SYSTEM AND
C     THE MODERATOR COEFFICIENT OF REACTIVITY ARE ESTABLISHED.
C
 70                            AA      =  -1.0E-06
                               ALPHM   =  -8.60E-06
                               TAU     =   5.0
                               TAUC    =   5.0
                               DN1     =   0.0
                               DN2     =   0.0
                               DVO     =   0.0
                               RIR     =   0.0

C
C           HERE THE SIX GROUPS OF DELAYED PRECURSORS AND DECAY CONSTANTS
C     AND THE DEFAULT VALUE OF THE DOPPLER COEFFICIENT OF REACTIVITY
C     ARE INITIALIZED, DEPENDING ON WHICH ISOTOPE WAS CHOSEN EARLIER.
C
        IF (IS .EQ. 1)  B(1)    =   2.2876E-04
        IF (IS .EQ. 1)  B(2)    =   7.9534E-04
        IF (IS .EQ. 1)  B(3)    =   6.7032E-04
        IF (IS .EQ. 1)  B(4)    =   7.3948E-04
        IF (IS .EQ. 1)  B(5)    =   1.3566E-04
```

```
      IF (IS .EQ. 1)  B(6)     =  9.0440E-05
      IF (IS .EQ. 1)  ALPHF    = -1.07E-05
      IF (IS .EQ. 2)  B(1)     =  2.1450E-04
      IF (IS .EQ. 2)  B(2)     =  1.4235E-03
      IF (IS .EQ. 2)  B(3)     =  1.2740E-03
      IF (IS .EQ. 2)  B(4)     =  2.5675E-03
      IF (IS .EQ. 2)  B(5)     =  7.4750E-04
      IF (IS .EQ. 2)  B(6)     =  2.7300E-04
      IF (IS .EQ. 2)  ALPHF    = -2.61E-05
      IF (IS .EQ. 3)  B(1)     =  7.4200E-05
      IF (IS .EQ. 3)  B(2)     =  6.3176E-04
      IF (IS .EQ. 3)  B(3)     =  4.4732E-04
      IF (IS .EQ. 3)  B(4)     =  6.9112E-04
      IF (IS .EQ. 3)  B(5)     =  1.8232E-04
      IF (IS .EQ. 3)  B(6)     =  9.3280E-05
      IF (IS .EQ. 3)  ALPHF    = -0.85E-05

C
C          HERE THE SUM TOTAL OF THE DELAYED PRECURSORS IS CALCULATED.
C     THIS WILL BE NEEDED LATER ON TO CONVERT REACTIVITY FROM UNITS OF
C     "CENTS" TO UNITS OF "((DK/K)/C)".
C
      DO 80 I = 1,6
         A      =       B(I) + A
 80   CONTINUE

C
C          HERE THE USER DECIDES WHETHER THE OPTION OF USING FREE REAC-
C     TOR KINETICS (THAT IS, NO FEEDBACK OR CONTROL SYSTEM) IS TO BE
C     USED.
C
 90   WRITE  (6, 100)
 100  FORMAT (X, '                DO YOU WISH TO HAVE FREE KINETICS ON
     + THIS RUN?  FREE KINETICS MEANS')
      WRITE  (6, 110)
 110  FORMAT (X, '                THAT REACTIVITY COEFFICIENTS ARE EQUAL
     + TO ZERO.  THERE WILL BE NO')
      WRITE  (6, 120)
 120  FORMAT (X, '                FEEDBACK AND NO CONTROL SYSTEM.  NOTE:
     +  THE PROMPT JUMP')
      WRITE  (6, 130)
 130  FORMAT (X, '                APPROXIMATION CAN BE SELECTED LATER ON
     + ONLY IF THE FREE KINETICS,')
      WRITE  (6, 140)
 140  FORMAT (X, '                OPTION IS SELECTED NOW.  IF YOU WANT
     + FREE KINETICS,')
      WRITE  (6, 150)
 150  FORMAT (X, '                TYPE IN 1;, IF NOT, TYPE IN 2')
      READ    160, KF
 160  FORMAT (I1)

      IF ((KF .EQ. 1) .OR. (KF .EQ. 2)) GO TO 200

      WRITE  (6, 170)
 170  FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 180)
```

```
  180 FORMAT (X, '                TO TRY AGAIN.')
      WRITE  (6, 190)
  190 FORMAT(' ')

      GO TO 90

  200 IF (KF .EQ. 1)  GO TO 1590

      WRITE  (6, 210)
  210 FORMAT (X, '                WHAT LEVEL (IN CENTS) PRE-PERTURBATION
     + REACTIVITY IS DESIRED?')
      WRITE  (6, 220)
  220 FORMAT (X, '                USE FORMAT F5.1    EXAMPLE: -05.0 EQUALS
     + -5 CENTS.')
      READ    230, RHON
  230 FORMAT (F4.1)

      IF ((RHON .GT. -100.0)  .AND.  (RHON .LT. 100.0)) GO TO 270

      WRITE  (6, 240)
  240 FORMAT (X, '                YOU HAVE SELECTED A VALUE THAT WILL
     + RESULT IN A PROMPT SUPERCRITICAL')
      WRITE  (6, 250)
  250 FORMAT (X, '                CONDITION.  THIS IS NOT ALLOWED, AND YOU
     + WILL HAVE TO TRY AGAIN.')
      WRITE  (6, 260)
  260 FORMAT (' ')
      GO TO 200

C
C        THIS EQUATION CONVERTS PRE-PERTURBATION REACTIVITY FROM
C   "CENTS" TO "((DK/K)/C)".
C
  270                 RHO   = 0.01 * RHON * A

C
C        THIS SECTION EXPLAINS REACTIVITY COEFFICIENTS.
C
      WRITE  (6, 280)
  280 FORMAT (X, '                DOPPLER                   COEFFICIENT OF
     + REACTIVITY IS')
      WRITE  (6, 290) ALPHF
  290 FORMAT (X, '                ', E10.3, 2X, '(dK / K) / C')
      WRITE  (6, 300)
  300 FORMAT (X, '                MODERATOR TEMPERATURE COEFFICIENT OF
     + REACTIVITY IS', 2X, E10.3)
      WRITE  (6, 310) ALPHM
  310 FORMAT (X, '                ', E10.3, 2X, '(dK / K) / C')
      IF (KF .EQ. 1)  GO TO 920

      WRITE  (6, 320)
  320 FORMAT (X, '                NOTE:  REACTIVITY COEFFICIENTS AS GIVEN
     + ABOVE ARE TYPICAL ONES.')
      WRITE  (6, 330)
  330 FORMAT (X, '                ADDITIONALLY, SINCE THEY ARE NEGATIVE
     + NUMBERS, FEEDBACK WILL ALSO')
      WRITE  (6, 340)
```

```
 340  FORMAT (X, '                BE NEGATIVE.  UNLESS YOU WANT TO CHANGE
     + THE VALUES OF THESE')
      WRITE  (6, 350)
 350  FORMAT (X, '                COEFFICIENTS, THEY WILL KEEP THE VALUES
     + ALREADY STATED. THE DOPPLER')
      WRITE  (6, 360)
 360  FORMAT (X, '                COEFFICIENT TYPICALLY VARIES FROM
     + -2.0E-05 TO 3.6E-05 FOR U-235.')
      WRITE  (6, 370)
 370  FORMAT (X, '                FOR OTHER FUELS IT IS SOMEWHAT LESS.
     + THE MODERATOR TEMPERATURE')
      WRITE  (6, 390)
 390  FORMAT (X, '                COEFFICIENT, WHICH IN THIS CASE IS A
     + COMBINED TEMPERATURE AND')
      WRITE  (6, 400)
 400  FORMAT (X, '             -  PRESSURE COEFFICIENT, TYPICALLY RANGES
     + FROM -3.2E-04 TO 1.7E-04')
      WRITE  (6, 410)
 410  FORMAT (X, '                (dK / K) / C')
      WRITE  (6, 420)
 420  FORMAT (' ')

C         HERE THE USER SELECTS A VALUE FOR PRE-PERTURBATION REACTIVITY.
C    (RHON)  USUALLY IT WILL BE ZERO.  HOWEVER, IT IS NOT ALLOWED TO BE
C    EITHER LESS THAN -90.0 CENTS, OR GREATER THAN + 90.0 CENTS, AS
C    THIS IS CLOSE TO A PROMPT CRITICALITY CONDITION.
C
 430  WRITE  (6, 440)
 440  FORMAT (X, '                IF YOU WISH TO INSERT YOUR OWN
     + REACTIVITY COEFFICIENTS, TYPE IN 1;')
      WRITE  (6, 450)
 450  FORMAT (X, '                IF NOT, TYPE IN 2')
      READ    460, ALA
 460  FORMAT (I1)

      IF (ALA .EQ. 2)  GO TO 630
      IF (ALA .EQ. 1)  GO TO 490

C
C         HERE THE USER CHOOSES WHETHER A CHANGE IN REACTIVITY COEFFI-
C    CIENTS IS DESIRED.  IF NO CHANGE IS DESIRED, THE DEFAULT VALUES
C    WILL REMAIN.  IF CHANGE IS DESIRED, THE PROGRAM WILL ENABLE THE
C    USER TO SELECT DESIRED VALUE(S).
C
      WRITE  (6, 470)
 470  FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 480)
 480  FORMAT (X, '                TO TRY AGAIN.')

      GO TO 430

 490  WRITE  (6, 500)
 500  FORMAT (X, '                TYPE IN DESIRED DOPPLER COEFFICIENT OF
     + REACTIVITY, USING FORMAT')
      WRITE  (6, 510)
 510  FORMAT (X, '                E10.3:      EX:  A REACTIVITY OF -8.61E-05
```

```
      + WOULD BE WRITTEN AS')
        WRITE  (6, 520) ALPHF
 520    FORMAT (X, '              -0.861E-04:  CURRENT VALUE IS', 2X,E10.3)

C
C           ALPHFS AND ALPHMS ARE HOLDING VARIABLES OF THE DEFAULT VALUES
C      OF THE REACTIVITY COEFFICIENTS.  IF THE USER FIRST SELECTS IM-
C      PERMISSIBLE VALUES FOR THE REACTIVITY, AND THEN, AFTER THE PROGRAM
C      JUMPS BACKWARD TO CHALLENGE THE CHOICE, DECIDES NOT TO CHANGE THE
C      THE VALUES AFTER ALL, THESE HOLDING VARIABLES PREVENTS THE DEFAULT
C      VALUES FROM BEING LOST.
C
            ALPHFS =        ALPHF

        READ    530, ALPHF
 530    FORMAT (E10.3)

        IF ((ALPHF .GE. -0.001)  .AND.  (ALPHF .LE. 0.001)) GO TO 560

            ALPHF  =        ALPHFS

        WRITE  (6, 540)
 540    FORMAT ('0', '             YOU CHOSE A REACTIVITY COEFFICIENT
      + THAT IS TOO LARGE TO BE')
        WRITE  (6, 550)
 550    FORMAT (X, '              USED.  YOU WILL HAVE TO SELECT ANOTHER.')

        GO TO 430

 560    WRITE  (6, 570)
 570    FORMAT (X, '              TYPE IN DESIRED TEMPERATURE COEFFICIENT,
      + USING FORMAT E10.3')
        WRITE  (6, 580)
 580    FORMAT (X, '              EX:  A REACTIVITY OF 7.22E-06 WOULD BE
      + WRITTEN AS 00.722E-05')

            ALPHMS =        ALPHM

        WRITE  (6, 590) ALPHM
 590    FORMAT (X, '              CURRENT VALUE IS', 2X, E10.3)
        READ    600, ALPHM
 600    FORMAT (E10.3)

        IF  ((ALPHM .GE. -0.0001)  .AND.  (ALPHM .LE. 0.0001))  GO TO 630

            ALPHM  =        ALPHMS

        WRITE  (6, 610)
 610    FORMAT (X, '              YOU HAVE CHOSEN A REACTIVITY COEFFICIENT
      + THAT IS TOO LARGE TO BE')
        WRITE  (6, 620)
 620    FORMAT (X, '              USED.  YOU WILL HAVE TO SELECT ANOTHER.')

        GO TO 560

 630    WRITE  (6, 640)
 640    FORMAT (X, '              DO YOU WISH TO UTILIZE A CONTROL
```

```
     + SYSTEM IN THE REACTOR, OR')
       WRITE  (6, 650)
  650  FORMAT (X, '                 NOT?  IF YOU WANT A CONTROL SYSTEM, TYPE
     + IN 1;  IF NOT, TYPE IN 2')
       READ    660, CS
  660  FORMAT (I1)

       IF (CS .EQ. 2)  GO TO 1590
       IF (CS .EQ. 1)  GO TO 700

       WRITE  (6, 670)
  670  FORMAT (X, '                 YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
       WRITE  (6, 680)
  680  FORMAT (X, '                 TO TRY AGAIN.')
       WRITE  (6, 690)
  690  FORMAT (' ')

       GO TO 630

C
C         HERE THE USER DECIDES IF A CONTROL SYSTEM IS DESIRED.
C
  700  WRITE  (6, 710)
  710  FORMAT (X, '                 IN THE CONTROL SYSTEM MODELED HERE,
     + PROPORTIONAL CONTROL AND')
       WRITE  (6, 720)
  720  FORMAT (X, '                 INTEGRAL CONTROL ARE USED.  WHEN THE
     + CONTROLLER "SENSES" A CHANGE')
       WRITE  (6, 730)
  730  FORMAT (X, '                 IN THE AVERAGE COOLANT TEMPERATURE OF
     + REACTOR, IT INSERTS A')
       WRITE  (6, 740)
  740  FORMAT (X, '                 REACTVITY STEP TO COUNTERACT IT.')

       WRITE  (6, 750)
  750  FORMAT ('0', '                 THERE ARE THREE PARAMETERS WHICH
     + TOGETHER DETERMINE THE RESPONSE')
       WRITE  (6, 760)
  760  FORMAT (X, '                 CHARACTERISTICS OF THE CONTROL SYSTEM:
     + THE GAIN, THE TIME CONSTANT')
       WRITE  (6, 770)
  770  FORMAT (X, '                 OF THE FIRST-ORDER DIFFERENTIAL EQUATION
     + WHICH DESCRIBES THE')
       WRITE  (6, 780)
  780  FORMAT (X, '                 ACTIONS OF THE MECHANICAL ACTUATOR,
     + AND A CONSTANT TAU USED TO')
       WRITE  (6, 790)
  790  FORMAT (X, '                 GOVERN THE ACTIONS OF THE INTEGRAL
     + CONTROLLER.  THE GAIN HAS UNITS')
       WRITE  (6, 800)
  800  FORMAT (X, '                 OF REACTIVITY / (DEG. C - SEC), AND
     + CONTROLS BOTH THE PROPORTIONAL')
       WRITE  (6, 810)
  810  FORMAT (X, '                 CONTROLLER AND THE INTEGRAL CONTROLLER
     + TOGETHER..  INCREASING THE')
       WRITE  (6, 820)
```

```
  820 FORMAT (X, '              VALUE OF THE GAIN CAUSES A DIRECTLY
     + PROPORTIONAL CHANGE IN THE')
      WRITE (6, 830)
  830 FORMAT (X, '              MAGNITUDE OF REACTIVITY.  FOR A GIVEN
     + ERROR, RESPONSE SPEED OF THE')
      WRITE (6, 840)
  840 FORMAT (X, '              CONTROL SYSTEM IS INCREASED BY DECREASING
     + THE TIME CONSTANT OF THE')
      WRITE (6, 850)
  850 FORMAT (X, '              DIFFERENTIAL EQUATION, AND VICE VERSA.
     + THE CONSTANT TAU ADJUSTS')
      WRITE (6, 860)
  860 FORMAT (X, '              THE EFFECT OF THE INTEGRAL PORTION OF
     + THE CONTROLLER.  INCREASING')
      WRITE (6, 870)
  870 FORMAT (X, '              TAU WILL DECREASE THE EFFECT OF THE
     + INTEGRAL CONTROLLER, AND VICE')
      WRITE (6, 880)
  880 FORMAT (X, '              VERSA.')
      WRITE (6, 890)
  890 FORMAT ('0', '             AS OPERATOR, YOU WILL BE ABLE TO ADJUST
     + THE ACTIONS OF THE')
      WRITE (6, 900)
  900 FORMAT (X, '              REACTOR CONTROL SYSTEM BY SELECTING YOUR
     + OWN VALUES FOR THE THREE')
      WRITE (6, 910)
  910 FORMAT (X, '              PARAMETERS LISTED ABOVE.')
  920 WRITE (6, 930)
  930 FORMAT ('0', '             CURRENT VALUE OF THE GAIN          IS
     + -1.0E-06;  ITS VALUE TYPICALLY')
      WRITE (6, 940)
  940 FORMAT (X, '                                         VARIES FROM
     + 0.0 TO -4.0E-06')
      WRITE (6, 950)
  950 FORMAT (X, '              CURRENT VALUE OF THE TIME CONSTANT IS
     + 5 SEC.;   ITS VALUE TYPICALLY')
      WRITE (6, 960)
  960 FORMAT (X, '                                         VARIES FROM
     + 2 SEC.  TO  20 SEC.')
      WRITE (6, 970)
  970 FORMAT (X, '              CURRENT VALUE OF THE PARAMETER TAU IS
     + 5 SEC.;  ITS VALUE TYPICALLY')
      WRITE (6, 980)
  980 FORMAT (X, '                                         VARIES FROM
     + 2 SEC.  TO  20 SEC.')
       IF (KF .EQ. 1)  GO TO 1590
C
C         HERE THE USER DECIDES IF ANY CHANGES TO ANY OF THE CONTROL
C    SYSTEM PARAMETERS ARE TO BE MADE.  IF NO CHANGES ARE MADE, DEFAULT
C    VALUES WILL REMAIN.  THE USER WILL BE ABLE TO CHANGE ANY OR ALL OF
C    THE PARAMETERS.
C
      WRITE (6, 990)
  990 FORMAT ('0', '             THE VALUES GIVEN ABOVE WILL REMAIN AS
```

```
     + THEY ARE UNLESS YOU')
       WRITE  (6, 1000)
  1000 FORMAT (X, '                CHANGE THEM.')
  1010 WRITE  (6, 1020)
  1020  FORMAT (X, '               DO YOU WISH TO CHANGE ANY OR ALL OF THE
     + VALUES OF THE PARAMETERS OF')
       WRITE  (6, 1030)
  1030  FORMAT (X, '               THE CONTROL SYSTEM?  IF SO, TYPE IN 1;
     + IF NOT, TYPE IN 2')
       WRITE  (6, 1040)
  1040 FORMAT (X, '                IF NOT, TYPE IN 2')
       READ    1050, CSC
  1050 FORMAT (I1)

       IF (CSC .EQ. 2)  GO TO 1590
       IF (CSC .EQ. 1)  GO TO 1090

       WRITE  (6, 1060)
  1060 FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
       WRITE  (6, 1070)
  1070 FORMAT (X, '                TO TRY AGAIN.')
       WRITE  (6, 1080)
  1080 FORMAT (' ')

       GO TO 1010

  1090 WRITE  (6, 1100)
  1100 FORMAT (X, '                DO YOU WISH TO INSERT YOUR OWN VALUE FOR
     + THE GAIN?')
       WRITE  (6, 1110)
  1110  FORMAT (X, '               CURRENT VALUE IS  -1.0E-06')
       WRITE  (6, 1120)
  1120 FORMAT (X, '                IF SO, TYPE IN 1;  IF NOT, TYPE IN 2')
       READ    1130, CSCG
  1130 FORMAT (I1)

       IF (CSCG .EQ. 2)  GO TO 1260
       IF (CSCG .EQ. 1)  GO TO 1170

       WRITE  (6, 1140)
  1140 FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
       WRITE  (6, 1150)
  1150 FORMAT (X, '                TO TRY AGAIN.')
       WRITE  (6, 1160)
  1160 FORMAT (' ')

       GO TO 1090

  1170 WRITE  (6, 1180)
  1180 FORMAT (X, '                WRITE IN YOUR OWN VALUE FOR THE GAIN,
     + USING FORMAT E8.1')
       WRITE  (6, 1190)
  1190 FORMAT (X, '                EX:  A GAIN OF -0.0004 PER DEG. C - SEC
     + WOULD BE WRITTEN AS -4.0E-04')
       WRITE  (6, 1200)
```

```
 1200 FORMAT (X, '              NOTE:  THE GAIN IS ALWAYS A NEGATIVE
     + NUMBER, AND NEVER SMALLER THAN')
      WRITE  (6, 1210)
 1210 FORMAT (X, '              0.0:  CURRENT VALUE IS  -1.0E-06')
      READ    1220, AA
 1220 FORMAT (E8.1)

      IF ((AA .LE. 0.0)  .AND.  (AA .GE. -4.0E-06)) GO TO 1260

C
C          THIS EQUATION "SAVES" THE DEFAULT VALUE OF THE GAIN IN CASE
C     THE USER CHANGES HIS OR HER MIND AND DECIDES NOT TO CHANGE T
C     VALUE AFTER ALL.  THE OTHER CONTROL SYSTEM PARAMETERS CAN ALSO BE
C     "SAVED" IN THE SAME WAY.
C
          AA     =       -5.0E-04

      WRITE  (6, 1230)
 1230 FORMAT (X, '              YOU HAVE CHOSEN A VALUE THAT IS OUTSIDE
     + THE PERMISSIBLE LIMITS.')
      WRITE  (6, 1240)
 1240 FORMAT (X, '              SELECT AGAIN, REMEMBERING THAT THE VALUE
     + SELECTED MUST LIE BETWEEN')
      WRITE  (6, 1250)
 1250 FORMAT (X, '              0.0 AND -2.0E-07:  CURRENT VALUE IS
     + -5.0E-04')
      GO TO 1090

 1260 WRITE  (6, 1270)
 1270 FORMAT (X, '              DO YOU WISH TO INSERT YOUR OWN VALUE FOR
     + THE TIME CONSTANT?')
      WRITE  (6, 1280)
 1280 FORMAT (X, '              CURRENT VALUE IS   5.0')
      WRITE  (6, 1290)
 1290 FORMAT (X, '              IF SO, TYPE IN 1;  IF NOT, TYPE IN 2')
      READ    1300, CSCTC
 1300 FORMAT (I1)

      IF (CSCTC .EQ. 2)  GO TO 1430
      IF (CSCTC .EQ. 1)  GO TO 1340

      WRITE  (6, 1310)
 1310 FORMAT (X, '              YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 1320)
 1320 FORMAT (X, '              TO TRY AGAIN.')
      WRITE  (6, 1330)
 1330 FORMAT (' ')

      GO TO 1260

 1340 WRITE  (6, 1350)
 1350 FORMAT (X, '              WRITE IN YOUR OWN VALUE FOR THE TIME
     + CONSTANT, USING FORMAT F3.1')
      WRITE  (6, 1360)
 1360 FORMAT (X, '              EX:  A VALUE OF 4 SECONDS WOULD BE
     + WRITTEN AS 4.0')
```

```
      WRITE  (6, 1370)
 1370 FORMAT (X, '                    CURRENT VALUE IS   5.0')
      READ    1380, TAUC
 1380 FORMAT (F3.1)

      IF ((TAUC .GE. 2.0)  .AND.  (TAUC .LE. 20.0))  GO TO 1430

          TAUC    =          5.0

      WRITE  (6, 1390)
 1390 FORMAT (X, '                    YOU HAVE CHOSEN A VALUE THAT IS OUTSIDE
     + THE PERMISSIBLE LIMITS.')
      WRITE  (6, 1400)
 1400 FORMAT (X, '                    SELECT AGAIN, REMEMBERING THAT THE VALUE
     + SELECTED MUST LIE BETWEEN')
      WRITE  (6, 1410)
 1410 FORMAT (X, '                    2.0 AND 20.0:  CURRENT VALUE IS 5.0')
      WRITE  (6, 1420)
 1420 FORMAT (' ')

      GO TO 1260

 1430 WRITE  (6, 1440)
 1440 FORMAT (X, '                    DO YOU WISH TO INSERT YOUR OWN VALUE FOR
     + THE PARAMETER TAU?')
      WRITE  (6, 1450)
 1450 FORMAT (X, '                    CURRENT VALUE IS   5.0')
      WRITE  (6, 1460)
 1460 FORMAT (X, '                    IF SO, TYPE IN 1;  IF NOT, TYPE IN 2')
      READ    1470, CSCT
 1470 FORMAT (I1)

      IF (CSCT .EQ. 2)  GO TO 1590
      IF (CSCT .EQ. 1)  GO TO 1510

      WRITE  (6, 1480)
 1480 FORMAT (X, '                    YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 1490)
 1490 FORMAT (X, '                    TO TRY AGAIN.')
      WRITE  (6, 1500)
 1500 FORMAT (' ')

      GO TO 1430

 1510 WRITE  (6, 1520)
 1520 FORMAT (X, '                    WRITE IN YOUR OWN VALUE FOR THE
     + PARAMETER TAU, USING FORMAT F3.1')
      WRITE  (6, 1530)
 1530 FORMAT (X, '                    CURRENT VALUE IS   5.0')
      READ    1540, TAU
 1540 FORMAT (F3.1)

      IF ((TAU .GE. 2.0)  .AND.  (TAU .LE. 20.0))  GO TO 1590

          TAU     =          5.0

```

```
      WRITE  (6, 1550)
 1550 FORMAT (X, '                  YOU HAVE CHOSEN A VALUE THAT IS OUTSIDE
     + THE PERMISSIBLE LIMITS.')
      WRITE  (6, 1560)
 1560 FORMAT (X, '                  SELECT AGAIN, REMEMBERING THAT THE VALUE
     + SELECTED MUST LIE BETWEEN')
      WRITE  (6, 1570)
 1570 FORMAT (X, '                  2.0 AND 20.0:  CURRENT VALUE IS 5.0')
      WRITE  (6, 1580)
 1580 FORMAT (' ')

      GO TO 1430

C          HERE, THE TWO-TIME-SCALE DECOUPLING ALGORITHM, THE PROMPT-JUMP
C     APPROXIMATION, AND THE RAMP-INPUT MODEL ARE ALL EXPLAINED.  THE
C     USER IS THEN GIVEN THE OPTION OF CHOOSING WHETHER THE DECOUPLING
C     ALGORITHM IS DESIRED.
C
 1590 WRITE  (6, 1600)
 1600 FORMAT (X, '                  BY DEFAULT, THE REACTOR IS CURRENTLY
     + USING A STEP INPUT MODEL THAT')
      WRITE  (6, 1610)
 1610 FORMAT (X, '                  SOLVES THE POINT KINETICS EQUATIONS
     + (WITHOUT EXTERNAL SOURCE)')
      WRITE  (6, 1620)
 1620 FORMAT (X, '                  DIRECTLY.  HOWEVER, IF YOU WISH YOU
     + CAN USE EITHER THE PROMPT')
      WRITE  (6, 1630)
 1630 FORMAT (X, '                  JUMP APPROXIMATION OR A RAMP INPUT
     + MODEL.  THE PROMPT JUMP')
      WRITE  (6, 1640)
 1640 FORMAT (X, '                  APPROXIMATION MODEL WOULD UTILIZE 6
     + DELAYED NEUTRON GROUPS.')
      WRITE  (6, 1650)
 1650 FORMAT (X, '                  IF YOU CHOOSE THE RAMP-INPUT MODEL, YOU
     + WILL BE ABLE TO CHOOSE THE')
      WRITE  (6, 1660)
 1660 FORMAT (X, '                  RAMP-INPUT RATE AND THE PERIOD OF TIME
     + OVER WHICH IT IS OPERATIVE.')

      WRITE  (6, 1670)
 1670 FORMAT ('0', '                  THIS PROGRAM ALSO HAS THE CAPABILITY OF
     + DECOUPLING THE 7X7 SYSTEM')
      WRITE  (6, 1680)
 1680 FORMAT (X, '                  OF POINT-KINETICS EQUATIONS (6 DELAYED -
     + NEUTRON GROUPS PLUS PROMPT')
      WRITE  (6, 1690)
 1690 FORMAT (X, '                  RESPONSE)  INTO 2 MODES:  THE SLOW-MODE
     + GROUP CONSISTING OF THE 6')
      WRITE  (6, 1700)
 1700 FORMAT (X,'                  DELAYED-NEUTRON GROUPS, AND THE FAST -
     + MODE GROUP CONSISTING OF THE')
      WRITE  (6, 1710)
 1710 FORMAT (X, '                  PROMPT RESPONSE.  THE POINT-KINETICS
     + EQUATIONS CAN THEN BE SOLVED')
      WRITE  (6, 1720)
 1720 FORMAT (X, '                  SEPARATELY FOR EACH OF THE TWO GROUPS.
```

```
          + SHORT TIME STEPS WOULD BE')
            WRITE  (6, 1730)
      1730 FORMAT (X, '              USED FOR THE FAST MODE, WHILE LONGER
          + TIME STEPS WOULD BE USED FOR')
            WRITE  (6, 1740)
      1740 FORMAT (X, '              THE SLOW MODE.  THIS WILL ACHIEVE
          + INCREASED ACCURACY AND DECREASED')
            WRITE  (6, 1750)
      1750 FORMAT (X,'              CPU TIME.  NOTE: IF YOU SELECT THIS
          + MATRIX DECOUPLING OPTION, YOU')
            WRITE  (6, 1760)
      1760 FORMAT (X, '              WILL NOT BE ABLE TO USE THE PROMPT JUMP
          + APPROXIMATION. THIS PROGRAM')
            WRITE  (6, 1780)
      1780 FORMAT (X, '              WILL PREVENT YOU FROM EVEN TRYING TO DO
          + SO.')

      1800 WRITE  (6, 1810)
      1810 FORMAT ('0', '              DO YOU WANT TO USE THE "MATRIX
          + DECOUPLING" OPTION?  IF SO,')
            WRITE  (6, 1820)
      1820 FORMAT (X, '              TYPE IN 1;  IF NOT, TYPE IN 2')
            READ     1830, EIG
      1830 FORMAT (I1)

     C
     C          HERE, IF A CONTROL SYSTEM IS NOT DESIRED, GAIN IS SET EQUAL TO
     C     ZERO, PREVENTING ANY FEEDBACK DUE TO A CONTROL SYSTEM.  IF THE -
     C     COUPLING OPTION IS DESIRED, THE PROMPT-JUMP APPROXIMATION IS NOT
     C     ALLOWED.
     C
            IF (CS  .EQ. 2)  AA  = 0.0
            IF (EIG .EQ. 2)  GO TO 1870
            IF (EIG .EQ. 1)  PJ  =  2
            IF (EIG .EQ. 1)  GO TO 1980

            WRITE  (6, 1840)
      1840 FORMAT (X, '              WHEN YOU WERE ASKED IF YOU WANTED TO
          + SELECT THE "MATRIX')
            WRITE  (6, 1850)
      1850 FORMAT (X, '              DECOUPLING" OPTION, YOU TYPED IN A
          + NUMBER THAT CANNOT BE UTILIZED.')
            WRITE  (6, 1860)
      1860 FORMAT (X, '              YOU WILL HAVE TO TRY AGAIN.')

            GO TO 1800

     C
     C          ASSUMING EARLIER DECISIONS DO NOT PRECLUDE THIS, THE USER DOES
     C     NOW HAVE THE OPTION OF SELECTING THE PROMPT-JUMP APPROXIMATION.
     C
      1870 WRITE  (6, 1880)
      1880 FORMAT (X, '              DO YOU WISH TO UTILIZE THE PROMPT JUMP
          + APPROXIMATION?  IF SO, TYPE')
            WRITE  (6, 1890)
      1890 FORMAT (X, '              IN 1;  IF NOT, TYPE IN 2')
            WRITE  (6, 1900)
```

```
 1900 FORMAT ('0', '              YOU CAN SELECT BOTH THE RAMP-INPUT
     + MODEL AND THE PROMPT JUMP')
      WRITE  (6, 1910)
 1910 FORMAT (X, '              APPROXIMATION.  BUT IF YOU DO, THE
     + PROMPT JUMP MODEL WILL NOT')
      WRITE  (6, 1920)
 1920 FORMAT (X, '              OPERATE UNTIL AFTER THE RAMP-INPUT
     + IS COMPLETE.  IF YOU SELECT THE')
      WRITE  (6, 1930)
 1930 FORMAT (X, '              PJ APPROXIMATION, THERE WILL BE NO
     + FEEDBACK AND NO CONTROL SYSTEM.')

      READ    1940, PJ
 1940 FORMAT (I1)

C
C        HERE, IF THE PROMPT-JUMP APPROXIMATION HAS BEEN SELECTED, THE
C   RAMP-INPUT MODEL WILL AUTOMATICALLY NOT BE ALLOWED, AND NERO WILL
C   JUMP BEYOND IT.
C
      IF ((PJ .EQ. 1) .OR. (PJ .EQ. 2))  GO TO 1980

      WRITE  (6, 1950)
 1950 FORMAT (X, '              YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 1960)
 1960 FORMAT (X, '              TO TRY AGAIN.')
      WRITE  (6, 1970)
 1970 FORMAT (' ')

      GO TO 1870

C
C        HERE, THE USER HAS THE CHOICE OF SELECTING, OR NOT SE
C   THE RAMP-INPUT MODEL (ASSUMING THAT EARLIER DECISIONS DO NOT-
C   CLUDE THIS).
C
 1980 WRITE  (6, 1990)
 1990 FORMAT (X, '              DO YOU WISH TO SELECT THE RAMP-INPUT
     + MODEL?  IF SO, TYPE IN 1;')
      WRITE  (6, 2000)
 2000 FORMAT (X, '              IF NOT, TYPE IN 2')
      READ    2010, RI
 2010 FORMAT (I1)

      IF ((RI .EQ. 1)  .OR.  (RI .EQ. 2))  GO TO 2050

      WRITE  (6, 2020)
 2020 FORMAT (X, '              YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 2030)
 2030 FORMAT (X, '              TO TRY AGAIN.')
      WRITE  (6, 2040)
 2040 FORMAT (' ')

      GO TO 1980
```

```
      2050 IF (RI .EQ. 2)  GO TO 2360

C
C            HERE, THE RAMP-INPUT MODEL IS EXPLAINED.  IN PARTICULAR, THE
C       USER WILL BE PREVENTED FROM ALLOWING A GIVEN RAMP-INPUT RATE FROM
C       RUNNING SO LONG THAT IT WILL RESULT IN A PROMPT CRITICALITY CON-
C       DITION.  THE RAMP-INPUT RATE MAY NOT EXCEED PLUS OR MINUS 10 CENTS
C       PER SECOND, AND TOTAL ACCUMULATED REACTIVITY MAY NOT EXCEED 90
C       CENTS.
C
           WRITE  (6, 2060)
      2060 FORMAT (X, '               SINCE YOU HAVE CHOSEN THE RAMP - INPUT
          + MODEL, YOU WILL NOW HAVE TO')
           WRITE  (6, 2070)
      2070 FORMAT (X, '               SELECT A RAMP - INPUT RATE AND A PERIOD
          + OF TIME DURING WHICH THIS')
           WRITE  (6, 2080)
      2080 FORMAT (X, '               RAMP - INPUT RATE IS IN EFFECT.  NOTE:
          + THE RAMP - INPUT RATE MAY')
           WRITE  (6, 2090)
      2090 FORMAT (X, '               NOT EXCEED PLUS OR MINUS 0.1 $/SEC.
          +  TOTAL ACCUMULATED REACTIVITY')
           WRITE  (6, 2100)
      2100 FORMAT (X, '               MAY NOT EXCEED 0.9$. IF YOU SELECT
          + VALUES OUTSIDE THESE PARAMETERS,')
           WRITE  (6, 2110)
      2110 FORMAT (X, '               YOUR CHOICE(S) WILL BE REJECTED, AND YOU
          + WILL HAVE TO TRY AGAIN.')

           WRITE  (6, 2120)
      2120 FORMAT ('0', '               EXAMPLE:  SUPPOSE YOU SELECT A RAMP -
          + INPUT RATE OF +10 CENTS/SEC.')
           WRITE  (6, 2130)
      2130 FORMAT (X, '               THIS IS WITHIN PERMISSIBLE LIMITS.  THEN
          + YOU SELECT A TIME PERIOD')
           WRITE  (6, 2140)
      2140 FORMAT (X, '               OF 11 SECONDS.  TEN CENTS OF REACTIVITY
          + PER SECOND FOR 11 SECONDS')
           WRITE  (6, 2150)
      2150 FORMAT (X, '               WOULD RESULT IN A TOTAL ACCUMULATED
          + REACTIVITY OF 110 CENTS.  THIS')
           WRITE  (6, 2160)
      2160 FORMAT (X, '               REPRESENTS A PROMPT SUPERCRITICALITY
          + CONDITION, IT VIOLATES THE')
           WRITE  (6, 2170)
      2170 FORMAT (X, '               CONDITION THAT TOTAL ACCUMULATED
          + REACTIVITY MAY NOT EXCEED PLUS OR')
           WRITE  (6, 2180)
      2180 FORMAT (X, '               MINUS 90 CENTS, AND IT WILL BE REJECTED.
          +  YOU WILL THEN HAVE TO TRY')
           WRITE  (6, 2190)
      2190 FORMAT (X, '               AGAIN.  WHEN YOU PICK YOUR OWN VALUES,
          + MULTIPLY THE RAMP - INPUT')
           WRITE  (6, 2200)
      2200 FORMAT (X, '               RATE (IN CENTS) BY THE TIME
          + (IN SECONDS).  THE PRODUCT MAY NOT ')
           WRITE  (6, 2210)
```

```
 2210 FORMAT (X, '              EXCEED PLUS OR MINUS 90.')

 2220 WRITE  (6, 2230)
 2230 FORMAT ('0', '               WRITE IN A RAMP - INPUT RATE
     + (IN CENTS/SEC), USING FORMAT F6.2:')
      WRITE  (6, 2240)
 2240 FORMAT (X, '              EX:  A RAMP - INPUT RATE OF -10 CENTS/SEC
     + WOULD BE WRITTEN AS -10.00')
      WRITE  (6, 2250)
 2250 FORMAT (X, '              A RAMP - INPUT RATE OF   5 CENTS/SEC
     + WOULD BE WRITTEN AS 005.00')

      READ    2260, RIR
 2260 FORMAT (F5.1)

      IF ((RIR .GE. -10.0)  .AND.  (RIR .LE. 10.0))  GO TO 2290

      WRITE  (6, 2270)
 2270 FORMAT (X, '              THE VALUE SELECTED MAY NOT EXCED PLUS
     + OR MINUS 10.  YOU WILL HAVE')
      WRITE  (6, 2280)
 2280 FORMAT (X, '              TO TRY AGAIN.')

      GO TO 2220

 2290 WRITE  (6, 2300)
 2300 FORMAT (X, '              WRITE IN THE TIME PERIOD FOR WHICH THE
     + RAMP - INPUT IS IN EFFECT,')
      WRITE  (6, 2310)
 2310 FORMAT (X, '              USING FORMAT F5.1')
      READ    2320, TTOT
 2320 FORMAT (F5.1)

          PROD    =        RIR  * TTOT
          RIR     =        0.01 * RIR * A

      IF ((PROD .GE. -90.0)  .AND.  (PROD .LE.  90.0))  GO TO 2450

      WRITE  (6, 2330)
 2330 FORMAT (X, '              YOU HAVE SELECTED VALUE(S) THAT ARE TOO
     + HIGH.  REMEMBER, THE PRODUCT')
      WRITE  (6, 2340)
 2340 FORMAT (X, '              OF RAMP - INPUT (IN CENTS) TIMES TOTAL
     + TIME PERIOD (IN SECONDS)')
      WRITE  (6, 2350)
 2350 FORMAT (X, '              MUST NOT EXCEED PLUS OR MINUS 90.0.  YOU
     + WILL HAVE TO TRY AGAIN.')

      GO TO 2220

 2360     RIR     =        0.01 * RIR * A
C
C         HERE THE USER SELECTS WHATEVER REACTIVITY STEP IS REQUIRED.
C
      WRITE  (6, 2370)
 2370 FORMAT (X, '              WHAT LEVEL (IN CENTS) REACTIVITY
```

```
      + PERTURBATION IS DESIRED?')
        WRITE (6, 2380)
 2380 FORMAT (X, '              USE FORMAT F5.1    EX: 010.0 = A STEP
      + INPUT OF +10 CENTS.')
        READ    2390, DRON
 2390 FORMAT (F4.1)

        IF (((DRON + RHON) .GT. -100.0)  .AND.  ((DRON + RHON) .LT.100.0))
      +      GO TO 2440

 2400  WRITE  (6, 2410)
 2410 FORMAT (X, '              YOU HAVE SELECTED A VALUE THAT WILL
      + RESULT IN A PROMPT SUPERCRITICAL')
        WRITE  (6, 2420)
 2420 FORMAT (X, '              CONDITION.  THIS IS NOT ALLOWED, AND YOU
      + WILL HAVE TO TRY AGAIN.')
        WRITE  (6, 2430)
 2430 FORMAT (' ')


        GO TO 2360

 2440      DRO     =          0.01 * DRON * A

C
C          HERE INITIAL POWER LEVEL IS SELECTED.
C
 2450 WRITE  (6, 2460)
 2460 FORMAT (X, '              WHAT INITIAL POWER LEVEL (IN MWT) IS
      + DESIRED?  USE FORMAT F6.1')
        WRITE  (6, 2470)
 2470 FORMAT (X, '              EXAMPLE:  IF INITIAL POWER IS 2000 MW,
      + TYPE IN 2000.0')
        READ    2480, N10
 2480 FORMAT (F6.1)

        IF ((N10 .GT. 0.0)  .AND.  (N10 .LE. 3000.0))  GO TO 2520

        WRITE  (6, 2490)
 2490 FORMAT (X, '              NEGATIVE POWER LEVELS OR POWER LEVELS
      + IN EXCESS OF 3000MW ARE NOT')
        WRITE  (6, 2500)
 2500 FORMAT (X, '              ALLOWED.  YOU WILL HAVE TO TRY AGAIN.')
        WRITE  (6, 2510)
 2510 FORMAT (' ')

        GO TO 2450
 2520 IF ((KF .EQ. 1)  .OR.  (RI  .EQ. 1)  .OR.
      +    (PJ .EQ. 1)  .OR.  (EIG .EQ. 1))  GO TO 2720


C
C          HERE THE USER DECIDES ON WHETHER TO SELECT A POWER STEP.  THIS
C     WILL MEAN NO REACTOR KINETICS, BUT A CONSTANT POWER AT SOME NEW
C     LEVEL.  THIS IS TO CHECK ON STEAM GENERATOR PERFORMANCE.
C
        WRITE  (6, 2530)
```

```
      2530 FORMAT (X, '                DO YOU WISH TO INSERT A FIXED POWER
     + CHANGE STEP?  THIS IS A STEP')
           WRITE  (6, 2540)
      2540 FORMAT (X, '                CHANGE IN REACTOR POWER IMPOSED BY YOU
     + AFTER THE INITIAL,')
           WRITE  (6, 2550)
      2550 FORMAT (X, '                STEADY-STATE CONDITIONS HAVE BEEN SET
     + UP.  IT REPLACES THE REACTOR')
           WRITE  (6, 2560)
      2560 FORMAT (X, '                KINETICS SUBROUTINE.  IF YOU CHOOSE THIS
     + OPTION, THERE WILL BE NO')
           WRITE  (6, 2570)
      2570 FORMAT (X, '                REACTOR KINETICS.  TOTAL POWER LEVEL
     + WILL BE PERMANENTLY FIXED.')
           WRITE  (6, 2580)
      2580 FORMAT (X, '                NATURALLY, TOTAL REACTOR POWER WILL NOT
     + BE ALLOWED TO BE GREATER')
           WRITE  (6, 2590)
      2590 FORMAT (X, '                THAN 3000MW OR LESS THAN ZERO.  IF YOU
     + WISH THIS OPTION, TYPE IN 1;')
           WRITE  (6, 2600)
      2600 FORMAT (X, '                IF NOT, TYPE IN 2')
           READ 2610, NK
      2610 FORMAT (I1)
           IF ((NK .EQ. 1)  .OR.  (NK .EQ. 2)) GO TO 2650
           WRITE  (6, 2620)
      2620 FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
           WRITE  (6, 2630)
      2630 FORMAT (X, '                TO TRY AGAIN.')
           WRITE  (6, 2640)
      2640 FORMAT (' ')
           GO TO 2520
      2650 IF (NK .EQ. 2)  GO TO 2710
           WRITE  (6, 2660)
      2660 FORMAT (X, '                WHAT POWER STEP CHANGE IS DESIRED?  USE
     + FORMAT F7.1')
           WRITE  (6, 2670)
      2670 FORMAT (X, '                EXAMPLE:  IF POWER STEP IS -100MW, TYPE
     + IN -0100.0')
           READ 2680, DN2
      2680 FORMAT (F7.1)
           IF (((DN2 + N10) .GE. 0.0)  .OR.  ((DN2 + N10) .LE. 3000.0))
     +        GO TO 2710
           WRITE  (6, 2690)
      2690 FORMAT (X, '                YOU HAVE SELECTED A POWER STEP WHICH
     + PLACES TOTAL POWER OUTSIDE')
           WRITE  (6, 2700)
```

```
      2700 FORMAT (X, '                 PERMISSIBLE PARAMETERS.  YOU WILL HAVE
          + TO TRY AGAIN.')

           GO TO 2520

      2710 DN1  =  DN2 * 1.0E06
           IF (NK .EQ. 1)   CS   =  2
           IF (NK .EQ. 1)   EIG  =  2
           IF (NK .EQ. 1)   PJ   =  2
           IF (NK .EQ. 1)   RI   =  2


      2720       VOSS    =         N10 / 3000.0
                 DVO     =         0.0
           IF (NK .EQ. 1)  GO TO 3050


C
C         HERE THE USER DECIDES ON WHETHER TO INDUCE A PERTURBATION BY
C    VARYING THE STEAM GENERATOR THROTTLE VALVE INSTEAD OF VARYING THE
C    REACTOR.  IF THIS IS SELECTED, ANY PREVIOUSLY SELECTED REACTIVITY
C    CHANGE WILL BE SET EQUAL TO ZERO.  MAXIMUM STEAM VALVE CHANGE IS
C    10%.
C
      2800 WRITE  (6, 2810) VOSS
      2810 FORMAT (X, '                 STEAM GENERATOR IS NOW', 3X, F5.3, ':',
          + 3X, 'THIS REPRESENTS PRESENT VALVE')
           WRITE  (6, 2820)
      2820 FORMAT (X, '                 POSITION.  0.000 MEANS CLOSED, WHILE
          + 1.000 MEANS OPEN.')
      2830 WRITE  (6, 2840)
      2840 FORMAT ('0', '                 DO YOU WISH TO INDUCE A PERTURBATION
          + BY CHANGING THE VALVE')
           WRITE  (6, 2850)
      2850 FORMAT (X, '                 POSITION?  IF "YES", TYPE 1:  IF "NO",
          + TYPE 2')

           WRITE  (6, 2860)
      2860 FORMAT ('0', '                 CAUTION:  IF A VALVE CHANGE IS THE
          + INITIAL PERTURBATION, THE STEAM')
           WRITE  (6, 2870)
      2870 FORMAT (X, '                 GENERATOR MODEL WILL STILL HAVE TO BE
          + USED.  AND ANY REACTIVITY')
           WRITE  (6, 2880)
      2880 FORMAT (X, '                 PERTURBATION WILL HAVE TO EQUAL
          + ZERO.  HOWEVER, DO NOT WORRY IF')
           WRITE  (6, 2890)
      2890 FORMAT (X, '                 YOU HAVE ALREADY CHOSEN A REACTIVITY
          + STEP STEAM GENERATOR MODEL.')
           WRITE  (6, 2900)
      2900 FORMAT (X, '                 OR A REACTIVITY STEP.  IF YOU
          + UTILIZE A VALVE PERTURBATION, THIS')
           WRITE  (6, 2910)
      2910 FORMAT (X, '                 PROGRAM WILL CHOOSE SET ANY REACTIVITY
          + INSERTION EQUAL TO ZERO.')
           READ     2930, ZZ
      2930 FORMAT (I1)
```

```
      IF ((ZZ .EQ. 1)  .OR.  (ZZ .EQ. 2))  GO TO 2960

      WRITE  (6, 2940)
 2940 FORMAT (X, '              YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 2950)
 2950 FORMAT (X, '              TO TRY AGAIN.')

      GO TO 2830

 2960 IF (ZZ .EQ. 1)    P = 1
      IF (ZZ .EQ. 1) DRO = 0.0
      IF (ZZ .EQ. 2)  GO TO 3050

      WRITE  (6, 2970)
 2970 FORMAT (X, '              IF "YES", TYPE IN THE PERCENT CHANGE,
     + USING FORMAT F5.1')
      WRITE  (6, 2980)
 2980 FORMAT (X, '              EX:   IF INITIAL VALVE POSITION IS 0.600
     + AND YOU WISH TO DECREASE')
      WRITE  (6, 2990)
 2990 FORMAT (X, '              IT 3%, TYPE IN -3.0   A PLUS 4% INCREASE
     +, ON THE OTHER HAND, WOULD')
      WRITE  (6, 3000)
 3000 FORMAT (X, '              BE TYPED IN AS 04.0:    CAUTION:  NO
     + CHANGE SHOULD EXCEED')
      WRITE  (6, 3010)
 3010 FORMAT (X, '              PLUS OR MINUS 10%')
      READ    3020, DVO
 3020 FORMAT (F5.1)

      IF ((DVO .GE. -10.0)  .AND.  (DVO .LE. 10.0))  GO TO 3050

      WRITE  (6, 3030)
 3030 FORMAT (X, '              YOU HAVE SELECTED A VALUE GREATER THAN
     + PLUS OR MINUS 30%.')
      WRITE  (6, 3040)
 3040 FORMAT (X, '              THIS IS NOT ALLOWED, AND YOU WILL HAVE
     + TO TRY AGAIN.')

      GO TO 2830

 3050 DVO          =     DVO  / 100.0
      VO           =     VOSS + (DVO * VOSS)


C
C          HERE THE INITIALIZATION OF SEVERAL VARIABLES TAKES PLACE.
C     THIS INCLUDES THE INTEGER VARIABLES USED IN THE ARRAYS, CPPAV AND
C     MDOTP, AND REACTOR INLET AND OUTLET TEMPERATURES, TIME (SET EQUAL
C     TO ZERO), MODERATOR AND FUEL TEMPERATURES, AND REACTOR HEAT TRA-
C     FER COEFFICIENT.
C
 3060        Y      =        1.0
             F      =        1
             K      =        2
```

```
                  Q       =          2
                  CPPAV   =     5916.241929 + 16.32498553 *EXP(6.488055E-04*N10)
                  MDOTP   =     3.0E09 / ((56.9 * 5.0 / 9.0) * CPPAV)
                  TI1     =     (565.0*5.0/9.0) - (28.45*5.0/9.0) * N10 / 3000.0
                  TO1     =     (565.0*5.0/9.0) + (28.45*5.0/9.0) * N10 / 3000.0
                DTF(1)    =        0.0
                DTI(1)    =        0.0
                DTM(1)    =        0.0
                DTO(1)    =        0.0
               DTSG(1)    =        0.0
                DTS(1)    =        0.0
                 PT(1)    =        N10 * 1.0E06
                 STMG(1)=          N10 * 1.0E06
                 TT(1)    =        0.0
                  ABN     =     0.2
                  FRAC    =     0.5  *  5932.566914 / CPPAV
                  LNT     =     1
                  TM1     =     FRAC * (TO1 - TI1) + TI1
                  TM      =        TM1
                  TW      =     0.0
                  TOTHO   =     0.0
                  CPM     =        4992.4097749 + 2.49340775E-04 * EXP
     +                                       (0.04825458 * TM1)
                  ROP     =       881.6309 - 2.86514041 * EXP(0.0133034152 *TM1)
                  VMOD    =       MDOTP / (5.26 * ROP)
                  KP      =       0.712768335 - 3.025E-03 * EXP(0.01321074*TM1)
                  MUP     =      -3.53438229E-07 * TM1 + 1.9978561E-04
                  NUP     =       MUP / ROP
                  REP     =       0.012 * VMOD / NUP
                  PRP     =       CPM * NUP * ROP / KP
                  NU      =       0.025 * (REP ** 0.8) * (PRP ** 0.6)
                  HP      =       NU * KP / 0.012
                  TC      =      (N10 * 1.0E06) / (HP * 5945.0) + TM1
                  TF1     =       25.6933661 * (TC - TM1) + TM1
                  STMGEN =        N10 * 1.0E06

                  IF (KF .EQ. 1)  AA     = 0.0
                  IF (KF .EQ. 1)  ALPHF = 0.0
                  IF (KF .EQ. 1)  ALPHM = 0.0
                  IF (KF .EQ. 1)  CS     =  2
                  IF (KF .EQ. 1)  RHO    = 0.0
                  IF (KF .EQ. 1)  RHO    = 0.0


C
C          HERE THE USER DECIDES WHETHER THE OUTPUT IS TO BE IN GRAPHICS
C    OR IN A TABLE.
C
 3070 WRITE  (6, 3080)
 3080 FORMAT ('0', '              DO YOU WANT OUTPUT IN GRAPHICS, OR IN A
     + TABLE?  TYPE 1 FOR GRAPHICS,')
      WRITE  (6, 3090)
 3090 FORMAT (X, '              2 FOR A TABLE.')
      READ     3100, NN
 3100 FORMAT (I1)

      IF ((NN .EQ. 1)  .OR.  (NN .EQ. 2))  GO TO 3130
```

```

      WRITE  (6, 3110)
 3110 FORMAT (X, '            YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 3120)
 3120 FORMAT (X, '            TO TRY AGAIN.')

      GO TO 3070

 3130 IF (NN .EQ. 1)  AB  = 2
      IF (NN .EQ. 1)  ABN = 0.2
      IF (NN .EQ. 1)  GO TO 3380

C
C        HERE THE USER IS ABLE TO CAUSE TABLE OUTPUT TO BE ABBREVIATED.
C
 3140 WRITE  (6, 3150)
 3150 FORMAT (X, '            SINCE YOU HAVE CHOSEN A TABLE, BE AWARE
     + THAT THE PRINTER BY DEFAULT')
      WRITE  (6, 3160)
 3160 FORMAT (X, '            WILL PRINT OUTPUT EVERY 0.2 SECONDS OF
     + SYSTEM TIME.  OVER A FULL 5')
      WRITE  (6, 3170)
 3170 FORMAT (X, '            - MINUTE RUN, THIS WOULD RESULT IN 24
     + PAGES OF OUTPUT.  YOU CAN,')
      WRITE  (6, 3180)
 3180 FORMAT (X, '            HOWEVER, ABBREVIATE THIS BY CAUSING THE
     + SYSTEM TO PRINT ONLY AT')
      WRITE  (6, 3190)
 3190 FORMAT (X, '            INTERVALS OF 1.0 SECONDS, 2.0 SECS., 4.0
     + SECS., 5.0 SECS.,')
      WRITE  (6, 3200)
 3200 FORMAT (X, '            OR 10 SECS., INSTEAD OF 0.2 SECONDS.')
      WRITE  (6, 3210)
 3210 FORMAT ('0', '            DO YOU WISH TO ABBREVIATE THE OUTPUT?
     +  IF SO, TYPE IN 1;  IF NOT, 2')
      WRITE  (6, 3220)
 3220 FORMAT (X, '            REMEMBER, IF YOU WISH TO TRUNCATE OUTPUT
     + ONLY, AND NOT ABBREVIATE,')
      WRITE  (6, 3230)
 3230 FORMAT (X, '            YOU WILL BE ABLE TO DO SO LATER.')
      READ    3240, AB
 3240 FORMAT (I1)

      IF ((AB .EQ. 1)  .OR.  (AB .EQ. 2)) GO TO 3280

      WRITE  (6, 3250)
 3250 FORMAT (X, '            YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 3260)
 3260 FORMAT (X, '            TO TRY AGAIN.')
      WRITE  (6, 3270)

 3270 FORMAT (' ')

      GO TO 3140
 3280 IF (AB .EQ. 2)  GO TO 3380
```

```
      3290 WRITE  (6, 3300)
      3300 FORMAT (X, '                YOU MAY HAVE OUTPUT PRINTED AT
          + INTERVALS OF 1, OR 2, OR 4,')
           WRITE  (6, 3310)
      3310 FORMAT (X, '                OR 5, OR 10 SECONDS  (SYSTEM TIME).
          +  CHOOSE ONE OF THESE, USING FORMAT')
           WRITE  (6, 3320)
      3320 FORMAT (X, '                F4.1 -- FOR INSTANCE, A DESIRED INTERVAL
          + OF 4 SECONDS WOULD BE')
           WRITE  (6, 3330)
      3330 FORMAT (X, '                WRITTEN AS 04.0')
           READ    3340, ABN
      3340 FORMAT (F4.1)

           IF ((ABN .EQ. 0.2) .OR. (ABN .EQ. 1.0) .OR. (ABN .EQ. 2.0) .OR.
          +    (ABN .EQ. 4.0) .OR. (ABN .EQ. 5.0) .OR. (ABN .EQ. 10.0))
          + GO TO 3380

           WRITE  (6, 3350)
      3350 FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
          + BE UTILIZED.  YOU WILL HAVE')
           WRITE  (6, 3360)
      3360 FORMAT (X, '                TO TRY AGAIN.')
           WRITE  (6, 3370)
      3370 FORMAT (' ')

           GO TO 3290

C
C          HERE THE USER IS ABLE TO LENGTHEN THE TIME OF THE RUN FROM 1
C     MINUTE UP TO 5 MINUTES.
C
      3380 WRITE  (6, 3390)
      3390 FORMAT (X, '                BY DEFAULT, THIS PROGRAM WILL RUN FOR 1
          + MINUTE OF "SYSTEM TIME".')
           WRITE  (6, 3400)
      3400 FORMAT (X, '                YOU CAN, HOWEVER, LENGTHEN THIS TO AS
          + MUCH AS 5 MINUTES.')
      3410 WRITE  (6, 3420)
      3420 FORMAT ('0', '                DO YOU WISH TO LENGTHEN THE TIME OF
          + THE COMPUTER RUN?  IF SO, TYPE')
           WRITE  (6, 3430)
      3430 FORMAT (X, '                IN 1;, IF NOT, TYPE IN 2')
           READ    3440, LN
      3440 FORMAT (I1)

           IF ((LN .EQ. 1)  .OR.  (LN .EQ. 2))  GO TO 3470

           WRITE  (6, 3450)
      3450 FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
          + BE UTILIZED.  YOU WILL HAVE')
           WRITE  (6, 3460)
      3460 FORMAT (X, '                TO TRY AGAIN.')

           GO TO 3410
      3470 IF (LN .EQ. 2)  GO TO 3560
```

```
3480 WRITE  (6, 3490)
3490 FORMAT (X, '                    TYPE IN THE AMOUNT OF TIME, IN MINUTES,
    + THAT YOU WANT THE SYSTEM TO')
     WRITE  (6, 3500)
3500 FORMAT (X, '                    RUN.  YOU MAY PICK ANY INTEGER (NOT REAL
    + NUMBER) FROM 1 TO 5.')
     WRITE  (6, 3510)
3510 FORMAT ('0', '                    TYPE IN AN INTEGER FROM 1 TO 5.  THIS
    + WILL BE THE TOTAL SYSTEM TIME,')
     WRITE  (6, 3520)
3520 FORMAT (X, '                    IN MINUTES.')
     READ    3530, LNT
3530 FORMAT (I1)

     IF ((LNT .EQ. 1) .OR. (LNT .EQ. 2) .OR. (LNT .EQ. 3) .OR.
    +    (LNT .EQ. 4) .OR. (LNT .EQ. 5)) GO TO 3560

     WRITE  (6, 3540)
3540 FORMAT (X, '                    YOU HAVE TYPED IN A NUMBER THAT CANNOT
    + BE UTILIZED.  YOU WILL HAVE')
     WRITE  (6, 3550)
3550 FORMAT (X, '                    TO TRY AGAIN.')

     GO TO 3480

3560   LND  =  300 * LNT

3570 WRITE  (6, 3580)
3580 FORMAT (X, '                    SINCE YOU HAVE CHOSEN A TABLE, DO YOU
    + WANT A HARD COPY, OR WILL THIS')
     WRITE  (6, 3590)
3590 FORMAT (X, '                    TERMINAL SUFFICE?  TYPE 8 FOR A HARD
    + COPY, 6 FOR THE TERMINAL.')
     READ    3600, YY
3600 FORMAT (I1)

     IF ((YY .EQ. 6) .OR. (YY .EQ. 8))  GO TO 3640

     WRITE  (6, 3610)
3610 FORMAT (X, '                    YOU HAVE TYPED IN A NUMBER THAT CANNOT
    + BE UTILIZED.  YOU MUST TYPE')
     WRITE  (6, 3620)
3620 FORMAT (X, '                    IN 8 FOR A HARD COPY, 6 FOR A TERMINAL
    + DISPLAY.')
     WRITE  (6, 3630)
3630 FORMAT (' ')

     GO TO 3570
3640 IF (IS .NE. 1)  GO TO 3660


C
C          HERE, THE OPTIONS THE USER HAS SELECTED ARE DISPLAYED.
C
     WRITE  (YY, 3650)
3650 FORMAT (X, '                    THE FUEL TO BE UTILIZED IS U-233.')
```

```
          GO TO 3700

 3660 IF (IS .NE. 2)  GO TO 3680
          WRITE  (YY, 3670)
 3670 FORMAT (X, '                THE FUEL TO BE UTILIZED IS U-235.')

          GO TO 3700

 3680 WRITE  (YY, 3690)
 3690 FORMAT (X, '                THE FUEL TO BE UTILIZED IS PU-239.')

 3700 IF (KF .EQ. 2)  GO TO 3740

          WRITE  (YY, 3710)
 3710 FORMAT ('0', '                FREE KINETICS WILL BE USED IN THIS RUN.
     + THERE')
          WRITE  (YY, 3720)
 3720 FORMAT (X, '                WILL BE NO FEEDBACK, NO CONTROL SYSTEM,
     + AND')
          WRITE  (YY, 3730)
 3730 FORMAT (X, '                REACTIVITY COEFFICIENTS WILL BE EQUAL TO
     + ZERO.')

          GO TO 3860

 3740 WRITE  (YY, 3750)
 3750 FORMAT ('0', '                FEEDBACK WITH REACTIVITY COEFFICIENTS
     + WILL BE')
          WRITE  (YY, 3760)
 3760 FORMAT (X, '                USED.')

          WRITE  (YY, 3770) RHON
 3770 FORMAT ('0', '                PRE-PERTURBATION REACTIVITY IS', 18X,
     + F4.1, 8X, 'CENTS')

          WRITE  (YY, 3780) ALPHF
 3780 FORMAT (X, '                DOPPLER COEFFICIENT OF REACTIVITY IS',
     + 12X, E10.3, 2X, '(DK/K)/C')
          WRITE  (YY, 3790) ALPHM
 3790 FORMAT (X, '                MODERATOR TEMP. COEFFICIENT OF
     + REACTIVITY IS', 4X, E10.3, 2X, '(DK/K)/C')

          IF (CS .EQ. 1)  GO TO 3810

          WRITE  (YY, 3800)
 3800 FORMAT ('0', '                A CONTROL SYSTEM WILL NOT BE USED.')

          GO TO 3860


 3810 WRITE  (YY, 3820)
 3820 FORMAT ('0', '                A CONTROL SYSTEM WILL BE USED.')
          WRITE  (YY, 3830) AA
 3830 FORMAT (X, '                GAIN             EQUALS', 27X, E8.1)
          WRITE  (YY, 3840) TAUC
 3840 FORMAT (X, '                TIME CONSTANT EQUALS', 28X, F3.1)
```

```
      WRITE  (YY, 3850) TAU
3850  FORMAT (X, '                    PARAMETER TAU EQUALS', 28X, F3.1)

3860  IF (EIG .EQ. 1)  GO TO 3890

      WRITE  (YY, 3870)
3870  FORMAT (X, '                    THE "MATRIX DECOUPLING" ALGORITHM
     + WILL NOT')
      WRITE  (YY, 3880)
3880  FORMAT (X, '                    BE USED.')

      GO TO 3930

3890  WRITE  (YY, 3900)
3900  FORMAT ('0', '                    THE "MATRIX DECOUPLING" ALGORITHM
     + WILL BE')
      WRITE  (YY, 3910)
3910  FORMAT (X, '                    USED.')
      WRITE  (YY, 3920)
3920  FORMAT (X, '                    THE PROMPT-JUMP APPROXIMATION WILL NOT
     + BE USED.')

3930  IF (PJ .EQ.1)  GO TO 3950

      WRITE  (YY, 3940)
3940  FORMAT ('0', '                    PROMPT-JUMP APROXIMATION WILL NOT BE
     + USED.')

      GO TO 3980

3950  WRITE  (YY, 3960)
3960  FORMAT ('0', '                    PROMPT-JUMP APPROXIMATION WILL BE USED')
      WRITE  (YY, 3970)
3970  FORMAT (X, '                    MATRIX DECOUPLING WILL NOT.')

      GO TO 4050

3980  IF (RI .EQ. 1)  GO TO 4000

      WRITE  (YY, 3990)
3990  FORMAT ('0', '                    RAMP-INPUT MODEL WILL NOT BE USED.')

      GO TO 4050

4000  WRITE  (YY, 4010)
4010  FORMAT ('0', '                    RAMP-INPUT MODEL WILL BE USED.')
      WRITE  (YY, 4020) RIR
4020  FORMAT (X, '                    RAMP-INPUT RATE IS', 28X, F5.1, 7X,
     + 'CENTS/S')
      WRITE  (YY, 4030)
4030  FORMAT (X, '                    DURATION OF RAMP-INPUT INSERTION IS',
     + 11X, F5.1, 7X, 'SECONDS')
      WRITE  (YY, 4040) PROD
4040  FORMAT (X, '                    TOTAL ACCUMULATED REACTIVITY IS', 17X,
     + F5.1, 7X, 'CENTS')

4050  WRITE  (YY, 4060) DRON
```

```
4060 FORMAT ('0', '                REACTIVITY PERTURBATION IS', 22X, F4.1,
    + 8X, 'CENTS')

     WRITE  (YY, 4070) N10
4070 FORMAT ('0', '                INITIAL POWER LEVEL IS', 26X, F6.1, 6X,
    + 'MW')

     WRITE  (YY, 4140) VO
4140 FORMAT ('0', '                STEAM GENERATOR VALVE POSITION IS',
    + 15X, F5.3, 7X, 'OPEN')

     IF (ZZ .EQ. 1)  GO TO 4160

     WRITE  (YY, 4150)
4150 FORMAT ('0', '               THERE IS NO VALVE INDUCED PERTURBATION')

     GO TO 4190

4160 WRITE  (YY, 4170)
4170 FORMAT ('0', '                SYSTEM PERTURBATION WILL BE A VALVE
    + CHANGE.')
     WRITE  (YY, 4180) DVO
4180 FORMAT (X, '              VALVE CHANGE IS', 34X, F4.1, 8X,
    + 'PERCENT.')

4190 IF (AB .EQ. 2)  GO TO 4220

     WRITE  (YY, 4200)
4200 FORMAT ('0', '                OUTPUT WILL BE ABBREVIATED.  PRINTING
    + WILL OCCUR EVERY')
     WRITE  (YY, 4210)  ABN
4210 FORMAT (X,                  F4.1, 3X, 'SECONDS, SYSTEM TIME.')

     GO TO 4250

4220 WRITE  (YY, 4230)
4230 FORMAT ('0', '                OUTPUT   IS NOT ABBREVIATED.  PRINTING
    + WILL OCCUR EVERY')
     WRITE  (YY, 4240)
4240 FORMAT (X, '              0.2 SECONDS, SYSTEM TIME.')
4250 WRITE  (YY, 4260)
4260 FORMAT ('0', '                TOTAL SYSTEM TIME OF THIS RUN IS',
    + 24X, I1, 4X, 'MINUTES')

     IF (NN .EQ. 1)  GO TO 4280

     WRITE  (YY, 4270)
4270 FORMAT ('0', '                OUTPUT WILL BE IN A TABLE')

     GO TO 4300

4280 WRITE  (YY, 4290)
4290 FORMAT ('0', '                OUTPUT WILL BE IN GRAPHICS.')

4300 IF (YY .EQ. 6)  GO TO 4320

     WRITE  (YY, 4310)
```

```
      4310 FORMAT ('0', '          HARD COPY WILL BE PRODUCED.')


           GO TO 4360

      4320 WRITE  (YY, 4330)
      4330 FORMAT ('0', '          OUTPUT OF GRAPHICS WILL BE ON TERMINAL
          + IN THE')
           WRITE  (YY, 4340)
      4340 FORMAT (X, '          IGL LAB.  TO GET A HARD COPY, YOU MUST
          + USE THE')
           WRITE  (YY, 4350)
      4350 FORMAT (X, '          4051 TEKTRONICS TERMINAL')
      4360 CONTINUE


      4370 WRITE  (YY, 4380)
      4380 FORMAT (' ')

           WRITE  (YY, 4390)
      4390 FORMAT (' ')

           IF (NN .EQ. 1)  GO TO 4460

           WRITE  (YY, 4400)
      4400 FORMAT (2X, 'TIME', 5X, 'FUEL', 6X, 'MODERATOR', 3X, 'RECTOR',
          +         4X, 'REACTOR', 5X, 'TURBINE', 3X, 'REACTOR')
           WRITE  (YY, 4410)
      4410 FORMAT (11X, 'TEMP', 6X, 'TEMP', 8X, 'INLET', 6X, 'OUTLET', 6X,
          +         'POWER', 5X, 'POWER')
           WRITE  (YY, 4420)
      4420 FORMAT (33X, 'TEMP', 7X, 'TEMP')
           WRITE  (YY, 4430)
      4430 FORMAT (X, '(SEC)', 4X, '(DEG. C)', 2X, '(DEG. C)', 4X,
          +         '(DEG. C)', 3X, '(DEG. C)', 4X, '(MW)', 6X, '(MW)')
           WRITE  (YY, 4440)
      4440 FORMAT (' ')
           WRITE  (YY, 4450)  TT(1), TF1, TM1, TI1, TO1, N10, N10
      4450 FORMAT (2X, F3.1, 5X, F8.3, 3X, F8.4, 4X, F8.4, 3X, F8.4,
          +         4X, F8.3, 2X, F8.3)

C
C          HERE REACTOR POWER IS CONVERTED FROM MW TO W, WHICH IS THE
C     FORMAT THAT WILL BE DISPLAYED IN THE OUTPUT.
C
 4460      N10     =       N10 * 1.0E06
           RP(K)   =       DN1 + N10

           DO 4540 I = 1,LND

C
C          HERE, OUTPUT IS DISPLAYED.  FIRST, REACTOR KINETICS IS COMPU-
C     TED FOR 0.005 SECONDS, IN THE GALBA SUBROUTINE.  THEN THERMAL HY-
C     DRAULICS IS COMPUTED IN OTHO.  CONTROL THEN PAASES BACK TO GALBA,
C     AND THIS PROCESS KEEPS ON GOING UNTIL 0.2 SECONDS HAS ELAPSED, AT
C     WHICH TIME CONTROL PASSES TO DMTN.  AFTER DMTN COMPUTES ST
C     GENERATOR OUTPUT, ONE ITERATION OF "DO" LOOP 4540 IS COMPLETE.
```

```
C    THIS "DO" LOOP KEEPS ON ITERATING UNTIL (LND MINUTES TIMES 300)
C    ITERATIONS ARE COMPLETE.  (LND IS THE AMOUNT OF TIME IN MINUTES
C    THAT THE PROGRAM WILL RUN.)  EVERY 0.2 SECONDS, THOSE QUANTITIES
C    WHICH CAN APPEAR IN THE OUTPUT ARE COMPUTED.  IF GRAPHICS IS
C    CALLED FOR, THE GRAPHICS SUBROUTINE VESPASIAN IS CALLED ON THE
C    VERY LAST ITERATION OF "DO" LOOP 4540.
C
 4470 IF (Y .EQ. 1.0) H = 0.000001
      IF (Y .EQ. 1.0) GO TO 4480
      IF (Y .NE. 1.0) H = 0.001

 4480 CALL GALBA

      CONTINUE
          Y       =            Y + 1.0
 4490 CALL OTHO
      TOTHO      =      TOTHO + 0.005
      IF (NK .EQ. 1)  TOTHO = 0.2
      IF (TOTHO .LE. 0.197)     GO TO 4480
      TOTHO      =        0.0
      IF (NN .EQ. 2)            GO TO 4510
      IF (I   .LE. (LND-1))     GO TO 4510
 4500 CALL VESPASIAN
 4510       TT(K)   =       TT(K-1) + 0.2
      CALL DMTN

            TF        =      TF1 + DTF(K)
            TM        =      TM1 + DTM(K)
      IF (K .LE. 55)  TI   =  TI1
      IF (K .GT. 55)  TI   =  TI1 + DTI(K-29)
            TO        =      TO1 + DTO(K)
            TW        =      TW  + 0.2
            N110      =     (N10 + DN1) / 1.0E06
            STMGN     =      STMGEN     / 1.0E06
            PT(K)     =      N10 + DN1
            STMG(K) =        STMGEN

      IF (ABN .GT. (TW + 0.01))  GO TO 4530
      IF   (NN .EQ. 1)   GO TO 4530

      WRITE  (YY, 4520)  TT(K), TF, TM, TI, TO, STMGN, N110
 4520 FORMAT (X, F5.2, 4X, F8.3, 3X, F8.4, 4X, F8.4, 3X, F8.4,
     +         4X, F8.3, 2X, F8.3)
 4530 CONTINUE
      IF (ABN .LE. (TW + 0.01))  TW  =  0.0
           K        =            K + 1
           Q        =            Q + 1
 4540 CONTINUE
      STOP
      END
```

```
C         POWER KINETICS EQUATIONS
C
      SUBROUTINE GALBA
C
C         THIS IS THE SUBROUTINE GALBA.  IT DEPICTS REACTOR KINETICS.
C     AS INPUT, IT RECEIVES THE FOLLOWING PARAMETERS FROM THE MAIN
C     PROGRAM NERO:
C              THE ISOTOPE TO BE USED.
C              WHETHER MATRX DECOUPLING IS TO BE USED.
C              WHETHER A CONTROL SYSTEM IS TO BE USED, AND IF SO, ITS
C              PARAMETERS.
C              WHETHER FEEDBACK IS TO  BE USED, AND IF SO, THE REAC-
C              TIVITY COEFFICCIENTS ASSOCIATED WITH IT.
C              INITIAL POWER LEVEL.
C              REACTIVITY STEP, IF ANY.
C              WHETHER THE PROMPT JUMP APPROXIMATION IS TO BE USED.
C              WHETHER THE RAMP-INPUT MODEL IS TO BE USED, AND IF SO,
C              THE MAGNITUDE OF THE RAMP AND HOW LONG IT LASTS.
C
C         GALBA THEN COMPUTES REACTOR POWER CHANGES (FROM INITIAL
C     LEVELS) AND THEN PASSES THIS INFORMATION ON TO THE THERMAL-
C     HYDRAULICS SUBROUTINE OTHO.  TIME STEPS IN GALBA ARE 0.001 SECONDS,
C     EXCEPT DURING THE FIRST 0.001 SECOND WHEN THEY ARE ARBITRARILY
C     SMALLER, AND FOR THE SLOW MODE OF THE MATRIX DECOUPLING ALGORITHM,
C     WHERE THE TIME STEPS ARE 200 TIMES THE SIZE OF THE FAST MODE TIME
C     STEP (ALWAYS 0.001 SECONDS, EXCEPT DURING THE FIRST 0.001 SECOND).
C     THE REASON THE TIME STEPS ARE ARBITRARILY SMALL AT FIRST IS THAT
C     THE EQUATIONS ARE EFFECTIVE FOR TIME STEPS IN WHICH POWER DOES
C     NOT VARY GREATLY.  SINCE POWER DOES VARY GREATLY DURING THE PROMPT
C     JUMP, SMALL TIME STEPS ARE USED SO THAT THE EQUATIONS CAN BE SOLVED
C     FOR A TIME STEP IN WHICH POWER DOES NOT VARY GREATLY.  GALBA WILL
C     EXECUTE FOR 0.005 SECONDS AT A TIME, THEN PASS CONTROL TO THE
C     THERMAL HYDRAULICS SUBROUTINE OTHO.  OTHO THEN COMPUTES
C     MODERATOR TEMPERATURE, REACTOR OUTLET TEMPERATURE AND FUEL TEMP-
C     ERATURE, AND IF THE TOTAL TIME IN THE CURRENT CYCLE IS LESS THAN
C     0.2 SECONDS, CONTROL IS PAASED BACK TO GALBA; OTHERWISE CONTROL IS
C     PASSED ON TO THE STEAM GENERATOR SUBROUTINE.
C
C         GALBA SOLVES THE POINT-KINETICS EQUATIONS FOR CHANGES IN REAC-
C     TOR POWER FROM STEADY-STATE CONDITIONS.  IN THE EQUATIONS USED
C     HERE, SIX GROUPS OF DELAYED-NEUTRON PRECURSORS ARE USED.  THE
C     POINT-KINETICS EQUATIONS ARE A SYSTEM OF TWO COUPLED DIFFERENTIAL
C     EQUATIONS IN WHICH ONE EQUATION SOLVES FOR THE DELAYED-NEUTRON
C     PRECURSORS, AND THE OTHER FOR REACTOR POWER.  IT IS POSSIBLE TO
C     SEPARATE, OR DECOUPLE, THE POINT-KINETICS EQUATIONS INTO TWO
C     INDEPENDENT EQUATIONS, ONE OF WHICH SOLVES FOR THE DELAYED-NEUTRON
C     PRECURSORS ALONE, THE OTHER FOR REACTOR POWER ALONE.  THIS IS
C     DONE BY MEANS OF A TRANSFORMATION MATRIX WHICH IS DEPENDENT UPON
C     THE EIGENVALUES AND EIGENVECTORS OF THE COEFFICIENT MATRIX OF
C     THE COUPLED POINT-KINETICS EQUATIONS.  SLOW TIME STEPS ARE USED
C     FOR THE DELAYED-NEUTRON PRECURSORS, FAST ONES FOR THE POWER
C     RESPONSE.  ARITHMETIC ERROR IS THUS AVOIDED IN SOLVING FOR THE
C     DELAYED-NEUTRONS, WHILE CPU TIME IS ALSO SAVED.  AN INVERSE
C     TRANSFORMATION MATRIX THEN CONVERTS THESE SOLUTIONS BACK INTO
C     THE ORIGINAL VARIABLES.
C
C         THE POINT-KINETICS EQUATIONS CAN ALSO BE SOLVED USING THE
```

```
C      PROMPT-JUMP APPROXIMATION AND THE RAMP-INPUT MODEL.  IN THE
C      PROMPT-JUMP APPROXIMATION, THE ASSUMPTION IS THAT THE PROMPT-
C      JUMP TAKES PLACE IN ZERO TIME RATHER THAN IN A VERY SMALL TIME,
C      AND THUS THE POWER LEVEL JUMPS INSTANTLY FROM THE INITIAL LEVEL
C      TO THE NEW LEVEL.  IN THE RAMP-INPUT MODEL, REACTIVITY IS INSERTED
C      OVER A PERIOD OF TIME INSTEAD OF INSTANTANEOUSLY.
C
        COMMON     AA, ADTI, ADTO, ALPHF, ALPHM, CPPAV, CS, DN1, DRO,
      +            DTF, DTI, DTM, DTO, DTS, DTSG, EIG, F, H, IS, K, KF,
      +            LND, MDOTP, NK, NN, N10, Q, PJ, PT, RHO, RI, RIR,
      +            RP, STMG, STMGEN, TAU, TAUC, TF1, TM, TT, TTOT, VO,
      +            VOSS, Y, ZZ
        REAL       AA, ALPHF, ALPHM, CPPAV, DN1, DRO, H, MDOTP, N10, RHO,
      +            RIR, STMGEN, TAU, TAUC, TF1, TM1, TTOT, VO, VOSS, Y
        INTEGER    CS, EIG, F, IS, J, K, KF, LND, NK, NN, PJ, Q, RI, ZZ
        REAL       ADTI(1510), ADTO(1510), DTF(1510), DTI(1510),DTM(1510),
      +            DTO (1510), DTSG(1510), DTS(1510), RP(1510), TT (1510),
      +            PT  (1510), STMG(1510)

C          IN ADDITION TO THE ONES USED IN THE COMMON BLOCK AND EXPLAINED
C      IN NERO, THE FOLLOWING VARIABLES ARE USED IN GALBA:
C                  BB       IS THE SUMMATION OF THE 6 GROUPS OF DELAYED
C                           PRECURSORS.
C                  DNI      IS THE CHANGE IN REACTOR POWER FROM STEADY-STATE.
C                  DRC      IS THE CHANGE IN REACTIVITY DUE TO THE CONTROL
C                           SYSTEM.
C                  DROIN    IS A REACTIVITY STEP.  IT IS ENTERED BY THE USER IN
C                           NERO.
C                  DRO1     IS REACTIVITY STEP USED IN PROMPT-JUMP APPROXIMA-
C                           TION, AFTER THE REACTIVITY STEP PASSED FROM NERO
C                           IS ADJUSTED FOR FEEDBACK DURING THE PROMPT JUMP.
C                  DRO2     IS THE DIFFERENCE BETWEEN DRO1 AND THE REACTIVITY
C                           STEP PASSED FROM NERO.  USED IN COMPUTING THE
C                           PROMPT-JUMP APPROXIMATION.
C                  DROT     IS THE DIFFERENCE BETWEEN DRO1 AND DRO2.  IF
C                           IT IS SMALL ENOUGH, THE PJ APPROXIMATION IS
C                           CONSIDERED TO BE COMPUTED.
C                  DTFGT    IS THE ESTIMATED CHANGE IN FUEL TEMPERATURE THAT
C                           OCCURS DURING ANY GIVEN INTERVAL OF 0.2 SECONDS.
C                           IT IS USED IN REACTIVITY CALCULATIONS.  ITS PURPOSE
C                           IS TO PROVIDE SOME FUEL TEMPERATURE CHANGE DATA
C                           DURING THE 0.2 SECOND CYCLE THAT GALBA RUNS BEFORE
C                           PASSING CONTROL ON TO OTHO.
C                  DTMGT    IS ESTIMATED CHANGE IN MODERATOR TEMPERATURE THAT
C                           OCCURS DURING ANY 0.2 SECOND CYCLE.  IT IS USED FOR
C                           THE SAME REASON THAT DTFGT IS. AFTER THE 0.2 SECOND
C                           INTERVAL, DTFGT AND DTMGT BOTH ARE SET EQUAL TO
C                           ZERO, AND NEW VALUES FOR THE FUEL AND MODERATOR
C                           TEMPERATURE CHANGES ARE COMPUTED IN OTHO.  WHAT
C                           THESE TWO VARIABLES DO IS PREVENT THE ABRUPT FUEL
C                           TEMPERATURE CHANGES (AND THEREBY,SUDDEN REACTIVITY
C                           AND POWER CHANGES) THAT WOULD OCCUR IF THE FUEL AND
C                           MODERATOR TEMPERATURES WERE ALLOWED TO CHANGE ONLY
C                           ONCE EVERY 0.2 SECONDS (THAT IS, EVERY TIME CONTROL
C                           IS PASSED TO OTHO).
C                  DTMT     IS THE SUM  OF ALL DTM(K).  USED IN THE INTEGRAL
C                           CONTROLLER OF THE CONTROL SYSTEM.
```

```
C              L       IS PROMPT NEUTRON LIFETIME.
C              LK      IS THE 1X1 MATRIX FORMED BY MULTIPLYING THE 1X6
C                      L MATRIX AND THE 6X1 K MATRIX IN THE MATRIX DE-
C                      COUPLING ALGORITHM.  IS USED TO COMPUTE ACTUAL
C                      REACTOR POWER FROM DECOUPLED VARIABLES.
C              PWRPJ   IS THE TOTAL REACTOR POWER, AS COMPUTED IN THE
C                      PROMPT-JUMP APPROXIMATION.
C              PWR1    IS THE CHANGE IN REACTOR POWER, AS COMPUTED DURING
C                      THE PREVIOUS ITERATION.
C              PWR2    IS THE DIFFERENCE BETWEEN CURRENT REACTOR POWER
C                      CHANGE AND PWR1.
C              SIG     IS THE 6-GROUP SUMMATION OF THE DELAYED NEUTRON
C                      DECAY CONSTANT TIMES THE CHANGE IN PRECURSOR
C                      DENSITY.  IS USED IN POINT-KINETICS EQUATIONS.
C              SIGPJ   IS ANALOGOUS TO SIG, ONLY IS USED IN THE PROMPT-
C                      JUMP APPROXIMATION.
C              T       IS THE TOTAL ELAPSED TIME IN THE CURRENT SERIES OF
C                      ITERATIONS.  WHEN IT EQUALS 0.1 SECONDS, GALBA
C                      CAUSES THE DECOUPLED ANALOGUE OF THE DELAYED-
C                      NEUTRON PRECURSORS TO BE COMPUTED.
C              TH      IS THE TOTAL TIME ELAPSED IN THE CURRENT SERIES
C                      OF ITERATIONS.  WHEN IT EQUALS 0.005, CONTROL IS
C                      PASSED TO THE SUBROUTINE OTHO.
C              X       IS A VARIABLE THAT WHOSE MAGNITUDE GOVERNS THE
C                      SIZE OF THE TIME STEP USED IN THE FIRST 0.2 SECONDS
C                      OF THE RUNNING OF GALBA.
C              XX      IS EQUAL TO -1.0.  IS USED IN THE COMPUTATION OF
C                      DTFGT AND DTMGT.
C              B(6)    IS THE THE DELAYED-NEUTRON FRACTON, OF EACH GROUP.
C              CIO(6)  IS THE STEADY-STATE DELAYED PRECURSOR DENSITY.
C              DCI(6)  IS THE CHANGE IN DELAYED PRECURSOR DENSITY FROM
C              LMB(6)  IS THE DECAY CONSTANT OF EACH OF THE 6 DELAYED-
C                      NEUTRON GROUPS.
C                      THE STEADY STATE.
C
C         THE FOLLOWING ARE VARIABLES USED IN MATRIX DECOUPLING:
C              A(7,7)  IS THE COEFFICIENT MATRIX OF THE COUPLED SYSTEM
C                      OF THE POINT-KINETICS EQUATIONS.  IS INPUT INTO
C                      THE EIGENVALUE-COMPUTING SUBROUTINE EIGEN.
C                      THE JORDAN CANONICAL  FORM OF THIS MATRIX IS
C                      A = MJQ, WHERE J IS THE DIAGONAL MATRIX CONSISTING
C                      OF THE EIGENVALUES OF A, Q IS THE FUNDAMENTAL
C                      MATRIX CONSISTING OF THE EIGENVECTORS OF A, AND
C                      M IS THE INVERSE OF Q.
C              AI(7,7)IS A 7X7 MATRIX CONSISTING OF THE IMAGINARY PORTION
C                      OF THE FUNDAMENTAL MATRIX Q OF THE COEFFICIENT
C                      MATRIX A.  FOR ANY COEFFICIENT MATRIX A THAT WILL
C                      BE ENCOUNTERED IN THIS PROGRAM, THERE ARE ONLY
C                      REAL EIGENVALUES AND EIGENVECTORS, AND ALL
C                      ELEMENTS OF AI(7,7) EQUAL ZERO.
C              AR(7,7)IS THE FUNDAMENTAL EIGENVECTOR MATRIX OF THE
C                      A MATRIX.  CORRESPONDS TO THE Q MATRIX MENTIONED
C                      EARLIER.  BOTH THE AI(7,7) AND AR(7,7) MATRICES
C                      ARE OUTPUT OF THE PORTLIBRARY SUBROUTINE EIGEN.
C                      EVEN THOUGH THE VALUES OF ALL OF THE ELEMENTS OF
C                      OF AI(7,7) AND ALL OF THE OTHER IMAGINARY MATRICES
C                      ARE ZERO, THE MATRICES MUST BE THUS DEFINED, AS
```

```
C                  THE SUCCESSFUL APPLICATION OF THE PORTLIBRARY
C                  SUBROUTINES EIGEN AND CLINQ DEMAND AT LEAST
C                  ZERO ENTRIES IN ALL OF THE POSITIONS OF THSES
C                  MATRICES.
C         BR(7,7)IS A 7X7 IDENTITY MATRIX USED AS INPUT IN THE
C                  PORTLIBRARY SUBROUTINE CLINQ TO HELP INVERT THE
C                  AR(7,7), OR Q, MATRIX INTO THE INVERSE FUNDAMENTAL
C                  EIGENVECTOR MATRIX M REFERRED TO EARLIER.
C         BI(7,7)IS THE IMAGINARY COMPANION TO BR(7,7).  ALL ITS
C                  ELEMENTS EQUAL ZERO.
C         B2       IS THE 1X1 B2 MATRIX CONSISTING OF THE COEFFICIENT
C                  OF THE FAST MODE SYSTEM OF DECOUPLED DIFFERENTIAL
C                  EQUATIONS.  BECAUSE IT IS 1X1, IT IS EQUAL TO THE
C                  LARGEST EIGENVALUE OF THE 7X7 COEFFICIENT MATRIX
C                  OF THE COUPLED POINT-KINETICS EQUATIONS.  THIS IS
C                  THE EIGENVALUE THAT CORRESPONDS TO THE FAST MODE.
C         DI(6)    IS A CORRECTION MATRIX.  IT IS 6X1 WHEN COMPUTING
C                  THE L MATRIX, 1X6 WHEN COMPUTING THE K MATRIX.
C                  IT IS COMPUTED TO SERVE AS A CORRECTION TO THE
C                  OLD MATRICES L OR K.  IT IS ADDED TO THE OLD
C                  MATRIX TO GENERATE A NEW ONE.
C         DY(7)    IS A 7X1 MATRIX REPRESENTING DECOUPLED VARIABLES
C                  CORRESPONDING TO EACH OF THE 7 VARIABLES SOLVED FOR
C                  IN THE 7X7 SYSTEM OF COUPLE POINT-KINETICS EQUA-
C                  TIONS  (6 DELAYED-NEUTRON GROUPS PLUS PROMPT RE-
C                  SPONSE).  IT REPRESENTS ALL OF THE VARIABLES THAT
C                  EXIST AFTER THE COUPLED SYSTEM OF POINT-KINETICS
C                  EQUATIONS IS DECOUPLED AND TRANSFORMED INTO THE
C                  DECOUPLED VARIABLES.  THE FIRST SIX ARE THE ANAL-
C                  OGUES OF THE SIX DELAYED-NEUTRON GROUPS, WHILE THE
C                  SEVENTH IS THE ANALOGUE OF THE PROMPT RESPONSE.
C         DYIN(6,18)  IS THE DY(I) MATRIX (MINUS THE 7TH ELEMENT.
C                  WHICH IS FOR THE PROMPT RESPONSE), AS USED DURING
C                  DURING THE PROMPT JUMP.  DURING THE PROMPT JUMP,
C                  TIME STEPS ARE OF VARIABLE.SIZE, AND THE DY(I)
C                  MATRIX IS DEFINED FREQUENTLY DURING THE FIRST
C                  0.2 SECONDS.  THUS IT IS NECESSARY TO HAVE A
C                  2-DIMENSIONAL MATRIX THAT CAN KEEP TRACK OF
C                  WHICH DY(I) IS CURRENTLY OF INTEREST.
C         INT      IS A "HOLDING VARIABLE".  WHEN CARRYNG OUT MULTI-
C                  PLICATION OPERATIONS IN THE MATRIX DECOUPLING
C                  ALGORITHM, IT IS USED TO STORE THE SUMS OF A ROW
C                  OF MULTIPLICATIONS.  AFTER THIS SUM IS USED IN
C                  FURTHER OPERATIONS, IT IS SET EQUAL TO ZERO FOR
C                  REUSE LATER.
C         H2       IS THE TIME STEP USED FOR THE DECOUPLED ANALOGUES
C                  OF THE DELAYED-NEUTRON PRECURSORS.  IT IS 200
C                  TIMES THE SIZE OF THE TIME STEP CURRENTLY IN
C                  USE FOR THE PROMPT RESPONSE.
C         H20(22) IS A VARIABLE H2 TIME STEP USED ONLY DURING THE
C                  FIRST 0.2 SECONDS.
C         KK       IS A DUMMY VALUE USED TO ARRANGE THE EIGENVALUES OF
C                  THE COEFFICINT MATRIX OF THE COUPLED POINT-
C                  KINETICS EQUATIONS IN ASCENDING ORDER OF THEIR
C                  ABSOLUTE VALUES.
C         KM(6)    IS THE 6X1 K MATRIX.  IT AND THE L MATRIX ARE FIRST
C                  USED TO TRANSFORM THE COUPLED VARIABLES OF THE
```

```
C                 POINT-KINETICS EQUATIONS INTO THEIR DECOUPLED ANAL-
C                 OGUES, THEN, AFTER SOLUTIONS FOR THE DECOUPLED VAR-
C                 IABLES ARE OBTAINED, ARE USED TO TRANSFORM THESE
C                 NEW VALUES FOR THE DECOUPLED VARIABLES INTO THEIR
C                 COUPLED, OR "REAL WORLD" VARIABLES.  IN PRACTICE,
C                 THE ONLY "REAL WORLD" VARIABLE OF INTEREST IS REAC-
C                 TOR POWER, AND IS THE ONLY DECOUPLED VARIABLE TO BE
C                 BACK-TRANSFORMED.  ONLY THE STEADY-STATE, INITIAL
C                 VALUES OF THE COUPLED VARIABLES ARE TRANSFORMED
C                 INTO THEIR DECOUPLED ANALOGUES;  THESE VALUES ARE
C                 NEEDED TO SOLVE THE DECOUPLED DIFFERENTIAL
C                 EQUATIONS.
C        KL(6,6)IS THE 6X6 MATRIX FORMED BY MULTIPLYING THE K
C                 MATRIX TIMES THE L MATRIX.  IT IS USED TO TRANSFORM
C                 THE INITIAL VALUES FOR THE SIX DELAYED-NEUTRON
C                 GROUPS INTO THEIR DECOUPLED ANALOGUES.
C        LM(6)    IS THE 1X6 L MATRIX.  IN ADDITION TO PERFORMING THE
C                 FUNCTIONS DESCRIBED IN THE KM(6) NARRATIVE, THE
C                 VALUES OF ITS ELEMENTS ARE ALSO USED TO COMPUTE THE
C                 B1 MATRIX.  (THE B2 MATRIX IS SIMPLY THE EIGENVALUE
C                 OF THE FAST MODE.)  THE K MATRIX IN TURN IS DERIVED
C                 BY SOLVING THE ALGEBRAIC RICCATI EQUATION.
C        N        THE ORDER OF THE MATRIX A.  IS INPUT TO THE PORT-
C                 LIBRARY SUBROUTINE EIGEN.
C        NB       IS THE NUMBER OF VARIABLES IN THE POINT-KINETICS
C                 EQUATIONS (7).  IS INPUT TO THE PORTLIBRARY SUB-
C                 ROUTINE CLINQ.
C        NM       IS THE ROW DIMENSION OF THE MATRIX A(7).  IS USED
C                 AS INPUT TO THE PORTLIBRARY SUBROUTINE EIGEN.
C        Q22      IS THE 1X1 SUBMATRIX IN THE (2,2) POSITION
C                 OF THE FUNDAMENTAL EIGENVECTOR MATRIX, SPLIT
C                 INTO 4 SUBMATRICES.
C        RIE      IS THE RESIDUAL ERROR MATRIX USED IN THE MATRIX
C                 DECOUPLING ALGORITHM.  IT IS THE EUCLIDEAN NORM OF
C                 THE 6X1 (WHEN COMPUTING THE "L" MATRIX) RESIDUAL
C                 ERROR MATRIX, WHICH IS 1X6 WHEN COMPUTING THE "K"
C                 MATRIX.
C        RMT      IS THE SQUARE OF RIE.
C        RM(6)    IS THE RESIDUAL ERROR MATRIX (6X1 WHEN COMPUTING
C                 THE LM(6) MATRIX, 1X6 WHEN COMPUTING THE KM(6)
C                 MATRIX).  IF ITS EUCLIDEAN NORM IS SMALL ENOUGH,
C                 THEN THE LM(6) AND THE KM(6) MATRICES ARE
C                 CONSIDERED TO HAVE CONVERGED TOWARD THEIR TRUE
C                 VALUES.
C        XR(7,7)IS THE 7X7 INVERSE FUNDAMENTAL EIGENVECTOR MATRIX
C                 OF THE COEFFICIENT MATRIX OF THE COUPLED POINT-
C                 KINETICS EQUATIONS.  ALSO KNOWN AS THE M MATRIX.
C                 IT IS OUTPUT FROM THE PORTLIBRARY SUBROUTINE CLINQ.
C        XI(7,7)IS THE 7X7 MATRIX WHICH IS THE IMAGINARY COMPANION
C                 TO XR(7,7).  THE VALUES OF ALL ITS ELEMENTS ARE
C                 EQUAL TO ZERO.
C        VAL      IS A HOLDING VARIABLE USED IN ARRANGING THE EIGEN-
C                 VALUES OF THE COEFFICIENT MATRIX OF THE COUPLED
C                 POINT-KINETICS EQUATIONS IN ASCENDING ORDER OF
C                 THEIR ABSOLUTE VALUES.
C        VALP     IS A HOLDING VARIABLE USED IN ARRANGING THE EIGEN-
C                 VALUES OF THE COEFFICIENT MATRIX OF THE COUPLED
```

```
C                     POINT-KINETICS EQUATIONS IN ASCENDING ORDER OF
C                     THEIR ABSOLUTE VALUES.  FOR A FURTHER EXPLANATION
C                     OF VAL & VALP, SEE THE COMMENTS IMMEDIATELY PRE-
C                     CEEDING DO LOOP 100.
C             VK      IS AN INTEGER WHOSE VALUE DURING THE FIRST 0.2
C                     SECONDS OF OPERATION DETERMINES WHETHER A DY(I)
C                     MATRIX IS COMPUTED OR NOT.
C             VL      IS THE SAME AS VK, BUT IS USED AT SLIGHTLY LATER
C                     TIMES.
C             WR(7)   IS THE 7X1 MATRIX CONSISTING OF THE EIGENVALUES OF
C                     THE A(7,7) MATRIX.  IS OUTPUT OF THE PORTLIBRARY
C                     SUBROUTINE EIGEN.
C             WI(7)   IS THE IMAGINARY COMANION TO WR(7).  THE VALUES OF
C                     ALL ITS ELEMENTS ARE EQUAL TO ZERO.
C             Z(7,7)  IS THE FUNDAMENTAL EIGENVECTOR MATRIX OF THE COEF-
C                     FICIENT MATRIX OF THE COUPLED POINT-KINETICS EQUA-
C                     TIONS.  IT IS OUTPUT FROM THE PORTLIBRARY SUBROU-
C                     TINE EIGEN, AND IS IDENTICAL TO THE AR(7,7) MATRIX.
C                     (SINCE THE CLINQ SUBROUTINE DEMANDS THAT ANY MATRIX
C                     TO BE INVERTED BE CALLED THE AR MATRIX, IT WAS
C                     NECESSARY TO "RELABEL" THE Z(7,7) MATRIX BY CREA-
C                     TING THE AR(7,7) MATRIX WHICH IS EXACTLY EQUAL TO
C                     IT.  IN THIS PROGRAM, THE FUNDAMENTAL EIGENVECTOR
C                     MATRIX IS INVERTED BECAUSE THIS MAKES THE INITIAL
C                     APPROXIMATION TO THE LM(6) MATRIX EASIER TO
C                     COMPUTE.)

      REAL      DNI, DRC, DROIN, DROD, DRODSQ, DROTSQ, DTFGT,
     +          DTMGT, DTMGTT, DTMT, LMBD, PTG, PTO, PWRPJ,
     +          PWRRI, PWR1, PWR2, SIG, SIGPJ, T, TTG, X
      REAL      CIO(6), DCI(6)
      REAL      DROR, Q22
      INTEGER   I, INT1, INT2, INT3, INT4, INT5, INT6, KK, M, N, NB,
     +          NM, SV, VK, VL
      REAL*8    A (7,7), AI(7,7), AR(7,7), BI(7,7), BR(7,7), B1(6,6),
     +          DI(6),   DY(7),   KM(6),   KL(6,6), LM(6),   M31(5),
     +          Q21(7),  RM(6),   SS(7),   XI(7,7), XR(7,7), WI(14),
     +          WR(14),  Z(7,7),  B(6),    LMB(6), BB, B2, INT, L, LK,
     +          RIE, RMT, VAL, VALP, XX
      REAL*8    DYIN(6,18), H2O(22)
      REAL      ADJF, ADJM, DRO1, DRO2, DROT, SSD, SSDN, SSDO, TH
      DIMENSION LPWR(7), MPWR(7), ZERO(1), TEN(2),  TWNT(2), THRT(2),
     +          FORT(2), FFTY(2), SIXT(2), DN1G(22), STM(22)


C         HERE, INITIALIZATIONS TAKE PLACE, AND ONLY DURING THE FIRST
C     0.2 SECONDS.  ALSO, IF THERE IS TO BE NO KINETICS, CONTROL IS
C     PASSED IMMEDIATELY TO OTHO.

      IF (NK .EQ. 1)  GO TO 480
      IF (Y .EQ. 1.0) TTG   = 0.0
      IF (Y .NE. 1.0) GO TO 50


            ADJF  =    1.0
            ADJM  =    1.0
            BB    =    0.0
```

```
                    DNI    =    0.0
                    DROIN =     DRO
                    DROR  =     DRO
                    DRC   =     0.0
                    DTFGT =     0.0
                    DTMGT =     0.0
                    DTMGTT=     0.0
                    DTMT  =     0.0
                    H2O(1)=     0.000001
                    INT   =     0.0
                    L     =     0.0001
                    PTG   =     NIO + DN1
                    PTO   =     PTG
                    PWRPJ =     N10
                    PWRRI =     0.0
                    PWR1  =     0.0
                    SIG   =     0.0
                    SIGPJ =     0.0
                    T     =     0.0
                    TH    =     0.0
                    TRI   =     0.0
                    TTG   =     0.0
                    VK    =     1
                    VL    =     2
                    X     =     1.0
                    XX    =    -1.0
                    A(7,7)=     (RHO - BB) / L

C         HERE, DEPENDING ON WHICH ISOTOPE WAS SELECTED IN NERO, THE
C     DELAYED-NEUTRON FRACTIONS AND DECAY CONSTANTS FOR EACH OF THE
C     SIX DELAYED-NEUTRON GROUPS IS SELECTED.

                IF (IS .EQ. 1)     B(1)     =   2.2876E-04
                IF (IS .EQ. 1)     B(2)     =   7.9534E-04
                IF (IS .EQ. 1)     B(3)     =   6.7032E-04
                IF (IS .EQ. 1)     B(4)     =   7.3948E-04
                IF (IS .EQ. 1)     B(5)     =   1.3566E-04
                IF (IS .EQ. 1)     B(6)     =   9.0440E-05
                IF (IS .EQ. 1)     LMB(1)   =   0.0126
                IF (IS .EQ. 1)     LMB(2)   =   0.0337
                IF (IS .EQ. 1)     LMB(3)   =   0.1390
                IF (IS .EQ. 1)     LMB(4)   =   0.3250
                IF (IS .EQ. 1)     LMB(5)   =   1.1300
                IF (IS .EQ. 1)     LMB(6)   =   2.5000
                IF (IS .EQ. 2)     B(1)     =   2.1450E-04
                IF (IS .EQ. 2)     B(2)     =   1.4235E-03
                IF (IS .EQ. 2)     B(3)     =   1.2740E-03
                IF (IS .EQ. 2)     B(4)     =   2.5675E-03
                IF (IS .EQ. 2)     B(5)     =   7.4750E-04
                IF (IS .EQ. 2)     B(6)     =   2.7300E-04
                IF (IS .EQ. 2)     LMB(1)   =   0.0124
                IF (IS .EQ. 2)     LMB(2)   =   0.0305
                IF (IS .EQ. 2)     LMB(3)   =   0.1110
                IF (IS .EQ. 2)     LMB(4)   =   0.3010
                IF (IS .EQ. 2)     LMB(5)   =   1.1400
                IF (IS .EQ. 2)     LMB(6)   =   3.0100
```

```
      IF (IS .EQ. 3)   B(1)    =  7.4200E-05
      IF (IS .EQ. 3)   B(2)    =  6.3176E-04
      IF (IS .EQ. 3)   B(3)    =  4.4732E-04
      IF (IS .EQ. 3)   B(4)    =  6.9112E-04
      IF (IS .EQ. 3)   B(5)    =  1.8232E-04
      IF (IS .EQ. 3)   B(6)    =  9.3280E-05
      IF (IS .EQ. 3)   LMB(1)  =  0.0128
      IF (IS .EQ. 3)   LMB(2)  =  0.0301
      IF (IS .EQ. 3)   LMB(3)  =  0.1240
      IF (IS .EQ. 3)   LMB(4)  =  0.3250
      IF (IS .EQ. 3)   LMB(5)  =  1.1200
      IF (IS .EQ. 3)   LMB(6)  =  2.6900


C          HERE, THE COEFFICIENT MATRIX FOR THE COUPLED POINT-KINETICS
C     EQUATIONS IS DEFINED.  ALSO, SOME OF THE INITIALIZATONS FOR
C     VARIABLES USED IN MATRIX DECOUPLING ARE MADE HERE.

              N         =  7
              NB        =  7
              NM        =  7
              RIE       =  0.0
              RMT       =  0.0
      DO 10 I = 1,6
              A(1,I)    =  0.0
              A(2,I)    =  0.0
              A(3,I)    =  0.0
              A(4,I)    =  0.0
              A(5,I)    =  0.0
              A(6,I)    =  0.0
              A(I,I)    = -LMB(I)
              A(7,I)    =  LMB(I)
              A(I,7)    =     B(I) / L
              M3I(I)    =  0.0
              DI (I)    =  0.0
              KM(I)     =  0.0
              RM (I)    =  0.0
   10 CONTINUE
      DO 30 I = 1,7
      DO 20 J = 1,7
              AI(I,J)   =  0.0
              BI(I,J)   =  0.0
              BR(I,J)   =  0.0
   20 CONTINUE
              BR(I,I)   =  1.0
              DY(I)     =  0.0
              DYIN(I,1)=  0.0
   30 CONTINUE


C          HERE, INITIAL VALUES FOR THE DELAYED-NEUTRON FRACTIONS ARE
C     DERIVED.  ALSO, THE DELAYED-NEUTRON FRACTIONS ARE SUMMED.

      DO 40 I = 1,6
              CIO(I)= N10 * B(I) / (L * LMB(I))
              BB    = BB  + B(I)
              DCI(I)= 0.0
```

```
 40   CONTINUE
           SSD    =    2.0 * DRO / BB
           SSDO   =    2.0 * DRO / BB


 50   CONTINUE

C        HERE, REACTIVITY EFFECTS DUE TO FEEDBACK AND THE REACTOR
C     CONTROL SYSTEM ARE COMPUTED.

      IF ((ZZ .EQ. 1) .AND.  (K .LE. 29.0))  GO TO 480
           DTMT       =    DTM(K-1) + DTMT
           IF ((X .NE. 1.0)  .AND.  (CS .EQ. 2))
     +     DRO        =    ALPHF * DTF(K-1) + ALPHM * DTM(K-1) + DROIN +
     +                     (RIR * TTG)
 60        IF ((X .NE. 1.0)  .AND.  (CS .EQ. 1))
     +     DRO        =    ALPHF * DTF(K-1) + ALPHM * DTM(K-1) + DROIN +
     +                     (RIR * TTG) + DRC
           DRC        =    0.2 * (AA * (DTM(K-1) + (0.2 * DTMT / TAU)) -
     +                     (DRC / TAUC)) + DRC
           IF ((X .NE. 1.0)  .AND.  (CS .EQ. 2)  .AND.  (ZZ .EQ. 1))
     +     DRO        =    ALPHF * DTF(K-1) + ALPHM * DTM(K-1)
           IF ((X .NE. 1.0)  .AND.  (CS .EQ. 1)  .AND.  (ZZ .EQ. 1))
     +     DRO        =    ALPHF * DTF(K-1) + ALPHM * DTM(K-1) + DRC
           DRC        =    0.2 * (AA * (DTM(K-1) + (0.2 * DTMT / TAU)) -
     +                     (DRC / TAUC)) + DRC


C
C        HERE, THE MATRIX DECOUPLING ALGORITHM OCCURS.  IF IT IS NOT
C     SELECTED, CONTROL IMMEDIATELY JUMPS TO STATEMENT 330.  IF IT IS,
C     IT STARTS HERE.  AS GALBA PROCEEDS THROUGH THIS ALGORITHM, AT
C     VARIOUS PLACES COMMENT STATEMENTS WILL BE INSERTED TO EXPLAIN
C     WHAT IS GOING ON.
C
 70   IF (EIG .EQ. 2)  GO TO 330
C
C        HERE, THE (7,7)TH ELEMENT OF THE COEFFICIENT MATRIX OF THE
C     COUPLED POINT-KINETICS EQUATIONS IS  DEFINED.  AS IT IS REACTIVITY
C     DEPENDENT, IT IS THE ONLY PART OF THIS MATRIX THAT VARIES.  THE
C     OTHER ELEMENTS OF THIS MATRIX ARE REDEFINED, TOO, BUT ALWAYS WITH
C     THE SAME VALUES.  THE REASON THEY NEED TO BE REDEFINED IS THAT
C     EXECUTION OF THE PORTLIBRARY SUBROUTINE EIGEN, WHICH TAKES PLACE
C     NEXT, OVERWRITES OR CANCELS ALL OF ITS INPUT, SO IT ALL NEEDS TO
C     BE REDEFINED PRIOR TO THE NEXT TIME EIGEN IS CALLED, EVEN IF
C     DOES NOT CHANGE.  THE FIRST TIME THE MATRIX DECOUPLING ALGORITHM IS
C     CALLED, HOWEVER, REACTIVITY IS CONSIDERED TO BE ZERO.  THIS IS TO
C     CALCULATE THE INITIAL VALUES FOR THE ANALOGUES OF THE VARIABLES
C     OF THE DELAYED-NEUTRON RESPONSE AND OF THE PROMPT RESPONSE.  AS
C     THIS PROGRAM IS NOW RUN ONLY ONCE, HOWEVER, NONE OF THE ABOVE NOW
C     APPLIES.  THIS INFORMATION IS INCLUDED FOR THOSE WHO MAY WISH TO
C     ALTER GALBA IN THE FUTURE SUCH THAT MATRIX DECOUPLING COULD BE
C     CALLED MORE THAN ONCE DURING A GIVEN RUN.  ALSO, THE K MATRIX IS
C     INITIALLY DEFINED IN TERMS OF THE A MATRIX, SO ITS VALUES WOULD
C     NEED TO BE REDEFINED ANYWAY.
C
      IF (Y .NE. 1.0)  GO TO 330
```

```
 80         A(7,7)    = (RHO - BB) / L
C
C          HERE, THE PORTLIBRARY SUBROUTINE "EIGEN" IS CALLED.  ITS
C     PURPOSE IS TO TAKE THE COEFFICIENT MATRIX OF THE COUPLED SYSTEM
C     OF POINT-KINETICS EQUATIONS AND DERIVE ITS EIGENVALUES AND
C     EIGENVECTORS.
C
       CALL DEIGEN (NM, N, A, WR, WI, Z)
C
C          HERE, THE REMAINING ELEMENTS OF THE COEFFICIENT MATRIX OF THE
C     THE SYSTEM OF COUPLED POINT-KINETICS EQUATIONS ARE REDEFINED, DUE
C     TO THEIR HAVING BEEN OVERWRITTEN DURING THE CALLING OF "EIGEN",
C     WHICH TOOK PLACE JUST PREVIOUSLY.
C
       DO 90 I = 1,6
            A(1,I)    =  0.0
            A(2,I)    =  0.0
            A(3,I)    =  0.0
            A(4,I)    =  0.0
            A(5,I)    =  0.0
            A(6,I)    =  0.0
            A(I,I)    = -LMB(I)
            A(7,I)    =  LMB(I)
            A(I,7)    =     B(I) / L
 90    CONTINUE
            RIE       =  0.0
C
C          HERE, THE EIGENVALUES THAT WERE JUST DERIVED FROM "EIGEN" ARE
C     REARRANGED, IN ASCENDING ORDER OF THEIR ABSOLUTE VALUES.  THIS IS
C     NECESSARY BECAUSE THE MATRIX DECOUPLING ALGORITHM DEMANDS THAT THE
C     FUNDAMENTAL EIGENVECTOR MATRIX CORRESPOND TO A DIAGONAL EIGENVALUE
C     MATRIX IN WHICH THIS HAS BEEN DONE, BUT "EIGEN" DOES NOT DO THIS.
C     FIRST, THE SQUARE ROOTS OF THE SQUARES OF THE EIGENVALUES ARE
C     DERIVED.  THIS WILL RESULT IN ALL POSITIVE VALUES ("DO" LOOP 100).
C     THEN EACH EIGEN VALUE IS EXAMINED AS DELIVERED FROM "EIGEN".  IT
C     IS ASSUMED THAT ANY EIGENVALUE EXAMINED IS THE SMALLEST OF THE LOT.
C     THE FIRST EIGENVALUE EXAMENED IS ASSIGNED THE VARIABLE NAME "VAL".
C     ANOTHER VARIABLE, "VALP", IS SET EQUAL TO IT.  THEN THE EIGENVALUE
C     IS EXAMINED AGAINST ALL THE OTHERS.  AS LONG AS IT IS SMALLER THAN
C     ANY OF THEM, THE ASSUMPTION IS CONSIDERED TO HOLD UP, AND THE
C     EIGENVALUE IS THEN COMPARED TO THE NEXT ONE.  (THIS TAKES PLACE IN
C     "DO" LOOP 110, WHICH IS NESTED IN 130.)  IF IT IS LARGER THAN THE
C     NEW EIGENVALUE BEING EXAMINED, THE NEW EIGENVALUE THEN BECOMES THE
C     "NEW" POSSIBLY SMALLEST EIGENVALUE, AND "VALP" IS SET EQUAL TO IT.
C     IF "VALP" IS SMALLER THAN "VAL", THEN "VAL" IS SET EQUAL TO "VALP".
C     THROUGH THIS MEANS, "VAL" IS SET, AFTER 7 ITERATIONS, EQUAL TO THE
C     SMALLEST OF THE EIGENVALUES.  SINCE THE VARIABLE "KK" IS SET EQUAL
C     TO "J" ONLY WHEN "VALP" IS LESS THAN "VAL", WHERE "J" IS THE SUB-
C     SCRIPT IN THE "DO" LOOP OF THE EIGENVALUE BEING COMPARED AT THE
C     MOMENT, NOT ONLY IS THE ABSOLUTELY SMALLEST EIGENVALUE IDENTIFIED,
C     BUT SO IS ITS EIGENVECTOR.  THEN, IN "DO" LOOP 120, THIS EIG-
C     VECCTOR IS PLACED IN THE LEFTMOST COLUMN IN THE FUNDAMENTAL EIGEN-
C     VECTOR MATRIX, WHERE IT WOULD BELONG IN ORDER TO CORRESPOND TO THE
C     POSITION IN THE DIAGONAL EIGENVALUE MATRIX GIVEN TO THE SMALLEST
C     EIGENVALUE.  THIS CONTINUES UNTIL ALL EIGENVALUES AND THE EIGEN-
C     VECTORS ASSOCIATED WITH THEM ARE PLACED IN ASCENDING ORDER OF
C     THEIR ABSOLUTE VALUES.
```

```
C
      DO 100 I = 1,7
            VAL      =  WR(I) ** 2.0
            WR(I)    =  DSQRT (VAL)
  100 CONTINUE
      DO 130 I = 1,7
            KK       =  I
            VAL      =  WR(I)
            VALP     =  VAL
      DO 110 J = I,7
      IF (WR(I) .LE. WR(J))  GO TO 110
      IF (WR(I) .GT. WR(J))  VALP = WR(J)
      IF (VALP  .LT.   VAL)  KK   = J
      IF (VALP  .LT.   VAL)  VAL  = VALP
  110 CONTINUE
      IF (KK .NE. I)       WR(KK) = WR(I)
      DO 120 J = 1,7
            AR(J,I)  =  Z(J,KK)
            Z(J,KK)  =  Z(J,I)
  120 CONTINUE
            WR(I)    =  VAL
  130 CONTINUE
C
C         HERE, THE B2 MATRIX IS DEFINED.  BY A HAPPY STROKE, SINCE THE
C     B2 MATRIX IS 1X1, IT IS PRECISELY EQUAL TO THE EIGENVALUE CORRES-
C     PONDING TO THE FAST MODE.  THE NEGATIVE SIGN IS NECESSARY BECAUSE
C     ALL THE EIGENVALUES OF THE COEFFICIENT MATRIX OF THE COUPLED POINT-
C     KINETICS EQUATIONS ARE NEGATIVE, BUT THEY WERE ALL CHANGED INTO
C     POSITIVE QUANTITIES WHEN THEY WERE REARRANGED INTO ASCENDING ORDER
C     OF THEIR ABSOLUTE VALUES.
C
            B2       =  -WR(7)
C
C         HERE, THE FUNDAMENTAL EIGENVECTOR MATRIX IS INVERTED VIA THE
C     PORTLIBRARY SUBROUTINE CLINQ.  THE INVERSE MATRIX IS THE XR MATRIX,
C     WHICH IS 7X7.  A FIRST APPROXIMATION FOR THE LM(I) MATRIX IS THEN
C     DERIVED BY DIVIDING THE COLUMN MATRIX IN THE 7TH COLUMN, FIRST SIX
C     ROWS, BY THE ELEMENT IN THE (7,7)TH POSITION IN THE XR MATRIX.
C
       IF (K .GE. 10)  GO TO 140
      CALL DCLINQ (N, AR, AI, BR, BI, NB, XR, XI)
      DO 140 I = 1,6
            LM(I)    =  XR(7,I) / XR(7,7)
  140 CONTINUE
  150       INT      =  0.0
C
C         HERE USING THE BEST VALUE FOR LM(I), THE RESIDUAL ERROR
C     MATRIX IS COMPUTED.
C
      DO 160 I = 1,6
            RM(I)    =  LMB(I) - (B2 * LM(I)) - (LM(I) * LMB(I))
  160 CONTINUE
      DO 170 I = 1,6
            RMT      =((RM(I)) ** 2.0) + RMT
  170 CONTINUE
            RIE      =  DSQRT (RMT)
            RMT      =  0.0
```

```
C
C         IF THE EUCLIDEAN NORM OF THE RESIDUAL ERROR MATRIX (WHICH
C    WHAT "RIE" IS), IS SMALLER THAN A CERTAIN VALUE, THE PROGRAM WILL
C    PASS CONTROL ON TO STATEMENT 190.  THE LM(I) MATRIX AT THIS POINT
C    WILL BE CONSIDERED TO BE WELL-DEFINED.
C
        IF (RIE .LE. 1.0E-10)  GO TO 190
C
C         SINCE THE LM(I) MATRIX IS NOT WELL-DEFINED AT THIS POINT, A
C    CORRECTION MATRIX DI(I) (WHICH IS 6X1) IS COMPUTED AND ADDED
C    THE LM(I) MATRIX.  THEN CONTROL IS PASSED TO STATEMENT 150, AND IT
C    IS THEN DETERMINED IF THIS STATEMENT IS WELL-DEFINED.
C
        DO 180 I = 1,6
              DI(I)     =  RM(I) / B2
              LM(I)     =  LM(I) + DI(I)
  180   CONTINUE
        GO TO 150
  190         RIE       =  0.0
C
C         SINCE THE B1 MATRIX IS DEFINED ONLY IN TERMS OF THE LM(I)
C    MATRIX AND THE COEFFICIENT MATRIX OF THE COUPLED POINT-KINETICS
C    EQUATIONS, THIS IS WHERE IT IS COMPUTED.  THE KM(I) MATRIX IS NOT
C    NEEDED HERE, AND WILL BE USED ONLY AT THE END, WHEN IT IS NECESSARY
C    TO CONVERT THE SOLUTION FOR THE DECOUPLED VARIABLES INTO ACTUAL
C    REACTOR POWER.  ALSO, FOR PURPOSES OF COMPUTING THE KM(I) MATRIX,
C    AT THIS POINT A RESIDUAL ERROR MATRIX IS COMPUTED.  IF ITS EUCLI-
C    DEAN NORM IS SMALL ENOUGH (WHICH ISN'T LIKELY AT FIRST), THEN ALL
C    OF THE ELEMENTS OF THE KM(I) MATRIX WILL EQUAL ZERO.  OTHERWISE,
C    SUCCESSIVE ITERATIONS OF A CORRECTION MATRIX WILL HAVE TO LEAD TO
C    A CORRECT KM(I) MATRIX.
C
        DO 210 I = 1,6
        DO 200 J = 1,6
              INT       = (B(J) / L) *   LM(I)
              B1(J,I)   =  XX * INT
        IF (I .EQ. J)     B1(I,J)    =  -LMB(I) - INT
  200   CONTINUE
              RM(I)     = -B(I) / L
  210   CONTINUE
              INT       =  0.0
  220   DO 230 I = 1,6
              RMT       =  RMT + ((RM(I)) ** 2.0)
  230   CONTINUE
              RIE       =  SQRT (RMT)
              RMT       =  0.0
C
C         HERE, IF THE EUCLIDEAN NORM OF THE RESIDUAL ERROR MATRIX IS
C    SMALL ENOUGH, CONTROL WILL PASS ON TO STATEMENT 270.  IF NOT, THE
C    SERIES OF CORRECTION MATRICES WILL HAVE TO BE COMPUTED.
C
        IF (RIE .LE. 1.0E-10)  GO TO 270
        DO 240 I = 1,6
              DI(I)     =  RM(I) / B2
              KM(I)     =  KM(I) + DI(I)
  240   CONTINUE
              INT       =  0.0
```

```
C
C        HERE, THE CORRECTION MATRIX DI(I) MATRIX IS COMPUTED.  IT
C     THEN ADDED TO THE MOST RECENT VERSION OF THE KM(I) MATRIX.
C     THE RESIDUAL ERROR MATRIX ASSOCIATED WITH THIS IMPROVED KM(I)
C     MATRIX IS COMPUTED, AND THE CONTROL SHIFTS BACK TO STATEMENT 220.
C
          DO 260 I = 1,6
          DO 250 J = 1,6
               INT       =  B1(I,J) * KM(J) + INT
  250     CONTINUE
               RM(I)     =-(KM(I)    * B2)    + INT - (B(I) / L)
               INT       =  0.0
  260     CONTINUE
          GO TO 220
  270          RIE       =  0.0
C
C        HERE, INITIALIZATIONS OF THE DECOUPLED ANALOGUES FOR THE
C     VARIABLES OF THE DELAYED-NEUTRON RESPONSE AND THE PROMPT RESPONSE
C     TAKE PLACE.
C
          CONTINUE
          DO 290 I = 1,6
          DO 280 J = 1,6
               KL(I,J)   =  KM(I)    * LM(J)
  280     CONTINUE
               KL(I,I)   =  KL(I,I) + 1.0
  290     CONTINUE
          DO 310 I = 1,6
          DO 300 J = 1,6
               INT       =  KL(I,J) * CIO(J) + INT
  300     CONTINUE
               SS(I)     = (KM(I)    * N10)    + INT
               INT       =  0.0
  310     CONTINUE
          DO 320 I = 1,6
               INT       =  LM(I)    * CIO(I) + INT
  320     CONTINUE
               SS(7)     =  N10      + INT
               INT       =  0.0
          DO 325 I = 1,6
               LK        =  LM(I) * KM(I) + LK
  325     CONTINUE
               LK        =  LK     + 1.0
C
C        HERE, THE SIZE OF THE TIME STEPS FOR THE PROMPT-JUMP APPROX-
C     ATION AND THE RAMP-INPUT MODEL IS DEFINED.  ALSO, IF THE PJ MODEL
C     IS SELECTED, CONTROL IS PASSED ON TO STATEMENT 390.
C


  330          IF  (PJ .EQ. 1)    H = 0.001
               IF  (RI .EQ. 1)    H = 0.001
               IF  (PJ .EQ. 1)    GO TO 390


C
C        GALBA SOLVES THE POINT-KINETICS EQUATIONS.  SINCE THEY ARE
```

```
C     COUPLED EQUATIONS, ANY NUMERICAL SOLUTION - WHICH AFTER ALL IS WHAT
C     GALBA DOES - IS VALID ONLY ON TIME INTERVALS IN WHICH NEITHER N-
C     TRON (REACTOR POWER) DENSITY NOR THE DELAYED PRECURSOR DENSITY
C     VARIES GREATLY.  HOWEVER, DURING THE PROMPT JUMP, BOTH DO VARY
C     GREATLY.  THE "WAY AROUND THIS" IS TO MAKE THE TIME STEPS ARBITRAR-
C     ILY SMALL UNTIL THE PERIOD OF THE PROMPT JUMP IS OVER.  THIS
C     WHAT GALBA ACTUALLY DOES.  THE TIME STEPS START AOUT BEING EQUAL
C     1 MICROSECOND.  BY THE TIME 1 MILLISECOND HAS PASSED, THE TIME STEP
C     ALSO EQUAL 1 MILLISECOND, WHERE IT REMAINS FOR THE DURATION OF THE
C     RUN.  X IS A NUMBER THAT CAUSES THE TIME STEPS TO ENLARGE THEM-
C     SELVES AFTER A CERTAIN LENGTH OF TIME HAS ELAPSED.  THE ALGEBRAIC
C     SUM OF ALL THE TIME STEPS IS 1 MILLISECOND.
C
  340   IF (X .LE. 10.0)  GO TO 341
        IF (X .LE. 13.0)  GO TO 342
        IF (X .LE. 16.0)  GO TO 343
        IF (X .GE. 17.0)  GO TO 344
  341        H       =     0.000001
             X       =     X  + 1.0
             VK      =     VK + 1
        GO TO 345
  342        H       =     0.00003
             X       =     X  + 1.0
             VK      =     VK + 1
        GO TO 345
  343        H       =     0.0003
             X       =     X  + 1.0
             VK      =     VK + 1
        GO TO 345
  344        H       =     0.001
             X       =     X  + 1.0
             VK      =     VK + 1
  345        T       =     T  + H
             TH      =     TH + H
C
C          HERE, THE DELAYED-PRECURSOR DENSITY IS COMPUTED.  SINCE THE
C     PRECURSORS CONSTITUTE THE SLOW MODE, THEIR TIME STEP IS 200 TIMES
C     THE SIZE OF THE FAST MODE TIME STEP (WHICH IS 0.001 SECONDS, EXCEPT
C     DURING THE PROMPT JUMP).  THIS SECTION, OF COURSE, IS FOR THE MA-
C     TRIX DECOUPLING ALGORITHM.  DURING THE FIRST 0.2 SECONDS, THE DURA-
C     TION OF THE PROMPT JUMP, THE DELAYED PRECURSORS ARE COMPUTED USING
C     SMALL TIME STEPS.  THEY ARE ALSO COMPUTED ONLY WHEN ENOUGH TIME
C     PASSED SO THAT THE TOTAL AMOUNT OF TIME IN THEIR TIME STEPS DOES
C     NOT EXCEED THE TOTAL AMOUNT OF TIME THAT HAS PASSED IN COMPUTING
C     THE PROMPT RESPONSE.  THIS IS WHAT TAKES PLACE FROM STATEMENTS 355
C     THROUGH 370.  AT ALL OTHER TIMES, THE PRECURSORS ARE COMPUTED
C     DURING THE MIDDLE OF EACH 0.2 SECTION, TO REFELCT THEIR AVERAGE
C     VALUES.  THE PRECURSORS THEMSELVES ARE NOT PART OF THE OUTPUT,
C     AND ARE COMPUTED ONLY BECAUSE THE POWER OUTPUT IS A FUNCTION OF
C     THEM.
C
        IF ((X .LE. 17.0)   .AND.  (EIG .EQ. 1))  GO TO 350
        IF ((T .GE. 0.0996) .AND.  (EIG .EQ. 1)   .AND.
     +      (T .LE. 0.1004) .AND.  (Y .NE. 1.0))  GO TO 350
        GO TO 380
  350          H2      =    H * 200.0
               H20(VK) =    H * 200.0
```

```
            IF (VK .GE. 20)    VK = 20
            IF (Y .GE. 1.1)    GO TO 365
            IF ((VK .GT.  6)  .AND.  (T .LT. 0.0011))  GO TO 375
            IF ((VL .GT. 11)  .AND.  (T .LT. 0.0019))  GO TO 375
            IF ((VL .GT. 11)  .AND.  (T .LT. 0.0079))  GO TO 375
            IF ((VL .GT. 12)  .AND.  (T .LT. 0.0139))  GO TO 375
            IF ((VL .GT. 13)  .AND.  (T .LT. 0.0199))  GO TO 375
            IF ((VL .GT. 14)  .AND.  (T .LT. 0.0799))  GO TO 375
            IF ((VL .GT. 15)  .AND.  (T .LT. 0.1399))  GO TO 375
            IF ((VL .GT. 16)  .AND.  (T .LT. 0.1999))  GO TO 375
            DO 360 I = 1,6
            DO 355 J = 1,6
                 INT       =   (B1(I,J)    * DYIN(J,(VL-1))) + INT
      355   CONTINUE
                 DYIN(I,VL)=   H2O(VL)     * (INT + (KM(I) * (N10 + DN1)
     +                    *    DRO / L)) +  DYIN(I,(VL-1))
                 INT       =   0.0
                 DY(I)     =   DYIN(I,VL)
      360   CONTINUE
                 VL        =   VL + 1
            IF ((VL .LE. 11) .AND.  (VK .GT. 6))  GO TO 350
            IF (Y .LE. 1.1)    GO TO 375
      365   DO 375 I = 1,6
            DO 370 J = 1,6
                 INT       =   (B1(I,J)    * DY(J)) + INT
      370   CONTINUE
                 DY(I)     =   H2 * (INT + (KM(I) *  (N10 + DN1)
     +                    *    DRO / L)) +  DY(I)
                 INT       =   0.0
      375   CONTINUE
      380   IF (EIG .EQ. 1)    GO TO 400
C
C           HERE THE POINT-KINETICS EQUATIONS ARE SOLVED, UNLESS THE
C       DECOUPLING OPTION HAS BEEN SELECTED.
C
                 DN1     =     H * (RHO * DN1 / L + DRO  * N10 / L + DRO *
     +                               DN1 / L - BB  * DN1 / L + SIG) + DN1
      390        SIG     =     0.0
                 SIGPJ   =     0.0
C
C           HERE THE CHANGE IN DELAYED-PRECURSOR DENSITY FROM THE STEADY-
C       STATE ARE COMPUTED, UNLESS THE DECOUPLING ALGORITHM HAS BEEN SELEC-
C       TED.
C
            DO 400 I     =     1,6
               DCI(I)    =     H * (B(I) * DN1 / L - LMB(I) * DCI(I)) + DCI(I)
                 SIG     =     SIG + LMB(I) * DCI(I)
                 SIGPJ   =     L    * LMB(I) *(DCI(I) + CIO(I)) + SIGPJ
      400   CONTINUE
                 TTG     =     TTG + H
C
C           HERE, THE PROMPT-JUMP APPROXIMATION OR THE RAMP-INPUT MODEL
C       ARE SELECTED.
C
            IF ((RI .EQ. 1)  .AND.  (PTO .NE. 0.0))  GO TO 415
            IF  (PJ .EQ. 2)                          GO TO 415
C
```

```
C          HERE, REACTOR POWER USING THE PROMPT-JUMP APPROXIMATION IS
C     COMPUTED.  ALSO, INTERIM REACTIVITY FROM FEEDBACK IS COMPUTED AND
C     APPLIED TO THE PROMPT JUMP BEFORE IT TAKES PLACE, THUS ALTERING ITS
C     MAGNITUDE.  THIS PREVENTS OSCILLATIONS THAT DESTROY THE SOLUTION
C     FROM TAKING PLACE BY THE FACT THAT IF AN UNMODIFIED PROMPT JUMP IS
C     ALLOWED, WHEN ITS FEEDBACK TAKES EFFECT, IT WILL PRODUCE SUCH LARGE
C     REACTIVITY THAT FURTHER SOLUTIONS WILL BE SO INACCURATE THAT OSCIL-
C     LATIONS OF INCREASING MAGNITUDE WILL TAKE PLACE.
C
            PWRPJ =        SIGPJ / (BB -(RHO + DRO))
        IF (X .GE. 2.9)  GO TO 410
            I     =        I + 1
  405       PWRPJ =        SIGPJ / (BB - (RHO + DRO1))
            DN1   =        PWRPJ - N10
            DTFGT =        DN1   * 9.4875E-10
            DTMGT =        DN1   * 2.0040E-11
            DRO2  =        DTFGT * ALPHF  +  DTMGT * ALPHM
            DROT  =        DRO1
            DRO1  =        DRO   + DRO2
            DROT  =        DROT  - DRO1
            DROT  =        DROT ** 2.0
            DROT  =        SQRT (DROT)
            I     =        I     + 1
        IF (I .GE. 100)          GO TO 410
        IF (DROT .GE. 1.0E-09)   GO TO 405
  410   CONTINUE
            DN1   =        PWRPJ -  N10
  415       PWR2  =        DN1   -  PWR1
C
C          HERE, INTERIM REACTIVITY THAT TAKES PLACE DURING A GIVEN
C     PERIOD OF 0.005 SECONDS IS COMPUTED.  AFTER THIS INTERVAL, A NEW
C     REACTIVITY BASED ON FEEDBACK IS COMPUTED, AND THE OLD INTERIM
C     FIGURE IS NO LONGER NEEDED.
C
        IF (DTF(K) .EQ. 0.0)  DTF(K)  = DTF(K-1).
        IF (DTM(K) .EQ. 0.0)  DTM(K)  = DTM(K-1)
            DTFGT =        DTF(K) - DTF(K-1)
            DTMGT =        DTM(K) - DTM(K-1)
        IF (X .LE. 20.0)  DTFGT = DN1 * 9.4875E-10
        IF (X .LE. 20.0)  DTMGT = DN1 * 2.0040E-11
        IF ((CS .EQ. 2)   .AND.      (KF .EQ. 2))
     +                     DRO   =   (ALPHF * (DTF(K-1) + (ADJF * DTFGT)))
     +                           +   (ALPHM * (DTM(K-1) + (ADJM * DTMGT)))
     +                           +    DROIN + (RIR * TTG)
        IF ((CS .EQ. 1)   .AND.      (KF .EQ. 2))
     +                     DRO   =   (ALPHF * (DTF(K-1) + (ADJF * DTFGT)))
     +                           +   (ALPHM * (DTM(K-1) + (ADJM * DTMGT)))
     +                           +    DROIN + (RIR * TTG) + (H * (AA *
     +                              ((DTMGT * ADJM) + (H * ADJM * DTMGT /
     +                                TAU)))) + DRC
        IF (TTG .GE. TTOT) TTG = TTOT


C
C          HERE, THE POWER RESPONSE BASED ON THE DECOUPLING ALGORITHM IS
C     IS COMPUTED.  ALSO, IN LINE 480, CURRENT TOTAL POWER IS COMPUTED.
C     THIS WILL BE USED TO ADJUST STEAM GENERATOR POWER IN THE SUBROUTINE
```

```
C    DMTN.
C
          IF (EIG .EQ. 2)  GO TO 440
                DY(7)     =  H    * ((B2 * DY(7)) + ((N10 + DN1)
     +                    *  DRO / L))   + DY(7)
                H2        =  H * 200.0
          DO 420 I = 1,6
                INT       =  LM(I) * DY(I) + INT
  420     CONTINUE
                DN1       = (DY(7) * LK)    - INT
                INT       =  0.0
  430     IF  (TH .LT. 0.0042)  GO TO 330
  440     IF  (NK .EQ. 1)  TTG = TTG + 0.2
          IF  (NK .EQ. 1)  T   = 0.2
  450     IF (T .GE. 0.1995)  GO TO 460
          IF (T .LT. 0.1995)  GO TO 470
  460           T      =     0.0
  470           TH     =     0.0
                PWR1   =     DN1
                PTO    =     PTG
  480           RP(K)  =     N10 + DN1
          RETURN
          END
```

```
C     THERMAL-HYDRAULICS EQUATIONS
C
C          THIS IS THE SUBROUTINE THAT MODELS THE THERMAL-HYDRAULICS
C     EQUATIONS.  AS INPUT, IT RECEIVES REACTOR POWER INFORMATION FROM
C     THE REACTOR KINETICS SUBROUTINE (GALBA) AND REACTOR INLET TEMP-
C     ERATURE INFORMATION FROM THE STEAM GENERATOR SUBROUTINE (DMTN).
C     AS OUTPUT, IT COMPUTES MODERATOR TEMPERATURE CHANGES AND REACTOR
C     OUTLET TEMPERATURE CHANGES AND FUEL TEMPERATURE CHANGES.  ALL ARE
C     DISPLAYED IN THE OUTPUT.
C
C          AS IN THE OTHER SUBROUTINES, OTHO SOLVES TWO COUPLED DIFFEREN-
C     TIAL EQUATIONS.  HERE, THE VARIABLES SOLVED FOR ARE MODERATOR
C     TEMPERATURE CHANGE AND THE SKIN TEMPERATURE CHANGE ON THE FUEL
C     ASSEMBLIES.  FROM THESE ARE COMPUTED REACTOR OUTLET TEMPERTURE
C     CHANGE AND FUEL TEMPERATURE CHANGE.  TIME STEPS IN OTHO ARE 0.005
C     SECONDS.


      SUBROUTINE OTHO

      COMMON   AA, ADTI, ADTO, ALPHF, ALPHM, CPPAV, CS, DN1, DRO,
     +         DTF, DTI, DTM, DTO, DTS, DTSG, EIG, F, H, IS, K, KF,
     +         LND, MDOTP, NK, NN, N10, Q, PJ, PT, RHO, RI, RIR,
     +         RP, STMG, STMGEN, TAU, TAUC, TF1, TM, TT, TTOT, VO,
     +         VOSS, Y, ZZ
      REAL     AA, ALPHF, ALPHM, CPPAV, DN1, DRO, H, MDOTP, N10, RHO,
     +         RIR, STMGEN, TAU, TAUC, TF1, TM1, TTOT, VO, VOSS, Y
      INTEGER  CS, EIG, F, IS, J, K, KF, LND, NK, NN, PJ, Q, RI, ZZ
      REAL     ADTI(1510), ADTO(1510), DTF(1510), DTI(1510), DTM(1510),
     +         DTO (1510), DTSG(1510), DTS(1510), RP(1510),  TT (1510),
     +         PT  (1510), STMG(1510)

C          IN ADDITION TO THOSE USED N THE COMMON BLOCK, THE FOLLOWING
C     VARIABLES ARE USED IN OTHO:
C
C               AR     IS THE HEAT TRANSFER AREA.
C               CPAV   IS THE HEAT CAPACITY OF THE REACTOR COOLANT
C               CPF    IS THE HEAT CAPACITY OF THE FUEL.
C               CPHS   IS THE HEAT CAPACITY OF THE REACTOR COOLANT AT THE
C                                                 REACTOR OUTLET.
C               CPLS   IS THE HEAT CAPACITY OF THE REACTOR COOLANT AT THE
C                                                 REACTOR INLET.
C               D      IS THE DIAMETER OF REACTOR COOLANT FLOW CHANNEL.
C               DTC    IS CHANGE INTEMPERATURE OF THE SURFACE OF THE FUEL
C                         ASSEMBLIES FROM THE STEADY-STATE TEMPERATURES.
C               DTMO   IS CHANGE IN REACTOR MODERATOR (COOLANT) TEMPERATURE
C                         FROM STEADY-STATE.
C               FRAC   IS A FACTOR WHICH ADJUSTS FOR THE FACT THAT CP DOES
C                         NOT VARY LINEARLY WITH TEMPERATURE CHANGE.  WITH
C                         FRAC, ONE CAN OBTAIN TRUE AVERAGE CP AND MODERA-
C                         TOR TEMPERATURE BY TAKING THE LINEAR AVERAGE OF
C                         OF THE EXTREMES AT INLET AND OUTLET CONDITIONS,
C                         AND MULTIPLYING THE DIFFERENCE BY FRAC, AND THEN
C                         ADDING TO THE LOW-EXTREME CONDITON.
C               HP     IS THE HEAT TRANSFER COEFFICIENT.  IT IS DERIVED VIA
C                         THE DITTIUS-BOELTER CORRELATION.
```

```
C               KP      IS THE THERMAL CONDUCTIVITY OF THE REACTOR COOLANT.
C               MF      IS THE MASS OF THE FUEL.
C               MM      IS THE MASS OF REACTOR COOLANT.
C               MUP     IS THE KINEMATIC VISCOSITY OF THE REACTOR COOLANT.
C               NU      IS THE NUSSELT   NUMBER    OF THE REACTOR COOLANT.
C               NUP     IS THE DYNAMIC   VISCOSITY OF THE REACTOR COOLANT.
C               PRP     IS THE PRANDTL   NUMBER    OF THE REACTOR COOLANT.
C               REP     IS THE REYNOLDS  NUMBER    OF THE REACTOR COOLANT.
C               ROP     IS THE DENSITY             OF THE REACTOR COOLANT.
C               T       IS THE AMOUNT OF TIME EXPIRED DURING  THE CURRENT
C                          CYCLE.  IS RESET TO ZERO AT THE BEGINNING OF
C                          EACH OTHO CYCLE.
C               TI1S    IS INITIAL REACTOR INLET   TEMPERATURE.
C               TO1S    IS INITIAL REACTOR OUTLET  TEMPERATURE.
C               TPAV    IS INITIAL AVERAGE COOLANT TEMPERATURE.  IS USED
C                          TO COMPUTE HEAT TRANSFER VARIABLES SUCH AS
C                          DENSITY.
C               TPAVS   IS THE SAME AS TPAV.
C               VMOD    IS VELOCITY OF MASS FLOW.
C               X       IS AN INTEGER, THE VALUE OF WHICH DETERMINES THE
C                          SIZE OF THE TIME STEP DURING THE FIRST CYCLE.

      REAL     AR, CPAV, CPF, CPHS, CPLS, D, DTC, DTMO, DTOO, FRAC,
     +         HP, KP, MF, MM, MUP, NU, NUP, PRP, REP, ROP, T,
     +         TI1S, TO1S, TPAV, TPAVS, VMOD
      INTEGER  X

C        INITIALIZATIONS TAKE PLACE HERE, DURING THE FIRST 0.2 SECONDS,
C        ONLY.

      IF (Y .GE. 3.0)  GO TO 10
          AR     = 5945.0
          CPF    =      0.18477 * (TF1 + DTF(K)) + 74.4
          D      =      0.012
          DTC    =      0.0
          DTMO   =      0.0
          DTOO   =      0.0
          TI1S   = (565.0 * 5.0 / 9.0) - (28.45 * 5.0/9.0) * N10/3.0E09
          TO1S   = (565.0 * 5.0 / 9.0) + (28.45 * 5.0/9.0) * N10/3.0E09
          CPHS   = 4992.4097749+ 2.49340775E-04 * EXP(0.04825458 *TO1S)
          CPLS   = 4992.4097749+ 2.49340775E-04 * EXP(0.04825458 *TI1S)
          FRAC   =      0.5 * 5932.2566914 / CPPAV
          MF     =      1.91E05
          T      =      0.0
          TPAVS  =      FRAC * (TO1S - TI1S) + TI1S
          TPAV   =      TPAVS
          X      =      1

   10 CONTINUE

C        HERE THE HEAT TRANSFER CORRELATIONS ARE DERIVED, IN TERMS OF
C        THE DITTIUS-BOELTER CORRELATION.

          CPAV   = 4992.4097749+ 2.49340775E-04 * EXP(0.04825458 *TPAV)
          ROP    =  881.6309    - 2.86514041     * EXP(0.013303415*TPAV)
          VMOD   =         MDOTP  /(5.26 * ROP)
          KP     =      0.7127683- 3.02500000E-03 * EXP(0.01321074 *TPAV)
```

```
               MUP    =    -3.53438229E-07 * TPAV      + 1.9978561E-04
               NUP    =       MUP / ROP
               REP    =       VMOD * D / NUP
               PRP    =       CPAV * NUP * ROP / KP
               NU     =    0.0250 * (REP ** 0.8) * (PRP ** 0.6)
               HP     =       NU   * KP / D
               MM     =       13275.0

C           AS IN GALBA, ARBITRARILY SMALL TIME STEPS ARE SELECTED DURING
C           THE FIRST CYCLE.

 20      IF (X .LE. 10)  GO TO 21
         IF (X .LE. 13)  GO TO 22
         IF (X .LE. 16)  GO TO 23
         IF (X .LE. 20)  GO TO 24
         IF (X .GE. 21)  GO TO 25
 21           H      =    0.000001
              X      =    X + 1
         GO TO 30
 22           H      =    0.00003
              X      =    X + 1
         GO TO 30
 23           H      =    0.0003
              X      =    X + 1
         GO TO 30
 24           H      =    0.001
              X      =    X + 1
         GO TO 30
 25           H      =    0.005

 30      CONTINUE

C           HERE THE THERMAL-HYDRAULICS EQUATIONS PROPER ARE SOLVED.
C           THE NOTEWORTHY FEATURE IS THAT THEY ARE SOLVED FOR CLADDING
C           SURFACE TEMPERATURE CHANGE (DTC) AND MODERATOR TEMPERATURE
C           CHANGE, INSTEAD OF FUEL AND MODERATOR TEMPERATURE CHANGES.
C           LATER, SINCE IT IS ASSUMED THAT AVERAGE FUEL TEMPERATURE
C           CHANGE IS PROPORTIONAL TO CLADDING TEMPERATURE CHANGE,
C           A FUEL TEMPERATURE CHANGE IS COMPUTED AS A FUNCTION OF
C           CLADDING TEMPERATURE CHANGE.

               CPF    =    0.18477 * (TF1 + DTF(K)) + 74.4
               DTC    =  H / (CPF   * MF) * (DN1 - (HP * AR * (DTC -
     +                   DTMO))) + DTC
               IF (K .GE. 56)  GO TO 40
               DTMO   =  H / (CPPAV * MM) * (HP * AR * (DTC - DTMO) -
     +                 (CPPAV * MDOTP * DTOO)) + DTMO
               GO TO 50
 40            DTMO   =  H / (CPPAV * MM) * ((HP * AR * (DTC - DTMO)) -
     +                   (CPPAV * MDOTP    *   (DTOO - DTI(K-29)))) +
     +                   DTMO
 50            T      = T + H

C           HERE, REACTOR OUTLET TEMPERATURE CHANGE IS COMPUTED.

         IF (K .LE. 55)   DTOO   = 2.0 * DTMO
         IF (K .GT. 55)   DTOO   = 2.0 * DTMO - DTI(K-29)
```

```
      IF ((T .LT. 0.2)  .AND.  (NK .EQ. 1))  GO TO 20
         TPAV     =  TPAVS + DTMO
         DTF(K)   =  10.0 * DTC
         DTM(K)   =  DTMO
         DTO(K)   =  DTOO
         H        =  0.2
         T        =  0.0
      RETURN
      END
```

```
C             SIMPLE STEAM GENERATOR MODEL
C
C         This is the subroutine that depicts the simple steam generator
C     model.  As input, it receives moderator temperature change (from
C     the steady-state) data from OTHO (the subroutine dealing with
C     reactor thermal hydraulics), or instructions on what the outlet
C     valve position is to be fixed at, from the main program NERO.
C     It computes reactor inlet temperature changes and steam generator
C     power as output.  Reactor inlet temperature changes are utilized in
C     OTHO to affect overall reactor moderator temperature, and both are
C     displayed as output in the main program, either in graphics or in a
C     table.
C
C         As in the other subroutines, DMTN solves two coupled diffen-
C     tial equations.  Here, the variables solved for are steam generator
C     temperature and steam temperature on the secondary side of the
C     steam generator.  By "steam generator" is meant all of the liquid
C     in the primary side of the steam generator plus all of the material
C     that makes up the steam generator itself.  The finite-difference
C     method is used to solve these differential equations.  In the case
C     of the steam temperature on the secondary side, it is assumed that
C     the control system will, at the end of each time step, automati-
C     cally adjust the flow of the coolant so as to restore the secondary
C     side back to equilibrium conditions.  Thus, the steam temperature
C     from the previous iteration is not used as input during the current
C     iteration.  Time steps in DMTN are 0.2 seconds.
C
C         From steam temperature changes are computed enthalpy changes
C     on the secondary side.  From this, power changes are computed, and
C     from this come secondary side flow changes and total steam genera-
C     tor output.  Reactor inlet temperature changes are computed as a
C     function of primary side temperature changes and reactor outlet
C     temperature changes (which is output into DMTN for OTHO, as
C     explained above).


      SUBROUTINE DMTN


      COMMON    AA, ADTI, ADTO, ALPHF, ALPHM, CPPAV, CS, DN1, DRO,
     +          DTF, DTI, DTM, DTO, DTS, DTSG, EIG, F, H, IS, K, KF,
     +          LND, MDOTP, NK, NN, N10, Q, PJ, PT, RHO, RI, RIR,
     +          RP, STMG, STMGEN, TAU, TAUC, TF1, TM, TT, TTOT, VO,
     +          VOSS, Y, ZZ
      REAL      AA, ALPHF, ALPHM, CPPAV, DN1, DRO, H, MDOTP, N10, RHO,
     +          RIR, STMGEN, TAU, TAUC, TF1, TM1, TTOT, VO, VOSS, Y
      INTEGER   CS, EIG, F, IS, J, K, KF, LND, NK, NN, PJ, Q, RI, X, ZZ
      REAL      ADTI(1510), ADTO(1510), DTF(1510), DTI(1510), DTM(1510),
     +          DTO (1510), DTSG(1510), DTS(1510), RP (1510),  TT(1510),
     +          PT  (1510), STMG(1510), B(6), LMB(6)
      REAL      AR, CPTM, DENTH, DENTC, DHFG, DPS, DTSD,
     +          DTSGD, HS, LMTD, MDOTS, MDOTCA, MDOTCO,
     +          MDOTSF, MMSG, PS, PSS, PWR, PWRCH, TI1, TO1, TOF, TS, TD

C         IN ADDITION TO THE VARIABLES CARRIED THROUGH IN THE COMMON
C     BLOCK, THE FOLLOWING VARIABLES ARE USED IN DMTN:
C
```

```
C                AR     IS THE HEAT TRANSFER AREA.
C                CPTM   IS A COMBINATION TERM.  IT EQUALS THE TOTAL HEAT
C                       CAPACITY TIMES THE TOTAL MASS OF WATER ON THE
C                       SECONDARY SIDE.
C                DENTH  IS THE CHANGE IN ENERGY ON THE SECONDARY SIDE CAUSED
C                       BY A CHANGE IN POWER TRANSFER RATE FROM THE PRIMARY
C                       SIDE.
C                DENTC  IS THE CHANGE IN ENERGY ON THE SECONDARY SIDE FROM
C                       ONE TIME STEP TO THE NEXT CAUSED BY THE "EXTRA POWER
C                       BOOST".  THIS BOOST IS INTRODUCED IN ORDER THAT THE
C                       CHANGE IN STEAM GENERATOR POWER BE TEMPORARILY GREA-
C                       TER THAN REACTOR POWER CHANGE.  THIS IS NECESSARY
C                       BECAUSE IMMEDIATELY AFTER THE PROMPT-JUMP, REACTOR
C                       POWER CHANGE IS GREATER THAN STEAM GENERATOR POWER
C                       CHANGE, AND THE ESTABLISHMENT OF A LONG-TERM ENERGY
C                       BALANCE REQUIRES THAT THE STEAM GENERATOR HAVE THIS
C                       "EXTRA POWER BOOST" IN ORDER TO "CATCH UP".
C                DHFG   IS TOTAL CHANGE IN ENTHALPY FOR A KILOGRAM OF WATER
C                       FROM THE INLET OF THE STEAM GENERATOR (235 C) TO
C                       OUTLET (311.1 C, INCLUDING 33.33 DEGREES OF
C                       SUPERHEAT).
C                DTSD   IS THE CHANGE OF TEMPERATURE ON THE SECONDARY SIDE.
C                       USED ONLY DURING THE FIRST 0.2 SECONDS.
C                DTSGD  IS THE CHANGE OF TEMPERATURE ON THE PRIMARY   SIDE.
C                       USED ONLY DURING THE FIRST 0.2 SECONDS.
C                HS     IS THE HEAT TRANSFER COEFFICIENT.  IT IS A FUNCTION
C                       OF MDOTS ** 0.806.
C                LMTD   IS THE LOGARITHMIC MEAN TEMPERATURE DIFFERENCE
C                       BETWEEN THE PRIMARY SIDE AND THE SECONDARY SIDE.
C                MDOTS  IS THE MASS FLOW RATE ON THE SECONDARY SIDE.
C                MDOTCA IS THE CHANGE IN MASS FLOW RATE ON THE SECONDARY
C                       SIDE.
C                MDOTCO IS THE VALUE FOR MDOTCA DURING THE PREVIOUS
C                       ITERATION.
C                MMSG   IS THE MASS OF WATER IN THE PRIMARY SIDE OF THE
C                       STEAM GENERATOR.
C                PWR    IS CURRENT POWER OUTPUT OF THE STEAM GENERATOR,
C                       INCLUDING THE "EXTRA POWER BOOST".
C                PWRCH  IS THE POWER ADDITION TO STEAM GENERATOR OUTPUT DUE
C                       TO THE "EXTRA POWER BOOST".
C                TD     IS THE DIFFERENCE BETWEEN INLET AND OUTLET
C                       TEMPERATURES.
C                TOF    IS THE OUTLET TEMPERATURE THAT THE REACTOR WOULD
C                       HAVE UNDER STEADY-STATE CONDITIONS AT A GIVEN POWER
C                       LEVEL.
C                TS     IS THE DIFFERENCE BETWEEN THE OUTLET TEMPERATURE
C                       THAT THE REACTOR WOULD HAVE UNDER STEADY-STATE
C                       CONDITIONS AT A GIVEN POWER LEVEL AND THE ACTUAL
C                       TEMPERATURE.



          IF (K .NE. 2)  GO TO 10
C            THIS STATEMENT WILL JUMP THE PROGRAM OVER THE INITIALIZATIONS
C      AT ALL TIMES EXCEPT DURING THE FIRST 0.2 SECONDS.

C            HERE ARE THE INITIALIZATIONS.
```

```
              AR      =  22400.0
              CPTM    =  2.545E08
              DENTH   =  0.0
              DHFG    =  1901744.167
              DTSD    =  0.0
              DTSGD   =  0.0
              DTU     =  0.0
              MDOTS   =  N10 / DHFG
              MDOTSF  =  1577.720759
              MDOTCA  =  0.0
              MDOTCO  =  0.0
              MMSG    =  43000.0
              PSS     =  6205550.04
              PWR     =  N10
              PWRCH   =  0.0
              TI1     =-(28.45*5.0/9.0) * N10 / 3.0E09 + (565.0*5.0/9.0)
              TO1     = (28.45*5.0/9.0) * N10 / 3.0E09 + (565.0*5.0/9.0)
              TD      =  TO1 - TI1
              WDOTS   =  MDOTS
              X       =  1


   10   CONTINUE

C           SINCE THERE IS NO POINT IN EXECUTING THE MAIN BODY OF THE
C      PROGRAM AS LONG AS THE DELAYED REACTOR OUTPUT TEMPERATURES ARE NOT
C      ARRIVING AT THE STEAM GENERATOR, THESE STATEMENTS DEFINE NORMAL
C      DMTN OUTPUT QUANTITIES AS ZERO, AND THEN SKIP OVER THE MAIN BODY OF
C      DMTN TO THE RETURN STATEMENT.  THIS WILL CONTINUE UNTIL REACTOR
C      OUTPUT STARTS ARRIVING.  HOWEVER, IF A THROTTLE VALVE PERTURBATION
C      IS SPECIFIED, NONE OF THIS APPLIES.

           IF (K .LE. 26)  DTS (K)    =  0.0
           IF (K .LE. 26)  DTSG(K)    =  0.0
           IF (K .LE. 26)  DTI (K)    =  0.0
           IF (K .LE. 26)  GO TO 50

   15   CONTINUE

C           HERE, QUANTITIES SUCH AS TOF, HS, LMTD, ARE ALL DEFINED AND
C      REDEFINED EVERY 0.2 SECONDS.

            TOF      =  ((28.45 * 5.0 * RP(K-25)) / (9.0 * 3.0E09)) +
          +                 (565.0 * 5.0 / 9.0)
            TS       =   (TO1 + DTO(K-25)) - TOF

C           DUE TO THE FACT THAT POWER JUMPS IN THE REACTOR CAN OCCUR MUCH
C      MORE QUICKLY THAN THE RESULTING TEMPERATURE CHANGES, IT IS POSSIBLE
C      THAT THE VARIABLE TS CAN ASSUME AN OPPOSITE SIGN FROM THE POWER
C      CHANGE.  SINCE THIS DOES NOT HAPPEN IN THE "REAL' WORLD", THIS
C      COMMAND WILL AUTOMATICALLY RESET TS TO EQUAL ZERO IF THIS SITUATION
C      ARISES EARLY IN THE RUN.

           IF ((K .LE. 35)   .AND.   (((DTO(K-25).GT.0.0).AND.(TS.LT.0.0))
          +                   .OR.    ((DTO(K-25).LT.0.0).AND.(TS.GT.0.0))))
```

```
     + TS      =  0.0
       LMTD    = ((TO1 + DTO(K-25) - 311.1)  - (TI1 + DTI(K-1)-235.0))
     +              / (LOG((TO1 + DTO(K-25) - 311.1) / (TI1 + DTI(K-1) -
     +                 235.0)))

C          IF ZZ = 1, THE STEAM GENERATOR VALVE OPENING FRACTION IS
C     PERMENENTLY FIXED.  BASED SOLELY UPON THE RESULTING MASS FLOW RATE,
C     LMTD AND HS WILL BE COMPUTED, AND SO WILL THE REACTOR INLET TEMP-
C     ERATURE.  THIS WILL TEST LOAD FOLLOWING WITHIN THE REACTOR.

        IF (ZZ .EQ. 1)    MDOTS = VO * MDOTSF
          HS      =  9.72604376 * ((MDOTS + MDOTCA) ** 0.806)


        IF (X .GE. 20)  H  =  0.2

C          HERE, ARBITRARILY SHORT TIME STEPS ARE TAKEN DURING THE FIRST
C     0.2 SECONDS, FOR THE SAME REASON AS IN GALBA.

 20     IF (K .NE. 26)  GO TO 30
        IF (X .LE. 10)  GO TO 21
        IF (X .LE. 19)  GO TO 22
        IF (X .LE. 20)  GO TO 23
 21       H         =   0.001
          X         =   X + 1
        GO TO 25
 22       H         =   0.01
          X         =   X + 1
        GO TO 25
 23       H         =   0.1
          X         =   X + 1


C          HERE, THE COUPLED EQUATIONS ARE SOLVED AND THE OUTPUT QUANTI-
C     TIES SUCH AS POWER AND REACTOR INLET TEMPERATURE ARE COMPUTED.  THE
C     NUMBER 50600.0 ARISES FROM THE FACT THAT EVERY DEGREE TEMPERATURE
C     CHANGE ON THE SECONDARY SIDE REQUIRES 50600.0 J OF ENERGY PER KG.

 25       DTSGD    =   (H / (MMSG * CPPAV)) * (((CPPAV * MDOTP) *
     +                 (DTO(K-25) + TD)) - (HS * AR * (DTSGD + LMTD +
     +             -    DTSD)))    +   DTSGD
          DTSD     =   (H / CPTM) *  ((HS * AR * (DTSGD + LMTD - DTSD))
     +             - ((MDOTS + MDOTCA) * DHFG))
          DENTH    =    50600.0 * DTSD
          DENTC    =    50600.0 * H * (MDOTCA - MDOTCO) * DHFG / CPTM
          TDENTH   =    DENTC  +  TDENTH
          PWRCH    =    MDOTCA * (TDENTH + DHFG)
          PWR      =    MDOTS  * (DENTH  + DHFG) + PWRCH
          PS       =    5509154.5 + (77756.49 * DTSD)  +  (538137557.7 /
     +                 (772.747 - (1.006 * DTSD)))
          DPS      =    PS  - PSS
          DVO      =    DPS / PSS
          VO       =    VOSS  + DVO
          MDOTS    =    VO  * MDOTSF
        IF (X .LE. 20) GO TO 20
          DTSG(K)  =    DTSGD
```

```
          DTS (K)  =   DTSD
        GO TO 40

C          HERE, THE COUPLED EQUATIONS ARE SOLVED AND THE OUTPUT QUANTI-
C     TIES SUCH AS POWER AND REACTOR INLET TEMPERATURE ARE COMPUTED.  THE
C     NUMBER 50600.0 ARISES FROM THE FACT THAT EVERY DEGREE TEMPERATURE
C     CHANGE ON THE SECONDARY SIDE REQUIRES 50600.0 J OF ENERGY PER KG.

 30       DTSG(K)  =  (H / (MMSG * CPPAV)) * (((CPPAV * MDOTP) *
     +                (DTO(K-25) + TD - DTI(K-1))) - (HS * AR * (DTSG(K-1)
     +             +   LMTD - DTS(K-1)))) + DTSG(K-1)
          DTS (K)  =  (H /   CPTM) * ((HS * AR * (DTSG(K-1) + LMTD))
     +             -  (MDOTS + MDOTCA) * DHFG)
          DENTH    =   50600.0 * DTS(K)
          DENTC    =   50600.0 * H * (MDOTCA - MDOTCO) * DHFG / CPTM
          TDENTH   =   DENTC  +  TDENTH
          PWRCH    =   MDOTCA * (TDENTH + DHFG)
          PWR      =   MDOTS  * (DENTH  + DHFG) + PWRCH
          IF (ZZ .EQ. 1)  GO TO 40
          PS       =   5509154.5 + (77756.49 * DTS(K-1))  + (538137557.7/
     +                (772.747 - (1.006 * DTS(K-1))))
          DPS      =   PS  - PSS
          DVO      =   DPS / PSS
          VO       =   VO  + DVO
          MDOTS    =   VO  * MDOTSF
          MDOTS    =  (PWR - PWRCH) / DHFG
 40       DTI (K)  =  2.0 * DTSG(K) - DTO(K-25)
          STMGEN   =  MDOTS * (DENTH + DHFG) + PWRCH
          MDOTCO   =   MDOTCA
          MDOTCA   =   49.9035 * TS
        IF (ZZ .EQ. 1)  MDOTCA = 0.0
 50     RETURN
        END
```

```
C          GRAPHICS SUBROUTINE
C
C          THIS IS THE SUBROUTINE THAT DOES THE GRAPHICS.  THERE ARE
C     THREE DISPLAYS, AND THE USER CAN SELECT EACH OF THEM SUCCESSIVELY.
C     ANY OF THE DISPLAYS CAN ALSO BE ENLARGED.  THE FIRST DISPLAY
C     DEPICTS POWER LEVELS -- THAT OF THE REACTOR AND THAT OF THE STEAM
C     STEAM GENERATOR.  THE SECOND DISPLAY SHOWS FUEL TEMPERATURE, AND
C     THE THIRD SHOWS CHANGE IN REACTOR OUTLET AND INLET TEMPRATURES,
C     AND CHANGE IN AVERAGE MODERATOR TEMPERATURE, FROM THE STEADY-STATE.
C     A HARD COPY CAPABILITY ALSO EXISTS, AND THIS SUBROUTINE (CALLED
C     VESPASIAN) ALLOWS THE USER TO SELECT THIS AT HIS OR HER DISCRETION.
C     THE GRAPHICS SYSTEM USED IS THE PLOT 10 SYSTEM OF THE INTERGRAPHICS
C     LIBRARY.  MOST OF THE BULK OF VESPASIAN CONCERNS ITSELF WITH
C     THE "PACKAGING" OF THE OUTPUT.  THE OUTPUT ITSELF COMES FROM THE
C     "CALL POLY" STATEMENTS, WHICH DRAW THE ARRAYS INTO WHICH THE OUTPUT
C     WAS PREVIOUSLY ARRANGED BY NERO AND THE OTHER SUBROUTINES.

       SUBROUTINE VESPASIAN
       COMMON    AA, ADTI, ADTO, ALPHF, ALPHM, CPPAV, CS, DN1, DRO,
     +           DTF, DTI, DTM, DTO, DTS, DTSG, EIG, F, H, IS, K, KF,
     +           LND, MDOTP, NK, NN, N10, Q, PJ, PT, RHO, RI, RIR,
     +           RP, STMG, STMGEN, TAU, TAUC, TF1, TM, TT, TTOT, VO,
     +           VOSS, Y, ZZ
       REAL      AA, ALPHF, ALPHM, CPPAV, DN1, DRO, H, MDOTP, N10, RHO,
     +           RIR, STMGEN, TAU, TAUC, TF1, TM1, TTOT, VO, VOSS, Y
       INTEGER   CS, DE, EIG, F, IS, J, K, KF, LND, NK, NN, PJ, Q, RI,
     +           PL, INT1, INT2, INT3, INT4, INT5, INT6, REP, TY, ZZ
       REAL      ER, LOWF, LOWP, LOWT, MIDF, MIDP, MIDT, TNX, TNY
       REAL      RSHA, LXF, LXP, LXT, HXF, HXP, HXT, MDF, MDP, MDT, TSF,
     +           TSP, TST
       REAL      ADTI(1510), ADTO(1510), DTF(1510), DTI(1510), DTM(1510),
     +           DTO (1510), DTSG(1510), DTS(1510), RP (1510), TT (1510),
     +           PT  (1510), STMG(1510)
       REAL      DTIV(1540), TMPF(1510)
       DIMENSION POST(9), NEGT(9), TIME(4),
     +            FT (16), HIFT(9), MDFT(9), TMC(19), TOC(19), TIC(19),
     +            LPWR(7), MPWR(7), ZERO(1),  TEN(2), TWNT(2), THRT(2),
     +            FORT(2), FFTY(2), SIXT(2), DN1G(22), STM(22),   SP(1),
     +              MW(2),  DEG(5)
       DATA       POST /'+', '1', '5', ' ', 'D', 'E', 'G', ' ', 'C'/,
     +            NEGT /'-', '1', '5', ' ', 'D', 'E', 'G', ' ', 'C'/,
     +            TIME /'T', 'I', 'M', 'E'/,     ZERO /'0'/,
     +            TMC  /'M', 'O', 'D', ' ', ' ', ' ', ' ', 'T', 'E',
     +                  'M', 'P', ' ', 'C', 'H', 'A', 'N', 'G', 'E',
     +                  ':'/,
     +            TOC  /'O', 'U', 'T', 'L', 'E', 'T', ' ', 'T', 'E',
     +                  'M', 'P', ' ', 'C', 'H', 'A', 'N', 'G', 'E',
     +                  ':'/,
     +            TIC  /'I', 'N', 'L', 'E', 'T', ' ', ' ', 'T', 'E',
     +                  'M', 'P', ' ', 'C', 'H', 'A', 'N', 'G', 'E',
     +                  ':'/,
     +            FT   /'F', 'U', 'E', 'L', ' ', 'T', 'E', 'M', 'P',
     +                  'E', 'R', 'A', 'T', 'U', 'R', 'E'/,
     +            HIFT /'6', '0', '0', ' ', 'D', 'E', 'G', ' ', 'C'/,
     +            MDFT /'4', '5', '0', ' ', 'D', 'E', 'G', ' ', 'C'/
       DATA       LPWR /'1', '0', '0', '0', ' ', 'M', 'W'/,
     +            MPWR /'2', '0', '0', '0', ' ', 'M', 'W'/,
```

```
     +              ZERO /'0'/,         TEN/'0', '0'/, TWNT/'2', '0'/,
     +              THRT /'3', '0'/, FORT/'4', '0'/, FFTY/'5', '0'/,
     +              SIXT /'6', '0'/,    MW/'M', 'W'/,    SP/' '/,
     +              DEG  /'D', 'E', 'G', ' ', 'C'/,
     +              DN1G /'R', 'E', 'A', 'C', 'T', 'O', 'R', ' ', ' ',
     +                    ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'P', 'O',
     +                    'W', 'E', 'R', ':'/,
     +              STM  /'S', 'T', 'M', '.', ' ', 'G', 'E', 'N', '.',
     +                    'O', 'U', 'T', 'P', 'U', 'T', ' ', 'P', 'O',
     +                    'W', 'E', 'R', ':'/

         TI1  =  (565.0*5.0/9.0) - (28.45*5.0/9.0) * (N10 / 3.0E09)
         TO1  =  (565.0*5.0/9.0) + (28.45*5.0/9.0) * (N10 / 3.0E09)
         REP  =   2
         ER   =   1.0
         MDT  =   0.0
         LXP  =   0.0
         HXP  =   3.0E09
         LXF  =  300.0
         HXF  =  720.0
         LXT  =  -25.0
         HXT  =   25.0
          INT1 =  LND / 30.0
          INT2 =  LND / 15.0
          INT3 =  LND / 10.0
          INT4 =  LND /  7.5
          INT5 =  LND /  6.0
          INT6 =  LND /  5.0
          RSHA =  LND /  5.0
       DO 10 I = 1,LND
       TMPF(I) =   TF1 + DTF(I)
 10    CONTINUE

 20    WRITE  (6, 30)
 30    FORMAT (X, '                  YOU HAVE SELECTED GRAPHICS.  SINCE THIS
     + IS SO, THERE ARE A FEW')
       WRITE  (6, 40)
 40    FORMAT (X, '                  THINGS YOU MUST KNOW.  FIRST, THREE
     + DIFFERENT PLOTTING WINDOWS WILL')
       WRITE  (6, 50)
 50    FORMAT (X, '                  BE SUCCESSIVELY DISPLAYED.  TO TRANSFER
     + FROM ONE WINDOW TO ANOTHER,')
       WRITE  (6, 60)
 60    FORMAT (X, '                  PRESS <RET>.  ANY TIME THE "ACTION"
     + STOPS, IT CAN BE RESUMED BY')
       WRITE  (6, 70)
 70    FORMAT (X, '                  PRESSING <RET>.  YOU CAN ALSO, AFTER THE
     + INITIAL DISPLAY, EXPAND THE')
       WRITE  (6, 80)
 80    FORMAT (X, '                  DISPLAY IN ORDER TO SHOW CONTRASTS
     + BETTER.  SECOND, IN ORDER TO')
       WRITE  (6, 90)
 90    FORMAT (X, '                  CLEAR THE GRAPHICS DISPLAYS AFTER THE
     + PROGRAM STOPS EXECUTING AND')
       WRITE  (6, 100)
 100   FORMAT (X, '                  TO RESTORE SCROLLING CAPABILITY, WHEN
     + THE PROGRAM RETURNS TO THE')
```

```
      WRITE  (6, 110)
  110 FORMAT (X, '                     COMMAND MODE (AFTER THE "FORTRAN STOP"
     + MESSAGE APPEARS ON THE')
      WRITE  (6, 120)
  120 FORMAT (X, '                     SCREEN), WRITE IN THE STATEMENT "@
     + GRAPHICS:CLEAR" (WITHOUT')
      WRITE  (6, 130)
  130 FORMAT (X, '                     THE QUOTATION MARKS).  YOU MUST BE ON A
     + TEKTRONICS TERMINAL.')
      WRITE  (6, 140)
  140 FORMAT ('0', '                     THIRD:  DO YOU WANT 4662 COPY
     + CAPABILITY OR NOT?  IF SO, TYPE IN 1;')
      WRITE  (6, 150)
  150 FORMAT (X, '                     IF NOT, 2')
      READ 160, PL
  160 FORMAT (I1)

      IF ((PL .EQ. 1)  .OR.  (PL .EQ. 2))  GO TO 190

      WRITE  (6, 170)
  170 FORMAT (X, '                     YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 180)
  180 FORMAT (X, '                     TO TRY AGAIN.')

      GO TO 20

  190 CONTINUE
  200 WRITE  (6, 210)
  210 FORMAT ('0', '                     AS CURRENTLY SET UP, REACTOR INLET
     + TEMPERATURE IS PLOTTED AS OF THE')
      WRITE  (6, 220)
  220 FORMAT (X, '                     MOMENT IT IS CREATED IN THE STEAM
     + GENERATOR.  HOWEVER, A GIVEN INLET')
      WRITE  (6, 230)
  230 FORMAT (X, '                     TEMPERATURE CHANGE DOES NOT AFFECT
     + REACTOR TEMPERATURES UNTIL AFTER SIX')
      WRITE  (6, 240)
  240 FORMAT (X, '                     SECONDS AFTER ITS CREATION.  IN ORDER
     + TO DEPICT REACTOR TEMPERATURE')
      WRITE  (6, 250)
  250 FORMAT (X, '                     CHANGES AT THE SAME TIME THAT THE INLET
     + TEMPERATURE CHANGES CAUSING')
      WRITE  (6, 260)
  260 FORMAT (X, '                     THEM ENTER THE REACTOR, TYPE IN 1.
     + OTHERWISE THE INLET TEMPERATURE')
      WRITE  (6, 270)
  270 FORMAT (X, '                     CHANGES WILL BE PLOTTED AS OF THE MOMENT
     + OF THEIR CREATION, WHICH')
      WRITE  (6, 280)
  280 FORMAT (X, '                     IS 6 SECONDS BEFORE THEY ARRIVE AT THE
     + INLET.  FOR THIS OPTION,')
      WRITE  (6, 290)
  290 FORMAT (X, '                     TYPE IN 2.')

      READ 300, DE
  300 FORMAT (I1)
```

```
      IF ((DE .EQ. 1)  .OR.  (DE .EQ. 2))  GO TO 330

      WRITE  (6, 310)
310   FORMAT (X, '              YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 320)
320   FORMAT (X, '              TO TRY AGAIN')

330   CONTINUE

      IF (DE .EQ. 2)  GO TO 350

      DO 340 I =  1,LND
      DTIV(I+29)= DTI(I)
340   CONTINUE
      DO 350 I =  1,30
      DTIV(I)  =  0.0
350   CONTINUE
      DO 355 I =  1,LND
      DTI(I)   =  DTIV(I)
355   CONTINUE


      IF (PL .EQ. 1)  GO TO 360
      CALL GRSTRT (4051,1)
      CALL NEWPAG
      GO TO 370

360   CALL GRSTRT (4662,1)
370   CALL VWPORT (0.0, 130.0,   0.0, 100.0)
      CALL WINDOW (0.0, 130.0,   0.0, 100.0)
      CALL MOVE (  0.0,  25.0)
      CALL DRAW (120.0,  25.0)
      CALL DRAW (120.0, 100.0)
      CALL DRAW (  0.0, 100.0)
      CALL DRAW (  0.0,  75.0)
      CALL DRAW (  3.0,  75.0)
      CALL MOVE (  6.0,  75.0)
      CALL TXICUR   (4)
      IF (REP .EQ. 1)  GO TO 380
      CALL TEXT (    7,  MPWR)
      GO TO 390
380   CALL RNUMBR (MIDP,    1,      6)
      CALL TEXT (    1,     SP)
      CALL TEXT (    2,     MW)
390   CALL DASHPT   (9)
      CALL MOVE ( 18.0,  75.0)
      CALL DRAW (120.0,  75.0)
      CALL MOVE (  0.0,  75.0)
      CALL DASHPT   (0)
      CALL DRAW (  0.0,  50.0)
      CALL DRAW (  3.0,  50.0)
      CALL MOVE (  6.0,  50.0)
      IF (REP .EQ. 1)  GO TO 400
      CALL TEXT (    7,  LPWR)
      GO TO 410
```

```
  400 CALL RNUMBR (LOWP,      1,        6)
      CALL TEXT (      1,     SP)
      CALL TEXT (      2,     MW)
  410 CALL MOVE (  18.0,   50.0)
      CALL DASHPT    (9)
      CALL DRAW (120.0,   50.0)
      CALL MOVE (   0.0,   50.0)
      CALL DASHPT    (0)
      CALL DRAW (   0.0,   25.0)
      CALL TXICUR    (2)
      CALL MOVE (  20.0,   25.0)
      CALL INUMBR (INT1,      2)
      CALL MOVE (  40.0,   25.0)
      CALL INUMBR (INT2,      3)
      CALL MOVE (  60.0,   25.0)
      CALL INUMBR (INT3,      3)
      CALL MOVE (  80.0,   25.0)
      CALL INUMBR (INT4,      3)
      CALL MOVE (100.0,   25.0)
      CALL INUMBR (INT5,      3)
      CALL TXICUR    (3)
      CALL MOVE (120.0,   25.0)
      CALL INUMBR (INT6,      3)
      CALL TXICUR    (4)
      CALL MOVE (   0.0,   20.0)
      CALL TEXT (     22,   DN1G)
      CALL MOVE (  45.0,   20.0)
      CALL DRAW (120.0,   20.0)
      CALL MOVE (   0.0,   16.0)
      CALL TEXT (     22,    STM)
      CALL MOVE (  45.0,   16.0)
      CALL DASHPT    (3)
      CALL DRAW (120.0,   16.0)
      CALL VWPORT (0.0, 120.0,   25.0, 100.0)
      CALL WINDOW (0.0,   RSHA,    LXP,     HXP)
      CALL MOVE    (0.0,      N10)
      CALL POLY (LND, TT, STMG)
      CALL DASHPT    (0)
      TNX   =   RSHA / 6.0
      TY    =   LND  / 6
      TNY   =   STMG (TY)
      CALL MOVE   (TNX,                        (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW   (TNX,                        (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW   (TNX,                        (TNY - (ER * 6.0E07)))
      TNX   =   RSHA / 3.0
      TY    =   LND  / 3
      TNY   =   STMG (TY)
      CALL MOVE   (TNX,                        (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW   (TNX,                        (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW   (TNX,                        (TNY - (ER * 6.0E07)))
      TNX   =   RSHA / 2.0
      TY    =   LND  / 2
      TNY   =   STMG (TY)
```

```
      CALL MOVE  (TNX,                        (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                        (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                        (TNY - (ER * 6.0E07)))
      TNX  =  RSHA * 2.0 / 3.0
      TY   =  LND  * 2   / 3
      TNY  =  STMG (TY)
      CALL MOVE  (TNX,                        (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                        (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                        (TNY - (ER * 6.0E07)))
      TNX  =  RSHA * 5.0 / 6.0
      TY   =  LND  * 5   / 6
      TNY  =  STMG (TY)
      CALL MOVE  (TNX,                        (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                        (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                        (TNY - (ER * 6.0E07)))
      TNX  =  RSHA / 6.0
      TY   =  LND  / 6
      TNY  =  PT (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      TNX  =  RSHA / 3.0
      TY   =  LND  / 3
      TNY  =  PT (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      TNX  =  RSHA / 2.0
      TY   =  LND  / 2
      TNY  =  PT (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      TNX  =  RSHA * 2.0 / 3.0
      TY   =  LND  * 2   / 3
      TNY  =  PT (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      TNX  =  RSHA * 5.0 / 6.0
      TY   =  LND  * 5   / 6
      TNY  =  PT (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
      CALL MOVE (  0.0,    N10)
```

```
              CALL POLY (LND, TT,    PT)

              IF (PL .EQ. 1)   GO TO 420
              CALL CMCLOS
              READ*
              CALL CMOPEN
              CALL NEWPAG
              GO TO 460
        420   CALL GRSTOP
              WRITE  (6, 430)
        430   FORMAT (X, '                YOU MUST NOW REMOVE THE COMPLETED
             + GRAPHICS DISPLAY, INSERT A NEW')
              WRITE  (6, 440)
        440   FORMAT (X, '                SHEET OF PAPER, AND TOTALLY RESET THE
             + 4662 PLOTTING MACHINE.')
              WRITE  (6, 450)
        450   FORMAT (X, '                PRESS <RET> WHEN THIS IS DONE.')
              READ*
        460   IF (REP .EQ. 1)  GO TO 540
        470   WRITE  (6, 480)
        480   FORMAT (X, '                DO YOU WISH TO EXPAND THIS DISPLAY?
             + IF SO, TYPE IN 1;  IF NOT, 2')
              READ 490, REP
        490   FORMAT (I1)
              IF (REP .EQ. 1)  GO TO 520
              IF (REP .EQ. 2)  GO TO 540
              WRITE  (6, 500)
        500   FORMAT (X, '                YOU HAVE TYPED IN A NUMBER THAT CANNOT
             + BE UTILIZED.  YOU WILL HAVE')
              WRITE  (6, 510)
        510   FORMAT (X, '                TO TRY AGAIN.')
              GO TO 470
        520   TSP  =   PT(LND) - PT(1)
              TSP  =   TSP ** 2.0
              TSP  =   SQRT(TSP)
              IF (PT(1) .LE. PT(LND))  MDP = (0.5 * TSP) + N10
              IF (PT(1) .GT. PT(LND))  MDP = (0.5 * TSP) + PT(LND)
              ER    =   1.2 * TSP   / 3.0E09
              LXP   =-(0.6 * TSP) + MDP
              HXP   = (0.6 * TSP) + MDP
              LOWP = (0.4 * TSP) + LXP
              MIDP = (0.8 * TSP) + LXP
              IF (NK .EQ. 1)  ER    =  ER * 2.0
              IF (ZZ .EQ. 1)  ER    =  ER * 4.0
              IF (NK .EQ. 1)  LXP   =-(1.2 * TSP) + MDP
              IF (NK .EQ. 1)  HXP   = (1.2 * TSP) + MDP
              IF (NK .EQ. 1)  LOWP = (0.8 * TSP) + LXP
              IF (NK .EQ. 1)  MIDP = (1.6 * TSP) + LXP
              IF (ZZ .EQ. 1)  LXP   =-(2.4 * TSP) + MDP
              IF (ZZ .EQ. 1)  HXP   = (2.4 * TSP) + MDP
              IF (ZZ .EQ. 1)  LOWP = (1.6 * TSP) + LXP
              IF (ZZ .EQ. 1)  MIDP = (3.2 * TSP) + LXP
              LOWP = LOWP / 1.0E06
              MIDP = MIDP / 1.0E06
              IF (PL .EQ. 2)  GO TO 530
              GO TO 360
        530   CONTINUE
```

```
      GO TO 370
540   REP  =  2
      ER   =  1.0
      IF (PL .EQ. 1)  GO TO 550
      GO TO 560

550   CALL GRSTRT (4662, 1)
560   CALL VWPORT (0.0, 130.0,    0.0, 100.0)
      CALL WINDOW (0.0, 130.0,    0.0, 100.0)
      CALL DASHPT     (0)
      CALL MOVE (   0.0,   25.0)
      CALL DRAW (120.0,   25.0)
      CALL DRAW (120.0,  100.0)
      CALL DRAW (   0.0,  100.0)
      CALL DRAW (   0.0,   25.0)
      CALL MOVE (  20.0,   25.0)
      CALL DRAW (  20.0,   28.0)
      CALL MOVE (  40.0,   28.0)
      CALL DRAW (  40.0,   25.0)
      CALL MOVE (  60.0,   25.0)
      CALL DRAW (  60.0,   30.0)
      CALL MOVE (  80.0,   28.0)
      CALL DRAW (  80.0,   25.0)
      CALL MOVE (100.0,   25.0)
      CALL DRAW (100.0,   28.0)
      CALL MOVE (   0.0,   24.0)
      CALL TXICUR     (7)
      CALL TEXT (      1,   ZERO)
      CALL TXICUR     (8)
      CALL MOVE (   7.0,   24.0)
      CALL TEXT (      4,   TIME)
      CALL MOVE (  20.0,   24.0)
      CALL INUMBR (INT1,      2)
      CALL MOVE (  40.0,   24.0)
      CALL INUMBR (INT2,      3)
      CALL MOVE (  60.0,   24.0)
      CALL INUMBR (INT3,      3)
      CALL MOVE (  80.0,   24.0)
      CALL INUMBR (INT4,      3)
      CALL MOVE (100.0,   24.0)
      CALL INUMBR (INT5,      3)
      CALL MOVE (120.0,   24.0)
      CALL TXICUR     (9)
      CALL INUMBR (INT6,      3)
      CALL MOVE (   0.0,   78.571)
      CALL DRAW (   3.0,   78.571)
      CALL TXICUR     (4)
      CALL MOVE (   6.0,   87.0)
      CALL TEXT (      9,     FT)
      CALL MOVE (   6.0,   78.571)
      IF (REP .EQ. 1)  GO TO 570
      CALL TEXT (      9,   HIFT)
      GO TO 580
570   CALL RNUMBR (MIDF,      1,      5)
      CALL TEXT (      5,   DEG)
580   CALL MOVE (  30.0,   78.571)
      CALL DASHPT     (9)
```

```
      CALL DRAW (120.0,  78.571)
      CALL MOVE (  0.0,  51.786)
      CALL DASHPT   (0)
      CALL DRAW (  3.0,  51.786)
      CALL MOVE (  6.0,  51.786)
      IF (REP .EQ. 1)  GO TO 590
      CALL TEXT (     9,  MDFT)
      GO TO 600
  590 CALL RNUMBR (LOWF,      1,       5)
      CALL TEXT (      5,  DEG)
  600 CALL MOVE ( 30.0,  51.786)
      CALL DASHPT   (9)
      CALL DRAW (120.0,  51.786)
      CALL VWPORT (0.0,  120.0,   25.0,  100.0)
      CALL WINDOW (0.0,  RSHA,    LXF,     HXF)
      CALL DASHPT   (0)
      TNX  =  RSHA / 6.0
      TY   =  LND  / 6
      TNY  =  TMPF (TY)
      CALL MOVE   (TNX,                            (TNY - (ER * 8.4)))
      CALL DRAW ((TNX + (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY + (ER * 8.4)))
      CALL DRAW ((TNX - (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY - (ER * 8.4)))
      TNX  =  RSHA / 3.0
      TY   =  LND  / 3
      TNY  =  TMPF (TY)
      CALL MOVE   (TNX,                            (TNY - (ER * 8.4)))
      CALL DRAW ((TNX + (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY + (ER * 8.4)))
      CALL DRAW ((TNX - (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY - (ER * 8.4)))
      TNX  =  RSHA / 2.0
      TY   =  LND  / 2
      TNY  =  TMPF (TY)
      CALL MOVE   (TNX,                            (TNY - (ER * 8.4)))
      CALL DRAW ((TNX + (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY + (ER * 8.4)))
      CALL DRAW ((TNX - (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY - (ER * 8.4)))
      TNX  =  RSHA * 2.0 / 3.0
      TY   =  LND  * 2   / 3
      TNY  =  TMPF (TY)
      CALL MOVE   (TNX,                            (TNY - (ER * 8.4)))
      CALL DRAW ((TNX + (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY + (ER * 8.4)))
      CALL DRAW ((TNX - (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY - (ER * 8.4)))
      TNX  =  RSHA * 5.0 / 6.0
      TY   =  LND  * 5   / 6
      TNY  =  TMPF (TY)
      CALL MOVE   (TNX,                            (TNY - (ER * 8.4)))
      CALL DRAW ((TNX + (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY + (ER * 8.4)))
      CALL DRAW ((TNX - (0.01250 * RSHA)),  TNY)
      CALL DRAW   (TNX,                            (TNY - (ER * 8.4)))
      CALL MOVE (  0.0,    TF1)
```

```
      CALL DASHPT   (5)
      CALL POLY (LND, TT, TMPF)

      IF (PL .EQ. 1)   GO TO 610
      CALL CMCLOS
      READ*
      CALL CMOPEN
      CALL NEWPAG
      GO TO 650
 610  CALL GRSTOP
      WRITE  (6, 620)
 620  FORMAT (X, '              YOU MUST NOW REMOVE THE COMPLETED
     + GRAPHICS DISPLAY, INSERT A NEW')
      WRITE  (6, 630)
 630  FORMAT (X, '              SHEET OF PAPER, AND TOTALLY RESET THE
     + 4662 PLOTTING MACHINE.')
      WRITE  (6, 640)
 640  FORMAT (X, '              PRESS <RET> WHEN THIS IS DONE.')
      READ*
 650  IF (REP .EQ. 1)  GO TO 730
 660  WRITE  (6, 670)
 670  FORMAT (X, '              DO YOU WISH TO EXPAND THIS DISPLAY?
     + IF SO, TYPE IN 1;  IF NOT, 2')
      READ 680, REP
 680  FORMAT (I1)
      IF (REP .EQ. 1)  GO TO 710
      IF (REP .EQ. 2)  GO TO 730
      WRITE  (6, 690)
 690  FORMAT (X, '              YOU HAVE TYPED IN A NUMBER THAT CANNOT
     + BE UTILIZED.  YOU WILL HAVE')
      WRITE  (6, 700)
 700  FORMAT (X, '              TO TRY AGAIN.')
      GO TO 660
 710  TSF  =  DTF(LND) - DTF(1)
      TSF  =  TSF ** 2.0
      TSF  =  SQRT (TSF)
      IF (DTF(1) .LE. DTF(LND))  MDF = (0.5 * TSF) + TF1
      IF (DTF(1) .GT. DTF(LND))  MDF =-(0.5 * TSF) + TF1
      ER   =  1.2 * TSF  / 420.0
      LXF  =-(0.6 * TSF) + MDF
      HXF  = (0.6 * TSF) + MDF
      LOWF = (0.4 * TSF) + LXF
      MIDF = (0.8 * TSF) + LXF
      IF (NK .EQ. 1)  ER   =  ER * 2.0
      IF (ZZ .EQ. 1)  ER   =  ER * 4.0
      IF (NK .EQ. 1)  LXF  =-(1.2 * TSF) + MDF
      IF (NK .EQ. 1)  HXF  = (1.2 * TSF) + MDF
      IF (NK .EQ. 1)  LOWF = (0.8 * TSF) + LXF
      IF (NK .EQ. 1)  MIDF = (1.6 * TSF) + LXF
      IF (ZZ .EQ. 1)  LXF  =-(2.4 * TSF) + MDF
      IF (ZZ .EQ. 1)  HXF  = (2.4 * TSF) + MDF
      IF (ZZ .EQ. 1)  LOWF = (1.6 * TSF) + LXF
      IF (ZZ .EQ. 1)  MIDF = (3.2 * TSF) + LXF
      IF (PL .EQ. 2)  GO TO 720
      GO TO 550
 720  CONTINUE
      GO TO 560
```

```
730   REP  =  2
      ER   =  1.0
      IF (PL .EQ. 1)  GO TO 740
      GO TO 750

740   CALL GRSTRT (4662, 1)
750   CALL VWPORT (0.0, 130.0,    0.0, 100.0)
      CALL WINDOW (0.0, 130.0,    0.0, 100.0)
      CALL DASHPT   (0)
      CALL MOVE (  0.0,  25.0)
      CALL DRAW (120.0,  25.0)
      CALL DRAW (120.0, 100.0)
      CALL DRAW (  0.0, 100.0)
      CALL DRAW (  0.0,  25.0)
      CALL MOVE (  0.0,  62.5)
      CALL DASHPT (9)
      CALL DRAW (120.0,  62.5)
      CALL DASHPT   (0)
      CALL MOVE (  0.0,  85.0)
      CALL DRAW (  3.0,  85.0)
      CALL MOVE (  6.0,  85.0)
      CALL TXICUR   (4)
      IF (REP .EQ. 1)  GO TO 760
      CALL TEXT (     9,  POST)
      GO TO 770
760   CALL RNUMBR (MIDT,      1,         5)
      CALL TEXT (     5,  DEG)
770   CALL MOVE ( 20.0,  64.5)
      CALL DRAW ( 20.0,  60.5)
      CALL MOVE ( 40.0,  64.5)
      CALL DRAW ( 40.0,  60.5)
      CALL MOVE ( 60.0,  66.5)
      CALL DRAW ( 60.0,  60.5)
      CALL MOVE ( 80.0,  64.5)
      CALL DRAW ( 80.0,  60.5)
      CALL MOVE (100.0,  64.5)
      CALL DRAW (100.0,  60.5)
      CALL MOVE (  0.0,  40.0)
      CALL DRAW (  3.0,  40.0)
      CALL MOVE (  6.0,  40.0)
      IF (REP .EQ. 1)  GO TO 780
      CALL TEXT (     9,  NEGT)
      GO TO 790
780   CALL RNUMBR (LOWT,      1,         5)
      CALL TEXT (     5,  DEG)
790   CALL MOVE (  0.0,  62.0)
      CALL TXICUR   (7)
      CALL TEXT (     1,  ZERO)
      CALL MOVE (  4.0,  24.0)
      CALL TEXT (     4,  TIME)
      CALL MOVE ( 20.0,  24.0)
      CALL TXICUR   (8)
      CALL INUMBR (INT1,      2)
      CALL MOVE ( 40.0,  24.0)
      CALL INUMBR (INT2,      3)
      CALL MOVE ( 60.0,  24.0)
      CALL INUMBR (INT3,      3)
```

```
      CALL MOVE ( 80.0,  24.0)
      CALL INUMBR (INT4,    3)
      CALL MOVE (100.0,  24.0)
      CALL INUMBR (INT5,    3)
      CALL MOVE (120.0,  24.0)
      CALL TXICUR   (9)
      CALL INUMBR (INT6,    3)
      CALL TXICUR   (4)
      CALL MOVE (  0.0,  20.0)
      CALL TEXT (    19,   TMC)
      CALL MOVE ( 42.0,  20.0)
      CALL DASHPT   (2)
      CALL DRAW (120.0,  20.0)
      CALL MOVE (  0.0,  16.0)
      CALL TEXT (    19,   TOC)
      CALL MOVE ( 42.0,  16.0)
      CALL DASHPT   (3)
      CALL DRAW (120.0,  16.0)
      CALL MOVE (  0.0,  12.0)
      CALL TEXT (    19,   TIC)
      CALL MOVE ( 42.0,  12.0)
      CALL DASHPT   (7)
      CALL DRAW (120.0,  12.0)
      CALL DASHPT   (9)
      CALL MOVE ( 30.0,  85.0)
      CALL DRAW (120.0,  85.0)
      CALL MOVE ( 30.0,  40.0)
      CALL DRAW (120.0,  40.0)
      CALL VWPORT (0.0, 120.0,  25.0, 100.0)
      CALL WINDOW (0.0,  RSHA,   LXT,   HXT)
      CALL MOVE (  0.0,  62.5)
      CALL DASHPT   (2)
      CALL POLY   (LND, TT, DTM)
      CALL DASHPT   (0)
      TNX  =  RSHA / 6.0
      TY   =  LND  / 6
      TNY  =  DTM (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      TNX  =  RSHA / 3.0
      TY   =  LND  / 3
      TNY  =  DTM (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      TNX  =  RSHA / 2.0
      TY   =  LND  / 2
      TNY  =  DTM (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      TNX  =  RSHA * 2.0 / 3.0
      TY   =  LND  * 2   / 3
```

```
      TNY  =  DTM (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      TNX  =  RSHA * 5.0 / 6.0
      TY   =  LND  * 5   / 6
      TNY  =  DTM (TY)
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
      CALL MOVE (  0.0,  62.5)
      CALL DASHPT   (3)
      CALL POLY   (LND, TT, DTO)
      CALL DASHPT   (0)
      TNX  =  RSHA / 6.0
      TY   =  LND  / 6
      TNY  =  DTO (TY)
      CALL MOVE  (TNX,                       (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY + (ER * 1.0)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY - (ER * 1.0)))
      TNX  =  RSHA / 3.0
      TY   =  LND  / 3
      TNY  =  DTO (TY)
      CALL MOVE  (TNX,                       (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY + (ER * 1.0)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY - (ER * 1.0)))
      TNX  =  RSHA / 2.0
      TY   =  LND  / 2
      TNY  =  DTO (TY)
      CALL MOVE  (TNX,                       (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY + (ER * 1.0)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY - (ER * 1.0)))
      TNX  =  RSHA * 2.0 / 3.0
      TY   =  LND  * 2   / 3
      TNY  =  DTO (TY)
      CALL MOVE  (TNX,                       (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY + (ER * 1.0)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY - (ER * 1.0)))
      TNX  =  RSHA * 5.0 / 6.0
      TY   =  LND  * 5   / 6
      TNY  =  DTO (TY)
      CALL MOVE  (TNX,                       (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY + (ER * 1.0)))
      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
      CALL DRAW  (TNX,                       (TNY - (ER * 1.0)))
      CALL DASHPT   (0)
```

```
      TNX  =  RSHA / 6.0
      TY   =  LND  / 6
      TNY  =  DTI  (TY)
      CALL MOVE  (TNX,                                (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW ((TNX - (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW  (TNX,                                (TNY - (ER * 1.0)))
      TNX  =  RSHA / 3.0
      TY   =  LND  / 3
      TNY  =  DTI  (TY)
      CALL MOVE  (TNX,                                (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW ((TNX - (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW  (TNX,                                (TNY - (ER * 1.0)))
      TNX  =  RSHA / 2.0
      TY   =  LND  / 2
      TNY  =  DTI  (TY)
      CALL MOVE  (TNX,                                (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW ((TNX - (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW  (TNX,                                (TNY - (ER * 1.0)))
      TNX  =  RSHA * 2.0 / 3.0
      TY   =  LND  * 2   / 3
      TNY  =  DTI  (TY)
      CALL MOVE  (TNX,                                (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW ((TNX - (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW  (TNX,                                (TNY - (ER * 1.0)))
      TNX  =  RSHA * 5.0 / 6.0
      TY   =  LND  * 5   / 6
      TNY  =  DTI  (TY)
      CALL MOVE  (TNX,                                (TNY - (ER * 1.0)))
      CALL DRAW ((TNX + (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW ((TNX - (0.01082 * RSHA)),  (TNY + (ER * 0.5)))
      CALL DRAW  (TNX,                                (TNY - (ER * 1.0)))
      CALL MOVE (  0.0,  62.5)
      CALL DASHPT   (7)
      CALL POLY   (LND, TT, DTI)

      IF (REP .EQ. 1)  GO TO 920
      IF (PL .EQ. 1)   GO TO 800
      CALL CMCLOS
      READ*
      CALL CMOPEN
      CALL NEWPAG
      GO TO 840
  800 CALL GRSTOP
      WRITE  (6, 810)
  810 FORMAT (X, '                YOU MUST NOW REMOVE THE COMPLETED
     + GRAPHICS DISPLAY, INSERT A NEW')
      WRITE  (6, 820)
  820 FORMAT (X, '                SHEET OF PAPER, AND TOTALLY RESET THE
     + 4662 PLOTTING MACHINE.')
      WRITE  (6, 830)
  830 FORMAT (X, '                PRESS <RET> WHEN THIS IS DONE.')
      READ*
  840 CONTINUE
```

```
850  WRITE  (6, 860)
860  FORMAT (X, '              DO YOU WISH TO EXPAND THIS DISPLAY?
    + IF SO, TYPE IN 1;  IF NOT, ')
     READ 870, REP
870  FORMAT (I1)
     IF (REP .EQ. 1)  GO TO 900
     IF (REP .EQ. 2)  GO TO 920
     WRITE  (6, 880)
880  FORMAT (X, '              YOU HAVE TYPED IN A NUMBER THAT CANNOT
    + BE UTILIZED.  YOU WILL HAVE')
     WRITE  (6, 890)
890  FORMAT (X, '              TO TRY AGAIN.')
     GO TO 850
900  TST  =  DTO(LND) - DTI(LND)
     TST  =  TST ** 2.0
     TST  =  SQRT (TST)
     ER   =  1.2 * TST / 50.0
     IF (NK .EQ. 1.0)  ER = 2.0
     LXT  = -0.6 * TST
     HXT  =  0.6 * TST
     LOWT = -0.3 * TST
     MIDT =  0.3 * TST
     IF (NK .EQ. 1)  LXT  =-1.2 * TST
     IF (NK .EQ. 1)  HXT  = 1.2 * TST
     IF (NK .EQ. 1)  LOWT =-0.6 * TST
     IF (NK .EQ. 1)  MIDT = 0.6 * TST
     IF (ZZ .EQ. 1)  LXT  =-2.4 * TST
     IF (ZZ .EQ. 1)  HXT  = 2.4 * TST
     IF (ZZ .EQ. 1)  LOWT =-1.2 * TST
     IF (ZZ .EQ. 1)  MIDT = 1.2 * TST
     IF (NK .EQ. 1)  ER   =  ER * 2.0
     IF (ZZ .EQ. 1)  ER   =  ER * 4.0
     IF (PL .EQ. 2)  GO TO 910
     GO TO 740
910  CONTINUE
     GO TO 750

920  CALL GRSTOP
     RETURN
     END
```