

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

---

Signatures have been redacted for privacy

Ames, Iowa

by

1984

## TABLE OF CONTENTS

NOMENCLATURE	vi
I. INTRODUCTION	1
II. PROPOSED OBJECTIVE AND SCOPE	2
III. LITERATURE REVIEW	4
IV. THE TWO-TIME-SCALE MATRIX DECOUPLING ALGORITHM	6
A. Computing the L Matrix	18
B. Computing the K Matrix	29
C. The Choice of Slow and Fast Variable	34
D. Solution of the Point-Kinetics Equations Utilizing Two-Time-Scale Decoupling Methods	38
V. MAIN DEVELOPMENT OF THE PROGRAMS	53
A. Main Program NERO	53
B. Reactor Kinetics Subroutine GALBA	55
C. Thermal Hydraulics Equations OTHO	73
D. Steam Generator Subroutine DMTN	84
E. Graphics Subroutine VESPASIAN	99
VI. TESTS AND RESULTS	101
VII. SUMMARY AND CONCLUSIONS	130
VIII. LITERATURE CITED	133
IX. ACKNOWLEDGMENTS	135
X. APPENDIX A: USER'S GUIDE	136
XI. APPENDIX B: PROGRAM LISTING	146

## LIST OF FIGURES

	Page
Figure 1. Reactor and steam generator power with feedback. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel	103
Figure 2. Reactor and steam generator power with initial power 2000 MW, 10 cents reactivity step insertion, U-233 fuel	108
Figure 3. Reactor and steam generator power with initial power 2000 MW, 10 cents reactivity step insertion, Pu-239 fuel	109
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	110
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	112
Figure 6. Reactor and steam generator power with feedback and ramp-input model. Initial power level 2000 MW, ramp-input model with ramp of 0.5 cents reactivity per second for 20 seconds, U-235 fuel	114
Figure 7. Reactor and steam generator power with ramp-input model. Time of run 3 minutes	115
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	116
Figure 9. Reactor and steam generator power with no feedback and no control system and using the prompt-jump approximation	117
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 $-1.0E-07$	119

	Page
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 $-1.0E-06$	120
Figure 12. Reactor and steam generator with load following and feedback. Initial power 2000 MW, throttle valve opened 10%, U-235 fuel	121
Figure 13. Reactor and steam generator with reactor power step change. Initial power 2000 MW with 100 MW step change	123
Figure 14. Reactor and steam generator power with reactor power step change. Initial power 2000 MW with -100 MW step change	124
Figure 15. Fuel temperature change. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel	126
Figure 16. Reactor inlet, outlet, and average moderator temperature changes. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel	127

## LIST OF TABLES

	Page
Table 1. Inverse eigenvector matrix of coefficient matrix of point-kinetics equations, with U-235 fuel	36a
Table 2. Correlations for physical constants of H <sub>2</sub> O as a function of temperature	83
Table 3. User-selected parameters, their default values and lower and upper limits	143

## 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 time 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 matrix
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 generator 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
$\alpha_f$	Doppler coefficient of reactivity
$\alpha_m$	moderator coefficient of reactivity
$\beta$	delayed-neutron fraction
$\gamma$	rampt-input rate
$\delta$	differential quantity
$\epsilon$	trial value for testing iteration cessation
$\Lambda$	neutron generation time
$\lambda$	delayed-neutron decay constant
$\mu$	kinematic viscosity
$\nu$	dynamic viscosity
$\rho$	reactivity; also density
$\rho_f$	fast decoupling ratio



$\rho_s$	slow decoupling ratio
$\rho_t$	total reactivity
$\tau$	constant of integral controller
$\tau_c$	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

$\delta X$  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 were 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

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

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

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

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

$$X = TY, \quad (4-3)$$

where T is a transformation matrix. Then

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

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

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

Now again,

$$X = TY. \quad (4-3)$$

This implies

$$Y = T^{-1}X, \quad (4-6)$$

which further implies

$$\dot{Y} = T^{-1}\dot{X}. \quad (4-7)$$

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

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

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

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

Expanding equation (4-9) yields

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

and

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

Introduce the algebraic Riccati equation

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

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

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

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

Introduce the vectors

$$[B_1] = [A_{11} - A_{12}L] \quad (4-14)$$

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

and the Lyapunov equation

$$KB_2 - B_1K + A_{12} = 0. \quad (4-16)$$

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

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

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

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

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

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

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

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

$$A = MJQ, \quad (4-19)$$

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,

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

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

Since

$$QM = I, \quad (4-21)$$

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

This implies that

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

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

which leads to the result that

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

or

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

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

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

Similarly, from Equation (4-24),

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

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

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

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

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

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

$$Q_{22}^{-1} \begin{matrix} -1 \\ 12 \end{matrix} - M_{22} M_{12}^{-1} = M_{21} M_{11}^{-1}$$

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

which leads to the identity

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

Both Equations (4-26) and (4-29) will be used in developing Theorem 1, in which the conditions under which the algebraic Riccati equation 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

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

and

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

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

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

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

$$X_1 = Y_1 - Y_2 K \quad (4-3a)$$

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

$$Y_1 = (I_{n_1} + LK) X_1 + K X_2 \quad (4-6a)$$

$$Y_2 = L X_1 + X_2 \quad (4-6b)$$

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

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

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,

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

Therefore,

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

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

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

$$\begin{aligned} &= A_{11}Y_1 - KA_{11}Y_2 + A_{12}(-Y_1L + (I_{n_1} + LK)Y_2) \\ &= A_{11}Y_1 - KA_{11}Y_2 - A_{12}LY_1 + A_{12}Y_2 + LKA_{12}Y_2 \\ &= (A_{11} - A_{12}L)(Y_1 - KY_2) + A_{12}Y_2 \end{aligned}$$

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

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

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

$$\dot{Y}_2 = B_2Y_2 \quad (4-18a)$$

Therefore, the transformation (4-33) reduces to

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

where

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

## 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)

$$Y = T^{-1} A T Y$$

or

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

where again,

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

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,

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

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

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

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

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

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

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

$$\begin{aligned}
 T^{*-1}MJQT^* &= \begin{bmatrix} I_{n_1} & 0 \\ -M_{21}M_{11}^{-1} & I_{n_2} \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} Q_{11}-Q_{12}Q_{22}^{-1}Q_{21} & Q_{12} \\ 0 & Q_{22} \end{bmatrix} \\
 &= \begin{bmatrix} M_{11} & M_{12} \\ 0 & -M_{21}M_{11}^{-1}M_{12} \\ & +M_{22} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} -Q_{12}Q_{22}^{-1}Q_{21} & Q_{12} \\ 0 & Q_{22} \end{bmatrix} \\
 &= \begin{bmatrix} M_{11} & M_{12} \\ 0 & Q_{22}^{-1} \end{bmatrix} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} \begin{bmatrix} M_{11}^{-1} & Q_{12} \\ 0 & Q_{22} \end{bmatrix} \\
 &= \begin{bmatrix} M_{11}J_1 & M_{12}J_2 \\ 0 & Q_{22}^{-1}J_2 \end{bmatrix} \begin{bmatrix} M_{11}^{-1} & Q_{12} \\ 0 & Q_{22} \end{bmatrix} \\
 &= \begin{bmatrix} M_{11}JM_{11}^{-1} & M_{12}J_2Q_{12}+M_{12}J_2Q_{22} \\ 0 & Q_{22}^{-1}J_2Q_{22} \end{bmatrix}, \tag{4-39}
 \end{aligned}$$

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, is 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

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

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  $6 \times 6$  matrix, the  $A_{22}$  matrix (and the  $M_{22}$  and the  $Q_{22}$  matrices) is  $1 \times 1$ , and the  $A_{12}$  and  $A_{21}$  matrices are, respectively,  $6 \times 1$  and  $1 \times 6$ . Also, the  $L$  matrix is  $1 \times 6$  and the  $K$  matrix is  $6 \times 1$ . 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

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

$$i = 0.$$

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

$$\|R_i\| \leq \epsilon,$$

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

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

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  $1 \times 1$ , 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  $1 \times 1$ ,

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

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

and

$$\begin{aligned} B_2 &= Q_{22}^{-1} J_2 Q_{22} \\ &= \frac{J_2 Q_{22}}{Q_{22}} \end{aligned}$$

and the elegant result that

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

is achieved. Of course,  $J_2$ , like  $B_2$ , is  $1 \times 1$ .

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

$$\begin{aligned}
 D_i &= B_2^{-1} R_i \\
 &= R_i / B_2.
 \end{aligned}
 \tag{4-43}$$

To continue with the description of the iteration process, let

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

Then recompute the residual matrix  $R_i$ ,

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

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

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

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

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

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

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

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

- 3) Evaluate

$$R_{i+1} = -K_{i+1} B_2 + B_1 K - A_{12}. \quad (4-45)$$

Stop if

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

- 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,  $\epsilon = 1.0 \times 10^{-6}$ .
- 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,

$$B_2 = J_2, \quad (4-42)$$

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

$$R_i = K_{i+1} B_2 - B_1 K_{i+1} + A_{12}. \quad (4-45)$$

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

$$0 = KB_2 - B_1K + A_{12} . \quad (4-16)$$

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

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

Now the situation is much different. For if

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

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.

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

or

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

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  $7 \times 7$  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  $\rho_s$  and the fast mode decoupling ratio  $\rho_f$

$$\rho_s = \frac{||M_{21}||}{||M_{11}||} \quad (4-48)$$

and

$$\rho_f = \frac{||M_{12}||}{||M_{22}||} \quad (4-49)$$

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 matrix 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- $\phi$ 3	-2.2450656E- $\phi$ 2	1.138648	-6.1259083E- $\phi$ 2
5.5526220E- $\phi$ 3	3.1158078E- $\phi$ 2	-4.7610782E-2	-1.119222
5.5012584E- $\phi$ 2	-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

$$\begin{aligned} \rho_f &= \frac{||M_{12}||}{||M_{22}||} \\ &= 0.564239945 \end{aligned}$$

$$\begin{aligned} \rho_s &= \frac{||M_{21}||}{||M_{11}||} \\ &= 0.017479834. \end{aligned}$$

Since  $\rho_f$  is very large compared to  $\rho_s$ , 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,  $\rho_f$  will start to



decrease, but on the other hand,  $\rho_s$  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  $\rho_f \gg \rho_s$ , but when  $s = 6$  and  $f = 1$ ,  $\rho_f$  is minimized and  $\rho_s$  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  $\rho_f$ , or  $\rho_s$ . When  $f = 1$ ,  $||M_{12}||$  is definitively minimized,  $||M_{21}||$  and  $\rho_s$  nearly so, and  $\rho_f$  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  $1 \times 6$  or  $6 \times 1$  or even  $1 \times 1$ , 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

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

where

$C_j$  is the delayed-neutron precursor number for the  $j$ th group of precursors

$n$  is reactor power

$\lambda_j$  is the delayed-neutron decay constant for the  $j$ th group

$\beta_j$  is the delayed-neutron fraction for the  $j$ th group

$\beta$  is the total delayed-neutron fraction for all groups

$\Lambda$  is neutron generation time

$\rho$  is reactivity.

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

Another way of expressing Equation (4-50) is

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

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_j \lambda_j C_j \quad (4-51b)$$

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

$$\frac{d}{dt}(C_{i0} + \delta C_i) = \frac{\beta_i}{\Lambda} (n_0 + \delta n) - \lambda_i (C_{i0} + \delta C_i), \quad (4-52)$$

where

$C_{j0}$  is steady-state precursor number

$\delta C_j$  is transient precursor number

$n_0$  is steady-state reactor power

$\delta n$  is transient reactor power.

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

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

$$\frac{d}{dt}(C_{i0}) = \frac{\beta_j}{\Lambda} (n_0) - \lambda_j (C_{j0}). \quad (4-53)$$

Since  $C_{j0}$  is a steady-state variable, it does not change with respect to time, and

$$\frac{d}{dt}(C_{j0}) = 0.$$

Therefore,

$$0 = \frac{\beta_j n_o}{\Lambda} - \lambda_j C_{j0},$$

which leads to the important result that

$$\lambda_j C_{j0} = \frac{\beta_j n_o}{\Lambda}$$

$$C_{j0} = \frac{\beta_j n_o}{\Lambda \lambda_j} \quad (4-54)$$

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,

$$\delta C_j(i+1) = h \left( \frac{\beta_j \delta n_i}{\Lambda} - \lambda_j \delta C_{ij} \right) + \delta C_{ij}. \quad (4-55)$$

Similarly,

$$\begin{aligned} \delta n_{i+1} = h \left( \frac{1}{\Lambda} (\rho_o \delta n + \delta \rho n_o + \delta \rho \delta n_i - \beta \delta n_i) \right. \\ \left. + \sum_j \lambda_j \delta C_{ij} \right) + \delta n_i, \end{aligned} \quad (4-56)$$

where

$\rho_0$  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),

$$X = TY \quad (4-3)$$

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

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

Since  $K$  is a  $6 \times 1$  matrix and  $L$  is  $1 \times 6$ , the lower right hand entry in the transformation matrix is  $1 \times 1$  and

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

Also, the upper left hand entry in the transformation matrix is a  $6 \times 6$  identity matrix.

Accordingly, restate Equation (4-57),

$$\begin{bmatrix} \delta C_1 \\ \delta C_2 \\ \delta C_3 \\ \delta C_4 \\ \delta C_5 \\ \delta C_6 \\ \delta n \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -K_1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -K_2 \\ 0 & 0 & 1 & 0 & 0 & 0 & -K_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & -K_4 \\ 0 & 0 & 0 & 0 & 1 & 0 & -K_5 \\ 0 & 0 & 0 & 0 & 0 & 1 & -K_6 \\ -L_1 & -L_2 & -L_3 & -L_4 & -L_5 & -L_6 & (1.0 + \sum_{i=1}^6 L_i K_i) \end{bmatrix} \begin{bmatrix} \delta Y_1 \\ \delta Y_2 \\ \delta Y_3 \\ \delta Y_4 \\ \delta Y_5 \\ \delta Y_6 \\ \delta Y_7 \end{bmatrix}.$$

(4-58)

The only term here that is of interest in program output is the reactor power term  $\delta n$ . The  $\delta C_i$  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  $\delta n$ .

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

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

and

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

(4-60)

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

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

or

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

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

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

$$h_s = 0.2 \text{ sec}$$

$$h_f = 0.001 \text{ sec}$$

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

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

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

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

$$\dot{X} = AX, \tag{4-1}$$

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  $\rho$  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  $\rho$  as a constant. Doing this will mean that every time  $\rho$  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  $\rho$  as a constant will mean that every time  $\rho$  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  $\rho$  calculated.

Since so many steps are needed for the calculation of  $\rho$ , and from this calculation further values for  $\rho$  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  $\rho$  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

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

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:

$$\begin{aligned} \delta n_{i+1} = h \left( \frac{1}{\Lambda} (\rho_0 \delta n + \delta \rho n_0 + \delta \rho n_i) \right. \\ \left. - \beta \delta n_i + \sum_j \lambda_j \delta C_{ij} \right) \\ + \delta n_i. \end{aligned} \quad (4-56)$$

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  $\delta n_i$ , 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  $\rho$  is expressed as a function of its steady-state value and its transient value:

$$\rho = \rho_0 + \delta\rho \quad (4-64)$$

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

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

$$\begin{aligned} \frac{d}{dt}(\delta n) = & \frac{1}{\Lambda}(\rho_0 \delta n + \delta\rho n_0 + \delta\rho \delta n - \beta \delta n) \\ & + \sum_j \lambda_j \delta C_j \end{aligned} \quad (4-66)$$

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

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

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

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

As before, restate the precursor variables  $\delta C_j$  as the slow mode variable  $\delta X_1$  and the power variable  $\delta n$  as the fast mode variable  $\delta X_2$ . Then, Equation (4-68) can be restated as

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

As before,

$$X = TY \quad (4-3)$$

$$Y = T^{-1}X \quad (4-6)$$

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

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

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

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

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

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

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

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

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

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

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:

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

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_j \lambda_j C_j, \quad (4-52)$$

where

$C_j$  is the delayed-neutron precursor number for the  $j$ th group of precursors

$n$  is reactor power

$\lambda_j$  is the delayed-neutron decay constant for the  $j$ th group

$\beta_j$  is the delayed-neutron fraction for the  $j$ th group

$\beta$  is the total delayed-neutron fraction for all groups

$\Lambda$  is neutron generation time

$\rho$  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:

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

and

$$\delta n_{i+1} = h \left( \frac{1}{\Lambda} (\rho_0 \delta n_i + \delta \rho n_0 + \delta \rho \delta n_i - \beta \delta n_i + \sum_j \lambda_j C_{ij}) \right) + \delta n_i, \quad (4-56)$$

where

$\rho_0$  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,

$$h = 0.001 \text{ sec}$$

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

With decoupling,

$$h_s = 0.2 \text{ sec},$$

$$h_f = 0.001 \text{ sec},$$

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

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

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  $\Lambda$ , 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  $1.0 \times 10^{-6}$  sec, but quickly increases to 0.001 sec. Where matrix decoupling is not used,  $h$  equals  $1.0 \times 10^{-6}$  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

$$\rho_t = \gamma t \quad (5-1)$$

and

$$\gamma = \rho/\text{sec}, \quad (5-2)$$

where

$\rho_t$  is total reactivity

$\gamma$  is ramp-insertion rate

$t$  is time

$\rho$  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),

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

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_j \lambda_j C_j \quad (4-52)$$

When  $\Lambda$  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  $(\sum_j \lambda_j C_j)$ . Under these circumstances,  $\frac{dn}{dt}$  can be set equal to zero, and Equation (4-52) can be restated as

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

Therefore,

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

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

and

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

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 restated as

$$(n_0 + \delta n) = \frac{\Lambda(\sum_j \lambda_j (C_j + \delta C_j))}{\beta - (\rho_0 + \delta \rho)} \quad (5-5)$$

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

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

Using finite-difference methods, the solutions are

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

and

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

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

$$X(t) = X(t_0)e^{A(t-t_0)} + B \int_{t_0}^t Y(t')e^{A(t-t')} dt'$$

$$Y(t) = Y(t_0)e^{D(t-t_0)} + C \int_{t_0}^t X(t')e^{D(t-t')} dt'.$$

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

$$X(t') \approx X(t_0)$$

and

$$Y(t') \approx Y(t_0)$$

The integrals may then be evaluated:

$$X(t) \approx X(t_0)e^{A(t-t_0)} + \frac{BY(t_0)}{A} [e^{A(t-t_0)} - 1]$$

and

$$Y(t) \approx Y(t_0)e^{D(t-t_0)} + \frac{CX(t_0)}{D} [e^{D(t-t_0)} - 1].$$

Letting

$$h = t - t_0$$

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

$$X(t_0) = X_i$$

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

$$Y(t_0) = Y_i,$$

where  $h$  is the time step,

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

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

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

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

(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

$$Ah = 0.1, \text{ then}$$

$$1 + Ah = 1.1,$$

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

and

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

$$\text{For } |Ah| < 0.2,$$

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

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

$$|Ah| < 0.1 \text{ will be used.}$$

Repeating the point-kinetics equations,

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_j \lambda_j C_j \quad (4-51b)$$

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda} n - \lambda_j C_j. \quad (4-51a)$$

Here,

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

$$D = \lambda_i.$$

For U-235,

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

For all cases,

$$\Lambda = 0.0001 \text{ seconds was used.}$$

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

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

For any isotope, then,

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

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

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  $1.53 \times 10^{-3}$  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  $K > 1.0$ ,  $\rho$  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  $\beta$  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,

$$\delta\rho_f = \alpha_f \delta T_f + \alpha_m \delta T_m, \quad (5-10)$$

where

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

$\alpha_f$  is Doppler coefficient of reactivity

$\delta T_f$  is change in fuel temperature

$\alpha_m$  is moderator temperature coefficient of reactivity  
 $\delta T_m$  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

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

where

$\rho_c$  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.

$\delta T_{av}$  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

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

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

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

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

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:

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

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

where

$\delta T_f$  is transient reactor fuel temperature

$\delta T_m$  is transient reactor coolant temperature

$\delta T_o$  is transient reactor coolant output temperature

$\delta T_i$  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

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

and

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

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.

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

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

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

where

$\delta T_{oi}$  is the transient reactor outlet temperature (that is, the difference between current temperature and initial temperature)

$\delta T_{ii}$  is the transient reactor inlet temperature

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

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

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),

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

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:

$$h_p = 34000 \text{ W/m}^2\text{K}$$

$$A = 5945.0 \text{ m}^2$$

$$C_{P_f} = 160 \text{ J/kgC}$$

$$M_f = 95000 \text{ kg}$$

$$C_{P_m} = 6000 \text{ J/kgC}$$

$$M_m = 13300 \text{ kg.}$$

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  $5945 \text{ m}^2$  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)

$\nu$  is kinematic viscosity ( $m^2/s$ ).

$$\nu = \mu/\rho$$

where

$\mu$  is dynamic viscosity (kg/m s)

$\rho$  is density of reactor coolant ( $Kg/m^3$ )

Correlations for  $\mu$  and  $\rho$  were derived from data points obtained in reference [10].

The Prandtl number is expressed as Pr.

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

where

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

$\nu$  is kinematic viscosity ( $\text{m}^2/\text{sec}$ )

$\rho$  is density of reactor coolant ( $\text{kg}/\text{m}^3$ )

$K$  is thermal conductivity of reactor coolant ( $\text{W}/\text{mC}$ ).

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],

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

Finally, the heat transfer coefficient  $h_p$  is given as

$$h_p = \text{Nu } k/D.$$

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<sub>2</sub>O as a function of temperature

Variable	Correlation	Type of fit	Pressure	R <sup>2</sup>
Kinematic viscosity ν	$6.1777117 \cdot 10^{-12} T^2 - 3.20997 \cdot 10^{-9} T + 5.5038552 \cdot 10^{-7}$ m <sup>2</sup> /s	Parabolic	15.514 MPa (2250 psia)	none (only 3 data points)
Thermal conductivity k	$0.7207553673 - 4.5873157 \cdot 10^{-3} \cdot \text{EXP}(0.012380238 T)$ W/m C	Exponential	"	0.9995
Heat capacity C <sub>p</sub>	$4992.4097749 + 2.49340775 \cdot 10^{-4} \cdot \text{EXP}(0.04825458 T)$ J/kg C	Exponential	15.514 MPa (2250 psia)	0.9999
Density ρ	$881.6309649 - 2.86514041 \cdot \text{EXP}(0.0133034152 T)$ kg/m <sup>3</sup>	Exponential	"	0.9999
Heat capacity at 15.514 MPa and 3.3189 C as function of power	$5916.241929 + 16.32498553 \cdot \text{EXP}(6.4880554 \cdot 10^{-10} \cdot \text{Power})$	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  $6.5 \text{ m}^2$ . Heat transfer area is  $22400 \text{ m}^2$ . 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

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

where

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

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

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

$T_{\text{sii}}$  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

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

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

$$(M_{msg} C_{pm}) \frac{d}{dt} T_{sg} = C_{pm} W_p (T_{sgo} - T_{sgi}) - h_s A (T_{sg} - T_s) \quad (5-24)$$

and

$$C_{ptm} \frac{d}{dt} T_s = h_s A (T_{sg} - T_s) - W_s D_{hfg}, \quad (5-25)$$

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  $2.54 \times 10^8$ , 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

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

and

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

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  $\delta T_{si}$  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).

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

where

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

$\delta T_{si}$  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).

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

where

$\delta T_{oi}$  is the change in reactor outlet temperature in a given time step

$\delta T_{mi}$  is change in reactor coolant temperature

$\delta T_{ii}$  is the change in reactor inlet temperature,

and

$$\begin{aligned} \delta T_{m(i+1)} = & \frac{h}{C_{pm} M_m} [(h_p A (\delta T_{ei} - \delta T_{mi})) \\ & - (C_{pm} W_p (\delta T_{oi} - \delta T_{ii}))]. \end{aligned} \quad (5-18)$$

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  $\delta T_{oi}$  and  $\delta T_{ii}$  would change by the same amount, albeit with opposite signs. For this reason,

$\delta T_{oi}$  and  $\delta T_{ii}$  would cancel each other out and  $\delta T_m$  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  $\delta T_{oi}$  and  $\delta T_{ii}$  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  $\delta T_{ii}$  still equals zero. Because  $\delta T_{ii}$  still equals zero, it does not now cancel  $\delta T_{oi}$  to produce an average moderator temperature change of zero (see Eq. 5-21). Because of this and the fact that  $\delta T_{ii}$  still equals zero, the  $\delta T_{oi}$  term in Equation (5-21) is twice as large as it would be if the  $\delta T_{ii}$  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  $\delta T_{oi}$  plus the negative of  $\delta T_{ii}$ , keeping in mind that under steady-state conditions,

$\delta T_{oi}$  is equal and opposite in sign to  $\delta T_{ii}$ . If  $\delta T_{ii}$  equals zero, the same level of power will be transferred if  $\delta T_{oi}$  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  $\delta T_{oi}$  coming into the steam generator is twice as large as its steady-state value for the new power level, then the new  $\delta T_{ii}$  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  $\delta T_{ii}$  is calculated by DMTN to be equal to zero,  $\delta T_{ii}$  remains zero when the coolant returns to the reactor. Thus, there is no  $\delta T_{ii}$  to reduce  $\delta T_{oi}$  back toward the value it would have under steady-state conditions at the new power level. Furthermore,  $\delta T_{mi}$  does not converge back toward zero, as it would if  $\delta T_{ii}$  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  $\delta T_{mi}$  other than zero.

The fact that  $\delta T_{oi}$  is twice as large as it "ought" to be, can be used by DMTN as a criterion for boosting  $W_s$  to compute a  $\delta T_{ii}$  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  $\delta T_{ii}$  will cause the moderator temperature change to converge back toward zero and the magnitude of  $\delta T_{oi}$  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  $\delta T_{oi}$  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  $\delta T_{oi}$  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  $\delta T_{oi}$  and the value that it would have under steady-state conditions at the new power level. It is the negative of what  $\delta T_{ii}$  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.

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

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 con-

verge toward the values they would have under steady-state conditions at the new power level. In actuality,  $\delta T_{ii}$  at this point still equals zero. Since  $\delta T_{oi}$ , 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  $\delta T_{ii}$  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  $\delta T_{oi}$ , 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  $\delta T_{ii}$  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

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

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:

$$TDENTC = \sum_{i=1} DENTC_i. \quad (5-31)$$

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

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

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

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

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

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

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  $\delta T_{sg}$ .

$\delta T_{ii}$  is now computed by means of Equation 5-35.

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

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

(5-18),  $\delta T_{mi}$  is brought back toward zero and  $\delta T_{oi}$  is brought back to the value it would have under steady-state conditions at the new power level. Coolant at the new temperature for  $\delta T_{oi}$  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  $\delta T_{ii}$  reaches the reactor, reactor power itself has changed, and the  $\delta T_{ii}$  signal is not powerful enough to quickly cause  $\delta T_{mi}$  to converge back to zero or  $\delta T_{oi}$  to converge toward its steady-state value at the new power level. Furthermore, any changes in  $\delta T_{mi}$  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.  $\delta T_{ii}$  will immediately assume some nonzero value, while  $\delta T_{oi}$  will still remain equal to zero. When the coolant at the new temperature of  $\delta T_{ii}$  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  $\delta T_{oi}$  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 phenomena 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

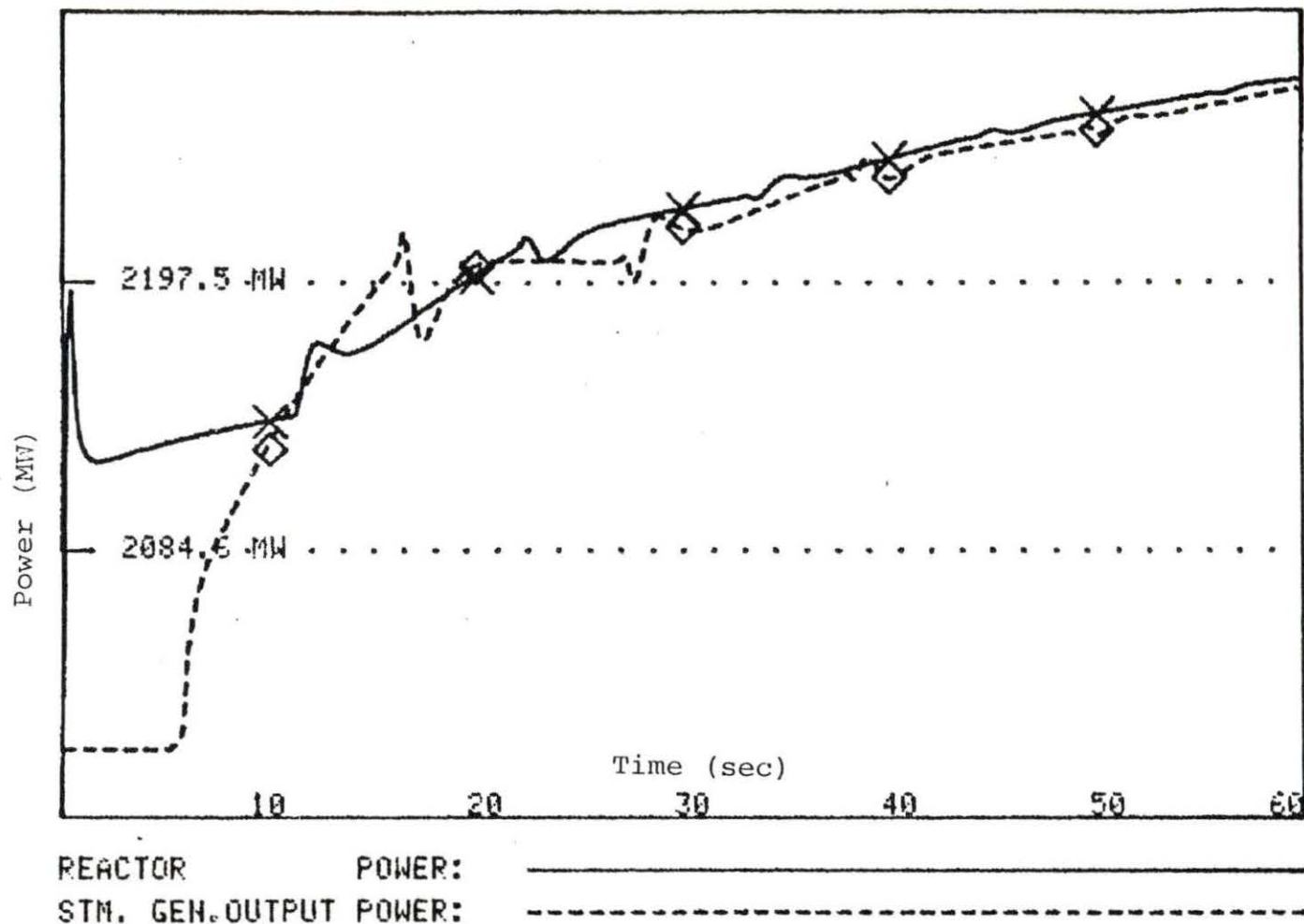


Figure 1. Reactor and steam generator power with feedback. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel

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.

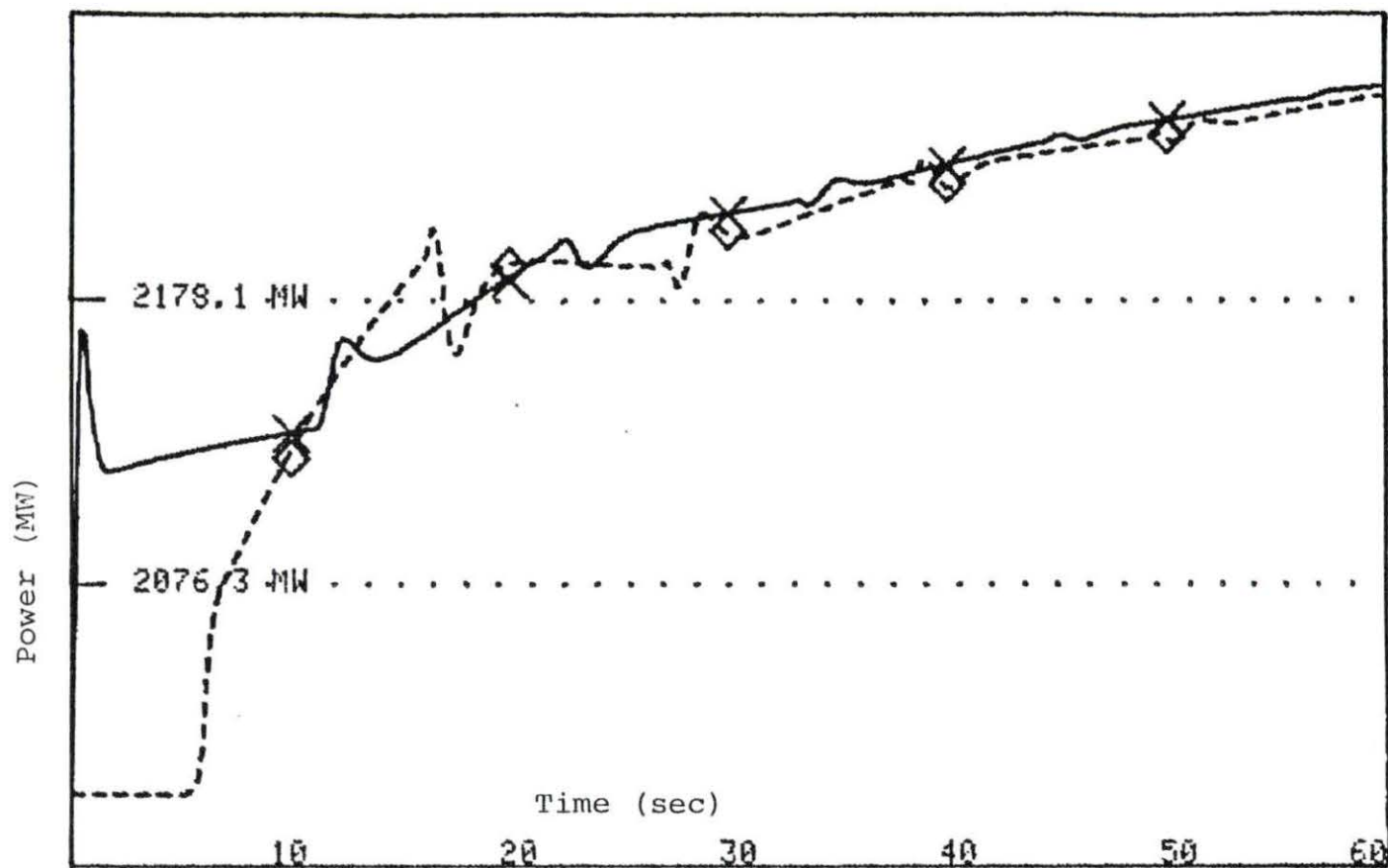
When this coolant with lowered temperatures arrives at the reactor at 11 seconds, it immediately lowers the average moderator temperature, as is shown in Figure 16. This in turn quickly lowers fuel temperature, as is seen in Figure 15. The rapid lowering of fuel and moderator temperatures adds a large 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.

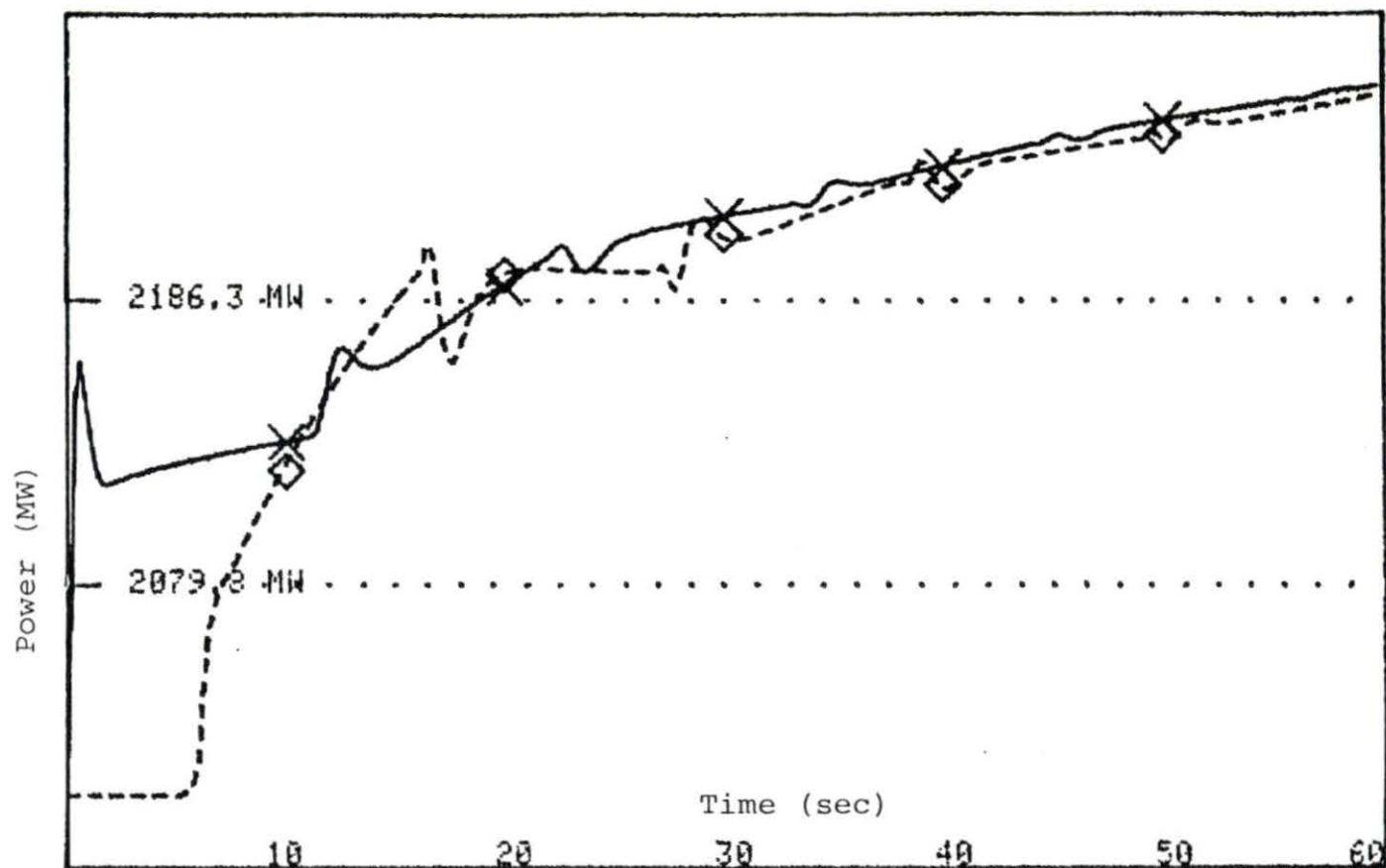
Figure 4 illustrates the workings of the prompt-jump approximation. Hetrick [7] shows the prompt-jump approximation will yield solutions of the point-kinetics equations that are slightly higher than those yielded by the unmodified point-kinetics equations. In this run, that was not the case. The reason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



REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

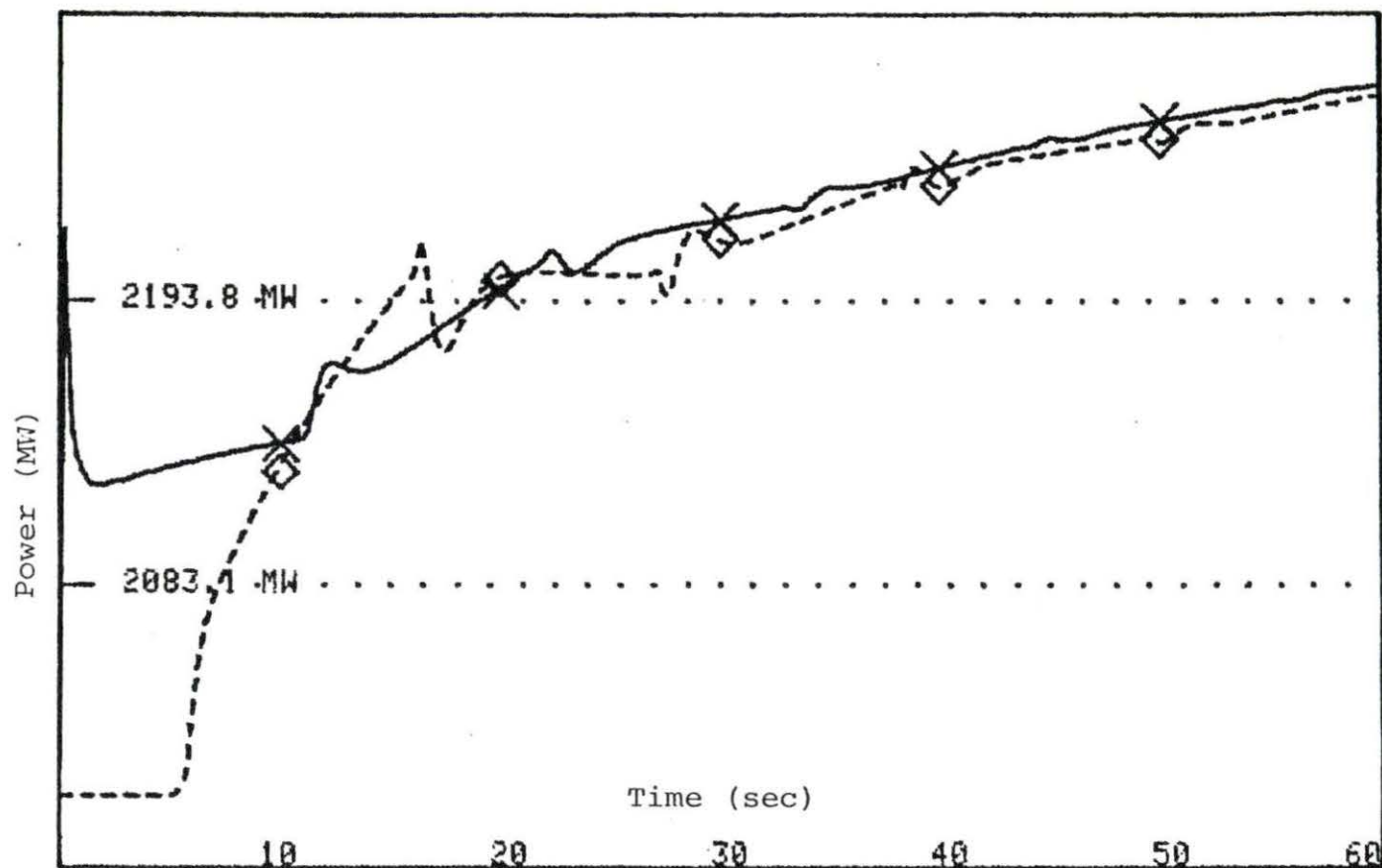
Figure 2. Reactor and steam generator power with initial power 2000 MW, 10 cents reactivity step insertion, U-233 fuel





REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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



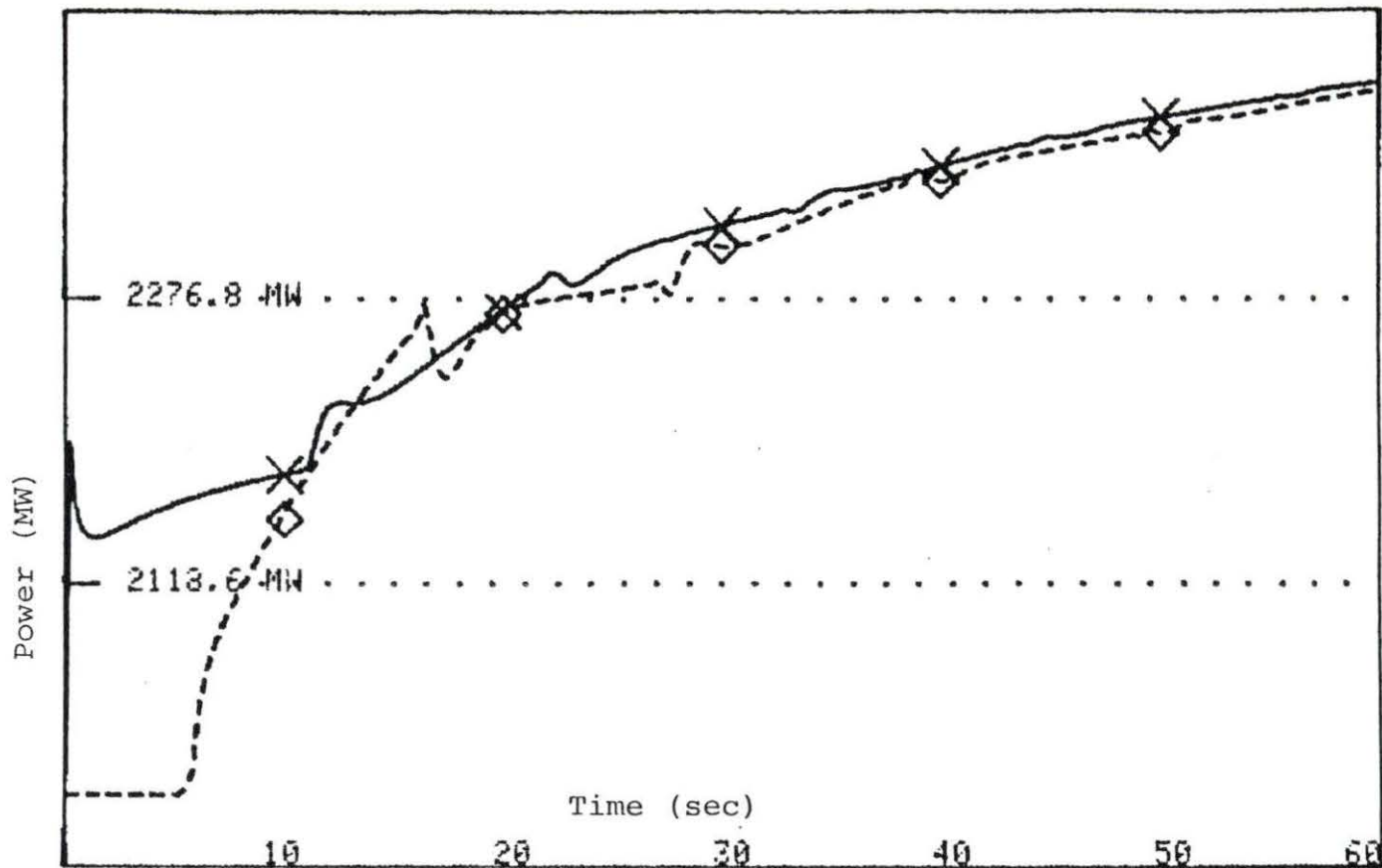
REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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,



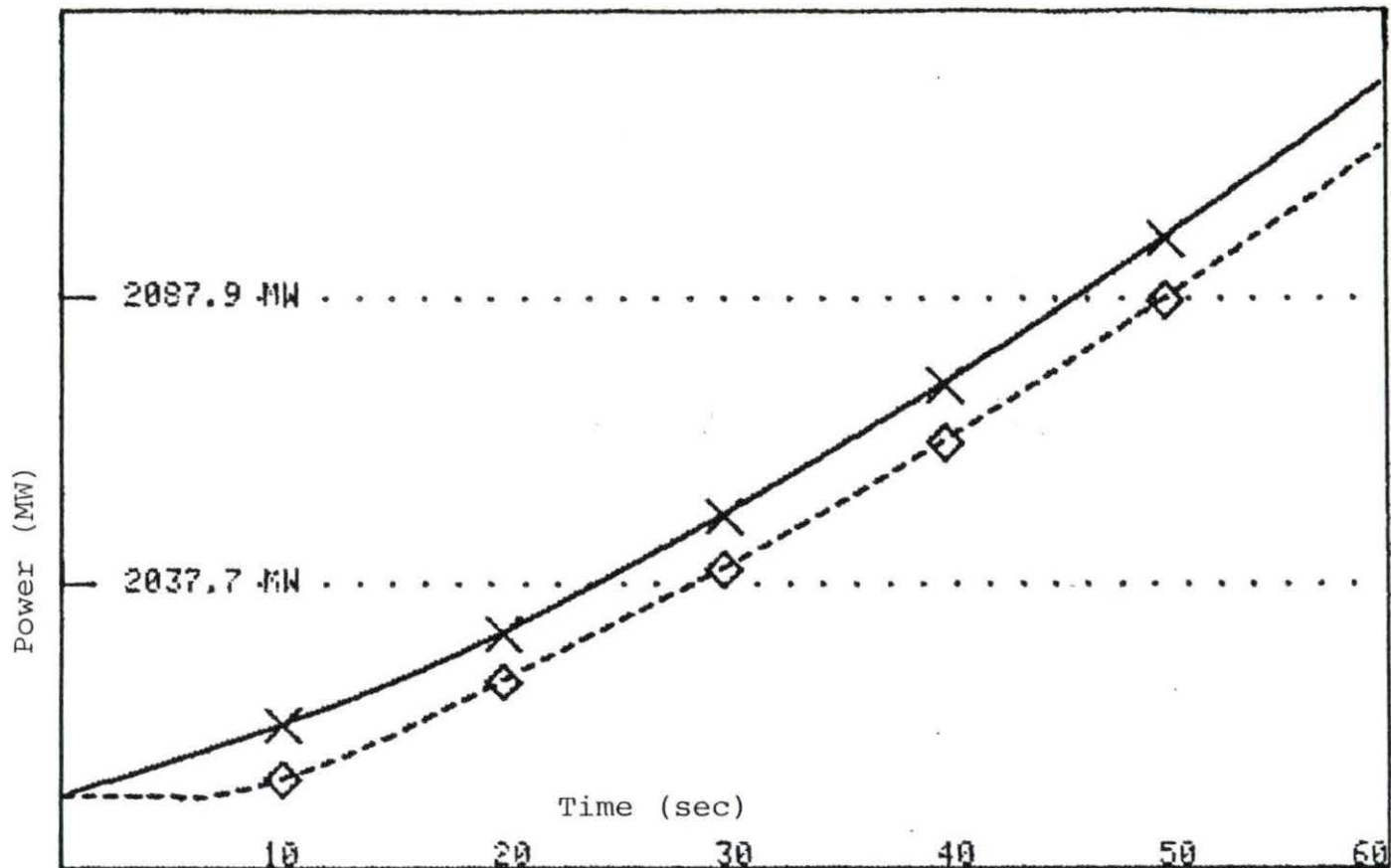
REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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



REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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

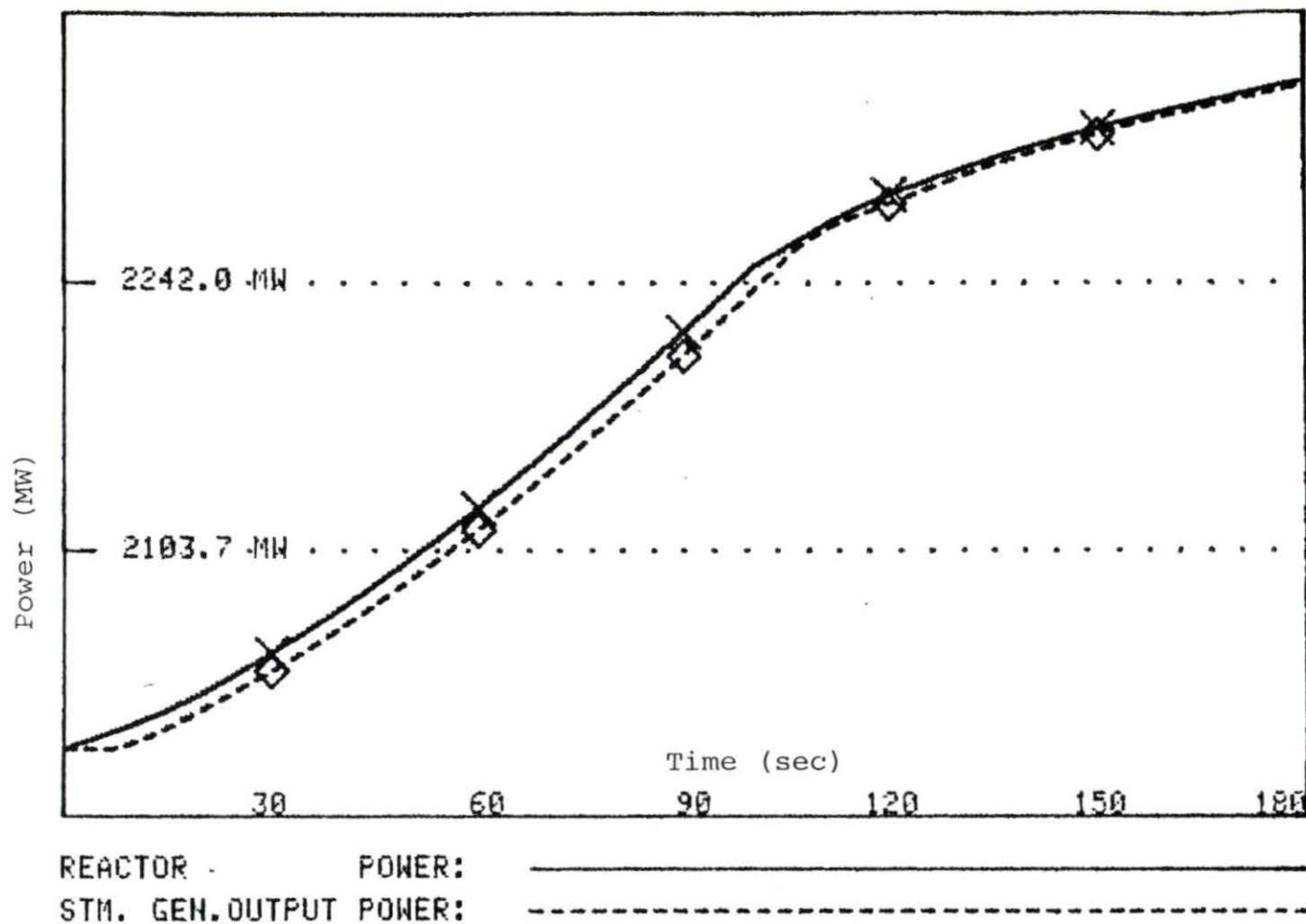
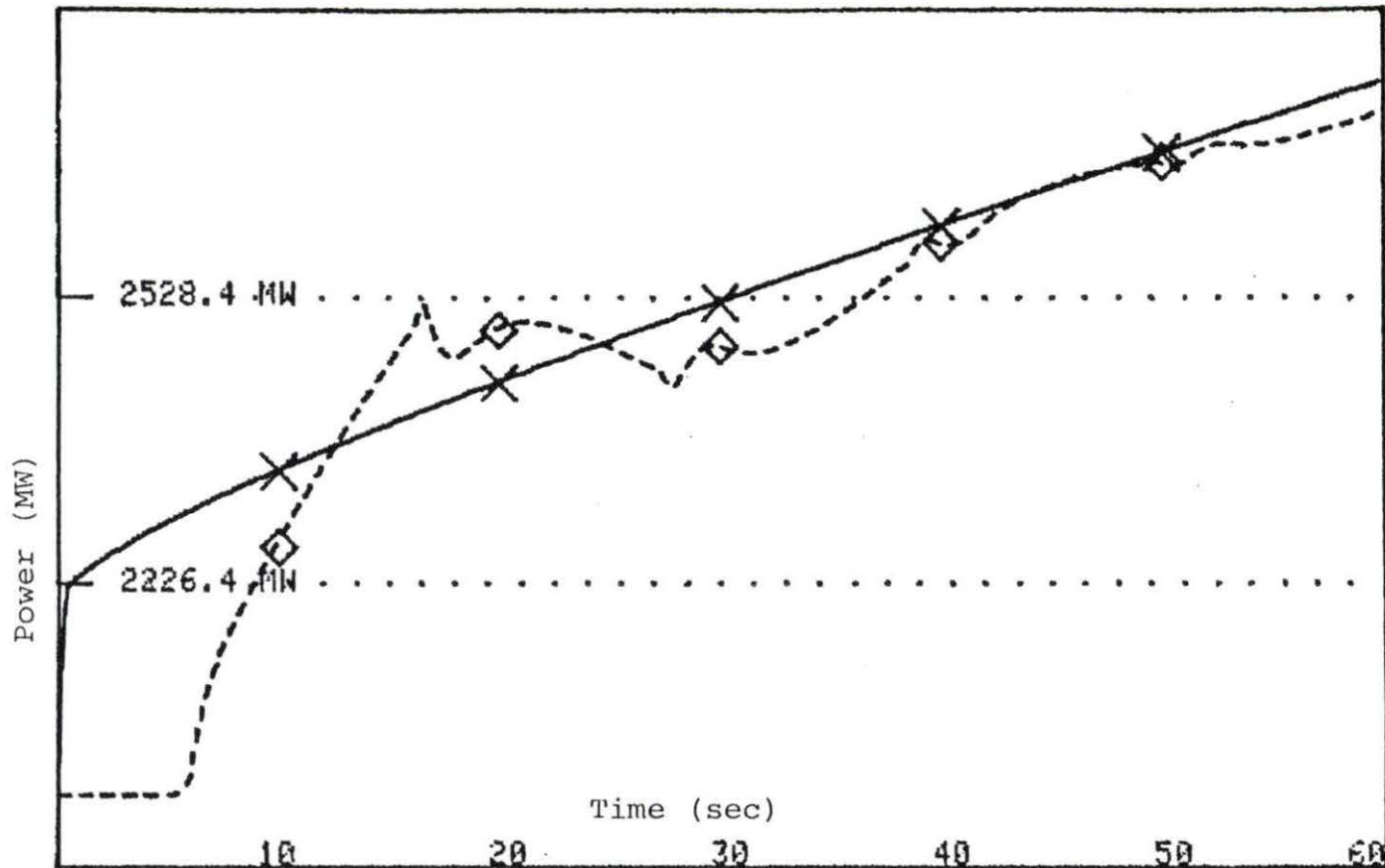


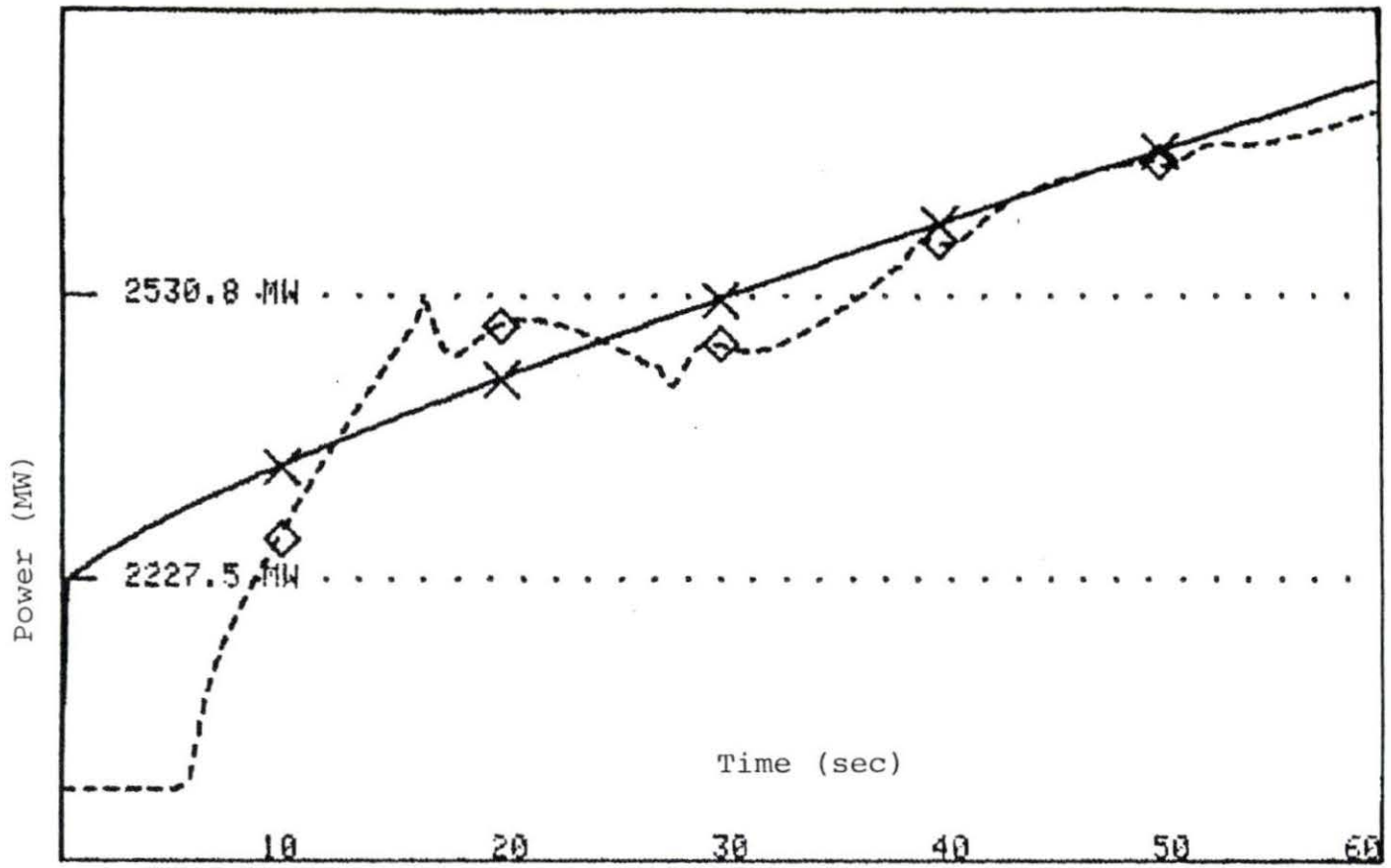
Figure 7. Reactor and steam generator power with ramp-input model.  
 Time of run 3 minutes



REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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





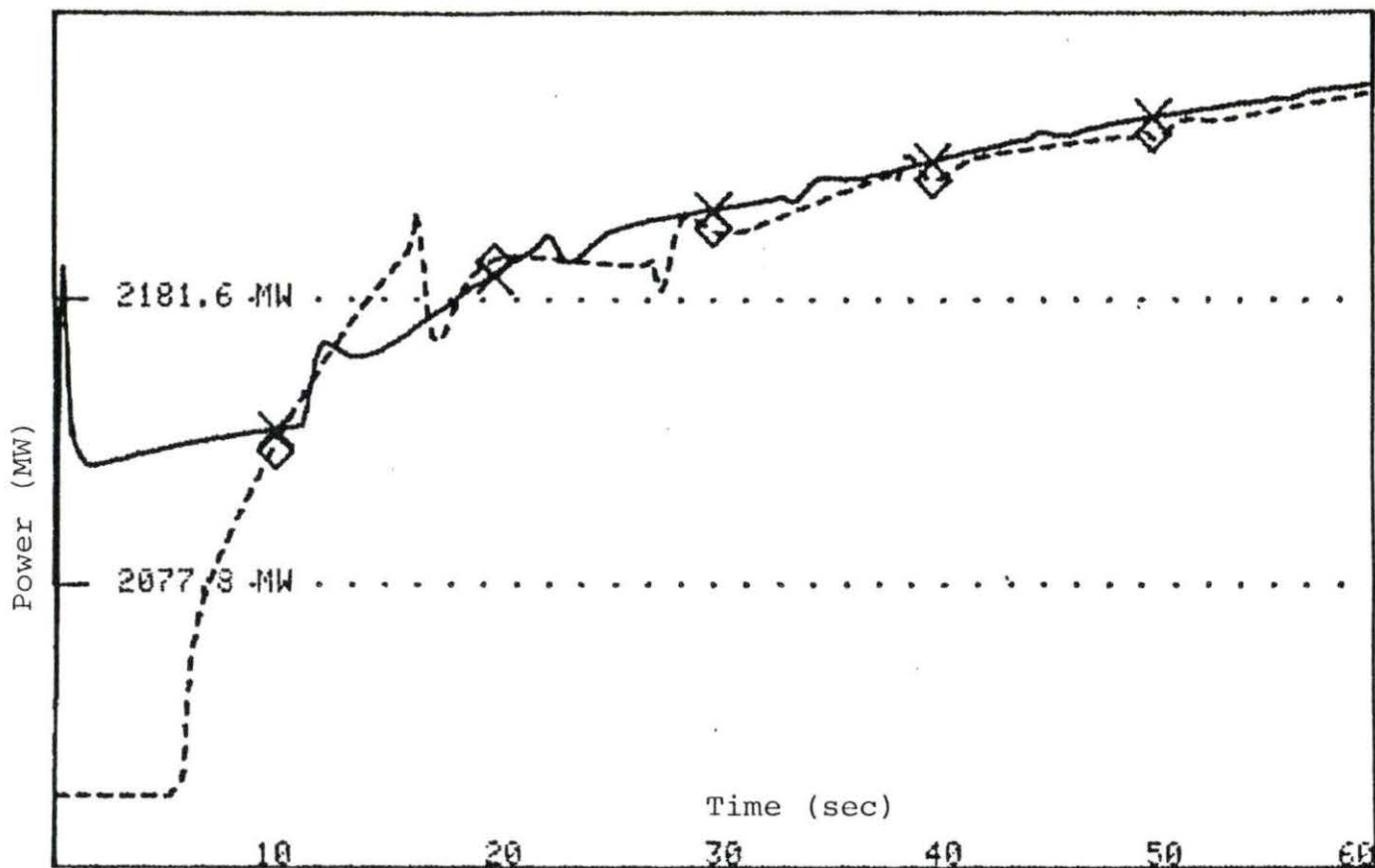
REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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

expected.

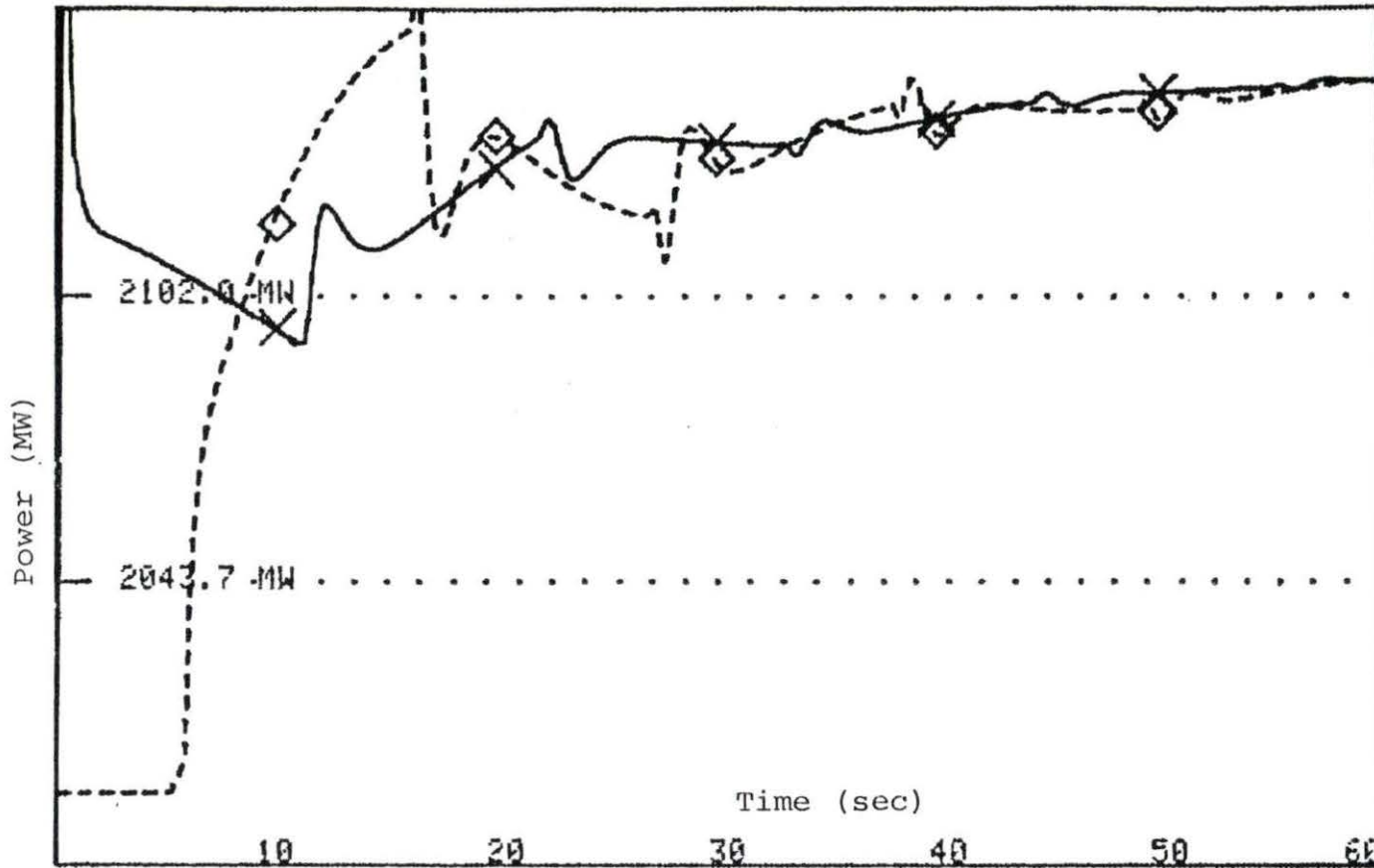
Figures 10 and 11 show the effects of a reactor control system. A step insertion of 2000 MW was used with a step insertion of 10 cents of reactivity. In Figure 10 a control system with a gain of  $-1.10^{-7}$  units of reactivity per degrees-second. In Figure 11, the gain is  $-1.10^{-6}$ . Comparison 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



REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT 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  $-1.0E-07$



REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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  $-1.0E-06$

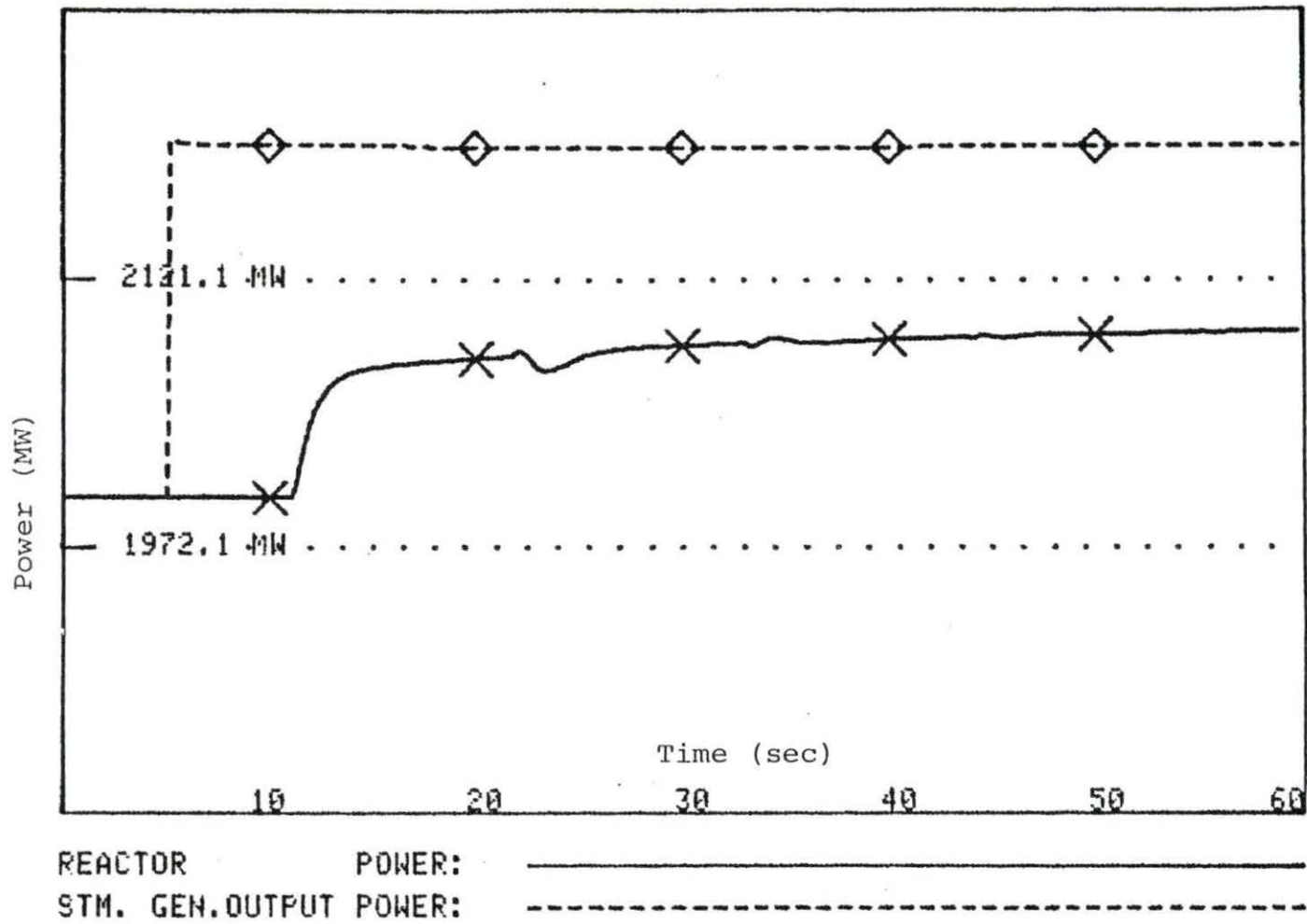


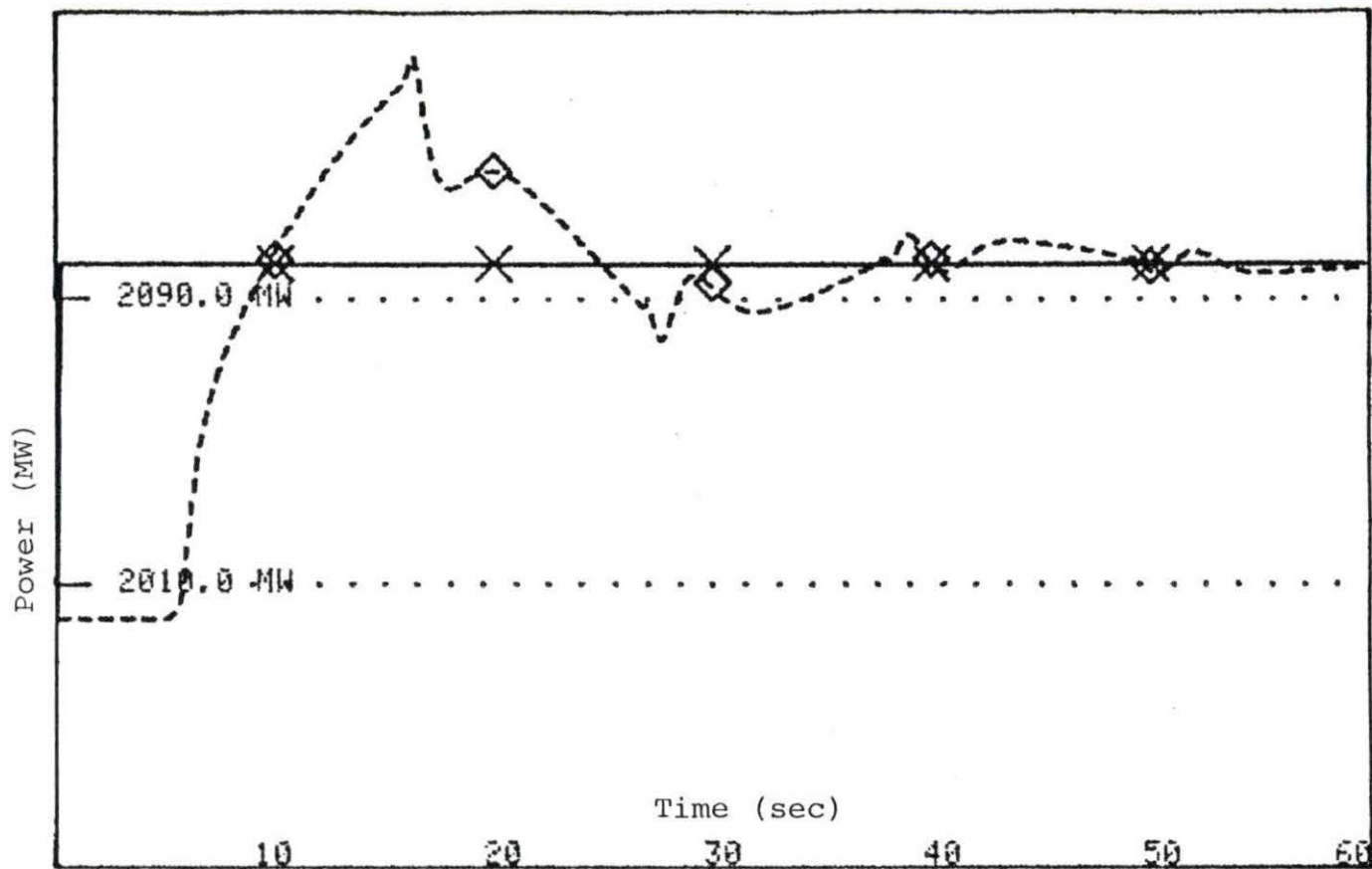
Figure 12. Reactor and steam generator power with load following and feedback. Initial power 2000 MW, throttle valve opened 10%, U-235 fuel

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.

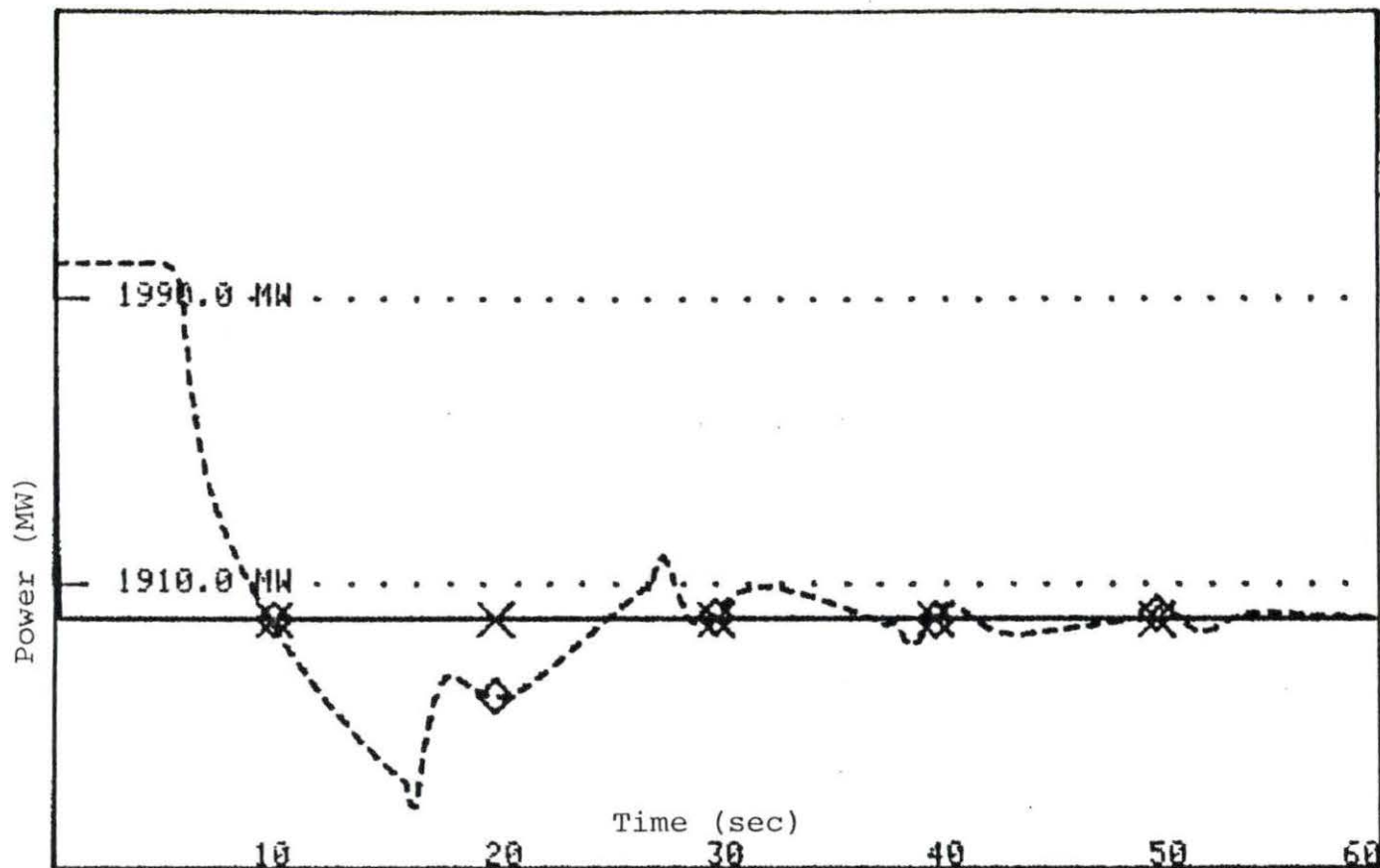
In Figure 13, reactor power jumps from 2000 MW to 2100 MW. After the five-second lag from reactor to steam generator, steam generator power rises to the reactor power level and exceeds it for a time. This reflects the fact that over a period of time, total steam generator energy output must equal reactor 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



REACTOR . POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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



REACTOR POWER: \_\_\_\_\_  
 STM. GEN. OUTPUT POWER: \_\_\_\_\_

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

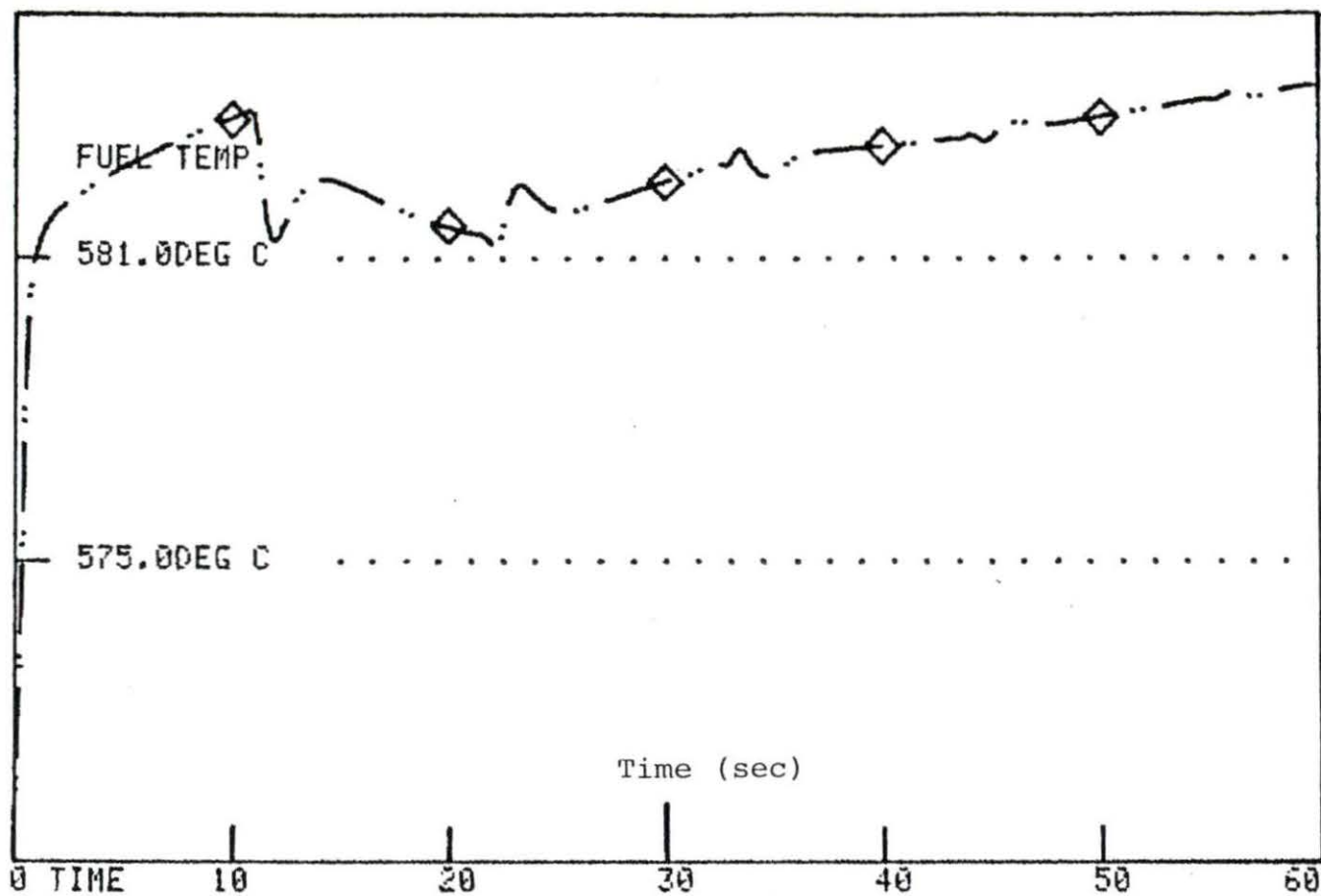
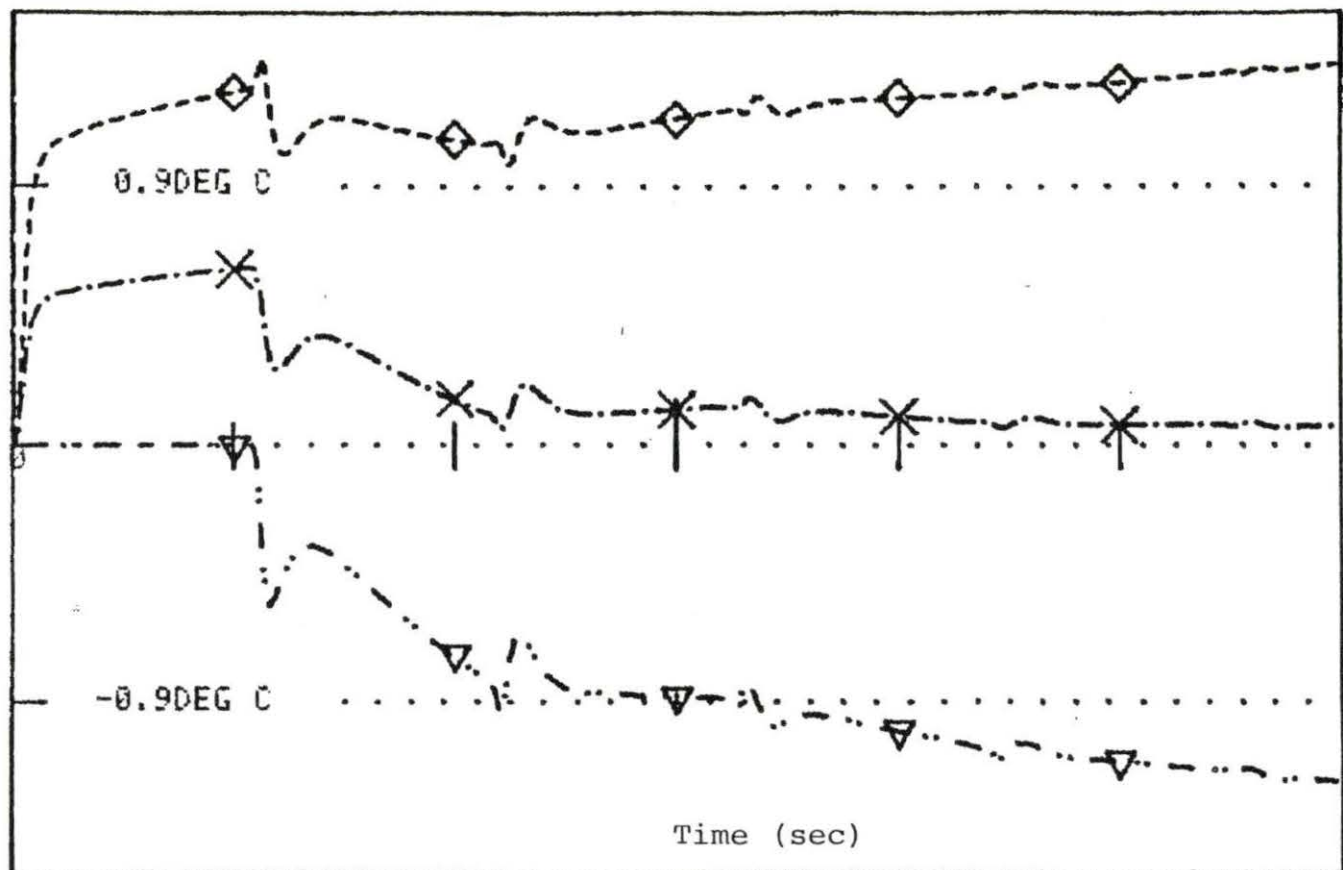


Figure 15. Fuel temperature change. Initial power 2000 MW, 10 cents reactivity step insertion, U-235 fuel



TIME 10 20 30 40 50 60  
 MOD TEMP CHANGE: .....  
 OUTLET TEMP CHANGE: .....  
 INLET TEMP CHANGE: .....

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 Tektronics 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 cassette tape labeled "MARK's EASY LOGON".

4. Press the AUTO LOAD key.

5. Eject the tape after the I/O 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 not 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 values and lower and upper limits

Parameter	Default value	Lower limit	Upper limit
Doppler coefficient of reactivity for U-233 ( $\alpha_f$ )	-1.07E-05 ( $\delta k/k$ )/C	-0.001 ( $\delta k/k$ )/C	+0.001 ( $\delta k/k$ )/C
Doppler coefficient of reactivity for U-235 ( $\alpha_f$ )	-2.61E-05 ( $\delta k/k$ )/C	-0.001 ( $\delta k/k$ )/C	+0.001 ( $\delta k/k$ )/C
Doppler coefficient of reactivity for Pu-239 ( $\alpha_f$ )	-0.85E-05 ( $\delta k/k$ )/C	-0.001 ( $\delta k/k$ )/C	+0.001 ( $\delta k/k$ )/C
Moderator coefficient of reactivity ( $\alpha_m$ )	-8.6E-06 ( $\delta k/k$ )/C	-0.0001 ( $\delta k/k$ )/C	+0.0001 ( $\delta k/k$ )/C
Pre-perturbation reactivity ( $\rho_0$ )	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 ( $\tau$ )	5 seconds	2 seconds	20 seconds
Control system time constant ( $\tau_c$ )	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

```

00025 0001 C MAIN PROGRAM NERO
00050 0002 C
00075 0003 C THIS IS THE MAIN PROGRAM NERO. IN THIS PROGRAM THE USER SE-
00100 0004 C LECTS THE PARAMETERS AND OPTIONS TO FIT THE SYSTEM WHOSE BEHAVIOR
00125 0005 C HE OR SHE WISHES TO EXAMINE. SOME OF THESE CHOICES ARE INCLUDED IN
00150 0006 C THE COMMON BLOCK AND CONTROL THE BEHAVIOR OF THE SUBROUTINES.
00175 0007 C OTHERS CONTROL THE FORMAT OF THE OUTPUT.
00200 0008 C
00225 0009 C THE SUBROUTINE GALBA CONTROLS REACTOR KINETICS.
00250 0010 C THE SUBROUTINE OTHO CONTROLS HEAT TRANSFER IN THE REACTOR
00275 0011 C CORE.
00300 0012 C THE SUBROUTINE VESPASIAN CONTROLS OUTPUT GRAPHICS.
00325 0013 C THE SUBROUTINE DMTN CONTROLS THE STEAM GENERATOR MODEL.
00350 0014 C THE SUBROUTINE EIGEN AND THE SUBROUTINE CLINQ ARE PORTLIBRARY
00375 0015 C SUBROUTINES USED IN GALBA TO COMPUTE
00400 0016 C EIGENVALUES, AND FUNDAMENTAL EIGEN-
00425 0017 C VECTOR MATRICES AND THEIR INVERSES.
00450 0018 C
00475 0019 C ALL DIMENSIONS ARE DEFINED IN THE SI SYSTEM OF MEASUREMENTS.
00500 0020 C
00525 0021 C
00550 0022 C THE FOLLOWING VARIABLES ARE USED IN THE COMMON BLOCK:
00575 0023 C
00600 0024 C AA IS THE GAIN OF THE REACTOR CONTROL SYSTEM.
00625 0025 C ALPHF IS THE DOPPLER COEFFICIENT OF REACTIVITY.
00650 0026 C ALPHM IS THE MODERATOR COEFFICIENT OF REACTIVITY.
00675 0027 C B IS THE 1X6 ARRAY FOR THE 6 GROUPS OF DELAYED-NEUTRON
00700 0028 C PRECURSORS.
00725 0029 C CPPAV IS THE AVERAGE HEAT CAPACITY OF THE REACTOR COOLANT.
00750 0030 C IT IS SLIGHTLY A FUNCTION OF REACTOR POWER.
00775 0031 C CS IS THE VARIABLE WHOSE VALUE, SELECTED BY THE USER,
00800 0032 C DETERMINES A REACTOR CONTROL SYSTEM IS USED.
00825 0033 C DN1 IS THE DIFFERENCE BETWEEN CURRENT REACTOR POWER AND
00850 0034 C THE INITIAL (STEADY-STATE) POWER.
00875 0035 C DRO IS THE DIFFERENCE BETWEEN CURRENT REACTIVITY AND
00900 0036 C THE INITIAL (STEADY-STATE) REACTIVITY.
00925 0037 C DTF IS THE DIFFERENCE BETWEEN CURRENT FUEL TEMPERATURE AND
00950 0038 C THE INITIAL (STEADY-STATE) TEMPERATURE.
00975 0039 C DTI IS THE DIFFERENCE BETWEEN CURRENT REACTOR INLET TEMP.
01000 0040 C AND INITIAL (STEADY-STATE) TEMPERATURE.
01025 0041 C DTM IS THE DIFFERENCE BETWEEN CURRENT MODERATOR TEMPERATURE
01050 0042 C AND THE INITIAL (STEADY-STATE) TEMPERATURE.
01075 0043 C DTO IS THE DIFFERENCE BETWEEN CURRENT REACTOR OUTLET TEMP.
01100 0044 C AND INITIAL (STEADY-STATE) TEMPERATURE.
01125 0045 C DTS IS THE DIFFERENCE BETWEEN CURRENT TEMPERATURE OF THE
01150 0046 C SECONDARY SIDE OF THE STEAM GENERATOR AND THE
01175 0047 C INITIAL (STEADY-STATE) TEMPERATURE.
01200 0048 C DTSG IS THE DIFFERENCE BETWEEN CURRENT TEMPERATURE OF THE
01225 0049 C PRIMARY SIDE OF THE STEAM GENERATOR AND THE
01250 0050 C INITIAL (STEADY-STATE) TEMPERATURE.
01275 0051 C EIG IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
01300 0052 C DETERMINES WHETHER THE TWO-TIME-SCALE MATRIX
01325 0053 C DECOUPLING ALGORITHM IS USED IN GALBA TO
01350 0054 C COMPUTE REACTOR KINETICS.
01375 0055 C H IS THE TIME STEP USED IN THE SUBROUTINES. IS ALLOWED
01400 0056 C TO VARY FROM SUBROUTINE TO SUBROUTINE.
01425 0057 C IS IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE

```

01450	0058	C		DETERMINES WHICH ISOTOPE OF FISSILE MATERIAL
01475	0059	C		WILL BE USED. ISOTOPES USED CAN INCLUDE U-233,
01500	0060	C		U-235, PU-239.
01525	0061	C	K	IS THE SUBSCRIPT OF THE ARRAYS USED IN DEPICTING
01550	0062	C		OUTPUT, WHETHER IN A TABLE OR IN GRAPHICS. IT
01575	0063	C		REPRESENTS TIME INCREMENTS OF 0.2 SECONDS. IT
01600	0064	C		IS ALSO USED IN SOME SUBROUTINES AS A "TIMER",
01625	0065	C		WHOSE VALUE CAN TRIGGER CERTAIN EVENTS.
01650	0066	C	KF	IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
01675	0067	C		DETERMINES WHETHER NO REACTOR CONTROL SYSTEM
01700	0068	C		AND NO REACTOR FEEDBACK IS USED.
01725	0069	C	LMB	IS THE 1X6 ARRAY FOR THE 6 GROUPS OF DELAYED-NEUTRON
01750	0070	C		DECAY COEFFICIENTS.
01775	0071	C	MDOTP	IS THE MASS FLOW RATE OF REACTOR COOLANT. IT IS
01800	0072	C		SLIGHTLY A FUNCTION OF INITIAL REACTOR POWER.
01825	0073	C	NK	IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
01850	0074	C		DETERMINES WHETHER REACTOR KINETICS IS USED,
01875	0075	C		OR WHETHER A SIMPLE POWER STEP IS POSTULATED.
01900	0076	C	NN	IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
01925	0077	C		DETERMINES WHETHER THE OUTPUT IS IN A TABLE
01950	0078	C		OR IN GRAPHICS.
01975	0079	C	N10	IS THE INITIAL POWER LEVEL OF THE REACTOR.
02000	0080	C	PJ	IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
02025	0081	C		DETERMINES WHETHER THE PROMPT-JUMP APPROXIMATION
02050	0082	C		IS USED OR NOT.
02075	0083	C	RI	IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE
02100	0084	C		DETERMINES WHETHER THE RAMP-INPUT MODEL IS USED
02125	0085	C		OR NOT.
02150	0086	C	RIR	IS THE RAMP-INPUT RATE.
02175	0087	C	RHO	IS THE INITIAL (STEADY-STATE) AMOUNT OF REACTIVITY.
02200	0088	C	RP	IS THE TOTAL REACTOR POWER AT ANY GIVEN TIME. THIS IS
02225	0089	C		USED IN DMTN AS A CRITERION FOR DECIDING HOW
02250	0090	C		MUCH TO ALTER SECONDARY SIDE FLUID FLOW IN ORDER
02275	0091	C		COUNTERACT ANY PERTURBATION CAUSED BY ALTERED
02300	0092	C		REACTOR POWER.
02325	0093	C	STMGEN	IS THE STEAM GENERATOR OUTPUT POWER.
02350	0094	C	TAU	IS ONE OF THE REACTOR CONTROL SYSTEM (RCS) PARAMETERS.
02375	0095	C		THIS ONE IS USED TO ADJUST THE INTEGRAL
02400	0096	C		CONTROLLER.
02425	0097	C	TAUC	IS ONE OF THE RCS PARAMETERS. THIS ONE IS USED TO
02450	0098	C		ADJUST THE DIFFERENTIAL CONTROLLER.
02475	0099	C	TF1	IS THE INITIAL FUEL TEMPERATURE.
02500	0100	C	TM1	IS THE INITIAL MODERATOR TEMPERATURE. ALWAYS EQUALS
02525	0101	C		313.89 DEG. C.
02550	0102	C	TT	IS THE TOTAL TIME. IT IS AN ARRAY WHOSE VALUES FORM
02575	0103	C		THE INDEPENDENT VARIABLE (THE "X-AXIS") IN GRAPHICS
02600	0104	C		AND IS PRINTED IN THE TABULAR OUTPUT AS WELL.
02625	0105	C	TTOT	IS THE TOTAL AMOUNT OF TIME THAT THE RAMP INPUT IS IN
02650	0106	C		EFFECT.
02675	0107	C	VO	IS THE FRACTIONAL AMOUNT OF OPENING OF THE THROTTLE
02700	0108	C		VALVE ON THE SECONDARY SIDE OF THE STEAM
02725	0109	C		GENERATOR.
02750	0110	C	VOSS	IS THE FRACTIONAL AMOUNT OF THE TOTAL OPENING OF THE
02775	0111	C		THROTTLE VALVE ON THE SECONDARY SIDE OF THE
02800	0112	C		STEAM GENERATOR, BEFORE ANY CHANGES ARE APPLIED.
02825	0113	C	Y	IS THE NUMBER THAT CHANGES BY ONE EVERY 0.2 SECONDS.
02850	0114	C		ITS FUNCTION IS THAT IT ALLOWS INITIALIZATIONS



02875	0115	C			
02900	0116	C			
02925	0117	C			
02950	0118	C			
02975	0119	C			
03000	0120	C			
03025	0121	C			
03050	0122	C			
03075	0123	C			
03100	0124	C			
03125	0125	C			
03150	0126	C			
03175	0127	C			
03200	0128	C			
03225	0129	C			
03250	0130	C			
03275	0131	C			
03300	0132	C			
03325	0133	C			
03350	0134	C			
03375	0135	C			
03400	0136	C			
03425	0137	C			
03450	0138	C			
03475	0139	C			
03500	0140	C			
03525	0141	C			
03550	0142	C			
03575	0143	C			
03600	0144	C			
03625	0145	C			
03650	0146	C			
03675	0147	C			
03700	0148	C			
03725	0149	C			
03750	0150	C			
03775	0151	C			
03800	0152	C			
03825	0153	C			
03850	0154	C			
03875	0155	C			
03900	0156	C			
03925	0157	C			
03950	0158	C			
03975	0159	C			
04000	0160	C			
04025	0161	C			
04050	0162	C			
04075	0163	C			
04100	0164	C			
04125	0165	C			
04150	0166	C			
04175	0167	C			
04200	0168	C			
04225	0169	C			
04250	0170	C			
04275	0171	C			

TO BE MADE IN GALBA DURING THE FIRST 0.2 SECONDS ONLY.  
 ZZ IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE DETERMINES WHETHER THROTTLE VALVE POSITION IS TO BE PERMANENTLY FIXED, AFTER IUT IS PERTURBED FROM SOME STEADY-STATE VALUE.

THE FOLLOWING REAL VARIABLES ARE USED IN NERO:

A IS THE SUMMATION OF THE 6 GROUPS OF DELAYED-NEUTRON PRECURSORS.  
 ABN IS THE TIME INTERVAL AT WHICH TABULAR OUTPUT IS PRINTED.  
 ALPHFS IS THE VARIABLE THAT RETAINS THE PREVIOUS VALUE FOR ALPHF, IN CASE THE USER LATER DECIDES NOT TO CHANGE IT AFTER ALL.  
 ALPHMS IS THE VARIABLE THAT RETAINS THE PREVIOUS VALUE FOR ALPHM, IN CASE THE USER LATER DECIDES NOT TO CHANGE IT AFTER ALL.  
 CPM IS THE HEAT CAPACITY OF THE REACTOR COOLANT, USED IN COMPUTING THE HEAT TRANSFER COEFFICIENT.  
 DRON IS THE REACTIVITY PERTURBATION, IN CENTS.  
 DVO IS THE VALVE PERTURBATION ON THE SECONDARY SIDE OF THE STEAM GENERATOR, IN PERCENT. AFTER CONVERSION TO A DECIMAL, IT IS ADDED TO THE STEADY-STATE QUANTITY.  
 FRAC IS THE VARIABLE USED IN DETERMINING THE REACTOR COOLANT TEMPERATURE SO THAT THE HEAT TRANSFER COEFFICIENT CAN BE COMPUTED. IT ABOUT EQUALS 0.5, BUT VARIES SLIGHTLY BECAUSE HEAT CAPACITY DOES NOT VARY LINEARLY WITH TEMPERATURE.  
 HP IS THE HEAT TRANSFER COEFFICIENT FOR THE REACTOR COOLANT.  
 KP IS THE THERMAL CONDUCTIVITY OF THE REACTOR COOLANT.  
 LNT IS THE TOTAL TIME THAT THE PROGRAM WILL RUN, IN MINUTES  
 MUP IS THE DYNAMIC VISCOSITY OF THE REACTOR COOLANT  
 NU IS THE NUSSELT NUMBER OF THE REACTOR COOLANT.  
 NUP IS THE KINEMATIC VISCOSITY OF THE REACTOR COOLANT.  
 N110 IS THE TOTAL REACTOR POWER, IN MW, AND IS PRINTED IN THE TABULAR OUTPUT.  
 PROD IS THE TOTAL AMOUNT OF REACTIVITY IN THE RAMP-INPUT MODEL, IN CENTS.  
 PRP IS THE PRANDTL NUMBER OF THE REACTOR COOLANT.  
 REP IS THE REYNOLDS NUMBER OF THE REACTOR COOLANT.  
 RHON IS THE PRE-PERTURBATION REACTIVITY (USUALLY ZERO), IN CENTS.  
 ROP IS THE DENSITY OF THE REACTOR COOLANT.  
 STMG IS THE CURRENT STEAM GENERATOR OUTPUT, IN MW. IT IS WHAT IS PRINTED IN THE TABULAR OUTPUT.  
 TC IS THE AVERAGE CLADDING SURFACE TEMPERATURE.  
 TF IS THE AVERAGE FUEL TEMPERATURE.  
 T11 IS THE INITIAL REACTOR INLET TEMPERATURE.  
 T01 IS THE INITIAL REACTOR OUTLET TEMPERATURE.  
 TOTH0 IS THE TIME MEASURE DETERMINING WHETHER NERO WILL PASS CONTROL FROM OTH0 TO GALBA. AT THE END OF EVERY 0.2 SECOND INTERVAL, IT WILL NOT; AT ALL OTHER TIMES IT WILL. THIS IS SO ACCURATE

```

04300 0172 C
04325 0173 C
04350 0174 C
04375 0175 C
04400 0176 C
04425 0177 C
04450 0178 C
04475 0179 C
04500 0180 C
04525 0181 C
04550 0182 C
04575 0183 C
04600 0184 C
04625 0185 C
04650 0186 C
04675 0187 C
04700 0188 C
04725 0189 C
04750 0190 C
04775 0191 C
04800 0192 C
04825 0193 C
04850 0194 C
04875 0195 C
04900 0196 C
04925 0197 C
04950 0198 C
04975 0199 C
05000 0200 C
05025 0201 C
05050 0202 C
05075 0203 C
05100 0204 C
05125 0205 C
05150 0206 C
05175 0207 C
05200 0208 C
05225 0209 C
05250 0210 C
05275 0211 C
05300 0212 C
05325 0213 C
05350 0214 C
05375 0215 C
05400 0216 C
05425 0217 C
05450 0218 C
05475 0219 C
05500 0220 C
05525 0221 C
05550 0222 C
05575 0223 C
05600 0224 C
05625 0225 C
05650 0226 C
05675 0227 C
05700 0228 C

```

REACTIVITY CALCULATIONS AFFECTING FEEDBACK CAN BE MADE.

TM IS THE AVERAGE FUEL MODERATOR TEMPERATURE. IT IS PRINTED IN TABULAR OUTPUT.

TW IS THE INTERVAL AT WHICH TABULAR OUTPUT IS PRINTED.

VMOD IS THE VELOCITY OF FLUID FLOW WHILE IN THE CORE.

THE FOLLOWING ARE INTEGER VARIABLES:

AB IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE DETERMINES WHETHER THE OUTPUT ABBREVIATION OPTION IS TO BE USED.

ALA IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE DETERMINES WHETHER ANY OR ALL OF THE REACTIVITY COEFFICIENTS WILL BE VARIED.

CSC IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE DETERMINES WHETHER THERE IS TO BE A CHANGE IN ANY OF THE REACTOR CONTROL SYSTEM PARAMETERS.

CSCG IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE DETERMINES WHETHER THE GAIN PARAMETER OF THE RCS WILL BE ALTERED.

CSCG IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE DETERMINES WHETHER THE GAIN PARAMETER WILL BE ALTERED.

CSCG IS THE VARIABLE, SELECTED BY THE USER, WHOSE VALUE DETERMINES WHETHER THE GAIN PARAMETER WILL BE ALTERED.

REAL AA, ALPHF, ALPHM, ALPHP, CPPAV, DN1, DRO, H, MDOTP, N10, RHO, RIR, STMG, 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, STMG, 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, STMG, TC, TF, T11, T01, TOHO, TM, TW, VMOD

+ INTEGER AB, ABC, ALA, CSC, CSCG, CSCT, CSCT, I, LN, LNT, P, YY

A = 0.0

GENERAL COMMENTS

THE PURPOSE OF THE MAIN PROGRAM NERO IS TO CONTROL THE OPERATION OF THE OTHER PROGRAMS, CONTROL THE FORMAT OF THE OUTPUT, AND ESTABLISH SOME OF THE INITIAL VALUES OF THE OPERATING PARAMETERS, AT THE USER'S OPTION.

NERO TYPICALLY OPERATES IN THE FOLLOWING MANNER: INSTRUCTIONS TO CHOOSE A PARAMETER ARE ISSUED BY NERO. IF THE SIGNIFICANCE OF

```

05725 0229 C THE PARAMETERS IS NOT IMMEDIATELY OBVIOUS. AN EXPLANATION S PRO-
05750 0230 C VIDED. THEN THE USER IS INSTRUCTED TO CHOOSE WHETHER HE OR SHE
05775 0231 C WANTS THE OPTION PRESENTED OR NOT. HE OR SHE MUST MAKE A CHOICE
05800 0232 C OR NERO WILL CHOOSE AN OPTION OR A VALUE BY DEFAULT. THEN A TEST
05825 0233 C IS APPLIED. IF THE USER'S CHOICE MEETS CERTAIN ELIGIBILITY CRI-
05850 0234 C TERIA, (THAT IS, IF THE CHOICE IS WITHIN PERMISSIBLE LIMITS)
05875 0235 C NERO JUMPS TO THE NEXT SET OF INSTRUCTIONS. IF THE CRITERIA ARE
05900 0236 C NOT MET, NERO PRINTS A MESSAGE TO THAT EFFECT, AND JUMPS BACK-
05925 0237 C WARD AND MAKES THE USER CHOOSE AGAIN.
05950 0238 C IN CHOOSING AN OPTION, WHEN THE USER TYPES IN 1, THE OPTION
05975 0239 C IS ELECTED; WHEN 2 IS TYPED IN, THE OPTION IS NOT SELECTED.
06000 0240 C
06025 0241 C
06050 0242 C HERE, THE CHOICE OF ISOTOPE IS MADE. THE USER MAY CHOOSE
06075 0243 C EITHER U-235, PU-239, OR U-233.
06100 0244 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
06500 0260 C
06525 0261 C
06550 0262 C HERE THE DEFAULT PARAMETERS OF THE REACTOR CONTROL SYSTEM AND
06575 0263 C THE MODERATOR COEFFICIENT OF REACTIVITY ARE ESTABLISHED.
06600 0264 C
06625 0265 C
06650 0266 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
06875 0275 C
06900 0276 C
06925 0277 C HERE THE SIX GROUPS OF DELAYED PRECURSORS AND DECAY CONSTANTS
06950 0278 C AND THE DEFAULT VALUE OF THE DOPPLER COEFFICIENT OF REACTIVITY
06975 0279 C ARE INITIALIZED, DEPENDING ON WHICH ISOTOPE WAS CHOSEN EARLIER.
07000 0280 C
07025 0281 C
07050 0282 C
07075 0283 C
07100 0284 C
07125 0285 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

```

```

07150 0286      IF (IS .EQ. 1) B(6) = 9.0440E-05
07175 0287      IF (IS .EQ. 1) ALPHF = -1.07E-05
07200 0288      IF (IS .EQ. 2) B(1) = 2.1450E-04
07225 0289      IF (IS .EQ. 2) B(2) = 1.4235E-03
07250 0290      IF (IS .EQ. 2) B(3) = 1.2740E-03
07275 0291      IF (IS .EQ. 2) B(4) = 2.5675E-03
07300 0292      IF (IS .EQ. 2) B(5) = 7.4750E-04
07325 0293      IF (IS .EQ. 2) B(6) = 2.7300E-04
07350 0294      IF (IS .EQ. 2) ALPHF = -2.61E-05
07375 0295      IF (IS .EQ. 3) B(1) = 7.4200E-05
07400 0296      IF (IS .EQ. 3) B(2) = 6.3176E-04
07425 0297      IF (IS .EQ. 3) B(3) = 4.4732E-04
07450 0298      IF (IS .EQ. 3) B(4) = 6.9112E-04
07475 0299      IF (IS .EQ. 3) B(5) = 1.8232E-04
07500 0300      IF (IS .EQ. 3) B(6) = 9.3280E-05
07525 0301      IF (IS .EQ. 3) ALPHF = -0.85E-05
07550 0302
07575 0303
07600 0304
07625 0305
07650 0306
07675 0307
07700 0308
07725 0309
07750 0310
07775 0311
07800 0312
07825 0313
07850 0314
07875 0315
07900 0316
07925 0317
07950 0318
07975 0319
08000 0320
08025 0321
08050 0322
08075 0323
08100 0324
08125 0325
08150 0326
08175 0327
08200 0328
08225 0329
08250 0330
08275 0331
08300 0332
08325 0333
08350 0334
08375 0335
08400 0336
08425 0337
08450 0338
08475 0339
08500 0340
08525 0341
08550 0342

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)

```

```

08575 0343 180 FORMAT (X, '          TO TRY AGAIN.')
```

```

08600 0344      WRITE (6, 190)
```

```

08625 0345 190 FORMAT(' ')
```

```

08650 0346      GO TO 90
```

```

08675 0347
```

```

08700 0348
```

```

08725 0349 200 IF (KF .EQ. 1) GO TO 1590
```

```

08750 0350
```

```

08775 0351      WRITE (6, 210)
```

```

08800 0352 210 FORMAT (X, '          WHAT LEVEL (IN CENTS) PRE-PERTURBATION
```

```

08825 0353      + REACTIVITY IS DESIRED?')
```

```

08850 0354      WRITE (6, 220)
```

```

08875 0355 220 FORMAT (X, '          USE FORMAT F5.1    EXAMPLE: -05.0 EQUALS
```

```

08900 0356      + -5 CENTS.')
```

```

08925 0357      READ 230, RHON
```

```

08950 0358 230 FORMAT (F4.1)
```

```

08975 0359      IF ((RHON .GT. -100.0) .AND. (RHON .LT. 100.0)) GO TO 270
```

```

09000 0360
```

```

09025 0361
```

```

09050 0362      WRITE (6, 240)
```

```

09075 0363 240 FORMAT (X, '          YOU HAVE SELECTED A VALUE THAT WILL
```

```

09100 0364      + RESULT IN A PROMPT SUPERCRITICAL')
```

```

09125 0365      WRITE (6, 250)
```

```

09150 0366 250 FORMAT (X, '          CONDITION. THIS IS NOT ALLOWED, AND YOU
```

```

09175 0367      + WILL HAVE TO TRY AGAIN.')
```

```

09200 0368      WRITE (6, 260)
```

```

09225 0369 260 FORMAT (' ')
```

```

09250 0370      GO TO 200
```

```

09275 0371
```

```

09300 0372 C
```

```

09325 0373 C      THIS EQUATION CONVERTS PRE-PERTURBATION REACTIVITY FROM
```

```

09350 0374 C "CENTS" TO "((DK/K)/C)".
```

```

09375 0375 C
```

```

09400 0376 270          RHO = 0.01 * RHON * A
```

```

09425 0377
```

```

09450 0378 C
```

```

09475 0379 C      THIS SECTION EXPLAINS REACTIVITY COEFFICIENTS.
```

```

09500 0380 C
```

```

09525 0381      WRITE (6, 280)
```

```

09550 0382 280 FORMAT (X, '          DOPPLER          COEFFICIENT OF
```

```

09575 0383      + REACTIVITY IS')
```

```

09600 0384      WRITE (6, 290) ALPHF
```

```

09625 0385 290 FORMAT (X, '          ', E10.3, 2X, '(dk / k) / C')
```

```

09650 0386      WRITE (6, 300)
```

```

09675 0387 300 FORMAT (X, '          MODERATOR TEMPERATURE COEFFICIENT OF
```

```

09700 0388      + REACTIVITY IS', 2X, E10.3)
```

```

09725 0389      WRITE (6, 310) ALPHM
```

```

09750 0390 310 FORMAT (X, '          ', E10.3, 2X, '(dk / k) / C')
```

```

09775 0391      IF (KF .EQ. 1) GO TO 920
```

```

09800 0392
```

```

09825 0393      WRITE (6, 320)
```

```

09850 0394 320 FORMAT (X, '          NOTE: REACTIVITY COEFFICIENTS AS GIVEN
```

```

09875 0395      + ABOVE ARE TYPICAL ONES.')
```

```

09900 0396      WRITE (6, 330)
```

```

09925 0397 330 FORMAT (X, '          ADDITIONALLY, SINCE THEY ARE NEGATIVE
```

```

09950 0398      + NUMBERS, FEEDBACK WILL ALSO')
```

```

09975 0399      WRITE (6, 340)
```

```

10000 0400 340 FORMAT (X, ' BE NEGATIVE. UNLESS YOU WANT TO CHANGE
10025 0401 + THE VALUES OF THESE')
10050 0402 WRITE (6, 350)
10075 0403 350 FORMAT (X, ' COEFFICIENTS, THEY WILL KEEP THE VALUES
10100 0404 + ALREADY STATED. THE DOPPLER')
10125 0405 WRITE (6, 360)
10150 0406 360 FORMAT (X, ' COEFFICIENT TYPICALLY VARIES FROM
10175 0407 + -2.0E-05 TO 3.6E-05 FOR U-235.')
10200 0408 WRITE (6, 370)
10225 0409 370 FORMAT (X, ' FOR OTHER FUELS IT IS SOMEWHAT LESS.
10250 0410 + THE MODERATOR TEMPERATURE')
10275 0411 WRITE (6, 390)
10300 0412 390 FORMAT (X, ' COEFFICIENT, WHICH IN THIS CASE IS A
10325 0413 + COMBINED TEMPERATURE AND')
10350 0414 WRITE (6, 400)
10375 0415 400 FORMAT (X, ' PRESSURE COEFFICIENT, TYPICALLY RANGES
10400 0416 + FROM -3.2E-04 TO 1.7E-04')
10425 0417 WRITE (6, 410)
10450 0418 410 FORMAT (X, ' (dK / K) / C')
10475 0419 WRITE (6, 420)
10500 0420 420 FORMAT (' ')
10525 0421
10550 0422 C HERE THE USER SELECTS A VALUE FOR PRE-PERTURBATION REACTIVITY.
10575 0423 C (RHON) USUALLY IT WILL BE ZERO. HOWEVER, IT IS NOT ALLOWED TO BE
10600 0424 C EITHER LESS THAN -90.0 CENTS, OR GREATER THAN + 90.0 CENTS, AS
10625 0425 C THIS IS CLOSE TO A PROMPT CRITICALITY CONDITION.
10650 0426 C
10675 0427 430 WRITE (6, 440)
10700 0428 440 FORMAT (X, ' IF YOU WISH TO INSERT YOUR OWN
10725 0429 + REACTIVITY COEFFICIENTS, TYPE IN 1;')
10750 0430 WRITE (6, 450)
10775 0431 450 FORMAT (X, ' IF NOT, TYPE IN 2')
10800 0432 READ 460, ALA
10825 0433 460 FORMAT (I1)
10850 0434
10875 0435 IF (ALA .EQ. 2) GO TO 630
10900 0436 IF (ALA .EQ. 1) GO TO 490
10925 0437
10950 0438 C
10975 0439 C HERE THE USER CHOOSES WHETHER A CHANGE IN REACTIVITY COEFFI-
11000 0440 C CIENTS IS DESIRED. IF NO CHANGE IS DESIRED, THE DEFAULT VALUES
11025 0441 C WILL REMAIN. IF CHANGE IS DESIRED, THE PROGRAM WILL ENABLE THE
11050 0442 C USER TO SELECT DESIRED VALUE(S).
11075 0443 C
11100 0444 WRITE (6, 470)
11125 0445 470 FORMAT (X, ' YOU HAVE TYPED IN A NUMBER THAT CANNOT
11150 0446 + BE UTILIZED. YOU WILL HAVE')
11175 0447 WRITE (6, 480)
11200 0448 480 FORMAT (X, ' TO TRY AGAIN.')
11225 0449
11250 0450 GO TO 430
11275 0451
11300 0452 490 WRITE (6, 500)
11325 0453 500 FORMAT (X, ' TYPE IN DESIRED DOPPLER COEFFICIENT OF
11350 0454 + REACTIVITY, USING FORMAT')
11375 0455 WRITE (6, 510)
11400 0456 510 FORMAT (X, ' E10.3: EX: A REACTIVITY OF -8.61E-05

```

```

11425 0457      + WOULD BE WRITTEN AS')
11450 0458      WRITE (6, 520) ALPHF
11475 0459      520  FORMAT (X, '          -0.861E-04:  CURRENT VALUE IS', 2X,E10.3)
11500 0460
11525 0461      C
11550 0462      C      ALPHFS AND ALPHMS ARE HOLDING VARIABLES OF THE DEFAULT VALUES
11575 0463      C      OF THE REACTIVITY COEFFICIENTS.  IF THE USER FIRST SELECTS IM-
11600 0464      C      PERMISSIBLE VALUES FOR THE REACTIVITY, AND THEN, AFTER THE PROGRAM
11625 0465      C      JUMPS BACKWARD TO CHALLENGE THE CHOICE, DECIDES NOT TO CHANGE THE
11650 0466      C      THE VALUES AFTER ALL, THESE HOLDING VARIABLES PREVENTS THE DEFAULT
11675 0467      C      VALUES FROM BEING LOST.
11700 0468      C
11725 0469      C      ALPHFS =      ALPHF
11750 0470
11775 0471      READ  530, ALPHF
11800 0472      530  FORMAT (E10.3)
11825 0473
11850 0474      IF ((ALPHF .GE. -0.001) .AND. (ALPHF .LE. 0.001)) GO TO 560
11875 0475
11900 0476      ALPHF =      ALPHFS
11925 0477
11950 0478      WRITE (6, 540)
11975 0479      540  FORMAT ('0', '          YOU CHOSE A REACTIVITY COEFFICIENT
12000 0480      + THAT IS TOO LARGE TO BE')
12025 0481      WRITE (6, 550)
12050 0482      550  FORMAT (X, '          USED.  YOU WILL HAVE TO SELECT ANOTHER.')
```

GO TO 430

```

12100 0484
12125 0485
12150 0486      560  WRITE (6, 570)
12175 0487      570  FORMAT (X, '          TYPE IN DESIRED TEMPERATURE COEFFICIENT,
12200 0488      + USING FORMAT E10.3')
12225 0489      WRITE (6, 580)
12250 0490      580  FORMAT (X, '          EX:  A REACTIVITY OF 7.22E-06 WOULD BE
12275 0491      + WRITTEN AS 00.722E-05')
```

ALPHMS = ALPHM

```

12300 0492
12325 0493
12350 0494
12375 0495      WRITE (6, 590) ALPHM
12400 0496      590  FORMAT (X, '          CURRENT VALUE IS', 2X, E10.3)
12425 0497      READ  600, ALPHM
12450 0498      600  FORMAT (E10.3)
12475 0499
12500 0500      IF ((ALPHM .GE. -0.0001) .AND. (ALPHM .LE. 0.0001)) GO TO 630
12525 0501
12550 0502      ALPHM =      ALPHMS
12575 0503
12600 0504
12625 0505      610  WRITE (6, 610)
12650 0506      610  FORMAT (X, '          YOU HAVE CHOSEN A REACTIVITY COEFFICIENT
12675 0507      + THAT IS TOO LARGE TO BE')
12700 0508      WRITE (6, 620)
12725 0509      620  FORMAT (X, '          USED.  YOU WILL HAVE TO SELECT ANOTHER.')
```

GO TO 560

```

12750 0510
12775 0511
12800 0512      630  WRITE (6, 640)
12825 0513      640  FORMAT (X, '          DO YOU WISH TO UTILIZE A CONTROL
```

```

12850 0514      + SYSTEM IN THE REACTOR, OR')
12875 0515      WRITE (6, 650)
12900 0516 650  FORMAT (X, '          NOT? IF YOU WANT A CONTROL SYSTEM, TYPE
12925 0517      + IN 1; IF NOT, TYPE IN 2')
12950 0518      READ 660, CS
12975 0519 660  FORMAT (I1)
13000 0520
13025 0521      IF (CS .EQ. 2) GO TO 1590
13050 0522      IF (CS .EQ. 1) GO TO 700
13075 0523
13100 0524      WRITE (6, 670)
13125 0525 670  FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
13150 0526      + BE UTILIZED. YOU WILL HAVE')
13175 0527      WRITE (6, 680)
13200 0528 680  FORMAT (X, '          TO TRY AGAIN.')
13225 0529      WRITE (6, 690)
13250 0530 690  FORMAT (' ')
13275 0531
13300 0532      GO TO 630
13325 0533
13350 0534
13375 0535
13400 0536
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')
720  WRITE (6, 720)
+ CONTROLLER "SENSES" A CHANGE')
730  WRITE (6, 730)
+ REACTOR, IT INSERTS A')
740  WRITE (6, 740)
+ IN THE AVERAGE COOLANT TEMPERATURE OF
+ REACTOR, IT INSERTS A')
750  WRITE (6, 750)
+ TOGETHER DETERMINE THE RESPONSE')
760  WRITE (6, 760)
+ CHARACTERISTICS OF THE CONTROL SYSTEM:
+ THE GAIN, THE TIME CONSTANT')
770  WRITE (6, 770)
+ OF THE FIRST-ORDER DIFFERENTIAL EQUATION
+ WHICH DESCRIBES THE')
780  WRITE (6, 780)
+ ACTIONS OF THE MECHANICAL ACTUATOR,
+ AND A CONSTANT TAU USED TO')
790  WRITE (6, 790)
+ GOVERN THE ACTIONS OF THE INTEGRAL
+ CONTROLLER. THE GAIN HAS UNITS')
800  WRITE (6, 800)
+ OF REACTIVITY / (DEG. C - SEC), AND
+ CONTROLS BOTH THE PROPORTIONAL')
810  WRITE (6, 810)
+ CONTROLLER AND THE INTEGRAL CONTROLLER
+ TOGETHER.. INCREASING THE')
      WRITE (6, 820)

```



```

14275 0571 820 FORMAT (X, ' VALUE OF THE GAIN CAUSES A DIRECTLY
14300 0572 + PROPORTIONAL CHANGE IN THE')
14325 0573 WRITE (6, 830)
14350 0574 830 FORMAT (X, ' MAGNITUDE OF REACTIVITY. FOR A GIVEN
14375 0575 + ERROR, RESPONSE SPEED OF THE')
14400 0576 WRITE (6, 840)
14425 0577 840 FORMAT (X, ' CONTROL SYSTEM IS INCREASED BY DECREASING
14450 0578 + THE TIME CONSTANT OF THE')
14475 0579 WRITE (6, 850)
14500 0580 850 FORMAT (X, ' DIFFERENTIAL EQUATION, AND VICE VERSA.
14525 0581 + THE CONSTANT TAU ADJUSTS')
14550 0582 WRITE (6, 860)
14575 0583 860 FORMAT (X, ' THE EFFECT OF THE INTEGRAL PORTION OF
14600 0584 + THE CONTROLLER. INCREASING')
14625 0585 WRITE (6, 870)
14650 0586 870 FORMAT (X, ' TAU WILL DECREASE THE EFFECT OF THE
14675 0587 + INTEGRAL CONTROLLER, AND VICE')
14700 0588 WRITE (6, 880)
14725 0589 880 FORMAT (X, ' VERSA.')
14750 0590
14775 0591 WRITE (6, 890)
14800 0592 890 FORMAT ('0', ' AS OPERATOR, YOU WILL BE ABLE TO ADJUST
14825 0593 + THE ACTIONS OF THE')
14850 0594 WRITE (6, 900)
14875 0595 900 FORMAT (X, ' REACTOR CONTROL SYSTEM BY SELECTING YOUR
14900 0596 + OWN VALUES FOR THE THREE')
14925 0597 WRITE (6, 910)
14950 0598 910 FORMAT (X, ' PARAMETERS LISTED ABOVE.')
14975 0599
15000 0600 920 WRITE (6, 930)
15025 0601 930 FORMAT ('0', ' CURRENT VALUE OF THE GAIN IS
15050 0602 + -1.0E-06; ITS VALUE TYPICALLY')
15075 0603 WRITE (6, 940)
15100 0604 940 FORMAT (X, ' VARIES FROM
15125 0605 + 0.0 TO -4.0E-06')
15150 0606 WRITE (6, 950)
15175 0607 950 FORMAT (X, ' CURRENT VALUE OF THE TIME CONSTANT IS
15200 0608 + 5 SEC.; ITS VALUE TYPICALLY')
15225 0609 WRITE (6, 960)
15250 0610 960 FORMAT (X, ' VARIES FROM
15275 0611 + 2 SEC. TO 20 SEC.')
15300 0612 WRITE (6, 970)
15325 0613 970 FORMAT (X, ' CURRENT VALUE OF THE PARAMETER TAU IS
15350 0614 + 5 SEC.; ITS VALUE TYPICALLY')
15375 0615 WRITE (6, 980)
15400 0616 980 FORMAT (X, ' VARIES FROM
15425 0617 + 2 SEC. TO 20 SEC.')
15450 0618 IF (KF .EQ. 1) GO TO 1590
15475 0619
15500 0620 C
15525 0621 C HERE THE USER DECIDES IF ANY CHANGES TO ANY OF THE CONTROL
15550 0622 C SYSTEM PARAMETERS ARE TO BE MADE. IF NO CHANGES ARE MADE, DEFAULT
15575 0623 C VALUES WILL REMAIN. THE USER WILL BE ABLE TO CHANGE ANY OR ALL OF
15600 0624 C THE PARAMETERS.
15625 0625 C
15650 0626 WRITE (6, 990)
15675 0627 990 FORMAT ('0', ' THE VALUES GIVEN ABOVE WILL REMAIN AS

```

```

15700 0628      + THEY ARE UNLESS YOU')
15725 0629      WRITE (6, 1000)
15750 0630      1000 FORMAT (X, '          CHANGE THEM.')
```

```

15775 0631      1010 WRITE (6, 1020)
15800 0632      1020 FORMAT (X, '          DO YOU WISH TO CHANGE ANY OR ALL OF THE
15825 0633      + VALUES OF THE PARAMETERS OF')
```

```

15850 0634      WRITE (6, 1030)
15875 0635      1030 FORMAT (X, '          THE CONTROL SYSTEM? IF SO, TYPE IN 1;
15900 0636      + IF NOT, TYPE IN 2')
```

```

15925 0637      WRITE (6, 1040)
15950 0638      1040 FORMAT (X, '          IF NOT, TYPE IN 2')
```

```

15975 0639      READ 1050, CSC
16000 0640      1050 FORMAT (11)
16025 0641
16050 0642      IF (CSC .EQ. 2) GO TO 1590
16075 0643      IF (CSC .EQ. 1) GO TO 1590
16100 0644
16125 0645      WRITE (6, 1060)
16150 0646      1060 FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
16175 0647      + BE UTILIZED. YOU WILL HAVE')
```

```

16200 0648      WRITE (6, 1070)
16225 0649      1070 FORMAT (X, '          TO TRY AGAIN.')
```

```

16250 0650      WRITE (6, 1080)
16275 0651      1080 FORMAT ('')
```

```

16300 0652
16325 0653      GO TO 1010
16350 0654
16375 0655      1090 WRITE (6, 1100)
16400 0656      1100 FORMAT (X, '          DO YOU WISH TO INSERT YOUR OWN VALUE FOR
16425 0657      + THE GAIN?')
```

```

16450 0658      WRITE (6, 1110)
16475 0659      1110 FORMAT (X, '          CURRENT VALUE IS -1.0E-06')
```

```

16500 0660      WRITE (6, 1120)
16525 0661      1120 FORMAT (X, '          IF SO, TYPE IN 1; IF NOT, TYPE IN 2')
```

```

16550 0662      READ 1130, CSCG
16575 0663      1130 FORMAT (11)
16600 0664
16625 0665      IF (CSCG .EQ. 2) GO TO 1260
16650 0666      IF (CSCG .EQ. 1) GO TO 1170
16675 0667
16700 0668      WRITE (6, 1140)
16725 0669      1140 FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
16750 0670      + BE UTILIZED. YOU WILL HAVE')
```

```

16775 0671      WRITE (6, 1150)
16800 0672      1150 FORMAT (X, '          TO TRY AGAIN.')
```

```

16825 0673      WRITE (6, 1160)
16850 0674      1160 FORMAT ('')
```

```

16875 0675
16900 0676      GO TO 1090
16925 0677
16950 0678      1170 WRITE (6, 1180)
16975 0679      1180 FORMAT (X, '          WRITE IN YOUR OWN VALUE FOR THE GAIN,
17000 0680      + USING FORMAT E8.1')
```

```

17025 0681      WRITE (6, 1190)
17050 0682      1190 FORMAT (X, '          EX: A GAIN OF -0.0004 PER DEG. C - SEC
17075 0683      + WOULD BE WRITTEN AS -4.0E-04')
```

```

17100 0684      WRITE (6, 1200)
```

```

17125 0685 1200 FORMAT (X, '          NOTE: THE GAIN IS ALWAYS A NEGATIVE
17150 0686      + NUMBER, AND NEVER SMALLER THAN')
17175 0687      WRITE (6, 1210)
17200 0688 1210 FORMAT (X, '          0.0: CURRENT VALUE IS  -1.0E-06')
17225 0689      READ 1220, AA
17250 0690 1220 FORMAT (E8.1)
17275 0691
17300 0692      IF ((AA .LE. 0.0) .AND. (AA .GE. -4.0E-06)) GO TO 1260
17325 0693
17350 0694 C
17375 0695 C      THIS EQUATION "SAVES" THE DEFAULT VALUE OF THE GAIN IN CASE
17400 0696 C      THE USER CHANGES HIS OR HER MIND AND DECIDES NOT TO CHANGE T
17425 0697 C      VALUE AFTER ALL. THE OTHER CONTROL SYSTEM PARAMETERS CAN ALSO BE
17450 0698 C      "SAVED" IN THE SAME WAY.
17475 0699 C
17500 0700      AA      =      -5.0E-04
17525 0701
17550 0702      WRITE (6, 1230)
17575 0703 1230 FORMAT (X, '          YOU HAVE CHOSEN A VALUE THAT IS OUTSIDE
17600 0704      + THE PERMISSIBLE LIMITS.')
17625 0705      WRITE (6, 1240)
17650 0706 1240 FORMAT (X, '          SELECT AGAIN, REMEMBERING THAT THE VALUE
17675 0707      + SELECTED MUST LIE BETWEEN')
17700 0708      WRITE (6, 1250)
17725 0709 1250 FORMAT (X, '          0.0 AND -2.0E-07: CURRENT VALUE IS
17750 0710      + -5.0E-04')
17775 0711      GO TO 1090
17800 0712
17825 0713 1260 WRITE (6, 1270)
17850 0714 1270 FORMAT (X, '          DO YOU WISH TO INSERT YOUR OWN VALUE FOR
17875 0715      + THE TIME CONSTANT?')
17900 0716      WRITE (6, 1280)
17925 0717 1280 FORMAT (X, '          CURRENT VALUE IS  5.0')
17950 0718      WRITE (6, 1290)
17975 0719 1290 FORMAT (X, '          IF SO, TYPE IN 1; IF NOT, TYPE IN 2')
18000 0720      READ 1300, CSCTC
18025 0721 1300 FORMAT (I1)
18050 0722
18075 0723      IF (CSCTC .EQ. 2) GO TO 1430
18100 0724      IF (CSCTC .EQ. 1) GO TO 1340
18125 0725
18150 0726      WRITE (6, 1310)
18175 0727 1310 FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
18200 0728      + BE UTILIZED. YOU WILL HAVE')
18225 0729      WRITE (6, 1320)
18250 0730 1320 FORMAT (X, '          TO TRY AGAIN.')
18275 0731      WRITE (6, 1330)
18300 0732 1330 FORMAT (' ')
18325 0733
18350 0734      GO TO 1260
18375 0735
18400 0736 1340 WRITE (6, 1350)
18425 0737 1350 FORMAT (X, '          WRITE IN YOUR OWN VALUE FOR THE TIME
18450 0738      + CONSTANT, USING FORMAT F3.1')
18475 0739      WRITE (6, 1360)
18500 0740 1360 FORMAT (X, '          EX: A VALUE OF 4 SECONDS WOULD BE
18525 0741      + WRITTEN AS 4.0')

```

```

18550 0742      WRITE (6, 1370)
18575 0743 1370 FORMAT (X, '          CURRENT VALUE IS 5.0')
18600 0744      READ 1380, TAUC
18625 0745 1380 FORMAT (F3.1)
18650 0746      IF ((TAUC .GE. 2.0) .AND. (TAUC .LE. 20.0)) GO TO 1430
18675 0747      TAUC = 5.0
18700 0748
18725 0749      WRITE (6, 1390)
18750 0750 1390 FORMAT (X, '          YOU HAVE CHOSEN A VALUE THAT IS OUTSIDE
18775 0751      + THE PERMISSIBLE LIMITS.')
18800 0752      WRITE (6, 1400)
18825 0753 1400 FORMAT (X, '          SELECT AGAIN, REMEMBERING THAT THE VALUE
18850 0754      + SELECTED MUST LIE BETWEEN')
18875 0755      WRITE (6, 1410)
18900 0756 1410 FORMAT (X, '          2.0 AND 20.0: CURRENT VALUE IS 5.0')
18925 0757      WRITE (6, 1420)
18950 0758 1420 FORMAT (' ')
18975 0759      GO TO 1260
19000 0760
19025 0761
19050 0762
19075 0763
19100 0764 1430 WRITE (6, 1440)
19125 0765 1440 FORMAT (X, '          DO YOU WISH TO INSERT YOUR OWN VALUE FOR
19150 0766      + THE PARAMETER TAU?')
19175 0767      WRITE (6, 1450)
19200 0768 1450 FORMAT (X, '          CURRENT VALUE IS 5.0')
19225 0769      WRITE (6, 1460)
19250 0770 1460 FORMAT (X, '          IF SO, TYPE IN 1; IF NOT, TYPE IN 2')
19275 0771      READ 1470, CSCT
19300 0772 1470 FORMAT (I1)
19325 0773      IF (CSCT .EQ. 2) GO TO 1590
19350 0774      IF (CSCT .EQ. 1) GO TO 1510
19375 0775
19400 0776
19425 0777      WRITE (6, 1480)
19450 0778 1480 FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
19475 0779      + BE UTILIZED. YOU WILL HAVE')
19500 0780      WRITE (6, 1490)
19525 0781 1490 FORMAT (X, '          TO TRY AGAIN.')
19550 0782      WRITE (6, 1500)
19575 0783 1500 FORMAT (' ')
19600 0784      GO TO 1430
19625 0785
19650 0786
19675 0787
19700 0788 1510 WRITE (6, 1520)
19725 0789 1520 FORMAT (X, '          WRITE IN YOUR OWN VALUE FOR THE
19750 0790      + PARAMETER TAU, USING FORMAT F3.1')
19775 0791      WRITE (6, 1530)
19800 0792 1530 FORMAT (X, '          CURRENT VALUE IS 5.0')
19825 0793      READ 1540, TAU
19850 0794 1540 FORMAT (F3.1)
19875 0795      IF ((TAU .GE. 2.0) .AND. (TAU .LE. 20.0)) GO TO 1590
19900 0796      TAU = 5.0
19925 0797
19950 0798

```

```

19975 0799          WRITE (6, 1550)
20000 0800 1550 FORMAT (X, '          YOU HAVE CHOSEN A VALUE THAT IS OUTSIDE
20025 0801      + THE PERMISSIBLE LIMITS.' )
20050 0802          WRITE (6, 1560)
20075 0803 1560 FORMAT (X, '          SELECT AGAIN, REMEMBERING THAT THE VALUE
20100 0804      + SELECTED MUST LIE BETWEEN' )
20125 0805          WRITE (6, 1570)
20150 0806 1570 FORMAT (X, '          2.0 AND 20.0: CURRENT VALUE IS 5.0' )
20175 0807          WRITE (6, 1580)
20200 0808 1580 FORMAT (' ')
20225 0809
20250 0810          GO TO 1430
20275 0811
20300 0812 C          HERE, THE TWO-TIME-SCALE DECOUPLING ALGORITHM, THE PROMPT-JUMP
20325 0813 C APPROXIMATION, AND THE RAMP-INPUT MODEL ARE ALL EXPLAINED. THE
20350 0814 C USER IS THEN GIVEN THE OPTION OF CHOOSING WHETHER THE DECOUPLING
20375 0815 C ALGORITHM IS DESIRED.
20400 0816
20425 0817 1590 WRITE (6, 1600)
20450 0818 1600 FORMAT (X, '          BY DEFAULT, THE REACTOR IS CURRENTLY
20475 0819      + USING A STEP INPUT MODEL THAT' )
20500 0820          WRITE (6, 1610)
20525 0821 1610 FORMAT (X, '          SOLVES THE POINT KINETICS EQUATIONS
20550 0822      + (WITHOUT EXTERNAL SOURCE' )
20575 0823          WRITE (6, 1620)
20600 0824 1620 FORMAT (X, '          DIRECTLY. HOWEVER, IF YOU WISH YOU
20625 0825      + CAN USE EITHER THE PROMPT' )
20650 0826          WRITE (6, 1630)
20675 0827 1630 FORMAT (X, '          JUMP APPROXIMATION OR A RAMP INPUT
20700 0828      + MODEL. THE PROMPT JUMP' )
20725 0829          WRITE (6, 1640)
20750 0830 1640 FORMAT (X, '          APPROXIMATION MODEL WOULD UTILIZE 6
20775 0831      + DELAYED NEUTRON GROUPS.' )
20800 0832          WRITE (6, 1650)
20825 0833 1650 FORMAT (X, '          IF YOU CHOOSE THE RAMP-INPUT MODEL, YOU
20850 0834      + WILL BE ABLE TO CHOOSE THE' )
20875 0835          WRITE (6, 1660)
20900 0836 1660 FORMAT (X, '          RAMP-INPUT RATE AND THE PERIOD OF TIME
20925 0837      + OVER WHICH IT IS OPERATIVE.' )
20950 0838
20975 0839          WRITE (6, 1670)
21000 0840 1670 FORMAT ('0', '          THIS PROGRAM ALSO HAS THE CAPABILITY OF
21025 0841      + DECOUPLING THE 7X7 SYSTEM' )
21050 0842          WRITE (6, 1680)
21075 0843 1680 FORMAT (X, '          OF POINT-KINETICS EQUATIONS (6 DELAYED -
21100 0844      + NEUTRON GROUPS PLUS PROMPT' )
21125 0845          WRITE (6, 1690)
21150 0846 1690 FORMAT (X, '          RESPONSE) INTO 2 MODES: THE SLOW-MODE
21175 0847      + GROUP CONSISTING OF THE 6' )
21200 0848          WRITE (6, 1700)
21225 0849 1700 FORMAT (X, '          DELAYED-NEUTRON GROUPS, AND THE FAST -
21250 0850      + MODE GROUP CONSISTING OF THE' )
21275 0851          WRITE (6, 1710)
21300 0852 1710 FORMAT (X, '          PROMPT RESPONSE. THE POINT-KINETICS
21325 0853      + EQUATIONS CAN THEN BE SOLVED' )
21350 0854          WRITE (6, 1720)
21375 0855 1720 FORMAT (X, '          SEPARATELY FOR EACH OF THE TWO GROUPS.

```

```

21400 0856      + SHORT TIME STEPS WOULD BE')
21425 0857      WRITE (6, 1730)
21450 0858 1730 FORMAT (X, '          USED FOR THE FAST MODE, WHILE LONGER
21475 0859      + TIME STEPS WOULD BE USED FOR')
21500 0860      WRITE (6, 1740)
21525 0861 1740 FORMAT (X, '          THE SLOW MODE. THIS WILL ACHIEVE
21550 0862      + INCREASED ACCURACY AND DECREASED')
21575 0863      WRITE (6, 1750)
21600 0864 1750 FORMAT (X, '          CPU TIME. NOTE: IF YOU SELECT THIS
21625 0865      + MATRIX DECOUPLING OPTION, YOU')
21650 0866      WRITE (6, 1760)
21675 0867 1760 FORMAT (X, '          WILL NOT BE ABLE TO USE THE PROMPT JUMP
21700 0868      + APPROXIMATION. THIS PROGRAM')
21725 0869      WRITE (6, 1780)
21750 0870 1780 FORMAT (X, '          WILL PREVENT YOU FROM EVEN TRYING TO DO
21775 0871      + SO.')
21800 0872
21825 0873 1800 WRITE (6, 1810)
21850 0874 1810 FORMAT ('0', '          DO YOU WANT TO USE THE "MATRIX
21875 0875      + DECOUPLING" OPTION? IF SO,')
21900 0876      WRITE (6, 1820)
21925 0877 1820 FORMAT (X, '          TYPE IN 1; IF NOT, TYPE IN 2')
21950 0878      READ 1830, EIG
21975 0879 1830 FORMAT (I1)
22000 0880
22025 0881 C
22050 0882 C      HERE, IF A CONTROL SYSTEM IS NOT DESIRED, GAIN IS SET EQUAL TO
22075 0883 C      ZERO, PREVENTING ANY FEEDBACK DUE TO A CONTROL SYSTEM. IF THE -
22100 0884 C      COUPLING OPTION IS DESIRED, THE PROMPT-JUMP APPROXIMATION IS NOT
22125 0885 C      ALLOWED.
22150 0886 C
22175 0887      IF (CS .EQ. 2) AA = 0.0
22200 0888      IF (EIG .EQ. 2) GO TO 1870
22225 0889      IF (EIG .EQ. 1) PJ = 2
22250 0890      IF (EIG .EQ. 1) GO TO 1980
22275 0891
22300 0892      WRITE (6, 1840)
22325 0893 1840 FORMAT (X, '          WHEN YOU WERE ASKED IF YOU WANTED TO
22350 0894      + SELECT THE "MATRIX')
22375 0895      WRITE (6, 1850)
22400 0896 1850 FORMAT (X, '          DECOUPLING" OPTION, YOU TYPED IN A
22425 0897      + NUMBER THAT CANNOT BE UTILIZED.')
22450 0898      WRITE (6, 1860)
22475 0899 1860 FORMAT (X, '          YOU WILL HAVE TO TRY AGAIN.')
22500 0900
22525 0901      GO TO 1800
22550 0902
22575 0903 C
22600 0904 C      ASSUMING EARLIER DECISIONS DO NOT PRECLUDE THIS, THE USER DOES
22625 0905 C      NOW HAVE THE OPTION OF SELECTING THE PROMPT-JUMP APPROXIMATION.
22650 0906 C
22675 0907 1870 WRITE (6, 1880)
22700 0908 1880 FORMAT (X, '          DO YOU WISH TO UTILIZE THE PROMPT JUMP
22725 0909      + APPROXIMATION? IF SO, TYPE')
22750 0910      WRITE (6, 1890)
22775 0911 1890 FORMAT (X, '          IN 1; IF NOT, TYPE IN 2')
22800 0912      WRITE (6, 1900)

```

```

22825 0913 1900 FORMAT ('0', '          YOU CAN SELECT BOTH THE RAMP-INPUT
22850 0914      + MODEL AND THE PROMPT JUMP')
22875 0915      WRITE (6, 1910)
22900 0916 1910 FORMAT (X, '          APPROXIMATION. BUT IF YOU DO, THE
22925 0917      + PROMPT JUMP MODEL WILL NOT')
22950 0918      WRITE (6, 1920)
22975 0919 1920 FORMAT (X, '          OPERATE UNTIL AFTER THE RAMP-INPUT
23000 0920      + IS COMPLETE. IF YOU SELECT THE')
23025 0921      WRITE (6, 1930)
23050 0922 1930 FORMAT (X, '          PJ APPROXIMATION, THERE WILL BE NO
23075 0923      + FEEDBACK AND NO CONTROL SYSTEM.')
23100 0924
23125 0925      READ 1940, PJ
23150 0926 1940 FORMAT (I1)
23175 0927
23200 0928 C
23225 0929 C      HERE, IF THE PROMPT-JUMP APPROXIMATION HAS BEEN SELECTED, THE
23250 0930 C      RAMP-INPUT MODEL WILL AUTOMATICALLY NOT BE ALLOWED, AND NERO WILL
23275 0931 C      JUMP BEYOND IT.
23300 0932 C
23325 0933      IF ((PJ .EQ. 1) .OR. (PJ .EQ. 2)) GO TO 1980
23350 0934
23375 0935      WRITE (6, 1950)
23400 0936 1950 FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
23425 0937      + BE UTILIZED. YOU WILL HAVE')
23450 0938      WRITE (6, 1960)
23475 0939 1960 FORMAT (X, '          TO TRY AGAIN.')
23500 0940      WRITE (6, 1970)
23525 0941 1970 FORMAT (' ')
23550 0942
23575 0943      GO TO 1870
23600 0944
23625 0945 C
23650 0946 C      HERE, THE USER HAS THE CHOICE OF SELECTING, OR NOT SE
23675 0947 C      THE RAMP-INPUT MODEL (ASSUMING THAT EARLIER DECISIONS DO NOT-
23700 0948 C      CLUDE THIS).
23725 0949 C
23750 0950 1980 WRITE (6, 1990)
23775 0951 1990 FORMAT (X, '          DO YOU WISH TO SELECT THE RAMP-INPUT
23800 0952      + MODEL? IF SO, TYPE IN 1;')
23825 0953      WRITE (6, 2000)
23850 0954 2000 FORMAT (X, '          IF NOT, TYPE IN 2')
23875 0955      READ 2010, RI
23900 0956 2010 FORMAT (I1)
23925 0957
23950 0958      IF ((RI .EQ. 1) .OR. (RI .EQ. 2)) GO TO 2050
23975 0959
24000 0960      WRITE (6, 2020)
24025 0961 2020 FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
24050 0962      + BE UTILIZED. YOU WILL HAVE')
24075 0963      WRITE (6, 2030)
24100 0964 2030 FORMAT (X, '          TO TRY AGAIN.')
24125 0965      WRITE (6, 2040)
24150 0966 2040 FORMAT (' ')
24175 0967
24200 0968      GO TO 1980
24225 0969

```

```

24250 0970      2050 IF (RI .EQ. 2) GO TO 2360
24275 0971
24300 0972
24325 0973
24350 0974
24375 0975
24400 0976
24425 0977
24450 0978
24475 0979
24500 0980
24525 0981
24550 0982
24575 0983
24600 0984
24625 0985
24650 0986
24675 0987
24700 0988
24725 0989
24750 0990
24775 0991
24800 0992
24825 0993
24850 0994
24875 0995
24900 0996
24925 0997
24950 0998
24975 0999
25000 1000
25025 1001
25050 1002
25075 1003
25100 1004
25125 1005
25150 1006
25175 1007
25200 1008
25225 1009
25250 1010
25275 1011
25300 1012
25325 1013
25350 1014
25375 1015
25400 1016
25425 1017
25450 1018
25475 1019
25500 1020
25525 1021
25550 1022
25575 1023
25600 1024
25625 1025
25650 1026

C
C      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
C      WRITE (6, 2060)
2060 FORMAT (X, '          SINCE YOU HAVE CHOSEN THE RAMP - INPUT
+ MODEL, YOU WILL NOW HAVE TO')
C      WRITE (6, 2070)
2070 FORMAT (X, '          SELECT A RAMP - INPUT RATE AND A PERIOD
+ OF TIME DURING WHICH THIS')
C      WRITE (6, 2080)
2080 FORMAT (X, '          RAMP - INPUT RATE IS IN EFFECT.  NOTE:
+ THE RAMP - INPUT RATE MAY')
C      WRITE (6, 2090)
2090 FORMAT (X, '          NOT EXCEED PLUS OR MINUS 0.1 $/SEC.
+ TOTAL ACCUMULATED REACTIVITY')
C      WRITE (6, 2100)
2100 FORMAT (X, '          MAY NOT EXCEED 0.9$.  IF YOU SELECT
+ VALUES OUTSIDE THESE PARAMETERS,')
C      WRITE (6, 2110)
2110 FORMAT (X, '          YOUR CHOICE(S) WILL BE REJECTED, AND YOU
+ WILL HAVE TO TRY AGAIN.')
```



```

25675 1027 2210 FORMAT (X, '          EXCEED PLUS OR MINUS 90.0')
25700 1028
25725 1029 2220 WRITE (6, 2230)
25750 1030 2230 FORMAT ('0', '          WRITE IN A RAMP - INPUT RATE
25775 1031      + (IN CENTS/SEC), USING FORMAT F6.2:')
25800 1032      WRITE (6, 2240)
25825 1033 2240 FORMAT (X, '          EX: A RAMP - INPUT RATE OF -10 CENTS/SEC
25850 1034      + WOULD BE WRITTEN AS -10.00')
25875 1035      WRITE (6, 2250)
25900 1036 2250 FORMAT (X, '          A RAMP - INPUT RATE OF 5 CENTS/SEC
25925 1037      + WOULD BE WRITTEN AS 005.00')
25950 1038
25975 1039      READ 2260, RIR
26000 1040 2260 FORMAT (F5.1)
26025 1041
26050 1042      IF ((RIR .GE. -10.0) .AND. (RIR .LE. 10.0)) GO TO 2290
26075 1043
26100 1044      WRITE (6, 2270)
26125 1045 2270 FORMAT (X, '          THE VALUE SELECTED MAY NOT EXCEED PLUS
26150 1046      + OR MINUS 10. YOU WILL HAVE')
26175 1047      WRITE (6, 2280)
26200 1048 2280 FORMAT (X, '          TO TRY AGAIN.0')
26225 1049
26250 1050      GO TO 2220
26275 1051
26300 1052 2290 WRITE (6, 2300)
26325 1053 2300 FORMAT (X, '          WRITE IN THE TIME PERIOD FOR WHICH THE
26350 1054      + RAMP - INPUT IS IN EFFECT,')
26375 1055      WRITE (6, 2310)
26400 1056 2310 FORMAT (X, '          USING FORMAT F5.1')
26425 1057      READ 2320, TTOT
26450 1058 2320 FORMAT (F5.1)
26475 1059
26500 1060      PROD = RIR * TTOT
26525 1061      RIR = 0.01 * RIR * A
26550 1062
26575 1063      IF ((PROD .GE. -90.0) .AND. (PROD .LE. 90.0)) GO TO 2450
26600 1064
26625 1065      WRITE (6, 2330)
26650 1066 2330 FORMAT (X, '          YOU HAVE SELECTED VALUE(S) THAT ARE TOO
26675 1067      + HIGH. REMEMBER, THE PRODUCT')
26700 1068      WRITE (6, 2340)
26725 1069 2340 FORMAT (X, '          OF RAMP - INPUT (IN CENTS) TIMES TOTAL
26750 1070      + TIME PERIOD (IN SECONDS)')
26775 1071      WRITE (6, 2350)
26800 1072 2350 FORMAT (X, '          MUST NOT EXCEED PLUS OR MINUS 90.0. YOU
26825 1073      + WILL HAVE TO TRY AGAIN.0')
26850 1074
26875 1075      GO TO 2220
26900 1076
26925 1077 2360      RIR = 0.01 * RIR * A
26950 1078
26975 1079
27000 1080 C          HERE THE USER SELECTS WHATEVER REACTIVITY STEP IS REQUIRED.
27025 1081 C
27050 1082      WRITE (6, 2370)
27075 1083 2370 FORMAT (X, '          WHAT LEVEL (IN CENTS) REACTIVITY

```

```

27100 1084      + PERTURBATION IS DESIRED?')
27125 1085      WRITE (6, 2380)
27150 1086 2380 FORMAT (X, '          USE FORMAT F5.1      EX: 010.0 = A STEP
27175 1087      + INPUT OF +10 CENTS.')
```

```

27200 1088      READ 2390, DRON
27225 1089 2390 FORMAT (F4.1)
27250 1090
27275 1091      IF (((DRON + RHON) .GT. -100.0) .AND. ((DRON + RHON) .LT.100.0))
27300 1092      +      GO TO 2440
27325 1093
27350 1094 2400 WRITE (6, 2410)
27375 1095 2410 FORMAT (X, '          YOU HAVE SELECTED A VALUE THAT WILL
27400 1096      + RESULT IN A PROMPT SUPERCRITICAL')
27425 1097      WRITE (6, 2420)
27450 1098 2420 FORMAT (X, '          CONDITION. THIS IS NOT ALLOWED, AND YOU
27475 1099      + WILL HAVE TO TRY AGAIN.')
```

```

27500 1100      WRITE (6, 2430)
27525 1101 2430 FORMAT (' ')
27550 1102
27575 1103
27600 1104      GO TO 2360
27625 1105
27650 1106 2440      DRO      =      0.01 * DRON * A
27675 1107
27700 1108
27725 1109 C          HERE INITIAL POWER LEVEL IS SELECTED.
27750 1110 C
27775 1111 2450 WRITE (6, 2460)
27800 1112 2460 FORMAT (X, '          WHAT INITIAL POWER LEVEL (IN MWT) IS
27825 1113      + DESIRED? USE FORMAT F6.1')
```

```

27850 1114      WRITE (6, 2470)
27875 1115 2470 FORMAT (X, '          EXAMPLE: IF INITIAL POWER IS 2000 MW,
27900 1116      + TYPE IN 2000.0')
```

```

27925 1117      READ 2480, N10
27950 1118 2480 FORMAT (F6.1)
27975 1119
28000 1120      IF ((N10 .GT. 0.0) .AND. (N10 .LE. 3000.0)) GO TO 2520
28025 1121
28050 1122      WRITE (6, 2490)
28075 1123 2490 FORMAT (X, '          NEGATIVE POWER LEVELS OR POWER LEVELS
28100 1124      + IN EXCESS OF 3000MW ARE NOT')
```

```

28125 1125      WRITE (6, 2500)
28150 1126 2500 FORMAT (X, '          ALLOWED. YOU WILL HAVE TO TRY AGAIN.')
```

```

28175 1127      WRITE (6, 2510)
28200 1128 2510 FORMAT (' ')
28225 1129
28250 1130      GO TO 2450
28275 1131 2520 IF ((KF .EQ. 1) .OR. (RI .EQ. 1) .OR.
28300 1132      +      (PJ .EQ. 1) .OR. (EIG .EQ. 1)) GO TO 2720
28325 1133
28350 1134
28375 1135
28400 1136 C          HERE THE USER DECIDES ON WHETHER TO SELECT A POWER STEP. THIS
28425 1137 C          WILL MEAN NO REACTOR KINETICS, BUT A CONSTANT POWER AT SOME NEW
28450 1138 C          LEVEL. THIS IS TO CHECK ON STEAM GENERATOR PERFORMANCE.
28475 1139 C
28500 1140      WRITE (6, 2530)
```

```

28525 1141 2530 FORMAT (X, ' DO YOU WISH TO INSERT A FIXED POWER
28550 1142 + CHANGE STEP? THIS IS A STEP')
28575 1143 WRITE (6, 2540)
28600 1144 2540 FORMAT (X, ' CHANGE IN REACTOR POWER IMPOSED BY YOU
28625 1145 + AFTER THE INITIAL,')
28650 1146 WRITE (6, 2550)
28675 1147 2550 FORMAT (X, ' STEADY-STATE CONDITIONS HAVE BEEN SET
28700 1148 + UP. IT REPLACES THE REACTOR')
28725 1149 WRITE (6, 2560)
28750 1150 2560 FORMAT (X, ' KINETICS SUBROUTINE. IF YOU CHOOSE THIS
28775 1151 + OPTION, THERE WILL BE NO')
28800 1152 WRITE (6, 2570)
28825 1153 2570 FORMAT (X, ' REACTOR KINETICS. TOTAL POWER LEVEL
28850 1154 + WILL BE PERMANENTLY FIXED.')
28875 1155 WRITE (6, 2580)
28900 1156 2580 FORMAT (X, ' NATURALLY, TOTAL REACTOR POWER WILL NOT
28925 1157 + BE ALLOWED TO BE GREATER')
28950 1158 WRITE (6, 2590)
28975 1159 2590 FORMAT (X, ' THAN 3000MW OR LESS THAN ZERO. IF YOU
29000 1160 + WISH THIS OPTION, TYPE IN 1;')
29025 1161 WRITE (6, 2600)
29050 1162 2600 FORMAT (X, ' IF NOT, TYPE IN 2')
29075 1163
29100 1164 READ 2610, NK
29125 1165 2610 FORMAT (I1)
29150 1166
29175 1167 IF ((NK .EQ. 1) .OR. (NK .EQ. 2)) GO TO 2650
29200 1168
29225 1169 WRITE (6, 2620)
29250 1170 2620 FORMAT (X, ' YOU HAVE TYPED IN A NUMBER THAT CANNOT
29275 1171 + BE UTILIZED. YOU WILL HAVE')
29300 1172 WRITE (6, 2630)
29325 1173 2630 FORMAT (X, ' TO TRY AGAIN.')
29350 1174 WRITE (6, 2640)
29375 1175 2640 FORMAT (' ')
29400 1176
29425 1177 GO TO 2520
29450 1178
29475 1179 2650 IF (NK .EQ. 2) GO TO 2710
29500 1180
29525 1181 WRITE (6, 2660)
29550 1182 2660 FORMAT (X, ' WHAT POWER STEP CHANGE IS DESIRED? USE
29575 1183 + FORMAT F7.1')
29600 1184 WRITE (6, 2670)
29625 1185 2670 FORMAT (X, ' EXAMPLE: IF POWER STEP IS -100MW, TYPE
29650 1186 + IN -0100.0')
29675 1187
29700 1188 READ 2680, DN2
29725 1189 2680 FORMAT (F7.1)
29750 1190
29775 1191 IF (((DN2 + N10) .GE. 0.0) .OR. ((DN2 + N10) .LE. 3000.0))
29800 1192 + GO TO 2710
29825 1193
29850 1194 WRITE (6, 2690)
29875 1195 2690 FORMAT (X, ' YOU HAVE SELECTED A POWER STEP WHICH
29900 1196 + PLACES TOTAL POWER OUTSIDE')
29925 1197 WRITE (6, 2700)

```

```

29950 1198 2700 FORMAT (X, ' PERMISSIBLE PARAMETERS. YOU WILL HAVE
29975 1199 + TO TRY AGAIN. ')
30000 1200
30025 1201 GO TO 2520
30050 1202
30075 1203 2710 DN1 = DN2 * 1.0E06
30100 1204 IF (NK .EQ. 1) CS = 2
30125 1205 IF (NK .EQ. 1) EIG = 2
30150 1206 IF (NK .EQ. 1) PJ = 2
30175 1207 IF (NK .EQ. 1) RI = 2
30200 1208
30225 1209
30250 1210 2720 VOSS = N10 / 3000.0
30275 1211 DVO = 0.0
30300 1212 IF (NK .EQ. 1) GO TO 3050
30325 1213
30350 1214
30375 1215
30400 1216 C
30425 1217 C HERE THE USER DECIDES ON WHETHER TO INDUCE A PERTURBATION BY
30450 1218 C VARYING THE STEAM GENERATOR THROTTLE VALVE INSTEAD OF VARYING THE
30475 1219 C REACTOR. IF THIS IS SELECTED, ANY PREVIOUSLY SELECTED REACTIVITY
30500 1220 C CHANGE WILL BE SET EQUAL TO ZERO. MAXIMUM STEAM VALVE CHANGE IS
30525 1221 C 10%.
30550 1222 C
30575 1223 2800 WRITE (6, 2810) VOSS
30600 1224 2810 FORMAT (X, ' STEAM GENERATOR IS NOW', 3X, F5.3, ':',
+ 3X, 'THIS REPRESENTS PRESENT VALVE')
30625 1225 WRITE (6, 2820)
30650 1226 2820 FORMAT (X, ' POSITION. 0.000 MEANS CLOSED, WHILE
+ 1.000 MEANS OPEN. ')
30675 1227
30700 1228 2830 WRITE (6, 2840)
30725 1229 2840 FORMAT ('0', ' DO YOU WISH TO INDUCE A PERTURBATION
+ BY CHANGING THE VALVE')
30750 1230 WRITE (6, 2850)
30775 1231 2850 FORMAT (X, ' POSITION? IF "YES", TYPE 1; IF "NO",
+ TYPE 2')
30800 1232
30825 1233
30850 1234 WRITE (6, 2860)
30875 1235 2860 FORMAT ('0', ' CAUTION: IF A VALVE CHANGE IS THE
+ INITIAL PERTURBATION, THE STEAM')
30900 1236 WRITE (6, 2870)
30925 1237 2870 FORMAT (X, ' GENERATOR MODEL WILL STILL HAVE TO BE
+ USED. AND ANY REACTIVITY')
30950 1238 WRITE (6, 2880)
30975 1239 2880 FORMAT (X, ' PERTURBATION WILL HAVE TO EQUAL
+ ZERO. HOWEVER, DO NOT WORRY IF')
31000 1240 WRITE (6, 2890)
31025 1241 2890 FORMAT (X, ' YOU HAVE ALREADY CHOSEN A REACTIVITY
+ STEP STEAM GENERATOR MODEL. ')
31050 1242 WRITE (6, 2900)
31075 1243 2900 FORMAT (X, ' OR A REACTIVITY STEP. , IF YOU
+ UTILIZE A VALVE PERTURBATION, THIS')
31100 1244 WRITE (6, 2910)
31125 1245 2910 FORMAT (X, ' PROGRAM WILL CHOOSE SET ANY REACTIVITY
+ INSERTION EQUAL TO ZERO. ')
31150 1246 READ 2930, ZZ
31175 1247
31200 1248 2930 FORMAT (I1)
31225 1249
31250 1250
31275 1251
31300 1252
31325 1253
31350 1254

```

31375	1255			
31400	1256			
31425	1257			
31450	1258			
31475	1259			
31500	1260			
31525	1261			
31550	1262			
31575	1263			
31600	1264			
31625	1265			
31650	1266			
31675	1267			
31700	1268			
31725	1269			
31750	1270			
31775	1271			
31800	1272			
31825	1273			
31850	1274			
31875	1275			
31900	1276			
31925	1277			
31950	1278			
31975	1279			
32000	1280			
32025	1281			
32050	1282			
32075	1283			
32100	1284			
32125	1285			
32150	1286			
32175	1287			
32200	1288			
32225	1289			
32250	1290			
32275	1291			
32300	1292			
32325	1293			
32350	1294			
32375	1295			
32400	1296			
32425	1297			
32450	1298			
32475	1299			
32500	1300			
32525	1301			
32550	1302			
32575	1303			
32600	1304			
32625	1305			
32650	1306			
32675	1307			
32700	1308			
32725	1309			
32750	1310			
32775	1311			

```

      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
```

```

32800 1312      Q      =      2
32825 1313      CPPAV =      5916.241929 + 16.32498553 *EXP(6.488055E-04*N10)
32850 1314      MDOTP =      3.0E09 / ((56.9 * 5.0 / 9.0) * CPPAV)
32875 1315      T11   =      (565.0*5.0/9.0) - (28.45*5.0/9.0) * N10 / 3000.0
32900 1316      TO1   =      (565.0*5.0/9.0) + (28.45*5.0/9.0) * N10 / 3000.0
32925 1317      DTF(1) =      0.0
32950 1318      DTI(1) =      0.0
32975 1319      DTM(1) =      0.0
33000 1320      DTO(1) =      0.0
33025 1321      DTSG(1) =      0.0
33050 1322      DTS(1) =      0.0
33075 1323      PT(1)  =      N10 * 1.0E06
33100 1324      STMG(1)=      N10 * 1.0E06
33125 1325      TT(1)  =      0.0
33150 1326      ABN    =      0.2
33175 1327      FRAC   =      0.5 * 5932.566914 / CPPAV
33200 1328      LNT    =      1
33225 1329      TM1   =      FRAC * (TO1 - T11) + T11
33250 1330      TM     =      TM1
33275 1331      TW     =      0.0
33300 1332      TOTHO  =      0.0
33325 1333      CPM    =      4992.4097749 + 2.49340775E-04 * EXP
33350 1334      +      (0.04825458 * TM1)
33375 1335      ROP    =      881.6309 - 2.86514041 * EXP(0.0133034152 *TM1)
33400 1336      VMOD   =      MDOTP / (5.26 * ROP)
33425 1337      KP     =      0.712768335 - 3.025E-03 * EXP(0.01321074*TM1)
33450 1338      MUP    =      -3.53438229E-07 * TM1 + 1.9978561E-04
33475 1339      NUP    =      MUP / ROP
33500 1340      REP    =      0.012 * VMOD / NUP
33525 1341      PRP    =      CPM * NUP * ROP / KP
33550 1342      NU     =      0.025 * (REP ** 0.8) * (PRP ** 0.6)
33575 1343      HP     =      NU * KP / 0.012
33600 1344      TC     =      (N10 * 1.0E06) / (HP * 5945.0) + TM1
33625 1345      TF1   =      25.6933661 * (TC - TM1) + TM1
33650 1346      STMGEN =      N10 * 1.0E06
33675 1347
33700 1348      IF (KF .EQ. 1) AA = 0.0
33725 1349      IF (KF .EQ. 1) ALPHF = 0.0
33750 1350      IF (KF .EQ. 1) ALPHM = 0.0
33775 1351      IF (KF .EQ. 1) CS = 2
33800 1352      IF (KF .EQ. 1) RHO = 0.0
33825 1353      IF (KF .EQ. 1) RHO = 0.0
33850 1354
33875 1355
33900 1356
33925 1357
33950 1358
33975 1359
34000 1360
34025 1361
34050 1362
34075 1363
34100 1364
34125 1365
34150 1366
34175 1367
34200 1368
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.')
```

34225	1369		
34250	1370		
34275	1371		
34300	1372		
34325	1373		
34350	1374		
34375	1375		
34400	1376		
34425	1377		
34450	1378		
34475	1379		
34500	1380		
34525	1381		
34550	1382		
34575	1383		
34600	1384		
34625	1385		
34650	1386		
34675	1387		
34700	1388		
34725	1389		
34750	1390		
34775	1391		
34800	1392		
34825	1393		
34850	1394		
34875	1395		
34900	1396		
34925	1397		
34950	1398		
34975	1399		
35000	1400		
35025	1401		
35050	1402		
35075	1403		
35100	1404		
35125	1405		
35150	1406		
35175	1407		
35200	1408		
35225	1409		
35250	1410		
35275	1411		
35300	1412		
35325	1413		
35350	1414		
35375	1415		
35400	1416		
35425	1417		
35450	1418		
35475	1419		
35500	1420		
35525	1421		
35550	1422		
35575	1423		
35600	1424		
35625	1425		

```

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  
C

```

HERE THE USER IS ABLE TO CAUSE TABLE OUTPUT TO BE ABBREVIATED.

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
```

```

35650 1426
35675 1427
35700 1428
35725 1429
35750 1430
35775 1431
35800 1432
35825 1433
35850 1434
35875 1435
35900 1436
35925 1437
35950 1438
35975 1439
36000 1440
36025 1441
36050 1442
36075 1443
36100 1444
36125 1445
36150 1446
36175 1447
36200 1448
36225 1449
36250 1450
36275 1451
36300 1452
36325 1453
36350 1454
36375 1455
36400 1456
36425 1457
36450 1458
36475 1459
36500 1460
36525 1461
36550 1462
36575 1463
36600 1464
36625 1465
36650 1466
36675 1467
36700 1468
36725 1469
36750 1470
36775 1471
36800 1472
36825 1473
36850 1474
36875 1475
36900 1476
36925 1477
36950 1478
36975 1479
37000 1480
37025 1481
37050 1482

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

```



```

37075 1483
37100 1484
37125 1485
37150 1486
37175 1487
37200 1488
37225 1489
37250 1490
37275 1491
37300 1492
37325 1493
37350 1494
37375 1495
37400 1496
37425 1497
37450 1498
37475 1499
37500 1500
37525 1501
37550 1502
37575 1503
37600 1504
37625 1505
37650 1506
37675 1507
37700 1508
37725 1509
37750 1510
37775 1511
37800 1512
37825 1513
37850 1514
37875 1515
37900 1516
37925 1517
37950 1518
37975 1519
38000 1520
38025 1521
38050 1522
38075 1523
38100 1524
38125 1525
38150 1526
38175 1527
38200 1528
38225 1529
38250 1530
38275 1531
38300 1532
38325 1533
38350 1534
38375 1535
38400 1536
38425 1537
38450 1538
38475 1539

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.')
```

```

38500 1540
38525 1541      GO TO 3700
38550 1542
38575 1543
38600 1544      3660 IF (IS .NE. 2) GO TO 3680
38625 1545      WRITE (YY, 3670)
38650 1546      3670 FORMAT (X, '          THE FUEL TO BE UTILIZED IS U-235.')
38675 1547      GO TO 3700
38700 1548
38725 1549      3680 WRITE (YY, 3690)
38750 1550      3690 FORMAT (X, '          THE FUEL TO BE UTILIZED IS PU-239.')
38775 1551
38800 1552      3700 IF (KF .EQ. 2) GO TO 3740
38825 1553
38850 1554      WRITE (YY, 3710)
38875 1555      3710 FORMAT ('0', '          FREE KINETICS WILL BE USED IN THIS RUN.
38900 1556      + THERE')
38925 1557      WRITE (YY, 3720)
38950 1558      3720 FORMAT (X, '          WILL BE NO FEEDBACK, NO CONTROL SYSTEM,
38975 1559      + AND')
39000 1560      WRITE (YY, 3730)
39025 1561      3730 FORMAT (X, '          REACTIVITY COEFFICIENTS WILL BE EQUAL TO
39050 1562      + ZERO.')
39075 1563
39100 1564      GO TO 3860
39125 1565
39150 1566      3740 WRITE (YY, 3750)
39175 1567      3750 FORMAT ('0', '          FEEDBACK WITH REACTIVITY COEFFICIENTS
39200 1568      + WILL BE')
39225 1569      WRITE (YY, 3760)
39250 1570      3760 FORMAT (X, '          USED.')
39275 1571
39300 1572      WRITE (YY, 3770) RHON
39325 1573      3770 FORMAT ('0', '          PRE-PERTURBATION REACTIVITY IS', 18X,
39350 1574      + F4.1, 8X, 'CENTS')
39375 1575
39400 1576      WRITE (YY, 3780) ALPHF
39425 1577      3780 FORMAT (X, '          DOPPLER COEFFICIENT OF REACTIVITY IS',
39450 1578      + 12X, E10.3, 2X, '(DK/K)/C')
39475 1579      WRITE (YY, 3790) ALPHM
39500 1580      3790 FORMAT (X, '          MODERATOR TEMP. COEFFICIENT OF
39525 1581      + REACTIVITY IS', 4X, E10.3, 2X, '(DK/K)/C')
39550 1582
39575 1583      IF (CS .EQ. 1) GO TO 3810
39600 1584
39625 1585      WRITE (YY, 3800)
39650 1586      3800 FORMAT ('0', '          A CONTROL SYSTEM WILL NOT BE USED.')
39675 1587
39700 1588      GO TO 3860
39725 1589
39750 1590
39775 1591      3810 WRITE (YY, 3820)
39800 1592      3820 FORMAT ('0', '          A CONTROL SYSTEM WILL BE USED.')
39825 1593      WRITE (YY, 3830) AA
39850 1594      3830 FORMAT (X, '          GAIN          EQUALS', 27X, E8.1)
39875 1595      WRITE (YY, 3840) TAUC
39900 1596      3840 FORMAT (X, '          TIME CONSTANT EQUALS', 28X, F3.1)

```

```

39925 1597          WRITE (YY, 3850) TAU
39950 1598 3850 FORMAT (X, '          PARAMETER TAU EQUALS', 28X, F3.1)
39975 1599
40000 1600 3860 IF (EIG .EQ. 1) GO TO 3890
40025 1601
40050 1602          WRITE (YY, 3870)
40075 1603 3870 FORMAT (X, '          THE "MATRIX DECOUPLING" ALGORITHM
40100 1604 + WILL NOT')
40125 1605          WRITE (YY, 3880)
40150 1606 3880 FORMAT (X, '          BE USED.')
```

```

40175 1607
40200 1608          GO TO 3930
40225 1609
40250 1610 3890 WRITE (YY, 3900)
40275 1611 3900 FORMAT ('0', '          THE "MATRIX DECOUPLING" ALGORITHM
40300 1612 + WILL BE')
40325 1613          WRITE (YY, 3910)
40350 1614 3910 FORMAT (X, '          USED.')
```

```

40375 1615          WRITE (YY, 3920)
40400 1616 3920 FORMAT (X, '          THE PROMPT-JUMP APPROXIMATION WILL NOT
40425 1617 + BE USED.')
```

```

40450 1618
40475 1619 3930 IF (PJ .EQ. 1) GO TO 3950
40500 1620
40525 1621          WRITE (YY, 3940)
40550 1622 3940 FORMAT ('0', '          PROMPT-JUMP APPROXIMATION WILL NOT BE
40575 1623 + USED.')
```

```

40600 1624
40625 1625          GO TO 3980
40650 1626
40675 1627 3950 WRITE (YY, 3960)
40700 1628 3960 FORMAT ('0', '          PROMPT-JUMP APPROXIMATION WILL BE USED')
```

```

40725 1629          WRITE (YY, 3970)
40750 1630 3970 FORMAT (X, '          MATRIX DECOUPLING WILL NOT.')
```

```

40775 1631
40800 1632          GO TO 4050
40825 1633
40850 1634 3980 IF (RI .EQ. 1) GO TO 4000
40875 1635
40900 1636          WRITE (YY, 3990)
40925 1637 3990 FORMAT ('0', '          RAMP-INPUT MODEL WILL NOT BE USED.')
```

```

40950 1638
40975 1639          GO TO 4050
41000 1640
41025 1641 4000 WRITE (YY, 4010)
41050 1642 4010 FORMAT ('0', '          RAMP-INPUT MODEL WILL BE USED.')
```

```

41075 1643          WRITE (YY, 4020) RIR
41100 1644 4020 FORMAT (X, '          RAMP-INPUT RATE IS', 28X, F5.1, 7X,
41125 1645 + 'CENTS/S')
```

```

41150 1646          WRITE (YY, 4030)
41175 1647 4030 FORMAT (X, '          DURATION OF RAMP-INPUT INSERTION IS',
41200 1648 + 11X, F5.1, 7X, 'SECONDS')
```

```

41225 1649          WRITE (YY, 4040) PROD
41250 1650 4040 FORMAT (X, '          TOTAL ACCUMULATED REACTIVITY IS', 17X,
41275 1651 + F5.1, 7X, 'CENTS')
```

```

41300 1652
41325 1653 4050 WRITE (YY, 4060) DRON
```

```

41350 1654 4060 FORMAT ('0', ' REACTIVITY PERTURBATION IS', 22X, F4.1,
41375 1655 + 8X, 'CENTS')
41400 1656
41425 1657 WRITE (YY, 4070) N10
41450 1658 4070 FORMAT ('0', ' INITIAL POWER LEVEL IS', 26X, F6.1, 6X,
41475 1659 + 'MW')
41500 1660
41525 1661 WRITE (YY, 4140) V0
41550 1662 4140 FORMAT ('0', ' STEAM GENERATOR VALVE POSITION IS',
41575 1663 + 15X, F5.3, 7X, 'OPEN')
41600 1664
41625 1665 IF (ZZ .EQ. 1) GO TO 4160
41650 1666
41675 1667 WRITE (YY, 4150)
41700 1668 4150 FORMAT ('0', ' THERE IS NO VALVE INDUCED PERTURBATION')
41725 1669
41750 1670 GO TO 4190
41775 1671
41800 1672 4160 WRITE (YY, 4170)
41825 1673 4170 FORMAT ('0', ' SYSTEM PERTURBATION WILL BE A VALVE
41850 1674 + CHANGE.')
41875 1675 WRITE (YY, 4180) DVO
41900 1676 4180 FORMAT (X, ' VALVE CHANGE IS', 34X, F4.1, 8X,
41925 1677 + 'PERCENT.')
41950 1678
41975 1679 4190 IF (AB .EQ. 2) GO TO 4220
42000 1680
42025 1681 WRITE (YY, 4200)
42050 1682 4200 FORMAT ('0', ' OUTPUT WILL BE ABBREVIATED. PRINTING
42075 1683 + WILL OCCUR EVERY')
42100 1684 WRITE (YY, 4210) ABN
42125 1685 4210 FORMAT (X, ' F4.1, 3X, 'SECONDS, SYSTEM TIME.')
42150 1686
42175 1687 GO TO 4250
42200 1688
42225 1689 4220 WRITE (YY, 4230)
42250 1690 4230 FORMAT ('0', ' OUTPUT IS NOT ABBREVIATED. PRINTING
42275 1691 + WILL OCCUR EVERY')
42300 1692 WRITE (YY, 4240)
42325 1693 4240 FORMAT (X, ' 0.2 SECONDS, SYSTEM TIME.')
42350 1694 4250 WRITE (YY, 4260)
42375 1695 4260 FORMAT ('0', ' TOTAL SYSTEM TIME OF THIS RUN IS',
42400 1696 + 24X, 11, 4X, 'MINUTES')
42425 1697
42450 1698 IF (NN .EQ. 1) GO TO 4280
42475 1699
42500 1700 WRITE (YY, 4270)
42525 1701 4270 FORMAT ('0', ' OUTPUT WILL BE IN A TABLE')
42550 1702
42575 1703 GO TO 4300
42600 1704
42625 1705 4280 WRITE (YY, 4290)
42650 1706 4290 FORMAT ('0', ' OUTPUT WILL BE IN GRAPHICS.')
42675 1707
42700 1708 4300 IF (YY .EQ. 6) GO TO 4320
42725 1709
42750 1710 WRITE (YY, 4310)

```

```

42775 1711 4310 FORMAT ('0', '          HARD COPY WILL BE PRODUCED.')
42800 1712
42825 1713
42850 1714          GO TO 4360
42875 1715
42900 1716 4320 WRITE (YY, 4330)
42925 1717 4330 FORMAT ('0', '          OUTPUT OF GRAPHICS WILL BE ON TERMINAL
42950 1718      + IN THE')
42975 1719      WRITE (YY, 4340)
43000 1720 4340 FORMAT (X, '          IGL LAB. TO GET A HARD COPY, YOU MUST
43025 1721      + USE THE')
43050 1722      WRITE (YY, 4350)
43075 1723 4350 FORMAT (X, '          4051 TEKTRONICS TERMINAL')
43100 1724 4360 CONTINUE
43125 1725
43150 1726
43175 1727 4370 WRITE (YY, 4380)
43200 1728 4380 FORMAT (' ')
43225 1729
43250 1730      WRITE (YY, 4390)
43275 1731 4390 FORMAT (' ')
43300 1732
43325 1733      IF (NN .EQ. 1) GO TO 4460
43350 1734
43375 1735      WRITE (YY, 4400)
43400 1736 4400 FORMAT (2X, 'TIME', 5X, 'FUEL', 6X, 'MODERATOR', 3X, 'RECTOR',
43425 1737      + 4X, 'REACTOR', 5X, 'TURBINE', 3X, 'REACTOR')
43450 1738      WRITE (YY, 4410)
43475 1739 4410 FORMAT (11X, 'TEMP', 6X, 'TEMP', 8X, 'INLET', 6X, 'OUTLET', 6X,
43500 1740      + 'POWER', 5X, 'POWER')
43525 1741      WRITE (YY, 4420)
43550 1742 4420 FORMAT (33X, 'TEMP', 7X, 'TEMP')
43575 1743      WRITE (YY, 4430)
43600 1744 4430 FORMAT (X, '(SEC)', 4X, '(DEG. C)', 2X, '(DEG. C)', 4X,
43625 1745      + '(DEG. C)', 3X, '(DEG. C)', 4X, '(MW)', 6X, '(MW)')
43650 1746      WRITE (YY, 4440)
43675 1747 4440 FORMAT (' ')
43700 1748      WRITE (YY, 4450) TT(1), TF1, TM1, TI1, TO1, N10, N10
43725 1749 4450 FORMAT (2X, F3.1, 5X, F8.3, 3X, F8.4, 4X, F8.4, 3X, F8.4,
43750 1750      + 4X, F8.3, 2X, F8.3)
43775 1751
43800 1752
43825 1753
43850 1754
43875 1755
43900 1756
43925 1757 4460      N10      =      N10 * 1.0E06
43950 1758      RP(K)      =      DN1 + N10
43975 1759
44000 1760      DO 4540 I = 1, LND
44025 1761
44050 1762
44075 1763
44100 1764
44125 1765
44150 1766
44175 1767
C          HERE REACTOR POWER IS CONVERTED FROM MW TO W, WHICH IS THE
C          FORMAT THAT WILL BE DISPLAYED IN THE OUTPUT.
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 PASSES 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.

```

```

44200 1768 C THIS "DO" LOOP KEEPS ON ITERATING UNTIL (LND MINUTES TIMES 300)
44225 1769 C ITERATIONS ARE COMPLETE. (LND IS THE AMOUNT OF TIME IN MINUTES
44250 1770 C THAT THE PROGRAM WILL RUN.) EVERY 0.2 SECONDS, THOSE QUANTITIES
44275 1771 C WHICH CAN APPEAR IN THE OUTPUT ARE COMPUTED. IF GRAPHICS IS
44300 1772 C CALLED FOR, THE GRAPHICS SUBROUTINE VESPASIAN IS CALLED ON THE
44325 1773 C VERY LAST ITERATION OF "DO" LOOP 4540.
44350 1774 C
44375 1775 4470 IF (Y .EQ. 1.0) H = 0.000001
44400 1776 IF (Y .EQ. 1.0) GO TO 4480
44425 1777 IF (Y .NE. 1.0) H = 0.001
44450 1778
44475 1779 4480 CALL GALBA
44500 1780
44525 1781 CONTINUE
44550 1782 Y = Y + 1.0
44575 1783 4490 CALL OTHO
44600 1784 TOTHO = TOTHO + 0.005
44625 1785 IF (NK .EQ. 1) TOTHO = 0.2
44650 1786 IF (TOTHO .LE. 0.197) GO TO 4480
44675 1787 TOTHO = 0.0
44700 1788 IF (NN .EQ. 2) GO TO 4510
44725 1789 IF (I .LE. (LND-1)) GO TO 4510
44750 1790 4500 CALL VESPASIAN
44775 1791 4510 TT(K) = TT(K-1) + 0.2
44800 1792 CALL DMTN
44825 1793
44850 1794 TF = TF + DTF(K)
44875 1795 TM = TM + DTM(K)
44900 1796 IF (K .LE. 55) TI = TI1
44925 1797 IF (K .GT. 55) TI = TI1 + DTI(K-29)
44950 1798 TO = TO1 + DTO(K)
44975 1799 TW = TW + 0.2
45000 1800 N110 = (N10 + DN1) / 1.0E06
45025 1801 STMGN = STMGEN / 1.0E06
45050 1802 PT(K) = N10 + DN1
45075 1803 STMG(K) = STMGEN
45100 1804
45125 1805 IF (ABN .GT. (TW + 0.01)) GO TO 4530
45150 1806 IF (NN .EQ. 1) GO TO 4530
45175 1807
45200 1808 WRITE (YY, 4520) TT(K), TF, TM, TI, TO, STMGN, N110
45225 1809 4520 FORMAT (X, F5.2, 4X, F8.3, 3X, F8.4, 4X, F8.4, 3X, F8.4,
45250 1810 + 4X, F8.3, 2X, F8.3)
45275 1811 4530 CONTINUE
45300 1812 IF (ABN .LE. (TW + 0.01)) TW = 0.0
45325 1813 K = K + 1
45350 1814 Q = Q + 1
45375 1815 4540 CONTINUE
45400 1816 STOP
45425 1817 END

```

```

00050 0001 C POWER KINETICS EQUATIONS
00100 0002 C
00150 0003 C SUBROUTINE GALBA
00200 0004 C
00250 0005 C THIS IS THE SUBROUTINE GALBA. IT DEPICTS REACTOR KINETICS.
00300 0006 C AS INPUT, IT RECEIVES THE FOLLOWING PARAMETERS FROM THE MAIN
00350 0007 C PROGRAM NERO:
00400 0008 C THE ISOTOPE TO BE USED.
00450 0009 C WHETHER MATRX DECOUPLING IS TO BE USED.
00500 0010 C WHETHER A CONTROL SYSTEM IS TO BE USED, AND IF SO, ITS
00550 0011 C PARAMETERS.
00600 0012 C WHETHER FEEDBACK IS TO BE USED, AND IF SO, THE REAC-
00650 0013 C TIVITY COEFFICENTS ASSOCIATED WITH IT.
00700 0014 C INITIAL POWER LEVEL.
00750 0015 C REACTIVITY STEP, IF ANY.
00800 0016 C WHETHER THE PROMPT JUMP APPROXIMATION IS TO BE USED.
00850 0017 C WHETHER THE RAMP-INPUT MODEL IS TO BE USED, AND IF SO,
00900 0018 C THE MAGNITUDE OF THE RAMP AND HOW LONG IT LASTS.
00950 0019 C
01000 0020 C GALBA THEN COMPUTES REACTOR POWER CHANGES (FROM INITIAL
01050 0021 C LEVELS) AND THEN PASSES THIS INFORMATION ON TO THE THERMAL-
01100 0022 C HYDRAULICS SUBROUTINE OTHO. TIME STEPS IN GALBA ARE 0.001 SECONDS,
01150 0023 C EXCEPT DURING THE FIRST 0.001 SECOND WHEN THEY ARE ARBITRARILY
01200 0024 C SMALLER, AND FOR THE SLOW MODE OF THE MATRIX DECOUPLING ALGORITHM,
01250 0025 C WHERE THE TIME STEPS ARE 200 TIMES THE SIZE OF THE FAST MODE TIME
01300 0026 C STEP (ALWAYS 0.001 SECONDS, EXCEPT DURING THE FIRST 0.001 SECOND).
01350 0027 C THE REASON THE TIME STEPS ARE ARBITRARILY SMALL AT FIRST IS THAT
01400 0028 C THE EQUATIONS ARE EFFECTIVE FOR TIME STEPS IN WHICH POWER DOES
01450 0029 C NOT VARY GREATLY. SINCE POWER DOES VARY GREATLY DURING THE PROMPT
01500 0030 C JUMP, SMALL TIME STEPS ARE USED SO THAT THE EQUATIONS CAN BE SOLVED
01550 0031 C FOR A TIME STEP IN WHICH POWER DOES NOT VARY GREATLY. GALBA WILL
01600 0032 C EXECUTE FOR 0.005 SECONDS AT A TIME, THEN PASS CONTROL TO THE
01650 0033 C THERMAL HYDRAULICS SUBROUTINE OTHO. OTHO THEN COMPUTES
01700 0034 C MODERATOR TEMPERATURE, REACTOR OUTLET TEMPERATURE AND FUEL TEMP-
01750 0035 C ERATURE, AND IF THE TOTAL TIME IN THE CURRENT CYCLE IS LESS THAN
01800 0036 C 0.2 SECONDS, CONTROL IS PAASED BACK TO GALBA; OTHERWISE CONTROL IS
01850 0037 C PASSED ON TO THE STEAM GENERATOR SUBROUTINE.
01900 0038 C
01950 0039 C GALBA SOLVES THE POINT-KINETICS EQUATIONS FOR CHANGES IN REAC-
02000 0040 C TOR POWER FROM STEADY-STATE CONDITIONS. IN THE EQUATIONS USED
02050 0041 C HERE, SIX GROUPS OF DELAYED-NEUTRON PRECURSORS ARE USED. THE
02100 0042 C POINT-KINETICS EQUATIONS ARE A SYSTEM OF TWO COUPLED DIFFERENTIAL
02150 0043 C EQUATIONS IN WHICH ONE EQUATION SOLVES FOR THE DELAYED-NEUTRON
02200 0044 C PRECURSORS, AND THE OTHER FOR REACTOR POWER. IT IS POSSIBLE TO
02250 0045 C SEPARATE, OR DECOUPLE, THE POINT-KINETICS EQUATIONS INTO TWO
02300 0046 C INDEPENDENT EQUATIONS, ONE OF WHICH SOLVES FOR THE DELAYED-NEUTRON
02350 0047 C PRECURSORS ALONE, THE OTHER FOR REACTOR POWER ALONE. THIS IS
02400 0048 C DONE BY MEANS OF A TRANSFORMATION MATRIX WHICH IS DEPENDENT UPON
02450 0049 C THE EIGENVALUES AND EIGENVECTORS OF THE COEFFICIENT MATRIX OF
02500 0050 C THE COUPLED POINT-KINETICS EQUATIONS. SLOW TIME STEPS ARE USED
02550 0051 C FOR THE DELAYED-NEUTRON PRECURSORS, FAST ONES FOR THE POWER
02600 0052 C RESPONSE. ARITHMETIC ERROR IS THUS AVOIDED IN SOLVING FOR THE
02650 0053 C DELAYED-NEUTRONS, WHILE CPU TIME IS ALSO SAVED. AN INVERSE
02700 0054 C TRANSFORMATION MATRIX THEN CONVERTS THESE SOLUTIONS BACK INTO
02750 0055 C THE ORIGINAL VARIABLES.
02800 0056 C
02850 0057 C THE POINT-KINETICS EQUATIONS CAN ALSO BE SOLVED USING THE

```

02900 0058 C PROMPT-JUMP APPROXIMATION AND THE RAMP-INPUT MODEL. IN THE  
02950 0059 C PROMPT-JUMP APPROXIMATION, THE ASSUMPTION IS THAT THE PROMPT-  
03000 0060 C JUMP TAKES PLACE IN ZERO TIME RATHER THAN IN A VERY SMALL TIME.  
03050 0061 C AND THUS THE POWER LEVEL JUMPS INSTANTLY FROM THE INITIAL LEVEL  
03100 0062 C TO THE NEW LEVEL. IN THE RAMP-INPUT MODEL, REACTIVITY IS INSERTED  
03150 0063 C OVER A PERIOD OF TIME INSTEAD OF INSTANTANEOUSLY.  
03200 0064 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)

03800 0076 C  
03850 0077 C IN ADDITION TO THE ONES USED IN THE COMMON BLOCK AND EXPLAINED  
03900 0078 C IN NERO, THE FOLLOWING VARIABLES ARE USED IN GALBA:  
03950 0079 C BB IS THE SUMMATION OF THE 6 GROUPS OF DELAYED  
04000 0080 C PRECURSORS.  
04050 0081 C DN1 IS THE CHANGE IN REACTOR POWER FROM STEADY-STATE.  
04100 0082 C DRC IS THE CHANGE IN REACTIVITY DUE TO THE CONTROL  
04150 0083 C SYSTEM.  
04200 0084 C DROIN IS A REACTIVITY STEP. IT IS ENTERED BY THE USER IN  
04250 0085 C NERO.  
04300 0086 C DRO1 IS REACTIVITY STEP USED IN PROMPT-JUMP APPROXIMA-  
04350 0087 C TION, AFTER THE REACTIVITY STEP PASSED FROM NERO  
04400 0088 C IS ADJUSTED FOR FEEDBACK DURING THE PROMPT JUMP.  
04450 0089 C DRO2 IS THE DIFFERENCE BETWEEN DRO1 AND THE REACTIVITY  
04500 0090 C STEP PASSED FROM NERO. USED IN COMPUTING THE  
04550 0091 C PROMPT-JUMP APPROXIMATION.  
04600 0092 C DROT IS THE DIFFERENCE BETWEEN DRO1 AND DRO2. IF  
04650 0093 C IT IS SMALL ENOUGH, THE PJ APPROXIMATION IS  
04700 0094 C CONSIDERED TO BE COMPUTED.  
04750 0095 C DTFGT IS THE ESTIMATED CHANGE IN FUEL TEMPERATURE THAT  
04800 0096 C OCCURS DURING ANY GIVEN INTERVAL OF 0.2 SECONDS.  
04850 0097 C IT IS USED IN REACTIVITY CALCULATIONS. ITS PURPOSE  
04900 0098 C IS TO PROVIDE SOME FUEL TEMPERATURE CHANGE DATA  
04950 0099 C DURING THE 0.2 SECOND CYCLE THAT GALBA RUNS BEFORE  
05000 0100 C PASSING CONTROL ON TO OTHO.  
05050 0101 C DTMGT IS ESTIMATED CHANGE IN MODERATOR TEMPERATURE THAT  
05100 0102 C OCCURS DURING ANY 0.2 SECOND CYCLE. IT IS USED FOR  
05150 0103 C THE SAME REASON THAT DTFGT IS. AFTER THE 0.2 SECOND  
05200 0104 C INTERVAL, DTFGT AND DTMGT BOTH ARE SET EQUAL TO  
05250 0105 C ZERO, AND NEW VALUES FOR THE FUEL AND MODERATOR  
05300 0106 C TEMPERATURE CHANGES ARE COMPUTED IN OTHO. WHAT  
05350 0107 C THESE TWO VARIABLES DO IS PREVENT THE ABRUPT FUEL  
05400 0108 C TEMPERATURE CHANGES (AND THEREBY, SUDDEN REACTIVITY  
05450 0109 C AND POWER CHANGES) THAT WOULD OCCUR IF THE FUEL AND  
05500 0110 C MODERATOR TEMPERATURES WERE ALLOWED TO CHANGE ONLY  
05550 0111 C ONCE EVERY 0.2 SECONDS (THAT IS, EVERY TIME CONTROL  
05600 0112 C IS PASSED TO OTHO).  
05650 0113 C DTMT IS THE SUM OF ALL DTM(K). USED IN THE INTEGRAL  
05700 0114 C CONTROLLER OF THE CONTROL SYSTEM.



05750	0115	C	L	IS PROMPT NEUTRON LIFETIME.
05800	0116	C	LK	IS THE 1X1 MATRIX FORMED BY MULTIPLYING THE 1X6
05850	0117	C		L MATRIX AND THE 6X1 K MATRIX IN THE MATRIX DE-
05900	0118	C		COUPLING ALGORITHM. IS USED TO COMPUTE ACTUAL
05950	0119	C		REACTOR POWER FROM DECOUPLED VARIABLES.
06000	0120	C	PWRPJ	IS THE TOTAL REACTOR POWER, AS COMPUTED IN THE
06050	0121	C		PROMPT-JUMP APPROXIMATION.
06100	0122	C	PWR1	IS THE CHANGE IN REACTOR POWER, AS COMPUTED DURING
06150	0123	C		THE PREVIOUS ITERATION.
06200	0124	C	PWR2	IS THE DIFFERENCE BETWEEN CURRENT REACTOR POWER
06250	0125	C		CHANGE AND PWR1.
06300	0126	C	SIG	IS THE 6-GROUP SUMMATION OF THE DELAYED NEUTRON
06350	0127	C		DECAY CONSTANT TIMES THE CHANGE IN PRECURSOR
06400	0128	C		DENSITY. IS USED IN POINT-KINETICS EQUATIONS.
06450	0129	C	SIGPJ	IS ANALOGOUS TO SIG, ONLY IS USED IN THE PROMPT-
06500	0130	C		JUMP APPROXIMATION.
06550	0131	C	T	IS THE TOTAL ELAPSED TIME IN THE CURRENT SERIES OF
06600	0132	C		ITERATIONS. WHEN IT EQUALS 0.1 SECONDS, GALBA
06650	0133	C		CAUSES THE DECOUPLED ANALOGUE OF THE DELAYED-
06700	0134	C		NEUTRON PRECURSORS TO BE COMPUTED.
06750	0135	C	TH	IS THE TOTAL TIME ELAPSED IN THE CURRENT SERIES
06800	0136	C		OF ITERATIONS. WHEN IT EQUALS 0.005, CONTROL IS
06850	0137	C		PASSED TO THE SUBROUTINE OTHO.
06900	0138	C	X	IS A VARIABLE THAT WHOSE MAGNITUDE GOVERNS THE
06950	0139	C		SIZE OF THE TIME STEP USED IN THE FIRST 0.2 SECONDS
07000	0140	C		OF THE RUNNING OF GALBA.
07050	0141	C	XX	IS EQUAL TO -1.0. IS USED IN THE COMPUTATION OF
07100	0142	C		DTFGT AND DTMTG.
07150	0143	C	B(6)	IS THE THE DELAYED-NEUTRON FRACTON, OF EACH GROUP.
07200	0144	C	CIO(6)	IS THE STEADY-STATE DELAYED PRECURSOR DENSITY.
07250	0145	C	DCI(6)	IS THE CHANGE IN DELAYED PRECURSOR DENSITY FROM
07300	0146	C	LMB(6)	IS THE DECAY CONSTANT OF EACH OF THE 6 DELAYED-
07350	0147	C		NEUTRON GROUPS.
07400	0148	C		THE STEADY STATE.
07450	0149	C		
07500	0150	C	THE FOLLOWING ARE VARIABLES USED IN MATRIX DECOUPLING:	
07550	0151	C	A(7,7)	IS THE COEFFICIENT MATRIX OF THE COUPLED SYSTEM
07600	0152	C		OF THE POINT-KINETICS EQUATIONS. IS INPUT INTO
07650	0153	C		THE EIGENVALUE-COMPUTING SUBROUTINE EIGEN.
07700	0154	C		THE JORDAN CANONICAL FORM OF THIS MATRIX IS
07750	0155	C		A = MJQ, WHERE J IS THE DIAGONAL MATRIX CONSISTING
07800	0156	C		OF THE EIGENVALUES OF A, Q IS THE FUNDAMENTAL
07850	0157	C		MATRIX CONSISTING OF THE EIGENVECTORS OF A, AND
07900	0158	C		M IS THE INVERSE OF Q.
07950	0159	C	AI(7,7)	IS A 7X7 MATRIX CONSISTING OF THE IMAGINARY PORTION
08000	0160	C		OF THE FUNDAMENTAL MATRIX Q OF THE COEFFICIENT
08050	0161	C		MATRIX A. FOR ANY COEFFICIENT MATRIX A THAT WILL
08100	0162	C		BE ENCOUNTERED IN THIS PROGRAM, THERE ARE ONLY
08150	0163	C		REAL EIGENVALUES AND EIGENVECTORS, AND ALL
08200	0164	C		ELEMENTS OF AI(7,7) EQUAL ZERO.
08250	0165	C	AR(7,7)	IS THE FUNDAMENTAL EIGENVECTOR MATRIX OF THE
08300	0166	C		A MATRIX. CORRESPONDS TO THE Q MATRIX MENTIONED
08350	0167	C		EARLIER. BOTH THE AI(7,7) AND AR(7,7) MATRICES
08400	0168	C		ARE OUTPUT OF THE PORTLIBRARY SUBROUTINE EIGEN.
08450	0169	C		EVEN THOUGH THE VALUES OF ALL OF THE ELEMENTS OF
08500	0170	C		OF AI(7,7) AND ALL OF THE OTHER IMAGINARY MATRICES
08550	0171	C		ARE ZERO, THE MATRICES MUST BE THUS DEFINED, AS

08600	0172	C	
08650	0173	C	
08700	0174	C	
08750	0175	C	
08800	0176	C	
08850	0177	C	
08900	0178	C	
08950	0179	C	
09000	0180	C	
09050	0181	C	
09100	0182	C	
09150	0183	C	
09200	0184	C	
09250	0185	C	
09300	0186	C	
09350	0187	C	
09400	0188	C	
09450	0189	C	
09500	0190	C	
09550	0191	C	
09600	0192	C	
09650	0193	C	
09700	0194	C	
09750	0195	C	
09800	0196	C	
09850	0197	C	
09900	0198	C	
09950	0199	C	
10000	0200	C	
10050	0201	C	
10100	0202	C	
10150	0203	C	
10200	0204	C	
10250	0205	C	
10300	0206	C	
10350	0207	C	
10400	0208	C	
10450	0209	C	
10500	0210	C	
10550	0211	C	
10600	0212	C	
10650	0213	C	
10700	0214	C	
10750	0215	C	
10800	0216	C	
10850	0217	C	
10900	0218	C	
10950	0219	C	
11000	0220	C	
11050	0221	C	
11100	0222	C	
11150	0223	C	
11200	0224	C	
11250	0225	C	
11300	0226	C	
11350	0227	C	
11400	0228	C	

THE SUCCESSFUL APPLICATION OF THE PORTLIBRARY SUBROUTINES EIGEN AND CLINQ DEMAND AT LEAST ZERO ENTRIES IN ALL OF THE POSITIONS OF THESE MATRICES.

BR(7,7) IS A 7X7 IDENTITY MATRIX USED AS INPUT IN THE PORTLIBRARY SUBROUTINE CLINQ TO HELP INVERT THE AR(7,7), OR Q, MATRIX INTO THE INVERSE FUNDAMENTAL EIGENVECTOR MATRIX M REFERRED TO EARLIER.

BI(7,7) IS THE IMAGINARY COMPANION TO BR(7,7). ALL ITS ELEMENTS EQUAL ZERO.

B2 IS THE 1X1 B2 MATRIX CONSISTING OF THE COEFFICIENT OF THE FAST MODE SYSTEM OF DECOUPLED DIFFERENTIAL EQUATIONS. BECAUSE IT IS 1X1, IT IS EQUAL TO THE LARGEST EIGENVALUE OF THE 7X7 COEFFICIENT MATRIX OF THE COUPLED POINT-KINETICS EQUATIONS. THIS IS THE EIGENVALUE THAT CORRESPONDS TO THE FAST MODE.

DI(6) IS A CORRECTION MATRIX. IT IS 6X1 WHEN COMPUTING THE L MATRIX, 1X6 WHEN COMPUTING THE K MATRIX. IT IS COMPUTED TO SERVE AS A CORRECTION TO THE OLD MATRICES L OR K. IT IS ADDED TO THE OLD MATRIX TO GENERATE A NEW ONE.

DY(7) IS A 7X1 MATRIX REPRESENTING DECOUPLED VARIABLES CORRESPONDING TO EACH OF THE 7 VARIABLES SOLVED FOR IN THE 7X7 SYSTEM OF COUPLE POINT-KINETICS EQUATIONS (6 DELAYED-NEUTRON GROUPS PLUS PROMPT RESPONSE). IT REPRESENTS ALL OF THE VARIABLES THAT EXIST AFTER THE COUPLED SYSTEM OF POINT-KINETICS EQUATIONS IS DECOUPLED AND TRANSFORMED INTO THE DECOUPLED VARIABLES. THE FIRST SIX ARE THE ANALOGUES OF THE SIX DELAYED-NEUTRON GROUPS, WHILE THE SEVENTH IS THE ANALOGUE OF THE PROMPT RESPONSE.

DYIN(6, 18) IS THE DY(I) MATRIX (MINUS THE 7TH ELEMENT, WHICH IS FOR THE PROMPT RESPONSE), AS USED DURING DURING THE PROMPT JUMP. DURING THE PROMPT JUMP, TIME STEPS ARE OF VARIABLE SIZE, AND THE DY(I) MATRIX IS DEFINED FREQUENTLY DURING THE FIRST 0.2 SECONDS. THUS IT IS NECESSARY TO HAVE A 2-DIMENSIONAL MATRIX THAT CAN KEEP TRACK OF WHICH DY(I) IS CURRENTLY OF INTEREST.

INT IS A "HOLDING VARIABLE". WHEN CARRYING OUT MULTIPLICATION OPERATIONS IN THE MATRIX DECOUPLING ALGORITHM, IT IS USED TO STORE THE SUMS OF A ROW OF MULTIPLICATIONS. AFTER THIS SUM IS USED IN FURTHER OPERATIONS, IT IS SET EQUAL TO ZERO FOR REUSE LATER.

H2 IS THE TIME STEP USED FOR THE DECOUPLED ANALOGUES OF THE DELAYED-NEUTRON PRECURSORS. IT IS 200 TIMES THE SIZE OF THE TIME STEP CURRENTLY IN USE FOR THE PROMPT RESPONSE.

H2O(22) IS A VARIABLE H2 TIME STEP USED ONLY DURING THE FIRST 0.2 SECONDS.

KK IS A DUMMY VALUE USED TO ARRANGE THE EIGENVALUES OF THE COEFFICIENT MATRIX OF THE COUPLED POINT-KINETICS EQUATIONS IN ASCENDING ORDER OF THEIR ABSOLUTE VALUES.

KM(6) IS THE 6X1 K MATRIX. IT AND THE L MATRIX ARE FIRST USED TO TRANSFORM THE COUPLED VARIABLES OF THE

11450	0229	C	
11500	0230	C	
11550	0231	C	
11600	0232	C	
11650	0233	C	
11700	0234	C	
11750	0235	C	
11800	0236	C	
11850	0237	C	
11900	0238	C	
11950	0239	C	
12000	0240	C	
12050	0241	C	
12100	0242	C	
12150	0243	C	
12200	0244	C	
12250	0245	C	
12300	0246	C	
12350	0247	C	
12400	0248	C	
12450	0249	C	
12500	0250	C	
12550	0251	C	
12600	0252	C	
12650	0253	C	
12700	0254	C	
12750	0255	C	
12800	0256	C	
12850	0257	C	
12900	0258	C	
12950	0259	C	
13000	0260	C	
13050	0261	C	
13100	0262	C	
13150	0263	C	
13200	0264	C	
13250	0265	C	
13300	0266	C	
13350	0267	C	
13400	0268	C	
13450	0269	C	
13500	0270	C	
13550	0271	C	
13600	0272	C	
13650	0273	C	
13700	0274	C	
13750	0275	C	
13800	0276	C	
13850	0277	C	
13900	0278	C	
13950	0279	C	
14000	0280	C	
14050	0281	C	
14100	0282	C	
14150	0283	C	
14200	0284	C	
14250	0285	C	

POINT-KINETICS EQUATIONS INTO THEIR DECOUPLED ANALOGUES, THEN, AFTER SOLUTIONS FOR THE DECOUPLED VARIABLES ARE OBTAINED, ARE USED TO TRANSFORM THESE NEW VALUES FOR THE DECOUPLED VARIABLES INTO THEIR COUPLED, OR "REAL WORLD" VARIABLES. IN PRACTICE, THE ONLY "REAL WORLD" VARIABLE OF INTEREST IS REACTOR POWER, AND IS THE ONLY DECOUPLED VARIABLE TO BE BACK-TRANSFORMED. ONLY THE STEADY-STATE, INITIAL VALUES OF THE COUPLED VARIABLES ARE TRANSFORMED INTO THEIR DECOUPLED ANALOGUES; THESE VALUES ARE NEEDED TO SOLVE THE DECOUPLED DIFFERENTIAL EQUATIONS.

KL(6,6) IS THE 6X6 MATRIX FORMED BY MULTIPLYING THE K MATRIX TIMES THE L MATRIX. IT IS USED TO TRANSFORM THE INITIAL VALUES FOR THE SIX DELAYED-NEUTRON GROUPS INTO THEIR DECOUPLED ANALOGUES.

LM(6) IS THE 1X6 L MATRIX. IN ADDITION TO PERFORMING THE FUNCTIONS DESCRIBED IN THE KM(6) NARRATIVE, THE VALUES OF ITS ELEMENTS ARE ALSO USED TO COMPUTE THE B1 MATRIX. (THE B2 MATRIX IS SIMPLY THE EIGENVALUE OF THE FAST MODE.) THE K MATRIX IN TURN IS DERIVED BY SOLVING THE ALGEBRAIC RICCATI EQUATION.

N THE ORDER OF THE MATRIX A. IS INPUT TO THE PORT-LIBRARY SUBROUTINE EIGEN.

NB IS THE NUMBER OF VARIABLES IN THE POINT-KINETICS EQUATIONS (7). IS INPUT TO THE PORTLIBRARY SUBROUTINE CLINQ.

NM IS THE ROW DIMENSION OF THE MATRIX A(7). IS USED AS INPUT TO THE PORTLIBRARY SUBROUTINE EIGEN.

Q22 IS THE 1X1 SUBMATRIX IN THE (2,2) POSITION OF THE FUNDAMENTAL EIGENVECTOR MATRIX, SPLIT INTO 4 SUBMATRICES.

RIE IS THE RESIDUAL ERROR MATRIX USED IN THE MATRIX DECOUPLING ALGORITHM. IT IS THE EUCLIDEAN NORM OF THE 6X1 (WHEN COMPUTING THE "L" MATRIX) RESIDUAL ERROR MATRIX, WHICH IS 1X6 WHEN COMPUTING THE "K" MATRIX.

RMT IS THE SQUARE OF RIE.

RM(6) IS THE RESIDUAL ERROR MATRIX (6X1 WHEN COMPUTING THE LM(6) MATRIX, 1X6 WHEN COMPUTING THE KM(6) MATRIX). IF ITS EUCLIDEAN NORM IS SMALL ENOUGH, THEN THE LM(6) AND THE KM(6) MATRICES ARE CONSIDERED TO HAVE CONVERGED TOWARD THEIR TRUE VALUES.

XR(7,7) IS THE 7X7 INVERSE FUNDAMENTAL EIGENVECTOR MATRIX OF THE COEFFICIENT MATRIX OF THE COUPLED POINT-KINETICS EQUATIONS. ALSO KNOWN AS THE M MATRIX. IT IS OUTPUT FROM THE PORTLIBRARY SUBROUTINE CLINQ.

XI(7,7) IS THE 7X7 MATRIX WHICH IS THE IMAGINARY COMPANION TO XR(7,7). THE VALUES OF ALL ITS ELEMENTS ARE EQUAL TO ZERO.

VAL IS A HOLDING VARIABLE USED IN ARRANGING THE EIGENVALUES OF THE COEFFICIENT MATRIX OF THE COUPLED POINT-KINETICS EQUATIONS IN ASCENDING ORDER OF THEIR ABSOLUTE VALUES.

VALP IS A HOLDING VARIABLE USED IN ARRANGING THE EIGENVALUES OF THE COEFFICIENT MATRIX OF THE COUPLED

```

14300 0286 C POINT-KINETICS EQUATIONS IN ASCENDING ORDER OF
14350 0287 C THEIR ABSOLUTE VALUES. FOR A FURTHER EXPLANATION
14400 0288 C OF VAL & VALP, SEE THE COMMENTS IMMEDIATELY PRE-
14450 0289 C CEEDING DO LOOP 100.
14500 0290 C VK IS AN INTEGER WHOSE VALUE DURING THE FIRST 0.2
14550 0291 C SECONDS OF OPERATION DETERMINES WHETHER A DY(1)
14600 0292 C MATRIX IS COMPUTED OR NOT.
14650 0293 C VL IS THE SAME AS VK, BUT IS USED AT SLIGHTLY LATER
14700 0294 C TIMES.
14750 0295 C WR(7) IS THE 7X1 MATRIX CONSISTING OF THE EIGENVALUES OF
14800 0296 C THE A(7,7) MATRIX. IS OUTPUT OF THE PORTLIBRARY
14850 0297 C SUBROUTINE EIGEN.
14900 0298 C WI(7) IS THE IMAGINARY COMANION TO WR(7). THE VALUES OF
14950 0299 C ALL ITS ELEMENTS ARE EQUAL TO ZERO.
15000 0300 C Z(7,7) IS THE FUNDAMENTAL EIGENVECTOR MATRIX OF THE COEF-
15050 0301 C FICIENT MATRIX OF THE COUPLED POINT-KINETICS EQUA-
15100 0302 C TIONS. IT IS OUTPUT FROM THE PORTLIBRARY SUBROU-
15150 0303 C TINE EIGEN, AND IS IDENTICAL TO THE AR(7,7) MATRIX.
15200 0304 C (SINCE THE CLINQ SUBROUTINE DEMANDS THAT ANY MATRIX
15250 0305 C TO BE INVERTED BE CALLED THE AR MATRIX, IT WAS
15300 0306 C NECESSARY TO "RELABEL" THE Z(7,7) MATRIX BY CREA-
15350 0307 C TING THE AR(7,7) MATRIX WHICH IS EXACTLY EQUAL TO
15400 0308 C IT. IN THIS PROGRAM, THE FUNDAMENTAL EIGENVECTOR
15450 0309 C MATRIX IS INVERTED BECAUSE THIS MAKES THE INITIAL
15500 0310 C APPROXIMATION TO THE LM(6) MATRIX EASIER TO
15550 0311 C COMPUTE.)
15600 0312 C
15650 0313 REAL DNI, DRC, DROIN, DROD, DRODSQ, DROTSQ, DTFGT,
15700 0314 + DIMGT, DIMGTT, DTMT, LMBD, PTG, PTO, PWRPJ,
15750 0315 + PWRRI, PWR1, PWR2, SIG, SIGPJ, T, TTG, X
15800 0316 REAL CIO(6), DCI(6)
15850 0317 REAL DROR, Q22
15900 0318 INTEGER I, INT1, INT2, INT3, INT4, INT5, INT6, KK, M, N, NB,
15950 0319 + NM, SV, VK, VL
16000 0320 REAL*8 A(7,7), AI(7,7), AR(7,7), BI(7,7), BR(7,7), B1(6,6),
16050 0321 + DI(6), DY(7), KM(6), KL(6,6), LM(6), M31(5),
16100 0322 + Q21(7), RM(6), SS(7), XI(7,7), XR(7,7), WI(14),
16150 0323 + WR(14), Z(7,7), B(6), LMB(6), BB, B2, INT, L, LK,
16200 0324 + RIE, RMT, VAL, VALP, XX
16250 0325 REAL*8 DYIN(6,18), H20(22)
16300 0326 REAL ADJF, ADJM, DRO1, DRO2, DROT, SSD, SSDN, SSDO, TH
16350 0327 DIMENSION LPWR(7), MPWR(7), ZERO(1), TEN(2), TWNT(2), THRT(2),
16400 0328 + FORT(2), FFTY(2), SIXT(2), DNIG(22), STM(22)
16450 0329
16500 0330
16550 0331 C HERE, INITIALIZATIONS TAKE PLACE, AND ONLY DURING THE FIRST
16600 0332 C 0.2 SECONDS. ALSO, IF THERE IS TO BE NO KINETICS, CONTROL IS
16650 0333 C PASSED IMMEDIATELY TO OTHO.
16700 0334
16750 0335 IF (NK .EQ. 1) GO TO 480
16800 0336 IF (Y .EQ. 1.0) TTG = 0.0
16850 0337 IF (Y .NE. 1.0) GO TO 50
16900 0338
16950 0339
17000 0340 ADJF = 1.0
17050 0341 ADJM = 1.0
17100 0342 BB = 0.0

```

17150	0343	DNI	=	0.0
17200	0344	DROIN	=	DRO
17250	0345	DROR	=	DRO
17300	0346	DRC	=	0.0
17350	0347	DTFGT	=	0.0
17400	0348	DTMGT	=	0.0
17450	0349	DTMGT	=	0.0
17500	0350	DTMT	=	0.0
17550	0351	H2O(1)	=	0.000001
17600	0352	INT	=	0.0
17650	0353	L	=	0.0001
17700	0354	PTG	=	N10 + DN1
17750	0355	PTO	=	PTG
17800	0356	PWRPJ	=	N10
17850	0357	PWRR1	=	0.0
17900	0358	PWR1	=	0.0
17950	0359	SIG	=	0.0
18000	0360	SIGPJ	=	0.0
18050	0361	T	=	0.0
18100	0362	TH	=	0.0
18150	0363	TRI	=	0.0
18200	0364	TTG	=	0.0
18250	0365	VK	=	1
18300	0366	VL	=	2
18350	0367	X	=	1.0
18400	0368	XX	=	-1.0
18450	0369	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.

18800	0376	IF (IS .EQ. 1)	B(1)	=	2.2876E-04
18850	0377	IF (IS .EQ. 1)	B(2)	=	7.9534E-04
18900	0378	IF (IS .EQ. 1)	B(3)	=	6.7032E-04
18950	0379	IF (IS .EQ. 1)	B(4)	=	7.3948E-04
19000	0380	IF (IS .EQ. 1)	B(5)	=	1.3566E-04
19050	0381	IF (IS .EQ. 1)	B(6)	=	9.0440E-05
19100	0382	IF (IS .EQ. 1)	LMB(1)	=	0.0126
19150	0383	IF (IS .EQ. 1)	LMB(2)	=	0.0337
19200	0384	IF (IS .EQ. 1)	LMB(3)	=	0.1390
19250	0385	IF (IS .EQ. 1)	LMB(4)	=	0.3250
19300	0386	IF (IS .EQ. 1)	LMB(5)	=	1.1300
19350	0387	IF (IS .EQ. 1)	LMB(6)	=	2.5000
19400	0388	IF (IS .EQ. 2)	B(1)	=	2.1450E-04
19450	0389	IF (IS .EQ. 2)	B(2)	=	1.4235E-03
19500	0390	IF (IS .EQ. 2)	B(3)	=	1.2740E-03
19550	0391	IF (IS .EQ. 2)	B(4)	=	2.5675E-03
19600	0392	IF (IS .EQ. 2)	B(5)	=	7.4750E-04
19650	0393	IF (IS .EQ. 2)	B(6)	=	2.7300E-04
19700	0394	IF (IS .EQ. 2)	LMB(1)	=	0.0124
19750	0395	IF (IS .EQ. 2)	LMB(2)	=	0.0305
19800	0396	IF (IS .EQ. 2)	LMB(3)	=	0.1110
19850	0397	IF (IS .EQ. 2)	LMB(4)	=	0.3010
19900	0398	IF (IS .EQ. 2)	LMB(5)	=	1.1400
19950	0399	IF (IS .EQ. 2)	LMB(6)	=	3.0100

```

20000 0400      IF (IS .EQ. 3)  B(1)  = 7.4200E-05
20050 0401      IF (IS .EQ. 3)  B(2)  = 6.3176E-04
20100 0402      IF (IS .EQ. 3)  B(3)  = 4.4732E-04
20150 0403      IF (IS .EQ. 3)  B(4)  = 6.9112E-04
20200 0404      IF (IS .EQ. 3)  B(5)  = 1.8232E-04
20250 0405      IF (IS .EQ. 3)  B(6)  = 9.3280E-05
20300 0406      IF (IS .EQ. 3)  LMB(1) = 0.0128
20350 0407      IF (IS .EQ. 3)  LMB(2) = 0.0301
20400 0408      IF (IS .EQ. 3)  LMB(3) = 0.1240
20450 0409      IF (IS .EQ. 3)  LMB(4) = 0.3250
20500 0410      IF (IS .EQ. 3)  LMB(5) = 1.1200
20550 0411      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 INITIALIZATIONS FOR
C VARIABLES USED IN MATRIX DECOUPLING ARE MADE HERE.

```

```

20900 0418      N          = 7
20950 0419      NB         = 7
21000 0420      NM         = 7
21050 0421      RIE        = 0.0
21100 0422      RMT        = 0.0
21150 0423      DO 10 I = 1,6
21200 0424          A(1,I) = 0.0
21250 0425          A(2,I) = 0.0
21300 0426          A(3,I) = 0.0
21350 0427          A(4,I) = 0.0
21400 0428          A(5,I) = 0.0
21450 0429          A(6,I) = 0.0
21500 0430          A(1,I) = -LMB(I)
21550 0431          A(7,I) = LMB(I)
21600 0432          A(1,7) = B(I) / L
21650 0433          M31(I) = 0.0
21700 0434          DI (I) = 0.0
21750 0435          KM(I)  = 0.0
21800 0436          RM (I)  = 0.0
21850 0437      10 CONTINUE
21900 0438          DO 30 I = 1,7
21950 0439          DO 20 J = 1,7
22000 0440              AI(I,J) = 0.0
22050 0441              BI(I,J) = 0.0
22100 0442              BR(I,J) = 0.0
22150 0443      20 CONTINUE
22200 0444              BR(1,1) = 1.0
22250 0445              DY(I)  = 0.0
22300 0446              DYIN(1,1) = 0.0
22350 0447      30 CONTINUE
22400 0448
22450 0449
22500 0450      C          HERE, INITIAL VALUES FOR THE DELAYED-NEUTRON FRACTIONS ARE
22550 0451      C DERIVED. ALSO, THE DELAYED-NEUTRON FRACTIONS ARE SUMMED.
22600 0452
22650 0453          DO 40 I = 1,6
22700 0454              CIO(I) = N10 * B(I) / (L * LMB(I))
22750 0455              BB    = BB  + B(I)
22800 0456              DCI(I) = 0.0

```

```

22850 0457 40 CONTINUE
22900 0458          SSD = 2.0 * DRO / BB
22950 0459          SSDO = 2.0 * DRO / BB
23000 0460
23050 0461
23100 0462 50 CONTINUE
23150 0463
23200 0464 C          HERE, REACTIVITY EFFECTS DUE TO FEEDBACK AND THE REACTOR
23250 0465 C CONTROL SYSTEM ARE COMPUTED.
23300 0466
23350 0467          IF ((ZZ .EQ. 1) .AND. (K .LE. 29.0)) GO TO 480
23400 0468          DTMT = DTM(K-1) + DTMT
23450 0469          IF ((X .NE. 1.0) .AND. (CS .EQ. 2))
23500 0470 +          DRO = ALPHF * DTF(K-1) + ALPHM * DTM(K-1) + DROIN +
23550 0471 +          (RIR * TTG)
23600 0472 60 - IF ((X .NE. 1.0) .AND. (CS .EQ. 1))
23650 0473 +          DRO = ALPHF * DTF(K-1) + ALPHM * DTM(K-1) + DROIN +
23700 0474 +          (RIR * TTG) + DRC
23750 0475          DRC = 0.2 * (AA * (DTM(K-1) + (0.2 * DTMT / TAU)) -
23800 0476 +          (DRC / TAUC)) + DRC
23850 0477          IF ((X .NE. 1.0) .AND. (CS .EQ. 2) .AND. (ZZ .EQ. 1))
23900 0478 +          DRO = ALPHF * DTF(K-1) + ALPHM * DTM(K-1)
23950 0479          IF ((X .NE. 1.0) .AND. (CS .EQ. 1) .AND. (ZZ .EQ. 1))
24000 0480 +          DRO = ALPHF * DTF(K-1) + ALPHM * DTM(K-1) + DRC
24050 0481 +          DRC = 0.2 * (AA * (DTM(K-1) + (0.2 * DTMT / TAU)) -
24100 0482 +          (DRC / TAUC)) + DRC
24150 0483
24200 0484 C
24250 0485 C          HERE, THE MATRIX DECOUPLING ALGORITHM OCCURS. IF IT IS NOT
24300 0486 C SELECTED, CONTROL IMMEDIATELY JUMPS TO STATEMENT 330. IF IT IS,
24350 0487 C IT STARTS HERE. AS GALBA PROCEEDS THROUGH THIS ALGORITHM, AT
24400 0488 C VARIOUS PLACES COMMENT STATEMENTS WILL BE INSERTED TO EXPLAIN
24450 0489 C WHAT IS GOING ON.
24500 0490 C
24550 0491 C
24600 0492 70 IF (EIG .EQ. 2) GO TO 330
24650 0493 C
24700 0494 C          HERE, THE (7,7)TH ELEMENT OF THE COEFFICIENT MATRIX OF THE
24750 0495 C COUPLED POINT-KINETICS EQUATIONS IS DEFINED. AS IT IS REACTIVITY
24800 0496 C DEPENDENT, IT IS THE ONLY PART OF THIS MATRIX THAT VARIES. THE
24850 0497 C OTHER ELEMENTS OF THIS MATRIX ARE REDEFINED, TOO, BUT ALWAYS WITH
24900 0498 C THE SAME VALUES. THE REASON THEY NEED TO BE REDEFINED IS THAT
24950 0499 C EXECUTION OF THE PORTLIBRARY SUBROUTINE EIGEN, WHICH TAKES PLACE
25000 0500 C NEXT, OVERWRITES OR CANCELS ALL OF ITS INPUT, SO IT ALL NEEDS TO
25050 0501 C BE REDEFINED PRIOR TO THE NEXT TIME EIGEN IS CALLED, EVEN IF
25100 0502 C DOES NOT CHANGE. THE FIRST TIME THE MATRIX DECOUPLING ALGORITHM IS
25150 0503 C CALLED, HOWEVER, REACTIVITY IS CONSIDERED TO BE ZERO. THIS IS TO
25200 0504 C CALCULATE THE INITIAL VALUES FOR THE ANALOGUES OF THE VARIABLES
25250 0505 C OF THE DELAYED-NEUTRON RESPONSE AND OF THE PROMPT RESPONSE. AS
25300 0506 C THIS PROGRAM IS NOW RUN ONLY ONCE, HOWEVER, NONE OF THE ABOVE NOW
25350 0507 C APPLIES. THIS INFORMATION IS INCLUDED FOR THOSE WHO MAY WISH TO
25400 0508 C ALTER GALBA IN THE FUTURE SUCH THAT MATRIX DECOUPLING COULD BE
25450 0509 C CALLED MORE THAN ONCE DURING A GIVEN RUN. ALSO, THE K MATRIX IS
25500 0510 C INITIALLY DEFINED IN TERMS OF THE A MATRIX, SO ITS VALUES WOULD
25550 0511 C NEED TO BE REDEFINED ANYWAY.
25600 0512 C
25650 0513          IF (Y .NE. 1.0) GO TO 330

```

```

25700 0514 80      A(7,7) = (RHO - BB) / L
25750 0515 C
25800 0516 C      HERE, THE PORTLIBRARY SUBROUTINE "EIGEN" IS CALLED. ITS
25850 0517 C      PURPOSE IS TO TAKE THE COEFFICIENT MATRIX OF THE COUPLED SYSTEM
25900 0518 C      OF POINT-KINETICS EQUATIONS AND DERIVE ITS EIGENVALUES AND
25950 0519 C      EIGENVECTORS.
26000 0520 C
26050 0521 C      CALL DEIGEN (NM, N, A, WR, WI, Z)
26100 0522 C
26150 0523 C      HERE, THE REMAINING ELEMENTS OF THE COEFFICIENT MATRIX OF THE
26200 0524 C      THE SYSTEM OF COUPLED POINT-KINETICS EQUATIONS ARE REDEFINED, DUE
26250 0525 C      TO THEIR HAVING BEEN OVERWRITTEN DURING THE CALLING OF "EIGEN",
26300 0526 C      WHICH TOOK PLACE JUST PREVIOUSLY.
26350 0527 C
26400 0528 C      DO 90 I = 1,6
26450 0529 C          A(1,1) = 0.0
26500 0530 C          A(2,1) = 0.0
26550 0531 C          A(3,1) = 0.0
26600 0532 C          A(4,1) = 0.0
26650 0533 C          A(5,1) = 0.0
26700 0534 C          A(6,1) = 0.0
26750 0535 C          A(1,1) = -LMB(1)
26800 0536 C          A(7,1) = LMB(1)
26850 0537 C          A(1,7) = B(1) / L
26900 0538 90      CONTINUE
26950 0539 C          RIE = 0.0
27000 0540 C
27050 0541 C      HERE, THE EIGENVALUES THAT WERE JUST DERIVED FROM "EIGEN" ARE
27100 0542 C      REARRANGED, IN ASCENDING ORDER OF THEIR ABSOLUTE VALUES. THIS IS
27150 0543 C      NECESSARY BECAUSE THE MATRIX DECOUPLING ALGORITHM DEMANDS THAT THE
27200 0544 C      FUNDAMENTAL EIGENVECTOR MATRIX CORRESPOND TO A DIAGONAL EIGENVALUE
27250 0545 C      MATRIX IN WHICH THIS HAS BEEN DONE, BUT "EIGEN" DOES NOT DO THIS.
27300 0546 C      FIRST, THE SQUARE ROOTS OF THE SQUARES OF THE EIGENVALUES ARE
27350 0547 C      DERIVED. THIS WILL RESULT IN ALL POSITIVE VALUES ("DO" LOOP 100).
27400 0548 C      THEN EACH EIGEN VALUE IS EXAMINED AS DELIVERED FROM "EIGEN". IT
27450 0549 C      IS ASSUMED THAT ANY EIGENVALUE EXAMINED IS THE SMALLEST OF THE LOT.
27500 0550 C      THE FIRST EIGENVALUE EXAMINED IS ASSIGNED THE VARIABLE NAME "VAL".
27550 0551 C      ANOTHER VARIABLE, "VALP", IS SET EQUAL TO IT. THEN THE EIGENVALUE
27600 0552 C      IS EXAMINED AGAINST ALL THE OTHERS. AS LONG AS IT IS SMALLER THAN
27650 0553 C      ANY OF THEM, THE ASSUMPTION IS CONSIDERED TO HOLD UP, AND THE
27700 0554 C      EIGENVALUE IS THEN COMPARED TO THE NEXT ONE. (THIS TAKES PLACE IN
27750 0555 C      "DO" LOOP 110, WHICH IS NESTED IN 130.) IF IT IS LARGER THAN THE
27800 0556 C      NEW EIGENVALUE BEING EXAMINED, THE NEW EIGENVALUE THEN BECOMES THE
27850 0557 C      "NEW" POSSIBLY SMALLEST EIGENVALUE, AND "VALP" IS SET EQUAL TO IT.
27900 0558 C      IF "VALP" IS SMALLER THAN "VAL", THEN "VAL" IS SET EQUAL TO "VALP".
27950 0559 C      THROUGH THIS MEANS, "VAL" IS SET, AFTER 7 ITERATIONS, EQUAL TO THE
28000 0560 C      SMALLEST OF THE EIGENVALUES. SINCE THE VARIABLE "KK" IS SET EQUAL
28050 0561 C      TO "J" ONLY WHEN "VALP" IS LESS THAN "VAL", WHERE "J" IS THE SUB-
28100 0562 C      SCRIPT IN THE "DO" LOOP OF THE EIGENVALUE BEING COMPARED AT THE
28150 0563 C      MOMENT, NOT ONLY IS THE ABSOLUTELY SMALLEST EIGENVALUE IDENTIFIED,
28200 0564 C      BUT SO IS ITS EIGENVECTOR. THEN, IN "DO" LOOP 120, THIS EIG-
28250 0565 C      VECTOR IS PLACED IN THE LEFTMOST COLUMN IN THE FUNDAMENTAL EIGEN-
28300 0566 C      VECTOR MATRIX, WHERE IT WOULD BELONG IN ORDER TO CORRESPOND TO THE
28350 0567 C      POSITION IN THE DIAGONAL EIGENVALUE MATRIX GIVEN TO THE SMALLEST
28400 0568 C      EIGENVALUE. THIS CONTINUES UNTIL ALL EIGENVALUES AND THE EIGEN-
28450 0569 C      VECTORS ASSOCIATED WITH THEM ARE PLACED IN ASCENDING ORDER OF
28500 0570 C      THEIR ABSOLUTE VALUES.

```



```

28550 0571 C
28600 0572 DO 100 I = 1,7
28650 0573 VAL = WR(I) ** 2.0
28700 0574 WR(I) = DSQRT (VAL)
28750 0575 100 CONTINUE
28800 0576 DO 130 I = 1,7
28850 0577 KK = I
28900 0578 VAL = WR(I)
28950 0579 VALP = VAL
29000 0580 DO 110 J = 1,7
29050 0581 IF (WR(I) .LE. WR(J)) GO TO 110
29100 0582 IF (WR(I) .GT. WR(J)) VALP = WR(J)
29150 0583 IF (VALP .LT. VAL) KK = J
29200 0584 IF (VALP .LT. VAL) VAL = VALP
29250 0585 110 CONTINUE
29300 0586 IF (KK .NE. I) WR(KK) = WR(I)
29350 0587 DO 120 J = 1,7
29400 0588 AR(J,I) = Z(J, KK)
29450 0589 Z(J, KK) = Z(J, I)
29500 0590 120 CONTINUE
29550 0591 WR(I) = VAL
29600 0592 130 CONTINUE
29650 0593 C
29700 0594 C HERE, THE B2 MATRIX IS DEFINED. BY A HAPPY STROKE, SINCE THE
29750 0595 C B2 MATRIX IS 1X1, IT IS PRECISELY EQUAL TO THE EIGENVALUE CORRES-
29800 0596 C PONDING TO THE FAST MODE. THE NEGATIVE SIGN IS NECESSARY BECAUSE
29850 0597 C ALL THE EIGENVALUES OF THE COEFFICIENT MATRIX OF THE COUPLED POINT-
29900 0598 C KINETICS EQUATIONS ARE NEGATIVE, BUT THEY WERE ALL CHANGED INTO
29950 0599 C POSITIVE QUANTITIES WHEN THEY WERE REARRANGED INTO ASCENDING ORDER
30000 0600 C OF THEIR ABSOLUTE VALUES.
30050 0601 C
30100 0602 B2 = -WR(7)
30150 0603 C
30200 0604 C HERE, THE FUNDAMENTAL EIGENVECTOR MATRIX IS INVERTED VIA THE
30250 0605 C PORTLIBRARY SUBROUTINE CLINQ. THE INVERSE MATRIX IS THE XR MATRIX,
30300 0606 C WHICH IS 7X7. A FIRST APPROXIMATION FOR THE LM(I) MATRIX IS THEN
30350 0607 C DERIVED BY DIVIDING THE COLUMN MATRIX IN THE 7TH COLUMN, FIRST SIX
30400 0608 C ROWS, BY THE ELEMENT IN THE (7,7)TH POSITION IN THE XR MATRIX.
30450 0609 C
30500 0610 IF (K .GE. 10) GO TO 140
30550 0611 CALL DCLINQ (N, AR, AI, BR, BI, NB, XR, XI)
30600 0612 DO 140 I = 1,6
30650 0613 LM(I) = XR(7,I) / XR(7,7)
30700 0614 140 CONTINUE
30750 0615 150 INT = 0.0
30800 0616 C
30850 0617 C HERE USING THE BEST VALUE FOR LM(I), THE RESIDUAL ERROR
30900 0618 C MATRIX IS COMPUTED.
30950 0619 C
31000 0620 DO 160 I = 1,6
31050 0621 RM(I) = LMB(I) - (B2 * LM(I)) - (LM(I) * LMB(I))
31100 0622 160 CONTINUE
31150 0623 DO 170 I = 1,6
31200 0624 RMT = ((PM(I)) ** 2.0) + RMT
31250 0625 170 CONTINUE
31300 0626 RIE = DSQRT (RMT)
31350 0627 RMT = 0.0

```

```

31400 0628 C
31450 0629 C
31500 0630 C
31550 0631 C
31600 0632 C
31650 0633 C
31700 0634 C
31750 0635 C
31800 0636 C
31850 0637 C
31900 0638 C
31950 0639 C
32000 0640 C
32050 0641 C
32100 0642 C
32150 0643 C
32200 0644 C
32250 0645 C
32300 0646 C
32350 0647 C
32400 0648 C
32450 0649 C
32500 0650 C
32550 0651 C
32600 0652 C
32650 0653 C
32700 0654 C
32750 0655 C
32800 0656 C
32850 0657 C
32900 0658 C
32950 0659 C
33000 0660 C
33050 0661 C
33100 0662 C
33150 0663 C
33200 0664 C
33250 0665 C
33300 0666 C
33350 0667 C
33400 0668 C
33450 0669 C
33500 0670 C
33550 0671 C
33600 0672 C
33650 0673 C
33700 0674 C
33750 0675 C
33800 0676 C
33850 0677 C
33900 0678 C
33950 0679 C
34000 0680 C
34050 0681 C
34100 0682 C
34150 0683 C
34200 0684 C

IF THE EUCLIDEAN NORM OF THE RESIDUAL ERROR MATRIX (WHICH
WHAT "RIE" IS), IS SMALLER THAN A CERTAIN VALUE, THE PROGRAM WILL
PASS CONTROL ON TO STATEMENT 190. THE LM(I) MATRIX AT THIS POINT
WILL BE CONSIDERED TO BE WELL-DEFINED.

IF (RIE .LE. 1.0E-10) GO TO 190

SINCE THE LM(I) MATRIX IS NOT WELL-DEFINED AT THIS POINT, A
CORRECTION MATRIX DI(I) (WHICH IS 6X1) IS COMPUTED AND ADDED
TO THE LM(I) MATRIX. THEN CONTROL IS PASSED TO STATEMENT 150, AND IT
IS THEN DETERMINED IF THIS STATEMENT IS WELL-DEFINED.

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

SINCE THE B1 MATRIX IS DEFINED ONLY IN TERMS OF THE LM(I)
MATRIX AND THE COEFFICIENT MATRIX OF THE COUPLED POINT-KINETICS
EQUATIONS, THIS IS WHERE IT IS COMPUTED. THE KM(I) MATRIX IS NOT
NEEDED HERE, AND WILL BE USED ONLY AT THE END, WHEN IT IS NECESSARY
TO CONVERT THE SOLUTION FOR THE DECOUPLED VARIABLES INTO ACTUAL
REACTOR POWER. ALSO, FOR PURPOSES OF COMPUTING THE KM(I) MATRIX,
AT THIS POINT A RESIDUAL ERROR MATRIX IS COMPUTED. IF ITS EUCLI-
DEAN NORM IS SMALL ENOUGH (WHICH ISN'T LIKELY AT FIRST), THEN ALL
OF THE ELEMENTS OF THE KM(I) MATRIX WILL EQUAL ZERO. OTHERWISE,
SUCCESSIVE ITERATIONS OF A CORRECTION MATRIX WILL HAVE TO LEAD TO
A CORRECT KM(I) MATRIX.

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

HERE, IF THE EUCLIDEAN NORM OF THE RESIDUAL ERROR MATRIX IS
SMALL ENOUGH, CONTROL WILL PASS ON TO STATEMENT 270. IF NOT, THE
SERIES OF CORRECTION MATRICES WILL HAVE TO BE COMPUTED.

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

```

```

34250 0685 C
34300 0686 C      HERE, THE CORRECTION MATRIX DI(I) MATRIX IS COMPUTED.  IT
34350 0687 C      THEN ADDED TO THE MOST RECENT VERSION OF THE KM(I) MATRIX.
34400 0688 C      THE RESIDUAL ERROR MATRIX ASSOCIATED WITH THIS IMPROVED KM(I)
34450 0689 C      MATRIX IS COMPUTED, AND THE CONTROL SHIFTS BACK TO STATEMENT 220.
34500 0690 C
34550 0691      DO 260 I = 1,6
34600 0692      DO 250 J = 1,6
34650 0693          INT      = B1(I,J) * KM(J) + INT
34700 0694 250 CONTINUE
34750 0695          RM(I)    = -(KM(I) * B2) + INT - (B(I) / L)
34800 0696          INT      = 0.0
34850 0697 260 CONTINUE
34900 0698      GO TO 220
34950 0699 270      RIE      = 0.0
35000 0700 C
35050 0701 C      HERE, INITIALIZATIONS OF THE DECOUPLED ANALOGUES FOR THE
35100 0702 C      VARIABLES OF THE DELAYED-NEUTRON RESPONSE AND THE PROMPT RESPONSE
35150 0703 C      TAKE PLACE.
35200 0704 C
35250 0705      CONTINUE
35300 0706      DO 290 I = 1,6
35350 0707      DO 280 J = 1,6
35400 0708          KL(I,J)  = KM(I) * LM(J)
35450 0709 280 CONTINUE
35500 0710          KL(I,I)  = KL(I,I) + 1.0
35550 0711 290 CONTINUE
35600 0712      DO 310 I = 1,6
35650 0713      DO 300 J = 1,6
35700 0714          INT      = KL(I,J) * CIO(J) + INT
35750 0715 300 CONTINUE
35800 0716          SS(I)    = (KM(I) * N10) + INT
35850 0717          INT      = 0.0
35900 0718 310 CONTINUE
35950 0719      DO 320 I = 1,6
36000 0720          INT      = LM(I) * CIO(I) + INT
36050 0721 320 CONTINUE
36100 0722          SS(7)    = N10 + INT
36150 0723          INT      = 0.0
36200 0724      DO 325 I = 1,6
36250 0725          LK      = LM(I) * KM(I) + LK
36300 0726 325 CONTINUE
36350 0727          LK      = LK + 1.0
36400 0728 C
36450 0729 C      HERE, THE SIZE OF THE TIME STEPS FOR THE PROMPT-JUMP APPROX-
36500 0730 C      ATION AND THE RAMP-INPUT MODEL IS DEFINED.  ALSO, IF THE PJ MODEL
36550 0731 C      IS SELECTED, CONTROL IS PASSED ON TO STATEMENT 390.
36600 0732 C
36650 0733 C
36700 0734 C
36750 0735 330      IF (PJ .EQ. 1)  H = 0.001
36800 0736          IF (RI .EQ. 1)  H = 0.001
36850 0737          IF (PJ .EQ. 1)  GO TO 390
36900 0738 C
36950 0739 C
37000 0740 C
37050 0741 C      GALBA SOLVES THE POINT-KINETICS EQUATIONS.  SINCE THEY ARE

```

```

37100 0742 C COUPLED EQUATIONS, ANY NUMERICAL SOLUTION - WHICH AFTER ALL IS WHAT
37150 0743 C GALBA DOES - IS VALID ONLY ON TIME INTERVALS IN WHICH NEITHER N-
37200 0744 C TRON (REACTOR POWER) DENSITY NOR THE DELAYED PRECURSOR DENSITY
37250 0745 C VARIES GREATLY. HOWEVER, DURING THE PROMPT JUMP, BOTH DO VARY
37300 0746 C GREATLY. THE "WAY AROUND THIS" IS TO MAKE THE TIME STEPS ARBITRAR-
37350 0747 C ILY SMALL UNTIL THE PERIOD OF THE PROMPT JUMP IS OVER. THIS
37400 0748 C WHAT GALBA ACTUALLY DOES. THE TIME STEPS START ABOUT BEING EQUAL
37450 0749 C 1 MICROSECOND. BY THE TIME 1 MILLISECOND HAS PASSED, THE TIME STEP
37500 0750 C ALSO EQUAL 1 MILLISECOND, WHERE IT REMAINS FOR THE DURATION OF THE
37550 0751 C RUN. X IS A NUMBER THAT CAUSES THE TIME STEPS TO ENLARGE THEM-
37600 0752 C SELVES AFTER A CERTAIN LENGTH OF TIME HAS ELAPSED. THE ALGEBRAIC
37650 0753 C SUM OF ALL THE TIME STEPS IS 1 MILLISECOND.
37700 0754 C
37750 0755 C 340 IF (X .LE. 10.0) GO TO 341
37800 0756 C IF (X .LE. 13.0) GO TO 342
37850 0757 C IF (X .LE. 16.0) GO TO 343
37900 0758 C IF (X .GE. 17.0) GO TO 344
37950 0759 C 341 H = 0.000001
38000 0760 C X = X + 1.0
38050 0761 C VK = VK + 1
38100 0762 C GO TO 345
38150 0763 C 342 H = 0.00003
38200 0764 C X = X + 1.0
38250 0765 C VK = VK + 1
38300 0766 C GO TO 345
38350 0767 C 343 H = 0.0003
38400 0768 C X = X + 1.0
38450 0769 C VK = VK + 1
38500 0770 C GO TO 345
38550 0771 C 344 H = 0.001
38600 0772 C X = X + 1.0
38650 0773 C VK = VK + 1
38700 0774 C 345 T = T + H
38750 0775 C TH = TH + H
38800 0776 C
38850 0777 C HERE, THE DELAYED-PRECURSOR DENSITY IS COMPUTED. SINCE THE
38900 0778 C PRECURSORS CONSTITUTE THE SLOW MODE, THEIR TIME STEP IS 200 TIMES
38950 0779 C THE SIZE OF THE FAST MODE TIME STEP (WHICH IS 0.001 SECONDS, EXCEPT
39000 0780 C DURING THE PROMPT JUMP). THIS SECTION, OF COURSE, IS FOR THE MA-
39050 0781 C TRIX DECOUPLING ALGORITHM. DURING THE FIRST 0.2 SECONDS, THE DURA-
39100 0782 C TION OF THE PROMPT JUMP, THE DELAYED PRECURSORS ARE COMPUTED USING
39150 0783 C SMALL TIME STEPS. THEY ARE ALSO COMPUTED ONLY WHEN ENOUGH TIME
39200 0784 C PASSED SO THAT THE TOTAL AMOUNT OF TIME IN THEIR TIME STEPS DOES
39250 0785 C NOT EXCEED THE TOTAL AMOUNT OF TIME THAT HAS PASSED IN COMPUTING
39300 0786 C THE PROMPT RESPONSE. THIS IS WHAT TAKES PLACE FROM STATEMENTS 355
39350 0787 C THROUGH 370. AT ALL OTHER TIMES, THE PRECURSORS ARE COMPUTED
39400 0788 C DURING THE MIDDLE OF EACH 0.2 SECTION, TO REFLECT THEIR AVERAGE
39450 0789 C VALUES. THE PRECURSORS THEMSELVES ARE NOT PART OF THE OUTPUT,
39500 0790 C AND ARE COMPUTED ONLY BECAUSE THE POWER OUTPUT IS A FUNCTION OF
39550 0791 C THEM.
39600 0792 C
39650 0793 C IF ((X .LE. 17.0) .AND. (EIG .EQ. 1)) GO TO 350
39700 0794 C IF ((T .GE. 0.0996) .AND. (EIG .EQ. 1) .AND.
39750 0795 C + (T .LE. 0.1004) .AND. (Y .NE. 1.0)) GO TO 350
39800 0796 C GO TO 380
39850 0797 C 350 H2 = H * 200.0
39900 0798 C H20(VK) = H * 200.0

```

```

39950 0799      IF (VK .GE. 20)      VK = 20
40000 0800      IF (Y .GE. 1.1)      GO TO 365
40050 0801      IF ((VK .GT. 6) .AND. (T .LT. 0.0011)) GO TO 375
40100 0802      IF ((VL .GT. 11) .AND. (T .LT. 0.0019)) GO TO 375
40150 0803      IF ((VL .GT. 11) .AND. (T .LT. 0.0079)) GO TO 375
40200 0804      IF ((VL .GT. 12) .AND. (T .LT. 0.0139)) GO TO 375
40250 0805      IF ((VL .GT. 13) .AND. (T .LT. 0.0199)) GO TO 375
40300 0806      IF ((VL .GT. 14) .AND. (T .LT. 0.0799)) GO TO 375
40350 0807      IF ((VL .GT. 15) .AND. (T .LT. 0.1399)) GO TO 375
40400 0808      IF ((VL .GT. 16) .AND. (T .LT. 0.1999)) GO TO 375
40450 0809      DO 360 I = 1,6
40500 0810      DO 355 J = 1,6
40550 0811          INT = (B1(I,J) * DYIN(J,(VL-1))) + INT
40600 0812 355 CONTINUE
40650 0813          DYIN(I,VL)= H2O(VL) * (INT + (KM(I) * (N10 + DN1)
40700 0814      +          * DRO / L)) + DYIN(I,(VL-1))
40750 0815          INT = 0.0
40800 0816          DY(I) = DYIN(I,VL)
40850 0817 360 CONTINUE
40900 0818          VL = VL + 1
40950 0819          IF ((VL .LE. 11) .AND. (VK .GT. 6)) GO TO 350
41000 0820          IF (Y .LE. 1.1) GO TO 375
41050 0821 365 DO 375 I = 1,6
41100 0822          DO 370 J = 1,6
41150 0823              INT = (B1(I,J) * DY(J)) + INT
41200 0824 370 CONTINUE
41250 0825              DY(I) = H2 * (INT + (KM(I) * (N10 + DN1)
41300 0826      +          * DRO / L)) + DY(I)
41350 0827              INT = 0.0
41400 0828 375 CONTINUE
41450 0829 380 IF (EIG .EQ. 1) GO TO 400
41500 0830      C
41550 0831      C      HERE THE POINT-KINETICS EQUATIONS ARE SOLVED, UNLESS THE
41600 0832      C      DECOUPLING OPTION HAS BEEN SELECTED.
41650 0833      C
41700 0834          DN1 = H * (RHO * DN1 / L + DRO * N10 / L + DRO *
41750 0835      +          DN1 / L - BB * DN1 / L + SIG) + DN1
41800 0836 390 SIG = 0.0
41850 0837          SIGPJ = 0.0
41900 0838      C
41950 0839      C      HERE THE CHANGE IN DELAYED-PRECURSOR DENSITY FROM THE STEADY-
42000 0840      C      STATE ARE COMPUTED, UNLESS THE DECOUPLING ALGORITHM HAS BEEN SELEC-
42050 0841      C      TED.
42100 0842      C
42150 0843          DO 400 I = 1,6
42200 0844              DCI(I) = H * (B(I) * DN1 / L - LMB(I) * DCI(I)) + DCI(I)
42250 0845              SIG = SIG + LMB(I) * DCI(I)
42300 0846              SIGPJ = L * LMB(I) * (DCI(I) + C10(I)) + SIGPJ
42350 0847 400 CONTINUE
42400 0848              TTG = TTG + H
42450 0849      C
42500 0850      C      HERE, THE PROMPT-JUMP APPROXIMATION OR THE RAMP-INPUT MODEL
42550 0851      C      ARE SELECTED.
42600 0852      C
42650 0853          IF ((RI .EQ. 1) .AND. (PTO .NE. 0.0)) GO TO 415
42700 0854          IF (PJ .EQ. 2) GO TO 415
42750 0855      C

```

```

42800 0856 C      HERE, REACTOR POWER USING THE PROMPT-JUMP APPROXIMATION IS
42850 0857 C      COMPUTED. ALSO, INTERIM REACTIVITY FROM FEEDBACK IS COMPUTED AND
42900 0858 C      APPLIED TO THE PROMPT JUMP BEFORE IT TAKES PLACE, THUS ALTERING ITS
42950 0859 C      MAGNITUDE. THIS PREVENTS OSCILLATIONS THAT DESTROY THE SOLUTION
43000 0860 C      FROM TAKING PLACE BY THE FACT THAT IF AN UNMODIFIED PROMPT JUMP IS
43050 0861 C      ALLOWED, WHEN ITS FEEDBACK TAKES EFFECT, IT WILL PRODUCE SUCH LARGE
43100 0862 C      REACTIVITY THAT FURTHER SOLUTIONS WILL BE SO INACCURATE THAT OSCIL-
43150 0863 C      LATIONS OF INCREASING MAGNITUDE WILL TAKE PLACE.
43200 0864 C
43250 0865       PWRPJ =      SIGPJ / (BB -(RHO + DRO))
43300 0866       IF (X .GE. 2.9) GO TO 410
43350 0867       I      =      I + 1
43400 0868 405     PWRPJ =      SIGPJ / (BB - (RHO + DRO1))
43450 0869       DN1  =      PWRPJ - N10
43500 0870       DTFGT =      DN1 * 9.4875E-10
43550 0871       DTMGT =      DN1 * 2.0040E-11
43600 0872       DRO2 =      DTFGT * ALPHF + DTMGT * ALPHM
43650 0873       DROT =      DRO1
43700 0874       DRO1 =      DRO + DRO2
43750 0875       DROT =      DROT - DRO1
43800 0876       DROT =      DROT ** 2.0
43850 0877       DROT =      SQRT (DROT)
43900 0878       I      =      I + 1
43950 0879       IF (I .GE. 100) GO TO 410
44000 0880       IF (DROT .GE. 1.0E-09) GO TO 405
44050 0881 410     CONTINUE
44100 0882       DN1  =      PWRPJ - N10
44150 0883 415     PWR2 =      DN1 - PWR1
44200 0884 C
44250 0885 C      HERE, INTERIM REACTIVITY THAT TAKES PLACE DURING A GIVEN
44300 0886 C      PERIOD OF 0.005 SECONDS IS COMPUTED. AFTER THIS INTERVAL, A NEW
44350 0887 C      REACTIVITY BASED ON FEEDBACK IS COMPUTED, AND THE OLD INTERIM
44400 0888 C      FIGURE IS NO LONGER NEEDED.
44450 0889 C
44500 0890       IF (DTF(K) .EQ. 0.0) DTF(K) = DTF(K-1).
44550 0891       IF (DTM(K) .EQ. 0.0) DTM(K) = DTM(K-1)
44600 0892       DTFGT =      DTF(K) - DTF(K-1)
44650 0893       DTMGT =      DTM(K) - DTM(K-1)
44700 0894       IF (X .LE. 20.0) DTFGT = DN1 * 9.4875E-10
44750 0895       IF (X .LE. 20.0) DTMGT = DN1 * 2.0040E-11
44800 0896       IF ((CS .EQ. 2) .AND. (KF .EQ. 2))
44850 0897 +         DRO = (ALPHF * (DTF(K-1) + (ADJF * DTFGT)))
44900 0898 +         + (ALPHM * (DTM(K-1) + (ADJM * DTMGT)))
44950 0899 +         + DROIN + (RIR * TTG)
45000 0900       IF ((CS .EQ. 1) .AND. (KF .EQ. 2))
45050 0901 +         DRO = (ALPHF * (DTF(K-1) + (ADJF * DTFGT)))
45100 0902 +         + (ALPHM * (DTM(K-1) + (ADJM * DTMGT)))
45150 0903 +         + DROIN + (RIR * TTG) + (H * (AA *
45200 0904 +         ((DTMGT * ADJM) + (H * ADJM * DTMGT /
45250 0905 +         TAU)))) + DRC
45300 0906       IF (TTG .GE. TTOT) TTG = TTOT
45350 0907
45400 0908
45450 0909 C
45500 0910 C      HERE, THE POWER RESPONSE BASED ON THE DECOUPLING ALGORITHM IS
45550 0911 C      IS COMPUTED. ALSO, IN LINE 480, CURRENT TOTAL POWER IS COMPUTED.
45600 0912 C      THIS WILL BE USED TO ADJUST STEAM GENERATOR POWER IN THE SUBROUTINE

```

```

45650 0913 C DMTN.
45700 0914 C
45750 0915 IF (EIG .EQ. 2) GO TO 440
45800 0916 DY(7) = H * ((B2 * DY(7)) + ((N10 + DN1)
45850 0917 + * DRO / L)) + DY(7)
45900 0918 H2 = H * 200.0
45950 0919 DO 420 I = 1,6
46000 0920 INT = LM(I) * DY(I) + INT
46050 0921 420 CONTINUE
46100 0922 DN1 = (DY(7) * LK) - INT
46150 0923 INT = 0.0
46200 0924 430 IF (TH .LT. 0.0042) GO TO 330
46250 0925 440 IF (NK .EQ. 1) TTG = TTG + 0.2
46300 0926 IF (NK .EQ. 1) T = 0.2
46350 0927 450 IF (T .GE. 0.1995) GO TO 460
46400 0928 IF (T .LT. 0.1995) GO TO 470
46450 0929 460 T = 0.0
46500 0930 470 TH = 0.0
46550 0931 PWR1 = DN1
46600 0932 PTO = PTG
46650 0933 480 RP(K) = N10 + DN1
46700 0934 RETURN
46750 0935 END

```

## THERMAL-HYDRAULICS EQUATIONS

THIS IS THE SUBROUTINE THAT MODELS THE THERMAL-HYDRAULICS EQUATIONS. AS INPUT, IT RECEIVES REACTOR POWER INFORMATION FROM THE REACTOR KINETICS SUBROUTINE (GALBA) AND REACTOR INLET TEMPERATURE INFORMATION FROM THE STEAM GENERATOR SUBROUTINE (DMTN). AS OUTPUT, IT COMPUTES MODERATOR TEMPERATURE CHANGES AND REACTOR OUTLET TEMPERATURE CHANGES AND FUEL TEMPERATURE CHANGES. ALL ARE DISPLAYED IN THE OUTPUT.

AS IN THE OTHER SUBROUTINES, OTHO SOLVES TWO COUPLED DIFFERENTIAL EQUATIONS. HERE, THE VARIABLES SOLVED FOR ARE MODERATOR TEMPERATURE CHANGE AND THE SKIN TEMPERATURE CHANGE ON THE FUEL ASSEMBLIES. FROM THESE ARE COMPUTED REACTOR OUTLET TEMPERATURE CHANGE AND FUEL TEMPERATURE CHANGE. TIME STEPS IN OTHO ARE 0.005 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)

IN ADDITION TO THOSE USED IN THE COMMON BLOCK, THE FOLLOWING VARIABLES ARE USED IN OTHO:

AR IS THE HEAT TRANSFER AREA.  
 CPAV IS THE HEAT CAPACITY OF THE REACTOR COOLANT  
 CPF IS THE HEAT CAPACITY OF THE FUEL.  
 CPHS IS THE HEAT CAPACITY OF THE REACTOR COOLANT AT THE REACTOR OUTLET.  
 CPLS IS THE HEAT CAPACITY OF THE REACTOR COOLANT AT THE REACTOR INLET.  
 D IS THE DIAMETER OF REACTOR COOLANT FLOW CHANNEL.  
 DTC IS CHANGE IN TEMPERATURE OF THE SURFACE OF THE FUEL ASSEMBLIES FROM THE STEADY-STATE TEMPERATURES.  
 DTMO IS CHANGE IN REACTOR MODERATOR (COOLANT) TEMPERATURE FROM STEADY-STATE.  
 FRAC IS A FACTOR WHICH ADJUSTS FOR THE FACT THAT CP DOES NOT VARY LINEARLY WITH TEMPERATURE CHANGE. WITH FRAC, ONE CAN OBTAIN TRUE AVERAGE CP AND MODERATOR TEMPERATURE BY TAKING THE LINEAR AVERAGE OF OF THE EXTREMES AT INLET AND OUTLET CONDITIONS, AND MULTIPLYING THE DIFFERENCE BY FRAC, AND THEN ADDING TO THE LOW-EXTREME CONDITION.  
 HP IS THE HEAT TRANSFER COEFFICIENT. IT IS DERIVED VIA THE DITTIUS-BOELTER CORRELATION.



```

05800 0058 C KP IS THE THERMAL CONDUCTIVITY OF THE REACTOR COOLANT.
05900 0059 C MF IS THE MASS OF THE FUEL.
06000 0060 C MM IS THE MASS OF REACTOR COOLANT.
06100 0061 C MUP IS THE KINEMATIC VISCOSITY OF THE REACTOR COOLANT.
06200 0062 C NU IS THE NUSSELT NUMBER OF THE REACTOR COOLANT.
06300 0063 C NUP IS THE DYNAMIC VISCOSITY OF THE REACTOR COOLANT.
06400 0064 C PRP IS THE PRANDTL NUMBER OF THE REACTOR COOLANT.
06500 0065 C REP IS THE REYNOLDS NUMBER OF THE REACTOR COOLANT.
06600 0066 C ROP IS THE DENSITY OF THE REACTOR COOLANT.
06700 0067 C T IS THE AMOUNT OF TIME EXPIRED DURING THE CURRENT
06800 0068 C CYCLE. IS RESET TO ZERO AT THE BEGINNING OF
06900 0069 C EACH OTHO CYCLE.
07000 0070 C T11S IS INITIAL REACTOR INLET TEMPERATURE.
07100 0071 C T01S IS INITIAL REACTOR OUTLET TEMPERATURE.
07200 0072 C TPAV IS INITIAL AVERAGE COOLANT TEMPERATURE. IS USED
07300 0073 C TO COMPUTE HEAT TRANSFER VARIABLES SUCH AS
07400 0074 C DENSITY.
07500 0075 C TPAVS IS THE SAME AS TPAV.
07600 0076 C VMOD IS VELOCITY (= MASS FLOW.
07700 0077 C X IS AN INTEGER, THE VALUE OF WHICH DETERMINES THE
07800 0078 C SIZE OF THE TIME STEP DURING THE FIRST CYCLE.
07900 0079
08000 0080 REAL AR, CPAV, CPF, CPHS, CPLS, D, DTC, DTMO, DTOO, FRAC,
+ HP, KP, MF, MM, MUP, NU, NUP, PRP, REP, ROP, T,
+ T11S, T01S, TPAV, TPAVS, VMOD
08200 0082 + INTEGER X
08300 0083
08400 0084 C INITIALIZATIONS TAKE PLACE HERE, DURING THE FIRST 0.2 SECONDS,
08500 0085 C ONLY.
08600 0086
08700 0087
08800 0088 IF (Y .GE. 3.0) GO TO 10
08900 0089 AR = 5945.0
09000 0090 CPF = 0.18477 * (TF1 + DTF(K)) + 74.4
09100 0091 D = 0.012
09200 0092 DTC = 0.0
09300 0093 DTMO = 0.0
09400 0094 DTOO = 0.0
09500 0095 T11S = (565.0 * 5.0 / 9.0) - (28.45 * 5.0/9.0) * N10/3.0E09
09600 0096 T01S = (565.0 * 5.0 / 9.0) + (28.45 * 5.0/9.0) * N10/3.0E09
09700 0097 CPHS = 4992.4097749+ 2.49340775E-04 * EXP(0.04825458 *T01S)
09800 0098 CPLS = 4992.4097749+ 2.49340775E-04 * EXP(0.04825458 *T11S)
09900 0099 FRAC = 0.5 * 5932.2566914 / CPPAV
10000 0100 MF = 1.91E05
10100 0101 T = 0.0
10200 0102 TPAVS = FRAC * (T01S - T11S) + T11S
10300 0103 TPAV = TPAVS
10400 0104 X = 1
10500 0105
10600 0106 10 CONTINUE
10700 0107
10800 0108 C HERE THE HEAT TRANSFER CORRELATIONS ARE DERIVED, IN TERMS OF
10900 0109 C THE DITTIUS-BOELTER CORRELATION.
11000 0110
11100 0111 CPAV = 4992.4097749+ 2.49340775E-04 * EXP(0.04825458 *TPAV)
11200 0112 ROP = 881.6309 - 2.86514041 * EXP(0.013303415*TPAV)
11300 0113 VMOD = MDOTP / (5.26 * ROP)
11400 0114 KP = 0.7127683- 3.02500000E-03 * EXP(0.01321074 *TPAV)

```

```

11500 0115      MUP = -3.53438229E-07 * TPAV      + 1.9978561E-04
11600 0116      NUP =      MUP / ROP
11700 0117      REP =      VMOD * D / NUP
11800 0118      PRP =      CPAV * NUP * ROP / KP
11900 0119      NU  =      0.0250 * (REP ** 0.8) * (PRP ** 0.6)
12000 0120      HP  =      NU * KP / D
12100 0121      MM  =      13275.0
12200 0122
12300 0123      C      AS IN GALBA, ARBITRARILY SMALL TIME STEPS ARE SELECTED DURING
12400 0124      C      THE FIRST CYCLE.
12500 0125
12600 0126      20     IF (X .LE. 10) GO TO 21
12700 0127          IF (X .LE. 13) GO TO 22
12800 0128          IF (X .LE. 16) GO TO 23
12900 0129          IF (X .LE. 20) GO TO 24
13000 0130          IF (X .GE. 21) GO TO 25
13100 0131      21     H      = 0.000001
13200 0132          X      = X + 1
13300 0133          GO TO 30
13400 0134      22     H      = 0.00003
13500 0135          X      = X + 1
13600 0136          GO TO 30
13700 0137      23     H      = 0.0003
13800 0138          X      = X + 1
13900 0139          GO TO 30
14000 0140      24     H      = 0.001
14100 0141          X      = X + 1
14200 0142          GO TO 30
14300 0143      25     H      = 0.005
14400 0144
14500 0145      30     CONTINUE
14600 0146
14700 0147      C      HERE THE THERMAL-HYDRAULICS EQUATIONS PROPER ARE SOLVED.
14800 0148      C      THE NOTEWORTHY FEATURE IS THAT THEY ARE SOLVED FOR CLADDING
14900 0149      C      SURFACE TEMPERATURE CHANGE (DTC) AND MODERATOR TEMPERATURE
15000 0150      C      CHANGE, INSTEAD OF FUEL AND MODERATOR TEMPERATURE CHANGES.
15100 0151      C      LATER, SINCE IT IS ASSUMED THAT AVERAGE FUEL TEMPERATURE
15200 0152      C      CHANGE IS PROPORTIONAL TO CLADDING TEMPERATURE CHANGE,
15300 0153      C      A FUEL TEMPERATURE CHANGE IS COMPUTED AS A FUNCTION OF
15400 0154      C      CLADDING TEMPERATURE CHANGE.
15500 0155
15600 0156          CPF = 0.18477 * (TF1 + DTF(K)) + 74.4
15700 0157          DTC = H / (CPF * MF) * (DN1 - (HP * AR * (DTC -
15800 0158      +          DTMO))) + DTC
15900 0159          IF (K .GE. 56) GO TO 40
16000 0160          DTMO = H / (CPPAV * MM) * (HP * AR * (DTC - DTMO) -
16100 0161      +          (CPPAV * MDOTP * DTOO)) + DTMO
16200 0162          GO TO 50
16300 0163      40     DTMO = H / (CPPAV * MM) * ((HP * AR * (DTC - DTMO)) -
16400 0164      +          (CPPAV * MDOTP * (DTMO - DT1(K-29)))) +
16500 0165      +          DTMO
16600 0166      50     T      = T + H
16700 0167
16800 0168      C      HERE, REACTOR OUTLET TEMPERATURE CHANGE IS COMPUTED.
16900 0169
17000 0170          IF (K .LE. 55) DTOO = 2.0 * DTMO
17100 0171          IF (K .GT. 55) DTOO = 2.0 * DTMO - DT1(K-29)

```

```
17200 0172      IF ((T .LT. 0.2) .AND. (NK .EQ. 1)) GO TO 20
17300 0173      TPAV  = TPAVS + DTMO
17400 0174      DTF(K) = 10.0 * DTC
17500 0175      DTM(K) = DTMO
17600 0176      DTO(K) = DT00
17700 0177      H     = 0.2
17800 0178      T     = 0.0
17900 0179      RETURN
18000 0180      END
```

```

00100 0001 C
00200 0002 C
00300 0003 C
00400 0004 C
00500 0005 C
00600 0006 C
00700 0007 C
00800 0008 C
00900 0009 C
01000 0010 C
01100 0011 C
01200 0012 C
01300 0013 C
01400 0014 C
01500 0015 C
01600 0016 C
01700 0017 C
01800 0018 C
01900 0019 C
02000 0020 C
02100 0021 C
02200 0022 C
02300 0023 C
02400 0024 C
02500 0025 C
02600 0026 C
02700 0027 C
02800 0028 C
02900 0029 C
03000 0030 C
03100 0031 C
03200 0032 C
03300 0033 C
03400 0034 C
03500 0035 C
03600 0036 C
03700 0037 C
03800 0038 C
03900 0039 C
04000 0040 C
04100 0041 C
04200 0042 C
04300 0043 C
04400 0044 C
04500 0045 C
04600 0046 C
04700 0047 C
04800 0048 C
04900 0049 C
05000 0050 C
05100 0051 C
05200 0052 C
05300 0053 C
05400 0054 C
05500 0055 C
05600 0056 C
05700 0057 C

```

SIMPLE STEAM GENERATOR MODEL

This is the subroutine that depicts the simple steam generator model. As input, it receives moderator temperature change (from the steady-state) data from OTHO (the subroutine dealing with reactor thermal hydraulics), or instructions on what the outlet valve position is to be fixed at, from the main program NERO. It computes reactor inlet temperature changes and steam generator power as output. Reactor inlet temperature changes are utilized in OTHO to affect overall reactor moderator temperature, and both are displayed as output in the main program, either in graphics or in a table.

As in the other subroutines, DMTN solves two coupled differential equations. Here, the variables solved for are steam generator temperature and steam temperature on the secondary side of the steam generator. By "steam generator" is meant all of the liquid in the primary side of the steam generator plus all of the material that makes up the steam generator itself. The finite-difference method is used to solve these differential equations. In the case of the steam temperature on the secondary side, it is assumed that the control system will, at the end of each time step, automatically adjust the flow of the coolant so as to restore the secondary side back to equilibrium conditions. Thus, the steam temperature from the previous iteration is not used as input during the current iteration. Time steps in DMTN are 0.2 seconds.

From steam temperature changes are computed enthalpy changes on the secondary side. From this, power changes are computed, and from this come secondary side flow changes and total steam generator output. Reactor inlet temperature changes are computed as a function of primary side temperature changes and reactor outlet temperature changes (which is output into DMTN for OTHO, as 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, MDOTC0,
+ MDOTSF, MMSG, PS, PSS, PWR, PWRCH, T11, T01, TOF, TS, TD

```

IN ADDITION TO THE VARIABLES CARRIED THROUGH IN THE COMMON BLOCK, THE FOLLOWING VARIABLES ARE USED IN DMTN:

05800	0058	C	AR	IS THE HEAT TRANSFER AREA.
05900	0059	C	CPTM	IS A COMBINATION TERM. IT EQUALS THE TOTAL HEAT CAPACITY TIMES THE TOTAL MASS OF WATER ON THE SECONDARY SIDE.
06000	0060	C		
06100	0061	C		
06200	0062	C	DENTH	IS THE CHANGE IN ENERGY ON THE SECONDARY SIDE CAUSED BY A CHANGE IN POWER TRANSFER RATE FROM THE PRIMARY SIDE.
06300	0063	C		
06400	0064	C		
06500	0065	C	DENTC	IS THE CHANGE IN ENERGY ON THE SECONDARY SIDE FROM ONE TIME STEP TO THE NEXT CAUSED BY THE "EXTRA POWER BOOST". THIS BOOST IS INTRODUCED IN ORDER THAT THE CHANGE IN STEAM GENERATOR POWER BE TEMPORARILY GREATER THAN REACTOR POWER CHANGE. THIS IS NECESSARY BECAUSE IMMEDIATELY AFTER THE PROMPT-JUMP, REACTOR POWER CHANGE IS GREATER THAN STEAM GENERATOR POWER CHANGE, AND THE ESTABLISHMENT OF A LONG-TERM ENERGY BALANCE REQUIRES THAT THE STEAM GENERATOR HAVE THIS "EXTRA POWER BOOST" IN ORDER TO "CATCH UP".
06600	0066	C		
06700	0067	C		
06800	0068	C		
06900	0069	C		
07000	0070	C		
07100	0071	C		
07200	0072	C		
07300	0073	C		
07400	0074	C		
07500	0075	C	DHFG	IS TOTAL CHANGE IN ENTHALPY FOR A KILOGRAM OF WATER FROM THE INLET OF THE STEAM GENERATOR (235 C) TO OUTLET (311.1 C, INCLUDING 33.33 DEGREES OF SUPERHEAT).
07600	0076	C		
07700	0077	C		
07800	0078	C		
07900	0079	C	DTSD	IS THE CHANGE OF TEMPERATURE ON THE SECONDARY SIDE. USED ONLY DURING THE FIRST 0.2 SECONDS.
08000	0080	C		
08100	0081	C	DTSGD	IS THE CHANGE OF TEMPERATURE ON THE PRIMARY SIDE. USED ONLY DURING THE FIRST 0.2 SECONDS.
08200	0082	C		
08300	0083	C	HS	IS THE HEAT TRANSFER COEFFICIENT. IT IS A FUNCTION OF MDOTS ** 0.806.
08400	0084	C		
08500	0085	C	LMTD	IS THE LOGARITHMIC MEAN TEMPERATURE DIFFERENCE BETWEEN THE PRIMARY SIDE AND THE SECONDARY SIDE.
08600	0086	C		
08700	0087	C	MDOTS	IS THE MASS FLOW RATE ON THE SECONDARY SIDE.
08800	0088	C	MDOTCA	IS THE CHANGE IN MASS FLOW RATE ON THE SECONDARY SIDE.
08900	0089	C		
09000	0090	C	MDOTCO	IS THE VALUE FOR MDOTCA DURING THE PREVIOUS ITERATION.
09100	0091	C		
09200	0092	C	MMSG	IS THE MASS OF WATER IN THE PRIMARY SIDE OF THE STEAM GENERATOR.
09300	0093	C		
09400	0094	C	PWR	IS CURRENT POWER OUTPUT OF THE STEAM GENERATOR, INCLUDING THE "EXTRA POWER BOOST".
09500	0095	C		
09600	0096	C	PWRCH	IS THE POWER ADDITION TO STEAM GENERATOR OUTPUT DUE TO THE "EXTRA POWER BOOST".
09700	0097	C		
09800	0098	C	TD	IS THE DIFFERENCE BETWEEN INLET AND OUTLET TEMPERATURES.
09900	0099	C		
10000	0100	C	TOF	IS THE OUTLET TEMPERATURE THAT THE REACTOR WOULD HAVE UNDER STEADY-STATE CONDITIONS AT A GIVEN POWER LEVEL.
10100	0101	C		
10200	0102	C		
10300	0103	C	TS	IS THE DIFFERENCE BETWEEN THE OUTLET TEMPERATURE THAT THE REACTOR WOULD HAVE UNDER STEADY-STATE CONDITIONS AT A GIVEN POWER LEVEL AND THE ACTUAL TEMPERATURE.
10400	0104	C		
10500	0105	C		
10600	0106	C		
10700	0107	C		
10800	0108	C		
10900	0109	C		
11000	0110	C		
11100	0111	C		
11200	0112	C		
11300	0113	C		
11400	0114	C		

IF (K .NE. 2) GO TO 10  
THIS STATEMENT WILL JUMP THE PROGRAM OVER THE INITIALIZATIONS  
AT ALL TIMES EXCEPT DURING THE FIRST 0.2 SECONDS.

HERE ARE THE INITIALIZATIONS.

```

11500 0115
11600 0116
11700 0117
11800 0118
11900 0119
12000 0120
12100 0121
12200 0122
12300 0123
12400 0124
12500 0125
12600 0126
12700 0127
12800 0128
12900 0129
13000 0130
13100 0131
13200 0132
13300 0133
13400 0134
13500 0135
13600 0136
13700 0137
13800 0138
13900 0139
14000 0140
14100 0141
14200 0142
14300 0143
14400 0144
14500 0145
14600 0146
14700 0147
14800 0148
14900 0149
15000 0150
15100 0151
15200 0152
15300 0153
15400 0154
15500 0155
15600 0156
15700 0157
15800 0158
15900 0159
16000 0160
16100 0161
16200 0162
16300 0163
16400 0164
16500 0165
16600 0166
16700 0167
16800 0168
16900 0169
17000 0170
17100 0171

```

```

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
T11 = -(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 - T11
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))))

```

```

17200 0172
17300 0173
17400 0174
17500 0175
17600 0176
17700 0177
17800 0178
17900 0179
18000 0180
18100 0181
18200 0182
18300 0183
18400 0184
18500 0185
18600 0186
18700 0187
18800 0188
18900 0189
19000 0190
19100 0191
19200 0192
19300 0193
19400 0194
19500 0195
19600 0196
19700 0197
19800 0198
19900 0199
20000 0200
20100 0201
20200 0202
20300 0203
20400 0204
20500 0205
20600 0206
20700 0207
20800 0208
20900 0209
21000 0210
21100 0211
21200 0212
21300 0213
21400 0214
21500 0215
21600 0216
21700 0217
21800 0218
21900 0219
22000 0220
22100 0221
22200 0222
22300 0223
22400 0224
22500 0225
22600 0226
22700 0227
22800 0228

+ 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 PERMANENTLY 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

```

```

22900 0229          DTS (K) = DTSD
23000 0230          GO TO 40
23100 0231
23200 0232
23300 0233
23400 0234          C      HERE, THE COUPLED EQUATIONS ARE SOLVED AND THE OUTPUT QUANTI-
23500 0235          C      TIES SUCH AS POWER AND REACTOR INLET TEMPERATURE ARE COMPUTED. THE
23600 0236          C      NUMBER 50600.0 ARISES FROM THE FACT THAT EVERY DEGREE TEMPERATURE
23700 0237          C      CHANGE ON THE SECONDARY SIDE REQUIRES 50600.0 J OF ENERGY PER KG.
23800 0238
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)
24100 0241          DTS (K) = (H / CPTM) * ((HS * AR * (DTSG(K-1) + LMTD))
+      +      -      (MDOTS + MDOTCA) * DHFG)
24300 0243          DENTH = 50600.0 * DTS(K)
24400 0244          DENTC = 50600.0 * H * (MDOTCA - MDOTCO) * DHFG / CPTM
24500 0245          TDENTH = DENTC + TDENTH
24600 0246          PWRCH = MDOTCA * (TDENTH + DHFG)
24700 0247          PWR = MDOTS * (DENTH + DHFG) + PWRCH
24800 0248          IF (ZZ .EQ. 1) GO TO 40
24900 0249          PS = 5509154.5 + (77756.49 * DTS(K-1)) + (538137557.7/
25000 0250          +      (772.747 - (1.006 * DTS(K-1))))
25100 0251          DPS = PS - PSS
25200 0252          DVO = DPS / PSS
25300 0253          VO = VO + DVO
25400 0254          MDOTS = VO * MDOTSF
25500 0255          MDOTS = (PWR - PWRCH) / DHFG
25600 0256          40      DTI (k) = 2.0 * DTSG(K) - DTO(K-25)
25700 0257          STMGEN = MDOTS * (DENTH + DHFG) + PWRCH
25800 0258          MDOTCO = MDOTCA
25900 0259          MDOTCA = 49.9035 * TS
26000 0260          IF (ZZ .EQ. 1) MDOTCA = 0.0
26100 0261          50      RETURN
26200 0262          END

```





```

02900 0058 + ZERO /'0'/, TEN/'0', '0'/, TWNT/'2', '0'/.
02950 0059 + THRT /'3', '0'/, FORT/'4', '0'/, FFTY/'5', '0'/.
03000 0060 + SIXT /'6', '0'/, MW/'M', 'W'/, SP/' '/,
03050 0061 + DEG /'D', 'E', 'G', 'A', 'C', 'T', 'O', 'R', 'I', 'P', 'O',
03100 0062 + DNIG /'R', 'E', 'A', 'C', 'T', 'O', 'R', 'I', 'P', 'O',
03150 0063 +
03200 0064 + STM /'S', 'T', 'M', 'I', 'G', 'E', 'N', 'I',
03250 0065 + /'O', 'U', 'T', 'P', 'U', 'T', 'I', 'P', 'O',
03300 0066 +
03350 0067 + /'W', 'E', 'R', ':',
03400 0068
03450 0069 T11 = (565.0*5.0/9.0) - (28.45*5.0/9.0) * (N10 / 3.0E09)
03500 0070 TO1 = (565.0*5.0/9.0) + (28.45*5.0/9.0) * (N10 / 3.0E09)
03550 0071 REP = 2
03600 0072 ER = 1.0
03650 0073 MDT = 0.0
03700 0074 LXP = 0.0
03750 0075 HXP = 3.0E09
03800 0076 LXF = 300.0
03850 0077 HXF = 720.0
03900 0078 LXT = -25.0
03950 0079 HXT = 25.0
04000 0080 INT1 = LND / 30.0
04050 0081 INT2 = LND / 15.0
04100 0082 INT3 = LND / 10.0
04150 0083 INT4 = LND / 7.5
04200 0084 INT5 = LND / 6.0
04250 0085 INT6 = LND / 5.0
04300 0086 RSHA = LND / 5.0
04350 0087 DO 10 I = 1, LND
04400 0088 TMPF(I) = TF1 + DTF(I)
04450 0089 10 CONTINUE
04500 0090
04550 0091 20 WRITE (6, 30)
04600 0092 30 FORMAT (X, ' YOU HAVE SELECTED GRAPHICS. SINCE THIS
04650 0093 + IS SO, THERE ARE A FEW')
04700 0094 WRITE (6, 40)
04750 0095 40 FORMAT (X, ' THINGS YOU MUST KNOW. FIRST, THREE
04800 0096 + DIFFERENT PLOTTING WINDOWS WILL')
04850 0097 WRITE (6, 50)
04900 0098 50 FORMAT (X, ' BE SUCCESSIVELY DISPLAYED. TO TRANSFER
04950 0099 + FROM ONE WINDOW TO ANOTHER,')
05000 0100 WRITE (6, 60)
05050 0101 60 FORMAT (X, ' PRESS <RET>. ANY TIME THE "ACTION"
05100 0102 + STOPS, IT CAN BE RESUMED BY')
05150 0103 WRITE (6, 70)
05200 0104 70 FORMAT (X, ' PRESSING <RET>. YOU CAN ALSO, AFTER THE
05250 0105 + INITIAL DISPLAY, EXPAND THE')
05300 0106 WRITE (6, 80)
05350 0107 80 FORMAT (X, ' DISPLAY IN ORDER TO SHOW CONTRASTS
05400 0108 + BETTER. SECOND, IN ORDER TO')
05450 0109 WRITE (6, 90)
05500 0110 90 FORMAT (X, ' CLEAR THE GRAPHICS DISPLAYS AFTER THE
05550 0111 + PROGRAM STOPS EXECUTING AND')
05600 0112 WRITE (6, 100)
05650 0113 100 FORMAT (X, ' TO RESTORE SCROLLING CAPABILITY, WHEN
05700 0114 + THE PROGRAM RETURNS TO THE')

```

```

05750 0115      WRITE (6, 110)
05800 0116 110  FORMAT (X, '          COMMAND MODE (AFTER THE "FORTRAN STOP"
05850 0117      + MESSAGE APPEARS ON THE')
05900 0118      WRITE (6, 120)
05950 0119 120  FORMAT (X, '          SCREEN), WRITE IN THE STATEMENT "@
06000 0120      + GRAPHICS: CLEAR" (WITHOUT')
06050 0121      WRITE (6, 130)
06100 0122 130  FORMAT (X, '          THE QUOTATION MARKS). YOU MUST BE ON A
06150 0123      + TEKTRONICS TERMINAL.')
06200 0124      WRITE (6, 140)
06250 0125 140  FORMAT ('0', '          THIRD: DO YOU WANT 4662 COPY
06300 0126      + CAPABILITY OR NOT? IF SO, TYPE IN 1;')
06350 0127      WRITE (6, 150)
06400 0128 150  FORMAT (X, '          IF NOT, 2')
06450 0129      READ 160, PL
06500 0130 160  FORMAT (I1)
06550 0131
06600 0132      IF ((PL .EQ. 1) .OR. (PL .EQ. 2)) GO TO 190
06650 0133
06700 0134      WRITE (6, 170)
06750 0135 170  FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
06800 0136      + BE UTILIZED. YOU WILL HAVE')
06850 0137      WRITE (6, 180)
06900 0138 180  FORMAT (X, '          TO TRY AGAIN.')
06950 0139
07000 0140      GO TO 20
07050 0141
07100 0142 190  CONTINUE
07150 0143 200  WRITE (6, 210)
07200 0144 210  FORMAT ('0', '          AS CURRENTLY SET UP, REACTOR INLET
07250 0145      + TEMPERATURE IS PLOTTED AS OF THE')
07300 0146      WRITE (6, 220)
07350 0147 220  FORMAT (X, '          MOMENT IT IS CREATED IN THE STEAM
07400 0148      + GENERATOR. HOWEVER, A GIVEN INLET')
07450 0149      WRITE (6, 230)
07500 0150 230  FORMAT (X, '          TEMPERATURE CHANGE DOES NOT AFFECT
07550 0151      + REACTOR TEMPERATURES UNTIL AFTER SIX')
07600 0152      WRITE (6, 240)
07650 0153 240  FORMAT (X, '          SECONDS AFTER ITS CREATION. IN ORDER
07700 0154      + TO DEPICT REACTOR TEMPERATURE')
07750 0155      WRITE (6, 250)
07800 0156 250  FORMAT (X, '          CHANGES AT THE SAME TIME THAT THE INLET
07850 0157      + TEMPERATURE CHANGES CAUSING')
07900 0158      WRITE (6, 260)
07950 0159 260  FORMAT (X, '          THEM ENTER THE REACTOR, TYPE IN 1.
08000 0160      + OTHERWISE THE INLET TEMPERATURE')
08050 0161      WRITE (6, 270)
08100 0162 270  FORMAT (X, '          CHANGES WILL BE PLOTTED AS OF THE MOMENT
08150 0163      + OF THEIR CREATION, WHICH')
08200 0164      WRITE (6, 280)
08250 0165 280  FORMAT (X, '          IS 6 SECONDS BEFORE THEY ARRIVE AT THE
08300 0166      + INLET. FOR THIS OPTION,')
08350 0167      WRITE (6, 290)
08400 0168 290  FORMAT (X, '          TYPE IN 2.')
08450 0169
08500 0170      READ 300, DE
08550 0171 300  FORMAT (I1)

```

```

08600 0172
08650 0173
08700 0174
08750 0175
08800 0176
08850 0177
08900 0178
08950 0179
09000 0180
09050 0181
09100 0182
09150 0183
09200 0184
09250 0185
09300 0186
09350 0187
09400 0188
09450 0189
09500 0190
09550 0191
09600 0192
09650 0193
09700 0194
09750 0195
09800 0196
09850 0197
09900 0198
09950 0199
10000 0200
10050 0201
10100 0202
10150 0203
10200 0204
10250 0205
10300 0206
10350 0207
10400 0208
10450 0209
10500 0210
10550 0211
10600 0212
10650 0213
10700 0214
10750 0215
10800 0216
10850 0217
10900 0218
10950 0219
11000 0220
11050 0221
11100 0222
11150 0223
11200 0224
11250 0225
11300 0226
11350 0227
11400 0228

      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

```

11450	0229	400	CALL RNUMBR (LOWP,	1,	6)		
11500	0230		CALL TEXT (	1,	SP)		
11550	0231		CALL TEXT (	2,	MW)		
11600	0232	410	CALL MOVE (	18.0,	50.0)		
11650	0233		CALL DASHPT	(9)			
11700	0234		CALL DRAW (	120.0,	50.0)		
11750	0235		CALL MOVE (	0.0,	50.0)		
11800	0236		CALL DASHPT	(0)			
11850	0237		CALL DRAW (	0.0,	25.0)		
11900	0238		CALL TXICUR	(2)			
11950	0239		CALL MOVE (	20.0,	25.0)		
12000	0240		CALL INUMBR (INT1,	2)			
12050	0241		CALL MOVE (	40.0,	25.0)		
12100	0242		CALL INUMBR (INT2,	3)			
12150	0243		CALL MOVE (	60.0,	25.0)		
12200	0244		CALL INUMBR (INT3,	3)			
12250	0245		CALL MOVE (	80.0,	25.0)		
12300	0246		CALL INUMBR (INT4,	3)			
12350	0247		CALL MOVE (	100.0,	25.0)		
12400	0248		CALL INUMBR (INT5,	3)			
12450	0249		CALL TXICUR	(3)			
12500	0250		CALL MOVE (	120.0,	25.0)		
12550	0251		CALL INUMBR (INT6,	3)			
12600	0252		CALL TXICUR	(4)			
12650	0253		CALL MOVE (	0.0,	20.0)		
12700	0254		CALL TEXT (	22,	DN1G)		
12750	0255		CALL MOVE (	45.0,	20.0)		
12800	0256		CALL DRAW (	120.0,	20.0)		
12850	0257		CALL MOVE (	0.0,	16.0)		
12900	0258		CALL TEXT (	22,	STM)		
12950	0259		CALL MOVE (	45.0,	16.0)		
13000	0260		CALL DASHPT	(3)			
13050	0261		CALL DRAW (	120.0,	16.0)		
13100	0262		CALL VWPORT (	0.0,	120.0,	25.0,	100.0)
13150	0263		CALL WINDOW (	0.0,	RSHA,	LXP,	HXP)
13200	0264		CALL MOVE (	0.0,	N10)		
13250	0265		CALL POLY (LND, TT, STMG)				
13300	0266		CALL DASHPT	(0)			
13350	0267		TNX =	RSHA / 6.0			
13400	0268		TY =	LND / 6			
13450	0269		TNY =	STMG (TY)			
13500	0270		CALL MOVE (TNX,		(TNY - (ER * 6.0E07)))		
13550	0271		CALL DRAW ((TNX + (0.01250 * RSHA)),	TNY)			
13600	0272		CALL DRAW (TNX,		(TNY + (ER * 6.0E07)))		
13650	0273		CALL DRAW ((TNX - (0.01250 * RSHA)),	TNY)			
13700	0274		CALL DRAW (TNX,		(TNY - (ER * 6.0E07)))		
13750	0275		TNX =	RSHA / 3.0			
13800	0276		TY =	LND / 3			
13850	0277		TNY =	STMG (TY)			
13900	0278		CALL MOVE (TNX,		(TNY - (ER * 6.0E07)))		
13950	0279		CALL DRAW ((TNX + (0.01250 * RSHA)),	TNY)			
14000	0280		CALL DRAW (TNX,		(TNY + (ER * 6.0E07)))		
14050	0281		CALL DRAW ((TNX - (0.01250 * RSHA)),	TNY)			
14100	0282		CALL DRAW (TNX,		(TNY - (ER * 6.0E07)))		
14150	0283		TNX =	RSHA / 2.0			
14200	0284		TY =	LND / 2			
14250	0285		TNY =	STMG (TY)			

```

14300 0286 CALL MOVE (TNX, (TNY - (ER * 6.0E07)))
14350 0287 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
14400 0288 CALL DRAW (TNX, (TNY + (ER * 6.0E07)))
14450 0289 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
14500 0290 CALL DRAW (TNX, (TNY - (ER * 6.0E07)))
14550 0291 TNX = RSHA * 2.0 / 3.0
14600 0292 TY = LND * 2 / 3
14650 0293 TNY = STMG (TY)
14700 0294 CALL MOVE (TNX, (TNY - (ER * 6.0E07)))
14750 0295 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
14800 0296 CALL DRAW (TNX, (TNY + (ER * 6.0E07)))
14850 0297 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
14900 0298 CALL DRAW (TNX, (TNY - (ER * 6.0E07)))
14950 0299 TNX = RSHA * 5.0 / 6.0
15000 0300 TY = LND * 5 / 6
15050 0301 TNY = STMG (TY)
15100 0302 CALL MOVE (TNX, (TNY - (ER * 6.0E07)))
15150 0303 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
15200 0304 CALL DRAW (TNX, (TNY + (ER * 6.0E07)))
15250 0305 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
15300 0306 CALL DRAW (TNX, (TNY - (ER * 6.0E07)))
15350 0307 TNX = RSHA / 6.0
15400 0308 TY = LND / 6
15450 0309 TNY = PT (TY)
15500 0310 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
15550 0311 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
15600 0312 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
15650 0313 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
15700 0314 TNX = RSHA / 3.0
15750 0315 TY = LND / 3
15800 0316 TNY = PT (TY)
15850 0317 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
15900 0318 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
15950 0319 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
16000 0320 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
16050 0321 TNX = RSHA / 2.0
16100 0322 TY = LND / 2
16150 0323 TNY = PT (TY)
16200 0324 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
16250 0325 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
16300 0326 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
16350 0327 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
16400 0328 TNX = RSHA * 2.0 / 3.0
16450 0329 TY = LND * 2 / 3
16500 0330 TNY = PT (TY)
16550 0331 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
16600 0332 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
16650 0333 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
16700 0334 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
16750 0335 TNX = RSHA * 5.0 / 6.0
16800 0336 TY = LND * 5 / 6
16850 0337 TNY = PT (TY)
16900 0338 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
16950 0339 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
17000 0340 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 6.0E07)))
17050 0341 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 6.0E07)))
17100 0342 CALL MOVE ( 0.0, N10)

```

```

17150 0343      CALL POLY (LND, TT,  PT)
17200 0344
17250 0345      IF (PL .EQ. 1)  GO TO 420
17300 0346      CALL CMCLOS
17350 0347      READ*
17400 0348      CALL CMOPEN
17450 0349      CALL NEWPAG
17500 0350      GO TO 460
17550 0351 420  CALL GRSTOP
17600 0352      WRITE (6, 430)
17650 0353 430  FORMAT (X, '          YOU MUST NOW REMOVE THE COMPLETED
17700 0354      + GRAPHICS DISPLAY, INSERT A NEW')
17750 0355      WRITE (6, 440)
17800 0356 440  FORMAT (X, '          SHEET OF PAPER, AND TOTALLY RESET THE
17850 0357      + 4662 PLOTTING MACHINE.')
17900 0358      WRITE (6, 450)
17950 0359 450  FORMAT (X, '          PRESS <RET> WHEN THIS IS DONE.')
18000 0360      READ*
18050 0361 460  IF (REP .EQ. 1) GO TO 540
18100 0362 470  WRITE (6, 480)
18150 0363 480  FORMAT (X, '          DO YOU WISH TO EXPAND THIS DISPLAY?
18200 0364      + IF SO, TYPE IN 1;  IF NOT, 2')
18250 0365      READ 490, REP
18300 0366 490  FORMAT (I1)
18350 0367      IF (REP .EQ. 1) GO TO 520
18400 0368      IF (REP .EQ. 2) GO TO 540
18450 0369      WRITE (6, 500)
18500 0370 500  FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
18550 0371      + BE UTILIZED.  YOU WILL HAVE')
18600 0372      WRITE (6, 510)
18650 0373 510  FORMAT (X, '          TO TRY AGAIN.')
18700 0374      GO TO 470
18750 0375 520  TSP = PT(LND) - PT(1)
18800 0376      TSP = TSP ** 2.0
18850 0377      TSP = SQRT(TSP)
18900 0378      IF (PT(1) .LE. PT(LND)) MDP = (0.5 * TSP) + N10
18950 0379      IF (PT(1) .GT. PT(LND)) MDP = (0.5 * TSP) + PT(LND)
19000 0380      ER = 1.2 * TSP / 3.0E09
19100 0381      LXP = -(0.6 * TSP) + MDP
19150 0382      HXP = (0.6 * TSP) + MDP
19200 0383      LOWP = (0.4 * TSP) + LXP
19300 0384      MIDP = (0.8 * TSP) + LXP
19305 0385      IF (NK .EQ. 1) ER = ER * 2.0
19309 0386      IF (ZZ .EQ. 1) ER = ER * 4.0
19313 0387      IF (NK .EQ. 1) LXP = -(1.2 * TSP) + MDP
19317 0388      IF (NK .EQ. 1) HXP = (1.2 * TSP) + MDP
19321 0389      IF (NK .EQ. 1) LOWP = (0.8 * TSP) + LXP
19325 0390      IF (NK .EQ. 1) MIDP = (1.6 * TSP) + LXP
19329 0391      IF (ZZ .EQ. 1) LXP = -(2.4 * TSP) + MDP
19333 0392      IF (ZZ .EQ. 1) HXP = (2.4 * TSP) + MDP
19337 0393      IF (ZZ .EQ. 1) LOWP = (1.6 * TSP) + LXP
19341 0394      IF (ZZ .EQ. 1) MIDP = (3.2 * TSP) + LXP
19345 0395      LOWP = LOWP / 1.0E06
19350 0396      MIDP = MIDP / 1.0E06
19400 0397      IF (PL .EQ. 2) GO TO 530
19450 0398      GO TO 360
19500 0399 530  CONTINUE

```

```

19550 0400
19600 0401
19650 0402
19700 0403
19750 0404
19800 0405
19850 0406
19900 0407
19950 0408
20000 0409
20050 0410
20100 0411
20150 0412
20200 0413
20250 0414
20300 0415
20350 0416
20400 0417
20450 0418
20500 0419
20550 0420
20600 0421
20650 0422
20700 0423
20750 0424
20800 0425
20850 0426
20900 0427
20950 0428
21000 0429
21050 0430
21100 0431
21150 0432
21200 0433
21250 0434
21300 0435
21350 0436
21400 0437
21450 0438
21500 0439
21550 0440
21600 0441
21650 0442
21700 0443
21750 0444
21800 0445
21850 0446
21900 0447
21950 0448
22000 0449
22050 0450
22100 0451
22150 0452
22200 0453
22250 0454
22300 0455
22350 0456

540 GO TO 370
    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)

```



```

22400 0457 CALL DRAW (120.0, 78.571)
22450 0458 CALL MOVE ( 0.0, 51.786)
22500 0459 CALL DASHPT (0)
22550 0460 CALL DRAW ( 3.0, 51.786)
22600 0461 CALL MOVE ( 6.0, 51.786)
22650 0462 IF (REP .EQ. 1) GO TO 590
22700 0463 CALL TEXT ( 9, MDFT)
22750 0464 GO TO 600
22800 0465 590 CALL RNUMBR (LOWF, 1, 5)
22850 0466 CALL TEXT ( 5, DEG)
22900 0467 600 CALL MOVE ( 30.0, 51.786)
22950 0468 CALL DASHPT (9)
23000 0469 CALL DRAW (120.0, 51.786)
23050 0470 CALL VWPORT (0.0, 120.0, 25.0, 100.0)
23100 0471 CALL WINDOW (0.0, RSHA, LXF, HXF)
23150 0472 CALL DASHPT (0)
23200 0473 TNX = RSHA / 6.0
23250 0474 TY = LND / 6
23300 0475 TNY = TMPF (TY)
23350 0476 CALL MOVE (TNX, (TNY - (ER * 8.4)))
23400 0477 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
23450 0478 CALL DRAW (TNX, (TNY + (ER * 8.4)))
23500 0479 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
23550 0480 CALL DRAW (TNX, (TNY - (ER * 8.4)))
23600 0481 TNX = RSHA / 3.0
23650 0482 TY = LND / 3
23700 0483 TNY = TMPF (TY)
23750 0484 CALL MOVE (TNX, (TNY - (ER * 8.4)))
23800 0485 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
23850 0486 CALL DRAW (TNX, (TNY + (ER * 8.4)))
23900 0487 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
23950 0488 CALL DRAW (TNX, (TNY - (ER * 8.4)))
24000 0489 TNX = RSHA / 2.0
24050 0490 TY = LND / 2
24100 0491 TNY = TMPF (TY)
24150 0492 CALL MOVE (TNX, (TNY - (ER * 8.4)))
24200 0493 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
24250 0494 CALL DRAW (TNX, (TNY + (ER * 8.4)))
24300 0495 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
24350 0496 CALL DRAW (TNX, (TNY - (ER * 8.4)))
24400 0497 TNX = RSHA * 2.0 / 3.0
24450 0498 TY = LND * 2 / 3
24500 0499 TNY = TMPF (TY)
24550 0500 CALL MOVE (TNX, (TNY - (ER * 8.4)))
24600 0501 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
24650 0502 CALL DRAW (TNX, (TNY + (ER * 8.4)))
24700 0503 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
24750 0504 CALL DRAW (TNX, (TNY - (ER * 8.4)))
24800 0505 TNX = RSHA * 5.0 / 6.0
24850 0506 TY = LND * 5 / 6
24900 0507 TNY = TMPF (TY)
24950 0508 CALL MOVE (TNX, (TNY - (ER * 8.4)))
25000 0509 CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
25050 0510 CALL DRAW (TNX, (TNY + (ER * 8.4)))
25100 0511 CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
25150 0512 CALL DRAW (TNX, (TNY - (ER * 8.4)))
25200 0513 CALL MOVE ( 0.0, TF1)

```

```

25250 0514      CALL DASHPT (5)
25300 0515      CALL POLY (LND, TT, TMPF)
25350 0516
25400 0517      IF (PL .EQ. 1) GO TO 610
25450 0518      CALL CMCLOS
25500 0519      READ*
25550 0520      CALL CMOPEN
25600 0521      CALL NEWPAG
25650 0522      GO TO 650
25700 0523 610    CALL GRSTOP
25750 0524      WRITE (6, 620)
25800 0525 620    FORMAT (X, '          YOU MUST NOW REMOVE THE COMPLETED
+ GRAPHICS DISPLAY, INSERT A NEW')
25850 0526      WRITE (6, 630)
25900 0527 630    FORMAT (X, '          SHEET OF PAPER, AND TOTALLY RESET THE
+ 4662 PLOTTING MACHINE.')
```

```

25950 0528      WRITE (6, 640)
26000 0529 640    FORMAT (X, '          PRESS <RET> WHEN THIS IS DONE.')
```

```

26100 0531      READ*
26150 0532 650    IF (REP .EQ. 1) GO TO 730
26200 0533 660    WRITE (6, 670)
26250 0534 670    FORMAT (X, '          DO YOU WISH TO EXPAND THIS DISPLAY?
+ IF SO, TYPE IN 1; IF NOT, 2')
```

```

26300 0535      READ 680, REP
26350 0536 680    FORMAT (I1)
26400 0537      IF (REP .EQ. 1) GO TO 710
26450 0538      IF (REP .EQ. 2) GO TO 730
26500 0539      WRITE (6, 690)
26550 0540 690    FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
+ BE UTILIZED. YOU WILL HAVE')
```

```

26600 0541      WRITE (6, 700)
26650 0542 700    FORMAT (X, '          TO TRY AGAIN.')
```

```

26700 0543      GO TO 660
26750 0544 710    TSF = DTF(LND) - DTF(1)
26800 0545      TSF = TSF ** 2.0
26850 0546      TSF = SQRT (TSF)
26900 0547      IF (DTF(1) .LE. DTF(LND)) MDF = (0.5 * TSF) + TF1
26950 0548      IF (DTF(1) .GT. DTF(LND)) MDF = -(0.5 * TSF) + TF1
27000 0549      ER = 1.2 * TSF / 420.0
27050 0550      LXF = -(0.6 * TSF) + MDF
27100 0551      HXF = (0.6 * TSF) + MDF
27150 0552      LOWF = (0.4 * TSF) + LXF
27200 0553      MIDF = (0.8 * TSF) + LXF
27250 0554      IF (NK .EQ. 1) ER = ER * 2.0
27300 0555      IF (ZZ .EQ. 1) ER = ER * 4.0
27350 0556      IF (NK .EQ. 1) LXF = -(1.2 * TSF) + MDF
27400 0557      IF (NK .EQ. 1) HXF = (1.2 * TSF) + MDF
27410 0558      IF (NK .EQ. 1) LOWF = (0.8 * TSF) + LXF
27420 0559      IF (NK .EQ. 1) MIDF = (1.6 * TSF) + LXF
27430 0560      IF (ZZ .EQ. 1) LXF = -(2.4 * TSF) + MDF
27440 0561      IF (ZZ .EQ. 1) HXF = (2.4 * TSF) + MDF
27450 0562      IF (ZZ .EQ. 1) LOWF = (1.6 * TSF) + LXF
27460 0563      IF (ZZ .EQ. 1) MIDF = (3.2 * TSF) + LXF
27470 0564      IF (PL .EQ. 2) GO TO 720
27480 0565      GO TO 550
27490 0566 720    CONTINUE
27500 0567      GO TO 560
27550 0568
27600 0570
```

27650	0571	730	REP = 2		
27700	0572		ER = 1.0		
27750	0573		IF (PL .EQ. 1) GO TO 740		
27800	0574		GO TO 750		
27850	0575				
27900	0576	740	CALL GRSTRT (4662, 1)		
27950	0577	750	CALL VWPORT (0.0, 130.0,	0.0, 100.0)	
28000	0578		CALL WINDOW (0.0, 130.0,	0.0, 100.0)	
28050	0579		CALL DASHPT (0)		
28100	0580		CALL MOVE ( 0.0, 25.0)		
28150	0581		CALL DRAW (120.0, 25.0)		
28200	0582		CALL DRAW (120.0, 100.0)		
28250	0583		CALL DRAW ( 0.0, 100.0)		
28300	0584		CALL DRAW ( 0.0, 25.0)		
28350	0585		CALL MOVE ( 0.0, 62.5)		
28400	0586		CALL DASHPT (9)		
28450	0587		CALL DRAW (120.0, 62.5)		
28500	0588		CALL DASHPT (0)		
28550	0589		CALL MOVE ( 0.0, 85.0)		
28600	0590		CALL DRAW ( 3.0, 85.0)		
28650	0591		CALL MOVE ( 6.0, 85.0)		
28700	0592		CALL TXICUR (4)		
28750	0593		IF (REP .EQ. 1) GO TO 760		
28800	0594		CALL TEXT ( 9, POST)		
28850	0595		GO TO 770		
28900	0596	760	CALL RNUMBR (MIDT, 1,	5)	
28950	0597		CALL TEXT ( 5, DEG)		
29000	0598	770	CALL MOVE ( 20.0, 64.5)		
29050	0599		CALL DRAW ( 20.0, 60.5)		
29100	0600		CALL MOVE ( 40.0, 64.5)		
29150	0601		CALL DRAW ( 40.0, 60.5)		
29200	0602		CALL MOVE ( 60.0, 66.5)		
29250	0603		CALL DRAW ( 60.0, 60.5)		
29300	0604		CALL MOVE ( 80.0, 64.5)		
29350	0605		CALL DRAW ( 80.0, 60.5)		
29400	0606		CALL MOVE (100.0, 64.5)		
29450	0607		CALL DRAW (100.0, 60.5)		
29500	0608		CALL MOVE ( 0.0, 40.0)		
29550	0609		CALL DRAW ( 3.0, 40.0)		
29600	0610		CALL MOVE ( 6.0, 40.0)		
29650	0611		IF (REP .EQ. 1) GO TO 780		
29700	0612		CALL TEXT ( 9, NEGT)		
29750	0613		GO TO 790		
29800	0614	780	CALL RNUMBR (LOWT, 1,	5)	
29850	0615		CALL TEXT ( 5, DEG)		
29900	0616	790	CALL MOVE ( 0.0, 62.0)		
29950	0617		CALL TXICUR (7)		
30000	0618		CALL TEXT ( 1, ZERO)		
30050	0619		CALL MOVE ( 4.0, 24.0)		
30100	0620		CALL TEXT ( 4, TIME)		
30150	0621		CALL MOVE ( 20.0, 24.0)		
30200	0622		CALL TXICUR (8)		
30250	0623		CALL INUMBR (INT1, 2)		
30300	0624		CALL MOVE ( 40.0, 24.0)		
30350	0625		CALL INUMBR (INT2, 3)		
30400	0626		CALL MOVE ( 60.0, 24.0)		
30450	0627		CALL INUMBR (INT3, 3)		

```

30500 0628 CALL MOVE ( 80.0, 24.0)
30550 0629 CALL INUMBR (INT4, 3)
30600 0630 CALL MOVE (100.0, 24.0)
30650 0631 CALL INUMBR (INT5, 3)
30700 0632 CALL MOVE (120.0, 24.0)
30750 0633 CALL TXICUR (9)
30800 0634 CALL INUMBR (INT6, 3)
30850 0635 CALL TXICUR (4)
30900 0636 CALL MOVE ( 0.0, 20.0)
30950 0637 CALL TEXT ( 19, TMC)
31000 0638 CALL MOVE ( 42.0, 20.0)
31050 0639 CALL DASHPT (2)
31100 0640 CALL DRAW (120.0, 20.0)
31150 0641 CALL MOVE ( 0.0, 16.0)
31200 0642 CALL TEXT ( 19, TOC)
31250 0643 CALL MOVE ( 42.0, 16.0)
31300 0644 CALL DASHPT (3)
31350 0645 CALL DRAW (120.0, 16.0)
31400 0646 CALL MOVE ( 0.0, 12.0)
31450 0647 CALL TEXT ( 19, TIC)
31500 0648 CALL MOVE ( 42.0, 12.0)
31550 0649 CALL DASHPT (7)
31600 0650 CALL DRAW (120.0, 12.0)
31650 0651 CALL DASHPT (9)
31700 0652 CALL MOVE ( 30.0, 85.0)
31750 0653 CALL DRAW (120.0, 85.0)
31800 0654 CALL MOVE ( 30.0, 40.0)
31850 0655 CALL DRAW (120.0, 40.0)
31900 0656 CALL VWPORT (0.0, 120.0, 25.0, 100.0)
31950 0657 CALL WINDOW (0.0, RSHA, LXT, HXT)
32000 0658 CALL MOVE ( 0.0, 62.5)
32050 0659 CALL DASHPT (2)
32100 0660 CALL POLY (LND, TT, DTM)
32150 0661 CALL DASHPT (0)
32200 0662 TNX = RSHA / 6.0
32250 0663 TY = LND / 6
32300 0664 TNY = DTM (TY)
32350 0665 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
32400 0666 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
32450 0667 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
32500 0668 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
32550 0669 TNX = RSHA / 3.0
32600 0670 TY = LND / 3
32650 0671 TNY = DTM (TY)
32700 0672 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
32750 0673 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
32800 0674 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
32850 0675 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
32900 0676 TNX = RSHA / 2.0
32950 0677 TY = LND / 2
33000 0678 TNY = DTM (TY)
33050 0679 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
33100 0680 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
33150 0681 CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
33200 0682 CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
33250 0683 TNX = RSHA * 2.0 / 3.0
33300 0684 TY = LND * 2 / 3

```

```

33350 0685      TNX = DTM (TY)
33400 0686      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
33450 0687      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
33500 0688      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
33550 0689      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
33600 0690      TNX = RSHA * 5.0 / 6.0
33650 0691      TY = LND * 5 / 6
33700 0692      TNY = DTM (TY)
33750 0693      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY - (ER * 1.0)))
33800 0694      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY + (ER * 1.0)))
33850 0695      CALL MOVE ((TNX - (RSHA * 0.01250)), (TNY + (ER * 1.0)))
33900 0696      CALL DRAW ((TNX + (RSHA * 0.01250)), (TNY - (ER * 1.0)))
33950 0697      CALL MOVE ( 0.0, 62.5)
34000 0698      CALL DASHPT (3)
34050 0699      CALL POLY (LND, TT, DTO)
34100 0700      CALL DASHPT (0)
34150 0701      TNX = RSHA / 6.0
34200 0702      TY = LND / 6
34250 0703      TNY = DTO (TY)
34300 0704      CALL MOVE (TNX, (TNY - (ER * 1.0)))
34350 0705      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
34400 0706      CALL DRAW (TNX, (TNY + (ER * 1.0)))
34450 0707      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
34500 0708      CALL DRAW (TNX, (TNY - (ER * 1.0)))
34550 0709      TNX = RSHA / 3.0
34600 0710      TY = LND / 3
34650 0711      TNY = DTO (TY)
34700 0712      CALL MOVE (TNX, (TNY - (ER * 1.0)))
34750 0713      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
34800 0714      CALL DRAW (TNX, (TNY + (ER * 1.0)))
34850 0715      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
34900 0716      CALL DRAW (TNX, (TNY - (ER * 1.0)))
34950 0717      TNX = RSHA / 2.0
35000 0718      TY = LND / 2
35050 0719      TNY = DTO (TY)
35100 0720      CALL MOVE (TNX, (TNY - (ER * 1.0)))
35150 0721      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
35200 0722      CALL DRAW (TNX, (TNY + (ER * 1.0)))
35250 0723      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
35300 0724      CALL DRAW (TNX, (TNY - (ER * 1.0)))
35350 0725      TNX = RSHA * 2.0 / 3.0
35400 0726      TY = LND * 2 / 3
35450 0727      TNY = DTO (TY)
35500 0728      CALL MOVE (TNX, (TNY - (ER * 1.0)))
35550 0729      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
35600 0730      CALL DRAW (TNX, (TNY + (ER * 1.0)))
35650 0731      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
35700 0732      CALL DRAW (TNX, (TNY - (ER * 1.0)))
35750 0733      TNX = RSHA * 5.0 / 6.0
35800 0734      TY = LND * 5 / 6
35850 0735      TNY = DTO (TY)
35900 0736      CALL MOVE (TNX, (TNY - (ER * 1.0)))
35950 0737      CALL DRAW ((TNX + (0.01250 * RSHA)), TNY)
36000 0738      CALL DRAW (TNX, (TNY + (ER * 1.0)))
36050 0739      CALL DRAW ((TNX - (0.01250 * RSHA)), TNY)
36100 0740      CALL DRAW (TNX, (TNY - (ER * 1.0)))
36150 0741      CALL DASHPT (0)

```

```

36200 0742 TNX = RSHA / 6.0
36250 0743 TY = LND / 6
36300 0744 TNY = DTI (TY)
36350 0745 CALL MOVE (TNX, (TNY - (ER * 1.0)))
36400 0746 CALL DRAW ((TNX + (0.01082 * RSHA)), (TNY + (ER * 0.5)))
36450 0747 CALL DRAW ((TNX - (0.01082 * RSHA)), (TNY + (ER * 0.5)))
36500 0748 CALL DRAW (TNX, (TNY - (ER * 1.0)))
36550 0749 TNX = RSHA / 3.0
36600 0750 TY = LND / 3
36650 0751 TNY = DTI (TY)
36700 0752 CALL MOVE (TNX, (TNY - (ER * 1.0)))
36750 0753 CALL DRAW ((TNX + (0.01082 * RSHA)), (TNY + (ER * 0.5)))
36800 0754 CALL DRAW ((TNX - (0.01082 * RSHA)), (TNY + (ER * 0.5)))
36850 0755 CALL DRAW (TNX, (TNY - (ER * 1.0)))
36900 0756 TNX = RSHA / 2.0
36950 0757 TY = LND / 2
37000 0758 TNY = DTI (TY)
37050 0759 CALL MOVE (TNX, (TNY - (ER * 1.0)))
37100 0760 CALL DRAW ((TNX + (0.01082 * RSHA)), (TNY + (ER * 0.5)))
37150 0761 CALL DRAW ((TNX - (0.01082 * RSHA)), (TNY + (ER * 0.5)))
37200 0762 CALL DRAW (TNX, (TNY - (ER * 1.0)))
37250 0763 TNX = RSHA * 2.0 / 3.0
37300 0764 TY = LND * 2 / 3
37350 0765 TNY = DTI (TY)
37400 0766 CALL MOVE (TNX, (TNY - (ER * 1.0)))
37450 0767 CALL DRAW ((TNX + (0.01082 * RSHA)), (TNY + (ER * 0.5)))
37500 0768 CALL DRAW ((TNX - (0.01082 * RSHA)), (TNY + (ER * 0.5)))
37550 0769 CALL DRAW (TNX, (TNY - (ER * 1.0)))
37600 0770 TNX = RSHA * 5.0 / 6.0
37650 0771 TY = LND * 5 / 6
37700 0772 TNY = DTI (TY)
37750 0773 CALL MOVE (TNX, (TNY - (ER * 1.0)))
37800 0774 CALL DRAW ((TNX + (0.01082 * RSHA)), (TNY + (ER * 0.5)))
37850 0775 CALL DRAW ((TNX - (0.01082 * RSHA)), (TNY + (ER * 0.5)))
37900 0776 CALL DRAW (TNX, (TNY - (ER * 1.0)))
37950 0777 CALL MOVE ( 0.0, 62.5)
38000 0778 CALL DASHPT (7)
38050 0779 CALL POLY (LND, TT, DTI)
38100 0780
38150 0781 IF (REP .EQ. 1) GO TO 920
38200 0782 IF (PL .EQ. 1) GO TO 800
38250 0783 CALL CMCLOS
38300 0784 READ*
38350 0785 CALL CMOPEN
38400 0786 CALL NEWPAG
38450 0787 GO TO 840
38500 0788 800 CALL GRSTOP
38550 0789 WRITE (6, 810)
38600 0790 810 FORMAT (X, ' YOU MUST NOW REMOVE THE COMPLETED
38650 0791 + GRAPHICS DISPLAY, INSERT A NEW')
38700 0792 WRITE (6, 820)
38750 0793 820 FORMAT (X, ' SHEET OF PAPER, AND TOTALLY RESET THE
38800 0794 + 4662 PLOTTING MACHINE.')
38850 0795 WRITE (6, 830)
38900 0796 830 FORMAT (X, ' PRESS <RET> WHEN THIS IS DONE.')
38950 0797 READ*
39000 0798 840 CONTINUE

```

```

39050 0799      850 WRITE (6, 860)
39100 0800      860 FORMAT (X, '          DO YOU WISH TO EXPAND THIS DISPLAY?
39150 0801      + IF SO, TYPE IN 1; IF NOT, ')
39200 0802      READ 870, REP
39250 0803      870 FORMAT (I1)
39300 0804      IF (REP .EQ. 1) GO TO 900
39350 0805      IF (REP .EQ. 2) GO TO 920
39400 0806      WRITE (6, 880)
39450 0807      880 FORMAT (X, '          YOU HAVE TYPED IN A NUMBER THAT CANNOT
39500 0808      + BE UTILIZED. YOU WILL HAVE')
39550 0809      WRITE (6, 890)
39600 0810      890 FORMAT (X, '          TO TRY AGAIN.')
39650 0811      GO TO 850
39700 0812      900 TST = DTO(LND) - DTI(LND)
39750 0813      TST = TST ** 2.0
39800 0814      TST = SQRT (TST)
39850 0815      ER = 1.2 * TST / 50.0
39900 0816      IF (NK .EQ. 1.0) ER = 2.0
39950 0817      LXT = -0.6 * TST
40000 0818      HXT = 0.6 * TST
40050 0819      LOWT = -0.3 * TST
40100 0820      MIDT = 0.3 * TST
40104 0821      IF (NK .EQ. 1) LXT = -1.2 * TST
40108 0822      IF (NK .EQ. 1) HXT = 1.2 * TST
40112 0823      IF (NK .EQ. 1) LOWT = -0.6 * TST
40116 0824      IF (NK .EQ. 1) MIDT = 0.6 * TST
40120 0825      IF (ZZ .EQ. 1) LXT = -2.4 * TST
40124 0826      IF (ZZ .EQ. 1) HXT = 2.4 * TST
40128 0827      IF (ZZ .EQ. 1) LOWT = -1.2 * TST
40132 0828      IF (ZZ .EQ. 1) MIDT = 1.2 * TST
40140 0829      IF (NK .EQ. 1) ER = ER * 2.0
40145 0830      IF (ZZ .EQ. 1) ER = ER * 4.0
40150 0831      IF (PL .EQ. 2) GO TO 910
40200 0832      GO TO 740
40250 0833      910 CONTINUE
40300 0834      GO TO 750
40350 0835
40400 0836      920 CALL GRSTOP
40450 0837      RETURN
40500 0838      END

```