THE IQS METHOD APPLIED TO
THERMAL REACTOR KINETICS

THE IMPROVED QUASISTATIC METHOD
APPLIED TO THERMAL REACTOR KINETICS

by

JOHN VINCENT ANTHONY MARCZAK, B.Sc.

An On-Campus Project Report
Submitted to the School of Graduate Studies
in Partial Fulfilment of the Requirements
for the Degree
Master of Engineering

McMaster University

August 1982

MASTER OF ENGINEERING (1982)
(Engineering Physics)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: The Improved Quasistatic Method
Applied to Thermal Reactor Kinetics

AUTHOR: John Vincent Anthony Marczak, B.Sc. (University of Waterloo)

SUPERVISOR: Professor O.A. Trojan

NUMBER OF PAGES: vii, 80

ABSTRACT

In this report, the theory for the Improved Quasistatic (IQS) method of solving the three-dimensional, two-neutron-energy group, time-dependent neutron diffusion equations is developed, and approximations appropriate to the CANDU-PHW reactor system are introduced. The theory is extended to a numerical formulation of the problem. The TM-2 computer program (written in FORTRAN 5), which employs the IQS method to numerically solve a two-dimensional form of the diffusion equations (with a correction to account for axial leakage), is outlined. Input and output descriptions for the TM-2 code are provided.

ACKNOWLEDGEMENTS

The author is grateful to Dr. O.A. Trojan, Professor of Engineering Physics at McMaster University and Manager of the Reactor Physics 'B' Branch of Atomic Energy of Canada Limited-Engineering Company (AECL-EC) for the guidance, supervision, and encouragement he provided throughout this project. Dr. Trojan's invaluable assistance and never-ending patience will always be remembered.

Appreciation is extended to the staff of the Physics Department of AECL-EC, especially Dr. B. Rouben, Dr. P. Akhtar and Mr. A.R. Dastur, for their helpful suggestions and detailed explanation of the CERBERUS and MATMAP computer codes, which are currently used in reactor kinetics studies at AECL-EC.

The flux calculations in the TM-2 program (described herein) have been adapted from the CERBERUS program with some modifications.

The author is indebted to Ms. Marianne Van der Wel, and Mr. W.W. Hutton of the McMaster Computer Consulting Staff for the assistance provided in resolving loader error problems and for preparing the COMPASS routine used by the TM-2 code for system resource unit (SRU) management.

TABLE OF CONTENTS

PAGE

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

THE IMPROVED QUASISTATIC METHOD APPLIED TO THERMAL REACTOR KINETICS

1.0 INTRODUCTION

For the purposes of reactor safety design and analysis, it is important to be able to numerically simulate the temporal behaviour of the neutron flux distribution in a reactor during any prescribed accident or otherwise induced transient. While the Linearized Boltzmann Transport Equation (LBTE)[1] can fully describe the neutronics of any possible reactor configuration during any sequence of events, the solution of the full seven-dimensional LBTE is generally intractable, and hence approximations and averaging over one or more of the variables is usually required.

The form of the equation or set of equations that results, and the method of solution, depend on the nature of the system, the type of event being simulated, the required accuracy and detail of the final solution, and of course the computational budget restraints.

In large thermal reactors, small local changes in the system may cause deformations of the neutron flux, and therefore, deviations from the point kinetics model. This situation has led to the development of a number of methods that account for spatial effects in reactor kinetics with varying levels of sophistication and accuracy(2). Space-dependent neutron transients in CANDU-PHW* systems result from a combination of:

1) Neutronic decoupling of reactor segments due to deliberate flattening of the power distribution,

2) Significant retardation of the power shape transient due to delayed neutron holdback, and

3) Asymmetric insertion of reactivity devices(3).

The flux distribution in a CANDU-PHW reactor changes as the reactor configuration and fuel composition changes. From a computational standpoint, these changes are classified according to the time scale over which they occur (4).

* CANDU-PHW: Canadian Deuterium Uranium - Pressurized Heavy Water

Changes in reactor configuration can result from the shifting of fuel bundles due to fuelling operations, the movement of control absorbers, the rapid insertion of shut-off rods or moderator poison, or accidental changes such as breaks in the primary heat transport system (PHTS) which lead to loss-of-coolant accidents (LOCAs)(4).

Similarly, changes in fuel composition can result from changes in the concentration of heavy isotopes (burnup), changes in fission product inventories, and changes in concentrations of delayed neutron precursors due to rapid changes in the neutron flux(4).

In this report, the modelling of transients such as those resulting from LOCAs and the subsequent insertion of shutdown devices, and startups following short shutdowns are considered, with explicit treatment of delayed neutrons and xenon effects.

2. MATHEMATICAL MODEL

The methods of solution of the neutron balance equations range from direct calculation of the full energy-space-time dependence of the neutron flux, to simple application of a point kinetics model with insertion of more properly calculated reactivity values(2). Between these extremes fall the modal or nodal expansions and quasistatic or adiabatic methods(2).

In this paper the Improved Quasistatic (IQS) method, introduced in Reference 2, is outlined, and details of a computer program, TM-2, which has been developed to solve the equations are given. The IQS method is particularly suited to handling CANDU-PHW transients for reasons outlined in Section 3.

2.1 THE IMPROVED QUASISTATIC METHOD

The three-dimensional, two-energy group, time-dependent neutron diffusion equations can be expressed as:

$$\frac{1}{\upsilon_1}\frac{\partial\Phi_1}{\partial t} = \nabla\cdot D_1\nabla\Phi_1-\Sigma_{a,1}\Phi_1-\Sigma_m\Phi_1+\nu\Sigma_{f,2}\Phi_2(1-\beta)+\sum_{i=1}^{N}\lambda_i C_i \tag{2.1-1}$$

$$\frac{1}{\upsilon_2}\frac{\partial\Phi_2}{\partial t} = \nabla\cdot D_2\nabla\Phi_2-\Sigma_{a,2}\Phi_2+\Sigma_m\Phi_1 \tag{2.1-2}$$

All symbols are explained in the Nomenclature. Fast fissions are accounted for by a correction to the value used for $\nu\Sigma_{f,2}$. The value for $\Sigma_{a,2}$ contains the effects due to all thermal absorptions, including xenon. No external sources are considered. All neutrons are assumed to be born in the fast group. All fluxes, material properties, and precursor concentrations are both time and space-dependent.

The implementation of the IQS method requires the factorization of the flux into a time-dependent amplitude and a space- and time-dependent shape function(2). That is:

$$\Phi_1(r,t)=A(t)\phi_1(r,t) \qquad (2.1\text{-}3)$$

$$\Phi_2(r,t)=A(t)\phi_2(r,t) \qquad (2.1\text{-}4)$$

where $A(0) = 1.0$ (2.1-5)

The factorization requires another separation condition for $t>0$ in order to become a unique definition. This splitting is to be made in such a way that A(t) contains the main time-dependence of the total flux, so that the time dependence of ϕ_1 and ϕ_2 accounts only for the relatively slow space-energy variations. There are many conditions that can be used, all of which involve constraining some space-energy integral of the shape functions to a constant value for all $t \geq 0$(2). The only basic requirement is that the shape function remain positive and bounded for all time and space. Henry imposed the requirement that:

$$\frac{1}{\upsilon_1}\frac{\partial}{\partial t}\int \phi_1\phi_1^* dV + \frac{1}{\upsilon_2}\frac{\partial}{\partial t}\int \phi_2\phi_2^* dV = 0 \qquad (2.1\text{-}6)$$

where ϕ_1^* and ϕ_2^* are the solution to the steady-state adjoint diffusion equations, and the integrals are taken over the entire reactor core.

The steady-state adjoint flux is space- and energy-dependent only, and has the physical significance of being proportional to the over-all power level which will result if neutrons with speed υ_1 or υ_2 (for $\phi_1^*(r)$ and $\phi_2^*(r)$, respectively) are introduced at point r in a critical reactor initially at zero power(5). The adjoint equations are considered further in Section 2.4.

Physically, since $\phi_i(r, t=0)/\upsilon_i$ (i=1,2) is just the number density of neutrons at point r, the integral in Equation 2.1-6 is proportional to the power level which will result asymptotically from the presence of these neutrons in the core for which ϕ_i^* is the adjoint(5).

The imposition of the restriction given in Equation 2.1-6 satisfies the requirements of the constraint and has the added advantage that it facilitates the transition to the point kinetics formulation as shall be illustrated below.

It is important to note that Equations 2.1-3 and 2.1-4 represent purely a definition and embody no physical assumptions. Equation 2.1-6 is a restrictive condition placed on the time-dependent shape function and again requires no physical assumption$^{(5)}$. If the time dependence of Φ_i is separable, ϕ_i will be independent of time and Equation 2.1-6 will be obeyed automatically.

2.1.1 MATHEMATICAL FORMULATION

2.1.1.1 THE AMPLITUDE EQUATION

If Equations 2.1-3 and 2.1-4 are substituted into Equations 2.1-1 and 2.1-2, and then, if Equation 2.1-1 is multiplied by ϕ_1^* and Equation 2.1-2 is multiplied by ϕ_2^* and both equations are integrated over the entire reactor, and then added together, one can express the resulting equation as:

$$\frac{dA}{dt} = \frac{\rho - \overline{\beta}}{\overline{\ell}} A + \sum_{i=1}^{N} \lambda_i \overline{C}_i / K \qquad (2.1\text{-}7)$$

with: $\rho = \{\int \nu\Sigma_{f,2}\phi_2\phi_1^* dV + \int(\nabla\cdot D_1\nabla\phi_1)\phi_1^* dV + \int(\nabla\cdot D_2\nabla\phi_2)\phi_2^* dV$

$-\int(\Sigma_{a,1} + \Sigma_m)\phi_1\phi_1^* dV - \int\Sigma_{a,2}\phi_2\phi_2^* dV + \int\Sigma_m\phi_1\phi_2^* dV\}$

$\div\{\int\nu\Sigma_{f,2}\phi_2\phi_1^* dV\}$ (2.1-8)

$$\overline{\beta} = \{\int\beta\nu\Sigma_{f,2}\phi_2\phi_1^* dV\} \;/\; \{\int\nu\Sigma_{f,2}\phi_2\phi_1^* dV\} \qquad (2.1\text{-}9)$$

$$\beta = \sum_{i=1}^{N} \beta_i \qquad (2.1\text{-}10)$$

$$\overline{\ell}= \{\frac{1}{\upsilon_1}\int\phi_1\phi_1^* dV + \frac{1}{\upsilon_2}\int\phi_2\phi_2^* dV\} / \{\int\nu\Sigma_{f,2}\phi_2\phi_1^* dV\} \qquad (2.1\text{-}11)$$

$$\overline{C}_i = \int C_i\phi_1^* dV \qquad (2.1\text{-}12)$$

$$K = \frac{1}{\upsilon_1}\int\phi_1\phi_1^* dV + \frac{1}{\upsilon_2}\int\phi_2\phi_2^* dV \qquad (2.1\text{-}13)$$

(If desired, K can be forced to equal 1.0.)

Hence the amplitude equation (2.1-7) takes the same form as the point kinetics equation, with the vital difference that the integral quantities must be averaged with the time-dependent shape functions. In the ideal case, where the flux transient is space-independent, the amplitude equation reduces to the point kinetics equation.

Note that weighting the neutron reaction rates by the neutron importance (adjoint flux) makes the fast and thermal reaction rates comparable on the basis of their ability to produce fission neutrons and allows the addition of the fast and thermal group equations without any loss of accuracy[(4)].

2.1.1.2 THE SHAPE EQUATIONS

The shape equations are readily derived from Equations 2.1-1 and 2.1-2 by insertion of Equations 2.1-3 and 2.1-4. Division by the amplitude gives:

$$\frac{1}{\upsilon_1}\left[\frac{\partial\phi_1}{\partial t} + \frac{\phi_1}{A}\frac{dA}{dt}\right] = \nabla\cdot D_1\nabla\phi_1 - \Sigma_{a,1}\phi_1 - \Sigma_m\phi_1 + \nu\Sigma_{f,2}\phi_2(1-\beta) + \frac{1}{A}\sum_{i=1}^{N}\lambda_i C_i \qquad (2.1\text{-}14)$$

$$\frac{1}{\upsilon_2}\left[\frac{\partial\phi_2}{\partial t} + \frac{\phi_2}{A}\frac{dA}{dt}\right] = \nabla\cdot D_2\nabla\phi_2 - \Sigma_{a,2}\phi_2 + \Sigma_m\phi_1 \qquad (2.1\text{-}15)$$

It can be seen that the amplitude and shape equations are coupled. They are therefore to be solved by an alternating procedure. This shall be discussed in Section 3.

2.2 CALCUALTION OF THE DELAYED NEUTRON PRECURSOR CONCENTRATIONS

The space and time-dependant precursor concentration for group "i" is obtained from the precursor balance equation:

$$\frac{\partial C_i}{\partial t} = \beta_i \frac{\nu\Sigma_{f,2}}{\text{k-eff}} \phi_2 A - \lambda_i C_i \qquad (2.2\text{-}1)$$

Upon integration this gives:

$$C_i(r,t)=\exp(-\lambda_i t)\left[\int_0^t \beta_i \frac{\nu\Sigma_{f,2}}{\text{k-eff}} \phi_2 A \exp(\lambda_i t')dt' + C_i(r,0)\right] \qquad (2.2\text{-}2)$$

where $C_i(r,0)$ is the concentration at t=0. For the most general case, β_i, $\nu\Sigma_{f,2}$, and ϕ_2 are all space and time-dependent, and must be included in the integral over time. The decay constant has been assumed to be independent of space and time. It should be noted that in the TM-2 code, all β_i and λ_i values are assumed to be constant in space and time. The value of $\nu\Sigma_{f,2}$ is assumed to be constant over the time-step (q.v. Section 3) considered.

The steady-state precursor concentrations are obtained by setting the left-hand-side of Equation 2.2-1 equal to zero. This gives:

$$C_i(r,0) = \frac{\beta_i}{\lambda_i}\,\frac{\nu\Sigma_{f,2}}{k\text{-eff}}\,\phi_2(r,0) \tag{2.2-3}$$

since A(0) = 1.0.

2.3 STEADY-STATE DIRECT FLUX EQUATIONS

The steady-state neutron diffusion equations provide the starting fluxes from which the transient begins. At t=0, the neutron flux satisfies the following equations:

$$-\nabla\cdot D_1\nabla\phi_1 + (\Sigma_{a,1} + \Sigma_m)\phi_1 = \nu\Sigma_{f,2}\phi_2 \;/\; k\text{-eff} \tag{2.3-1}$$

$$-\nabla\cdot D_2\nabla\phi_2 + \Sigma_{a,2}\phi_2 = \Sigma_m\phi_1 \tag{2.3-2}$$

This can also be expressed in matrix form as:

$$\begin{bmatrix} \nabla\cdot D_1\nabla - (\Sigma_{a,1} + \Sigma_m) & \nu\Sigma_{f,2} \;/\; k\text{-eff} \\ \Sigma_m & \nabla\cdot D_2\nabla - \Sigma_{a,2} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.3-3}$$

For all physical systems, k-eff is exactly unity. Recall that in steady-state the amplitude is unity.

2.4 STEADY-STATE ADJOINT FLUX EQUATIONS

The corresponding adjoint equation is:

$$\begin{bmatrix} \nabla\cdot D_1\nabla - (\Sigma_{a,1} + \Sigma_m) & \Sigma_m \\ \nu\Sigma_{f,2} \,/\, k\text{-eff} & \nabla\cdot D_2\nabla - \Sigma_{a,2} \end{bmatrix} \begin{bmatrix} \phi_1^* \\ \phi_2^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (2.4\text{-}1)$$

The solution of Equation 2.4-1 gives the weighting functions used in the amplitude equation.

2.5 XENON CONCENTRATION

The thermal neutron absorption cross section contains a component due to absorption by xenon. Since the xenon level depends on the flux, it may be prudent to calculate the xenon concentration explicitly during a transient.

The iodine and xenon balance equations are:

$$\frac{\partial I}{\partial t} = \gamma_I \Sigma_{f,2} \Phi_2 - \lambda_I I \qquad (2.5\text{-}1)$$

$$\frac{\partial X}{\partial t} = \gamma_X \Sigma_{f,2} \Phi_2 + \lambda_I I - \{\sigma\Phi_2 + \lambda_X\} X \qquad (2.5\text{-}2)$$

2.5.1 STEADY-STATE XENON AND IODINE CONCENTRATIONS

Under steady-state conditions, the left-hand-sides of Equations 2.5-1 and 2.5-2 are zero. Consequently at steady-state:

$$I(r) = \gamma_I \Sigma_{f,2} \Phi_2 \; / \; \lambda_I \qquad (2.5\text{-}3)$$

$$X(r) = (\gamma_I + \gamma_X) \Sigma_{f,2} \Phi_2 \; / \; (\sigma\Phi_2 + \lambda_X) \qquad (2.5\text{-}4)$$

These equations must be solved simultaneously with the steady-state flux equations.

2.5.2 XENON CONCENTRATION FOR DYNAMIC FLUX CALCULATIONS

For dynamic flux calculations, the xenon concentration can be approximated by dividing the simulation time into small sub-intervals and determining the xenon concentration to be used during a given sub-interval with the assumption that during that "time-step", the flux remains equal to its value at the start of the sub-interval.

With Φ_1 and Φ_2 being independent of time in Equations 2.5-1 and 2.5-2, one can integrate the equations to find the xenon and iodine concentrations at time "t" during the subinterval:

$$I(r,t) = \gamma_I \Sigma_{f,2} \Phi_2 \{1 - \exp(-\lambda_I t)\} \; / \; \lambda_I + I(r,0)\exp(-\lambda_I t) \qquad (2.5\text{-}5)$$

$$\begin{aligned} X(r,t) = {} & (\gamma_I + \gamma_X) \Sigma_{f,2} \Phi_2 \{1 - \exp(-\lambda_X t - \sigma\Phi_2 t)\} \; / \; (\lambda_X + \sigma\Phi_2) \\ & + (\lambda_I I(r,0) - \gamma_I \Sigma_{f,2} \Phi_2) \{\exp(-\lambda_I t) - \exp(-\lambda_X t - \sigma\Phi_2 t)\} \; / \\ & (\sigma\Phi_2 + \lambda_X - \lambda_I) + X(r,0)\exp(-\lambda_X t - \sigma\Phi_2 t) \qquad (2.5\text{-}6) \end{aligned}$$

where I(r,0) and X(r,0) are respectively the iodine and xenon concentrations at the start of the sub-interval.

3.0 NUMERICAL FORMULATION

Up to this point, the factorizing approach does not introduce any approximation. However, it has led to a more complicated formulation than the original, as represented by Equations 2.1-1 and 2.1-2. The advantage is gained when one makes use of the assumption that the time-dependence of the shape function is of lesser importance than the time-dependence of the amplitude function[(2)].

The factorization of the flux (q.v. Equations 2.1-3 and 2.1-4) with the restriction given by Equation 2.1-6 minimizes the number of times that the shape functions must be calculated during a transient. Due to the neutronic decoupling in CANDU-PHW reactors, the major computational expense lies in the calculation of the flux shape as opposed to the flux level, and minimizing the frequency of the flux shape calculation results in a dramatic reduction in cost[(4)].

The TM-2 computer program has been written to take advantage of this factorizing approach. The code has been written in FORTRAN 5, and operates under the NOS system. In this section the numerical procedure used by the TM-2 program is outlined.

3.1 SIMULATION MODEL

The TM-2 code uses a finite difference approximation to the time and space derivatives in the shape calculations and hence the reactor must be divided into rectangular mesh cells. The user must lay out a series of perpendicular mesh lines, providing a border of mesh cells around the outer region to allow the specification of external boundary conditions.

The amplitude and related integrals are calculated using DGEAR, an IMSL routine for solving sets of stiff differential equations[(6)]. The code requires the user to input basic fuel type properties for each mesh point in the region of solution. Incremental device properties can be added to the basic fuel properties. The flux is calculated at the centre, rather than at the boundary of each cell. This avoids the problem of specifying cross sections at the interface of materials having widely differing neutronic properties such as at the interface of a shutoff rod and a regular lattice cell.

3.2 THE TREATMENT OF SPATIAL DERIVATIVES

The TM-2 computer code allows only for two spatial dimensions, but has provisions for an axial leakage correction. The diffusion coefficients are assumed to be constant over the reactor. For energy group "i" (with i=1, or 2), the leakage terms in the diffusion equations are expressed in finite difference form as:

$$\nabla \cdot D_i \nabla \phi_i \longrightarrow D_i \nabla^2 \phi_i \longrightarrow D_i \{(\phi_R - 2\phi_C + \phi_L) / (\Delta x)^2 + (\phi_T - 2\phi_C + \phi_B) / (\Delta y)^2 + B_i^2 \phi_C\} \qquad (3.2\text{-}1)$$

where the subscripts C, R, L, T, and B represent the centre, right, left, top and bottom, respectively, and refer to the flux for group "i" at a mesh point, C, and the four surrounding cells. The axial leakage is approximated by $D_i B_i^2 \phi_C$ where B_i^2 is the axial buckling for group "i".

3.3 THE TREATMENT OF TIME DERIVATIVES

In the amplitude calculation, the time derivatives are handled using the Adam's predictor-corrector method in DGEAR(6). Since the amplitude calculation handles most of the time-dependence of the flux, sufficient accuracy can only be attained using this or similar higher-order methods.

In the shape calculations, where there is a much weaker time dependence, the TM-2 code approximates the time derivative as a backward difference. At time (t + T):

$$\frac{\partial \phi_i}{\partial t} \longrightarrow \frac{\phi_i(t+T) - \phi_i(t)}{T} \qquad (3.3\text{-}1)$$

Since the shape function is slowly varying, this first-order difference form can be applied over a much larger time interval than would be possible for Φ_i. This form was primarily chosen for reasons of simplicity and ease of application to a computer code. Secondary reasons were its stability with large time-step lengths(2),(4) and the fact that a linear relationship over T results in Equation 2.1-6 being satisfied automatically within the range(2).

3.4 STEADY-STATE DIRECT AND ADJOINT FLUX EQUATIONS

The steady-state direct and adjoint flux equations are solved iteratively using both Liebmann over-relaxation and Aitken acceleration. The fast and thermal equations are solved simultaneously. If required, the xenon concentration is calculated before every flux calculation at each mesh point. The flux is normalized to a maximum thermal flux input to the code.

After every iteration (beyond iteration KNT (q.v. Section 5.16)) in the steady-state direct flux calculation, the eigenvalue, k-eff, is calculated from:

$$\text{k-eff}\Big|_{\text{c.i.}} = \text{k-eff}\Big|_{\text{p.i.}} * \frac{\sum\limits_{\substack{\text{mesh}\\ \text{cells}}} \{\phi_1 + \phi_2\}\Big|_{\text{c.i.}}}{\sum\limits_{\substack{\text{mesh}\\ \text{cells}}} \{\phi_1 + \phi_2\}\Big|_{\text{p.i.}}} \qquad (3.4\text{-}1)$$

where "c.i." stands for "current iteration", and "p.i." stands for "previous iteration". k-eff is kept at its steady-state value for the adjoint calculation. The iterations terminate when a maximum number of iterations have been completed or when a specified convergence criterion has been met (q.v. Section 5.16).

3.5 DYNAMIC FLUX EQUATIONS

Once the steady-state direct and adjoint flux shapes have been obtained, the distribution of materials in the reactor is changed to represent the configuration after time T. The flux at this later time must then be calculated.

3.5.1 SHAPE CALCULATIONS

As a first approximation, the value of the amplitude and its derivative at the beginning of the time interval (in this case at t=0) is used in the shape calculations (Equations 2.1-14 and 2.1-15). For a very small time-step, this is a good approximation. However, for convenience and economy, it is advantageous to use larger time-steps. Hence once the flux shapes are calculated using A(0) (which is 1.0) and its derivative at t=0 (which is 0.0), the integrals for ρ, $\bar{\ell}$, $\bar{C}$, and $\bar{\beta}$ can be recalculated (q.v Section 2.1.1.1) and Equation 2.1-7 can be solved for A and its derivatives at T using DGEAR. These modified values can be used to once again iteratively solve the shape equations.

The processes can be repeated until the amplitude and shape equations are self-consistent. Then the flux at the end of the next time interval can be calculated, again using the amplitude information from time T as a first approximation.

The transient calculation is carried to later times using a similar procedure.

To improve the efficiency of the calculations, only a partially converged solution of the flux shape is used to calculate the amplitude[4]. This saves a considerable amount of computer time, since the code is not forced to completely converge on a solution based on an incorrect amplitude. This method does, however, require a larger number of amplitude calculations, due to the use of partially converged flux shapes. Nevertheless, the total computational effort is reduced. The code performs an amplitude calculation every fifteen shape iterations, or every time OCON (q.v. Section 8.18) changes by a factor of four, whichever is smaller.

3.5.2 DELAYED NEUTRON PRECURSOR CALCULATIONS

In the flux shape calculations, the shape functions are assumed to vary linearly between the initial and final values. Hence, if one considers a travelling time origin that is situated at the beginning of each time-step, the following equation results for the precursor concentration to be used in Equation 2.1-14. From Equation 2.2-2 with ϕ_2 varying linearly between $\phi_2(r,0)$ and $\phi_2(r,T)$, where T is the time-step size:

$$C_i(r,T) = \exp(-\lambda_i T) \left\{ \phi_2(r,0) \int_0^T \beta_i \frac{\nu\Sigma_{f,2}}{k\text{-eff}} \exp(\lambda_i t) A(t) dt + \right.$$

$$\left. \left[\frac{\phi_2(r,T) - \phi_2(r,0)}{T}\right] \int_0^T \beta_i \frac{\nu\Sigma_{f,2}}{k\text{-eff}} \exp(\lambda_i t) A(t) t dt + C_i(r,0) \right\} \qquad (3.5\text{-}1)$$

The integrals in Equation 3.5-1 are calculated using DGEAR at the same time that the amplitude calculations take place.

3.5.3 XENON AND IODINE CONCENTRATIONS

Xenon and iodine concentrations, when required, are calculated at the beginning of each time-step. The calculation is done using the equations developed in Section 2.5.2, with t replaced by T.

3.5.4 AMPLITUDE CALCULATIONS

In order to obtain A and $\frac{dA}{dt}$ at the end of a given time-step, and the integrals from Equation 3.5-1, the amplitude equation (Equation 2.1-7) must be accurately solved over each time-interval.

In principle, ρ, $\bar{\beta}$, $\bar{\ell}$ and $\bar{C}_i$ are all time-dependent, and Equation 2.1-7 becomes very cumbersome to solve. Instead, it is assumed that $\bar{C}_i$ and $\bar{\ell}$ are constant over a given time-step, and equal to the values at the end of the time-step. The delayed neutron fraction, β is assumed to be constant in space and time. Consequently the value of $\bar{\beta}$ remains independent of time and equal to β.

The reactivity, ρ, is assumed to vary linearly from its value at the beginning of the time-step to its value at the end of the time-step. Hence, the amplitude equation for t between the travelling origin and the end of the time-step, T, becomes:

$$\frac{dA(t)}{dt} = \frac{\left\{\left[\frac{\rho(T) - \rho(0)}{T}\right] t + \rho(0) - \overline{\beta}\right\}}{\overline{\ell}(T)} A(t) + \sum_{i=1}^{N} \lambda_i \overline{C}_i(t)/K \qquad (3.5\text{-}2)$$

3.6 BOUNDARY CONDITIONS

The division of the reactor into rectangular mesh cells introduces four external faces. In addition, the reactor, and hence the region of solution, may not extend over the entire rectangular area delineated by the mesh lines. For example, to model a cylindrical reactor, the four corner regions of the rectangular area will not be part of the region of solution, even though they are delineated by mesh lines.

The code treats internal and external boundaries differently.

3.6.1 TREATMENT OF INTERNAL BOUNDARIES

The code sets the flux equal to zero in all internal mesh cells outside the region of solution. This effectively adds a one-half mesh space extrapolation distance beyond the region of solution.

3.6.2 TREATMENT OF EXTERNAL BOUNDARIES

The external faces of the model can be either mirror image planes or external boundaries of the reactor. If an external face represents a mirror image plane, the fluxes in the boundary mesh spaces are set equal to the fluxes in the adjacent cells, and a zero neutron current situation results.

If a boundary represents the edge of the reactor, the user has three choices:

1) The flux can be set equal to zero at a distance of $0.7104\lambda_{tr}$ beyond the region of solution.

2) The flux can be set equal to zero at the edge of the reactor.

3) The flux can be set equal to zero at the centre of the border mesh cells along a given face of the rectangular model region.

The fluxes in the border mesh cells (which must be outside the region of solution) are determined by the code to yield the appropriate boundary conditions.

3.7 POWER CALCULATION

The power generated per unit volume at r and T is obtained from:

$$P(r,T) = H(r,T)A(T)\phi_2(r,T)\left\{\frac{\text{POWER}}{\int H(r,0)A(0)\phi_2(r,0)\ dV}\right\} \qquad (3.6\text{-}1)$$

$$= H(r,T)A(T)\phi_2(r,T)\{S\} \qquad (3.6\text{-}2)$$

where the integral is taken over the reactor core,

POWER is the steady-state reactor power,

H is the relative power per unit flux and volume,

H(r,0) is the steady-state relative power per unit flux and volume, and

S is the normalization factor.

If T is the time-step size, and the travelling origin is placed at the start of the time-step, an assumed linear variation in ϕ_2 yields the following energy equation:

$$E(r,T) = E(r,0) + S{*}H(r,T)\left\{\frac{\phi_2(r,T)-\phi_2(r,0)}{T}\int_0^T tA(t)dt + \phi_2(r,0)\int_0^T A(t)dt\right\} \qquad (3.6\text{-}3)$$

where E(r,T) is the energy produced at "r" per unit volume, from the start of the transient up to the present, E(r,0) is the energy produced at "r" per unit volume, from the start of the transient until the start of the present time-step (or end of the previous time-step), and the integrals are calculated by DGEAR at the same time as the amplitude is calculated in the flux routine.

In the TM-2 program, the user supplies the H values required for the power and energy calculations. The H values must be kept consistent (though not necessarily constant) throughout the transient since the normalization multiplier is calculated only for the steady-state case.

4.0 TRANSIENT MODELLING WITH TM-2

The first stage of the calculation involves two consecutive executions of TM-2 to determine the steady-state direct and adjoint fluxes. The user is then ready to extend the solution of his problem over time. The total time interval over which the transient is to be studied must first be subdivided into a number of subintervals (time-steps). The subintervals do not have to be of equal size. The program explicitly calculates the flux distribution, amplitude, delayed neutron precursor concentrations, and xenon and iodine concentrations at the end of each time-step.

The sequential executions must be carried out in chronological order. However, they need not be performed in a single job or run. In fact, it may be advisable to execute only one time-step per job. In any case, provided that the proper files at any stage (see Section 7) are catalogued or stored on magnetic tape, the solution of the transient may be stopped at any time and later continued from there. Backtracking and restarting (to improve convergence of the flux shape, for example) is also possible, provided that the proper files are available.

The length of the time-step appropriate for a given problem depends on the total duration of the transient, the severity of the perturbation, and the accuracy desired(4).

4.1 MODEL SET-UP AND SIZE

Before any flux calculations are carried out, the user must provide a complete specification of the reactor model. The configuration of the reactor will be modified during the transient as operating conditions and control rod positions are altered. These modifications must be reflected in the model geometry at the appropriate time-steps.

The model configuration is represented by appropriate neutronic cross sections being applied to the mesh cells inside the region of solution. The code requires that all mesh cells have the same dimensions, and since only one set of cross sections can be applied to a given mesh cell, the size of the mesh spacings in the x and y directions must be chosen to provide sufficient detail in the model. Further, the finite difference approximation to the spatial derivatives is more accurate with smaller mesh spaces. The benefits of detail and accuracy must be weighed against the extra computer costs due to the larger number of mesh cells, and an optimal choice must be made.

The user can supply "basic fuel type" properties to be assigned to such regions as the inner core, outer core, and reflector, for example. The influence of reactivity devices on mesh properties can be modelled by supplying incremental device properties to be added to the basic fuel type properties in the appropriate mesh cells.

The user must supply additional information required by the code in order to fully specify the problem. Energy group velocities, delayed neutron data, xenon-related data, maximum thermal flux, reactor power, diffusion coefficients and axial buckling parameters are required to set up the model. (It is important to note the xenon and delayed neutron data need not be supplied if the user wishes to simplify the problem.)

5.0 INPUT DESCRIPTION

This section consists of the input description for the TM-2 code, providing the user with the information required to carry out transient simulations using TM-2.

5.1 IDENTIFICATION CARD

Number of Cards: 1

Required: in all cases

Format: (A8, I2, 2A10)

Columns	Variable	Type	Description
1 to 8	IDENT	Alphanumeric	Eight letter case identification code.
8 to 10	NCASE	Integer	Case number identification = 1 Steady-state direct flux calculation. = 2 Steady-state adjoint flux calculation. ≥ 3 Dynamic calculation.
11 to 20	NAME (1)	Alphanumeric	User's name.
21 to 30	NAME (2)		

Notes:

1. The data from the identification card is printed on each page of output and is written to all data tapes.
2. NCASE determines the type of calculation to be carried out during the case. Dynamic cases can be numbered 3,4,5,..., as convenient.
3. If IDENT consists of 8 blank characters the program terminates execution.

5.2 TITLE CARD

Number of Cards: 1

Required: in all cases

Format: (8A10)

Columns	Variable	Type	Description
1 to 10	TITL (1)	Alphanumeric	Case title.
.	.		
.	.		
.	.		
71 to 80	TITL (8)		

Notes:

1. The case title is printed on each page of output and is written to all data tapes.

5.3 CONTROL CARD

Number of cards: 1

Required: in all cases

Format: (11I5, E15.7)

Columns	Variable	Type	Description
1 to 5	CNTRL	Integer	= 0 No flux calculation required. No flux maps will be printed. = 1 Initial run of steady state flux calculation. Start with a flat flux guess. = 2 New case - adjoint or dynamic. Take initial starting flux guess from TAPE 8. For dynamic cases obtain adjoint from TAPE 3. = 3 Restart case. Carry on with the calculation started in a previous run. = 4 No flux calculation, but a flux map from a previous run can be read from TAPE 8 and printed, and power distributions can be calculated.
6 to 10	MAPP	Integer	= 0 Allowed only if NCASE=1 and CNTRL = 1. Read in complete materials map and properties.

Columns	Variable	Type	Description
			= 1 Materials map and properties taken soley from TAPE1.
			= 2 Materials map and properties taken from TAPE1 and then modified.
11 to 15	IPOW	Integer	= 0 Perform a power calculation based on fluxes calculated during this case.
			= 1 Do not perform a power calculation.
			= 2 For CNTRL = 0 or 4, perform a power calculation based on flux on TAPE8, and flux for previous time-step (dynamic cases only) taken from TAPE10.
16 to 20	IXOP	Integer	= 1 Xenon is not calculated explicitly since the input cross sections already include the effects of xenon.
			= 2 Xenon concentrations are read from TAPE18 and maintained at those values.
			= 3 Xenon concentrations from the previous time-step are read from TAPE18 and re-calculated for a time ZD later.

Columns	Variable	Type	Description
21 to 25	PRT (1)	Integer	= 0 Print materials map. = 1 Do not print materials map or basic fuel map. = 2 Print basic fuel map and materials map.
26 to 30	PRT (2)	Integer	= 0 Print fast flux map. = 1 Do not print fast flux map.
31 to 35	PRT (3)	Integer	= 0 Print fast flux map. = 1 Do not print fast flux map.
36 to 40	PRT (4)	Integer	= 0 Print power map. = 1 Do not print power map.
41 to 45	PRT (5)	Integer	= 0 Print energy map. = 1 Do not print energy map.
46 to 50	PRT (6)	Integer	= 0 Print xenon map. = 1 Do not print xenon map.
51 to 55	PRT (7)	Integer	= 0 Print iodine map. = 1 Do not print iodine map.
56 to 70	ZD	Real	= Time step. (seconds)

Notes:

1. The flux calculation will take place at time ZD after the previous calculation.

2. Control of output maps is effected through appropriate values for the members of the PRT array.

3. If the user simply wishes to run a case that sets up the model and does not perform a flux calculation, CNTRL can be set equal to zero. If the user wishes to process some data from a previous case — specifically, print flux maps and power maps — CNTRL should be set equal to 4. If CNTR = 4, then for dynamic cases TAPE8 should contain the present flux, TAPE10 should contain the flux from the previous time step, and TAPE16 should contain the energy and power from the previous time step, if IPOW = 2.

4. If, for example, a previous run did not achieve the desired convergence level, or an error was made, one can continue the run by setting CNTRL = 3.

5. ZD should be zero for steady-state and adjoint calculations.

6. IXOP can only decrease or remain at its original value from case to case.

5.4 SCALING DATA

Number of Cards: 1

Required: IF((NCASE.EQ.1).AND.(CNTRL.EQ.1))

Format: (I5, 3E15.7)

Columns	Variable	Type	Description
1 to 5	NDG	Integer	Number of delayed neutron groups. Default is zero.
6 to 20	POWER	Real	Total steady-state thermal reactor power (MW). Default is 1.0 MW(th).
21 to 35	P2MAX	Real	Maximum steady-state thermal cell flux (n/cm^2/s). Default is 1.0 n/cm^2/s.
36 to 50	G1BYG2	Real	Initial value of fast to thermal flux ratio. Default is 0.5.

Notes:

1. These data cannot be altered during the transient.

2. POWER, P2MAX, and G1BYG2 will revert to their default values if they are ≤ 0.

3. NDG will be set equal to zero if it is negative.

4. For the initial steady-state run, the iterations begin with a flat fast flux, the thermal flux being set equal to P2MAX, and the fast flux equal to P2MAX * G1BYG2.

5.5 DIMENSION DATA

Number of Cards: 1

Required: IF (MAPP.EQ.0)

Format: (5I5, 2E15.7)

Columns	Variable	Type	Description
1 to 5	NBFT	Integer	Number of basic fuel types.
6 to 10	NDT	Integer	Number of device types.
11 to 15	ND	Integer	Number of devices.
16 to 20	NI	Integer	Number of mesh spaces in the x direction.
21 to 25	NJ	Integer	Number of mesh spaces in the y direction.
26 to 40	DX	Real	Mesh space size in the x direction (cm).
41 to 55	DY	Real	Mesh space size in the y direction (cm).

NOTES:

1. The number of mesh spaces cannot be altered during the transient.
2. The minimum number of mesh points required in any one direction is such as to properly delineate the material configuration. The user must ensure that there is a sufficient number of mesh lines to provide the accuracy required in the finite difference approximation and to properly delineate the devices that are or will be present in the core during the transient being simulated.

3. The number of mesh points specified in each dimension should include the boundary mesh points which are required for specifying the external boundary conditions and therefore cannot be within the region of solution. Therefore, the minimum number of mesh points possible in the simplest model is 9. Mesh spaces 1 and NI, in the x direction, and 1 and NJ in the y direction, lie outside the physical reactor.

5.6 MATERIALS MODIFICATION CONTROL CARD

Number of Cards: 1

Required: IF (MAPP.EQ.2)

Format: (11I5)

Columns	Variable	Type	Description
1 to 5	NUMCHNG	Integer	= 0 Do not optimize the material numbering scheme. = 1 Optimize the material numbering scheme, deleting unused materials.
6 to 10	NBFTADD	Integer	Number of new basic fuel types to be added to the model.
11 to 15	NDTADD	Integer	Number of new device types to be added to the model.
16 to 20	NDADD	Integer	Number of new devices to be added to the model.
21 to 25	NBFCHNG	Integer	= 0 Basic fuel type properties from previous case will not be altered. = 1 Basic fuel type properties from previous case will be altered.
26 to 30	NDTCHNG	Integer	= 0 Device type properties from previous case will not be altered. = 1 Device type properties from previous case will be altered.

Columns	Variable	Type	Description
31 to 35	NBMAPCH	Integer	= 0 Basic fuel type map from previous case will not be altered. = 1 Basic fuel type map from previous case will be altered.
36 to 40	NDMCHNG	Integer	= 0 Device map from previous case will not be altered. = 1 Device map from previous case will be altered.
41 to 45	NDCHNG	Integer	= 0 Diffusion coefficients from previous case will not be altered. = 1 Diffusion coefficients from previous case will be altered.
46 to 50	NBCHNG	Integer	= 0 Axial bucklings from previous case will not be altered. = 1 Axial bucklings from previous case will be altered.
51 to 55	NBOUNDC	Integer	= 0 Boundary markers from previous case will not be altered. = 1 Boundary markers from previous case will be altered.

1. The NUMCHNG option should be used with some care. While it does serve to rid the code of any unneeded material properties, it may cause a rearrangement of material numbering making it somewhat more difficult to compare results from one case to the next. For example, if one had five materials and then no longer required the material identified as "3", the NUMCHNG option would rename Material 4 to Material 3 and Material 5 to Material 4, with the properties changing correspondingly. While this means that the code no longer has to hold material properties for a Material 5, it does cause a rearrangement of the numbering that could be a nuisance if one wished to, for example, compare absorptions in a particular material over a number of cases.

5.7 BASIC FUEL TYPE MAP

Number of Cards: Case dependent

Required: IF((MAPP.EQ.0)).OR.((MAPP.EQ.2).AND.((NBFTADD.NE.0)
.OR.(NBMAPCH.NE.0))))

Format: (5I5)

Terminate this set of cards with one blank card.

Columns	Variable	Type	Description
1 to 5	M	Integer	Basic fuel type number.
6 to 10	IL	Integer	First mesh space in x direction.
11 to 15	IH	Integer	Last mesh space in x direction.
16 to 20	JL	Integer	First mesh space in y direction.
21 to 25	JH	Integer	Last mesh space in y direction.

NOTES:

1. Data on later basic fuel type map cards will overwrite previous input.
2. The entire basic fuel type map is preset to zero, or to the last previously determined values.
3. Regions for which no solution is required should be identified by zero values for the basic fuel type number.

4. $O < M \leq NBFT$

5. $1 < IL$; $IH < NI$; $IH \geq IL$

6. $1 < JL$; $JH < NJ$; $JH \geq JL$

7. Only modifications and additions to the map have to be included.

5.8 DEVICE MAP

Number of Cards: Case dependent

Required: IF((ND.GE.0).AND.(((NDADD.GT.0).OR.(NDMCHNG.GT.0))

.AND.(MAPP.EQ.2)))

Format: (6I5)

Terminate this set of cards with one blank card.

Columns	Variable	Type	Description
1 to 5	N	Integer	Device number. $(1 \le N \le ND)$
6 to 10	M	Integer	Device type. $(1 \le M \le NDT)$
11 to 15	IL	Integer	First mesh space in x direction.
16 to 20	IH	Integer	Last mesh space in x direction.
21 to 25	JL	Integer	First mesh space in y direction.
26 to 30	JH	Integer	Last mesh space in y direction.

NOTES:

1. All devices are rectangular in shape.
2. If subsequent cards with the same N value are read, the corresponding limits and device type supercede the previously read values.
3. Devices cannot be located in regions where the basic fuel type is zero.

4. Devices can overlap each other. A given mesh cell can have up to 6 devices.

5. Only modifications and additions to the map must be included.

6. If M is zero, an existing device, N, will be removed.
   In this case columns 11 to 30 can be blank.

7. If N is a device number from a previous case, and the user wishes to modify either the device type or location, it is necessary to put N equal to its previous value, M equal to the device type number, and only the changed coordinate limits must be input. For example, if only IL changes from the previous case, it is necessary to input N, M and the new value of IL, but columns 16 to 30 can be left blank.

5.9 BOUNDARY CONDITIONS AND GLOBAL MATERIAL PROPERTIES

Number of Cards: 1

Required: IF(MAPP.EQ.0).OR.((NDCHNG.NE.0).OR.(NBCHNG.NE.0)
.OR.(NBOUNDC.NE.0)))

Format: (4E15.7,5I2)

	Variable			
Columns	MAPP=0	Otherwise	Type	Description
1 to 15	D1	D1M	Real	Fast diffusion coefficient (cm).
16 to 30	D2	D2M	Real	Thermal diffusion coefficient (cm).
31 to 45	B1	B1M	Real	Fast axial buckling(cm^{-2}).
46 to 60	B2	B2M	Real	Thermal axial buckling (cm^{-2}).
61 to 62	IS	ISM	Integer	Low x-coordinate boundary marker.
63 to 64	IR	IRM	Integer	High x-coordinate boundary marker.
65 to 66	JS	JSM	Integer	Low y-coordinate boundary marker.
67 to 68	JR	JRM	Integer	High y-coordinate boundary marker.
69 to 70	KEXTR	KEXTRM	Integer	Extrapolation option.

NOTES:

1. At the outer edges of the model, the user can force the flux to zero, or reflection boundary conditions can be applied.

2. If a boundary marker is set equal to zero, an exterior boundary is assumed to exist at the particular face under consideration. The flux is accordingly set equal to zero at a distance of 0.7104 λ_{tr} beyond the edge of the last (or first) mesh line in the region of solution, or at the centre or inside edge of the boundary mesh cells, depending on the value of KEXTR.

3. If a boundary marker is set equal to 1, a mirror boundary condition (zero current) is assumed to exist at the particular face.

4. If KEXTR = 0, extrapolation distance of 0.7104 λ_{tr} will be automatically added to the faces of the reactor in the flux calculation. If KEXTR=1, the fluxes will be set equal to zero at the reactor faces. If KEXTR=2, an extrapolation distance equal to one-half the mesh size perpendicular to the boundary is assumed. The value of KEXTR only has an effect at the faces with boundary markers equal to zero.

5. If MAPP=0, all data must be initialized since the default is 0.

6. IF(NDCHNG.NE.0)THEN
   IF(D1M.GT.0) D1=D1M
   IF(D2M.GT.0) D2=D2M
   Otherwise, the values from the previous case are maintained.

7. IF(NBCHNG.NE.0)THEN
   IF(B1M.GT.0) B1=B1M
   IF(B2M.GT.0) B2=B2M
   Otherwise, the values from the previous case are maintained.

8.
```
IF (NBOUNDC.NE.0)THEN
IS = ISM
IR = IRM
JS = JSM
JR = JRM
KEXTR = KEXTRM
```

9. It is necessary to read in only the variables that change values. The exception is when NBOUNDC $\neq 0$. Then all boundary markers and the extrapolation option must be read in.

10. If, for example, one wished only to alter the value of the fast diffusion coefficient from the previous case, then, in Section 5.6, NDCHNG should be set to 1, and on this card, columns 1 to 15 should contain the new value for the coefficient. The rest of the card can be left blank.

5.10 XENON DATA

Number of Cards: 1

Required: IF(IXOP.GT.1)

Format: (6E10.3)

Columns	Variable	Type	Description
1 to 10	GI	Real	Iodine-135 yeild per fission.
11 to 20	GX	Real	Xenon-135 yield per fission.
21 to 30	LI	Real	Decay constant for Iodine-135 (s^{-1}).
31 to 40	LX	Real	Decay constant for Xenon-135 (s^{-1}).
41 to 50	NU	Real	Neutron yield per fission.
51 to 60	SIG	Real	Xenon microscopic absorption cross section for thermal neutrons(cm^2).

NOTES:

1. If TAPE1 is not available, these values will default to 0.0 if left blank or set ≤ 0.
2. If TAPE1 is available, these values will default to the TAPE1 values, if left blank or set ≤ 0.
3. Only values altered from TAPE1 are required to have non-blank fields. That is, a blank card is interpretted as meaning that the xenon data can be obtained directly from TAPE1 (or default values if no TAPE1 is available).

5.11 GROUP VELOCITIES

Number of Cards: 1

Required: IF(NCASE.EQ.1).AND.(CNTRL.EQ.1))

Format: (3X, 2E11.3)

Columns	Variable	Type	Description
1 to 3	—	—	Blank.
4 to 14	V1	Real	Fast group neutron velocity.
15 to 25	V2	Real	Thermal group neutron velocity.

NOTES:

1. Velocities are in cm/s.

2. Velocities must be greater than zero.

3. These data cannot be changed during the transient.

5.12 DELAYED NEUTRON FRACTIONS

Number of Cards: $\{ \frac{NDG}{7} \}^*$ cards.

Required: IF((NCASE.EQ.1).AND.(CNTRL.EQ.1).AND.(NDG.GT.0))

Format: (3X, 7E11.3)

Columns	Variable	Type	Description
1 to 3	—	—	Blank.
4 to 14	BETA(1)	Real	Delayed neutron fraction for group 1.
15 to 25	BETA(2)	Real	Delayed neutron fraction for group 2.
.	.		
.	.		
.	.		
70 to 80	BETA(8)	Real	Delayed neutron fraction for group 8.

NOTES:

1. If necessary, subsequent cards with BETA(9) to BETA(16), etc., can be input.

2. These data cannot be altered as the transient proceeds.

3. Delayed neutron data is not allowed to vary with position in the reactor.

* { X } is here defined as identical to X if X is an integer, otherwise the value of X is rounded up to the next highest integer.

5.13 DELAYED NEUTRON DECAY CONSTANTS

Number of Cards: $\{ \frac{NDG}{7} \}^*$ cards.

Required: IF((NCASE.EQ.1).AND.(CNTRL.EQ.1).AND.(NDG.GT.0))

Format: (3X, 7E11.3)

Columns	Variable	Type	Description
1 to 3	—	—	Blank.
4 to 14	LAMDA(1)	Real	Decay constant for group 1 (s^{-1}).
15 to 25	LAMDA(2)	Real	Decay constant for group 2 (s^{-1}).
.	.		
.	.		
.	.		
70 to 80	LAMDA(8)	Real	Decay constant for group 8 (s^{-1}).

NOTES:

1. If necessary, subsequent cards with LAMDA(9) to LAMDA(16), etc., can be input.
2. These data cannot be altered as the transient proceeds.
3. Delayed neutron data is not allowed to vary with position in the reactor.

* { X } is here defined as identical X if X is an integer, otherwise the value of X is rounded up to the next highest integer.

5.14 BASIC FUEL TYPE PROPERTIES

Number of Cards: Case dependent

Required: IF((MAPP.EQ.0).OR.(MAPP.EQ.2).AND.((NBFTADD.GT.0)
.OR.(NBFCHNG.GT.0))))

Format: (I5, E15.7)

Terminate this set of cards with one blank card.

Columns	Variable	Type	Description
1 to 5	I	Integer	Basic fuel type.(1≤I≤NBFT)
6 to 20	BPROP(I,1)	Real	Fast absorption cross section.
21 to 35	BPROP(I,2)	Real	Thermal absorption cross section.
36 to 50	BPROP(I,3)	Real	Moderation cross section.
51 to 65	BPROP(I,4)	Real	Production cross section.
66 to 80	BPROP(I,5)	Real	Relative H factor. (power to thermal cell flux ratio)

NOTES:

1. Cross sections should be in units of cm^{-1}.

2. If blank or zero, properties will take on their values from the previous case, or will remain zero if no TAPE1 is available. Hence, only additions and modifications need to be included.

3. Basic fuel type numbers must correspond to the numbers used in the basic fuel type map.

4. If xenon is treated explicitly by the code (IXOP≠0), BPROP(I,2) should not contain any component due to xenon.

5.15 DEVICE TYPE INCREMENTAL PROPERTIES

Number of Cards: Case dependent

Required: IF((NDT.GT.0).AND.((MAPP.EQ.0).OR.((MAPP.EQ.2)
.AND.((NDTADD.NE.0).OR.(NDADD.NE.0)))))

Format: (I5, 5E15.7)

Terminate this set of cards with one blank card.

Columns	Variable	Type	Description
1 to 5	I	Integer	Device type. $(1 \leq I \leq NDT)$
6 to 20	DPROP(I,1)	Real	Incremental fast absorption cross section.
21 to 35	DPROP(I,2)	Real	Incremental thermal absorption cross section.
36 to 50	DPROP(I,3)	Real	Incremental moderation cross section.
51 to 65	DPROP(I,4)	Real	Incremental production cross section.
66 to 80	DPROP(I,5)	Real	Incremental H factor.

NOTES:

1. Incremental cross sections should be in units of cm^{-1}, and will be added to the basic fuel type properties in the appropriate mesh cells.
2. If blank, or zero, properties will take on their values from the previous case, or will remain zero if no TAPE1 is available. Hence, only additions and modifications need to be included.

3. Device type numbers must correspond to the numbers in the device map.

4. If xenon is treated explicitly by the code ($IXOP\neq 0$), DPROP(I,2) should not contain any component due to xenon.

5.16 ITERATION-RELATED DATA

Number of Cards: 1

Required: IF((CNTRL.NE.4).AND.(CNTRL.NE.0))

Format: (3I5, 3E15.7)

Columns	Variable	Type	Description
1 to 5	LIM	Integer	Maximum number of flux-shape iterations allowed.
6 to 10	KNT	Integer	Number of shape iterations at start of calculation over which k-eff is not allowed to depart from the initially assumed value. KNT should not be smaller than 1.
11 to 15	IOP	Integer	= 0 Aitken acceleration is not used in the steady-state or ajoint calculation. = 1 Aitken acceleration is used in the steady state or adjoint calculation.
16 to 30	KE	Real	Initial value of k-eff for use in steady-state direct flux calculation.
31 to 45	ALFA	Real	Liebmann over-relaxation parameter (1 ALFA<2).
46 to 60	EPS	Real	Convergence criterion.

NOTES:

1. Convergence is attained when at each mesh point in the region of solution:

$$\left| \frac{\phi_1 \text{ (present iteration) } + \phi_2 \text{ (present iteration)}}{\varphi_1 \text{ (previous iteration) } + \phi_2 \text{ (previous iteration)}} -1.0 \right| \le \text{EPS}$$

2. Default values for all iteration-related parameters are those from the previous case and shall be selected if an input parameter is 0. If values from a previous case are not available the following default values are used:

LIM = 250

KNT = 5

IOP = 1

KE = 1.0

ALFA = 1.4

EPS = 1.E-6

Hence only modifications need to be included.

3. KE is used as a first guess for the eigenvalue in the steady-state direct flux iterations only. It should be left blank for adjoint and dynamic cases.

5.17 DATA TERMINATION CARD

Either input a subsequent data set (q.v. Section 5.1 to 5.16) (another TM-2 case) or put in one blank card to terminate program execution.

6.0 MEMORY REQUIREMENTS

The TM-2 program makes extensive use of variably-dimensioned arrays and as such, the user can make optimal use of the central memory, storing arrays that are only as large as necessary. While dynamic allocation of memory is not a user-available feature of the NOS system, for which this code was designed to operate under, the placement of all variably-dimensioned arrays into an unlabelled (or blank) common block (with a nominal length of 4000 decimal words) allows for simple modifications to increase the maximum model size. Only two lines of programming need to be altered if the length of the blank common area is insufficient.

The use of variably-dimensional arrays allows the user to construct a model of a maximum size such that:

6 + A + B + 8*IJ + 5*NBFT + 6*NMAT + 3*NDG + ND + NDT*5 ≤ 4000

where 4000 is the length of the unlabelled common block,

A is the maximum of (NMAT *2 + (19+N)*N) or (14*NMAT),

B is zero if IXOP = 1 (q.v. Section 5.3),

B is 2*IJ if IXOP≠1,

IJ is (NI*NJ) where NI and NJ are the number of mesh spaces in the x and y directions, respectively,

NMAT is the number of materials in the model,

NDG is the number of delayed neutron groups considered,

NBFT is the number of basic fuel types,

ND is the number of devices in the model,

NDT is the number of device types in the model, and

N is 3 + 3*NDG.

The TM-2 program is divided into six overlays:

(0,0): Main control and calling overlay,

(1,0): Overlay to produce materials map,

(2,0): Overlay to process materials properties,

(3,0): Overlay to calculate steady-state direct and adjoint fluxes,

(4,0): Overlay to calculate dynamic fluxes, and

(5,0): Overlay to calculate power distribution.

The use of overlays minimizes the amount of central memory required for the logistics of the program. This then leaves more central memory for array storage. The unlabelled common block is stored in the main overlay and as such, remains in central memory during the execution of all primary overlays.

7.0 FILE MANIPULATION

In addition to an input file (TAPE5) and an output file (TAPE6), the TM-2 code makes use of several other local files. Several of these files must be retained in order to proceed to the next point in the transient calculations. The TM-2 program stores the model, the flux shape, the power shape, the precursor concentrations, and xenon and iodine concentrations on separate files to allow the user to mix and match files or provide user-specified data. Not all files have to be catalogued (stored on tape or as permanent files). All data files are assumed to be positioned properly at the start of a TM-2 run, and all files (except TAPE5 and TAPE6) are rewound at the end of each run. The following additional files are used in TM-2:

TAPE1: Holds model data including materials map and properties. This tape is read in a steady-state direct flux restart case or in any adjoint or dynamic case and is written in all cases (after any modifications are made).

TAPE2: In a multicase run, TAPE2 will contain the series of TAPE1s created in the run. That is, while TAPE1 is rewound after each case, TAPE2 is not.

TAPE3: For dynamic flux calculations, the user must attach a TAPE3 which contains the steady-state adjoint flux. TAPE3 is written by the code only if NCASE=2 (q.v. Section 5.1).

TAPE8: This tape is read to obtain a starting guess for the flux calculation, if required. The flux at the end of the iterations is written to TAPE8 (except if NCASE=2).

TAPE9: In a multicase run, TAPE9 will contain the series of TAPE8s created in the run. That is, while TAPE8 is rewound after each case, TAPE9 is not.

TAPE10: In a restart of a dynamic case (to improve convergence of the flux shape for example), TAPE8 should contain the almost converged flux from which the iterations are continued, and TAPE10 should contain the flux shape from the previous time-step. For steady-state direct and adjoint cases, TAPE10 is not required.

TAPE11: This file is read to obtain precursor densities from the previous time-step (even in a restart). After the flux iterations, the precursor data is written to TAPE11.

TAPE12: In a multicase run, TAPE12 will contain the series of TAPE11s created in the run. That is, while TAPE11 is rewound after each case, TAPE12 is not.

TAPE13: This is a scratch file for precursor information and should not be catalogued.

TAPE16: This file is read to obtain power and energy distributions from the previous time-step (even in a restart). After the power calculations, the new power distributions are written to TAPE16.

TAPE17: In a multicase run, TAPE17 will contain the series of TAPE16s created in the run. That is, while TAPE16 is rewound after each case, TAPE17 is not.

TAPE18: This file is read to obtain xenon and iodine distributions from the previous time-step, except in a restart case, when TAPE18 should contain the xenon and iodine distributions for the present case. TAPE18 is written once the new xenon distributions are calculated.

TAPE19: In a multicase run, TAPE19 will contain the series of TAPE18s created in the run. That is, while TAPE18 is rewound after each case, TAPE19 is not.

Note that TAPE2, TAPE9, TAPE12, TAPE17, and TAPE19 are provided as safety features to protect information that might otherwise be lost if an error occurs part of the way through a transient. These tapes are not usually attached or catalogued. Only special circumstances will warrant their use.

7.1 NORMAL CATALOGUING SEQUENCE

To start a steady-state direct flux calculation, no files must be attached. TAPE1 , TAPE16 and TAPE18 should be catalogued. (If various options such as implicit xenon or no delayed neutrons are chosen, the corresponding data tapes need not be catalogued. This applies to other cases as well.)

If convergence is satisfactory, an adjoint calculation comes next. The steady-state files: TAPE1, TAPE8, and TAPE18, should be attached from the steady-state case. After the adjoint case has been run, TAPE3 and TAPE11 should be catalogued.

To start the first dynamic flux calculation, TAPE1, TAPE3, TAPE8, TAPE11, TAPE16 and TAPE18 should be attached. All except TAPE3 and TAPE11 should come from the steady-state direct flux case. The materials map and properties are then adjusted as required and the first dynamic case is done. The newly written TAPE1, TAPE8, TAPE11, TAPE16, and TAPE18 should be catalogued.

To perform calculations at subsequent time steps, TAPE1, TAPE8, TAPE11, TAPE16 and TAPE18 should be attached from the previous time-step. TAPE3 from the adjoint calculation must also be attached. The newly written TAPE1, TAPE8, TAPE11, TAPE16, and TAPE18 should be catalogued after the run.

7.2 RESTART CASES

7.2.1 RESTART OF THE STEADY-STATE FLUX CALCULATION

To restart the steady-state calculation, TAPE1 and TAPE8 should be attached from the first run. TAPE1, TAPE8, TAPE16, and TAPE18 should be catalogued.

7.2.2 RESTART OF THE ADJOINT FLUX CALCULATION

To restart the adjoint calculation, the unconverged adjoint flux from the first run should be attached as TAPE8. TAPE1, TAPE11 and TAPE16 should be attached from the steady-state case. TAPE8 from the steady-state case should be attached as TAPE10. TAPE3 and TAPE11 should be catalogued. A restart of the adjoint flux calculation must be the last case in a particular run.

7.2.3 RESTART OF A DYNAMIC FLUX CALCULATION

To restart a dynamic flux calculation, the flux and power distribution from the previous time-step should be attached as TAPE10 and TAPE16 respectively. The precursor information from the previous time-step should be attached as TAPE11. The modified materials map and properties from the first run of the case should be attached as TAPE1. The adjoint flux should be attached as TAPE3 and the unconverged flux, and xenon data should be attached as TAPE8, and TAPE18 respectively. After the calculations, the new TAPE1 (if modified), TAPE8, TAPE11, and TAPE16 and TAPE18 should be catalogued.

8.0 OUTPUT DESCRIPTION

The output generated by TM-2 is organized into sections. These are now described in sequential order. Abnormal termination of TM-2 executions is described in Section 9.

8.1 TITLE PAGE

The first page of each case is a title page giving the name of the program and its version identifier. The case title, user name and case identifiers are printed on each page of output. Each page of output also has the page number, date, clock-time, run time (CPU seconds) and the number of system resource units (SRUs) used and remaining to be used in the job.

8.2 CONTROL CARD DATA

The contents of the control card (q.v. Section 5.3) are printed. The variable names correspond to those from Section 5.3.

8.3 MODEL SIZE AND GEOMETRY

The number of basic fuel types, number of device types, and number of devices in the model are printed. The mesh space sizes in the x and y directions are printed. The units are assumed to be centimeters. This is followed by the number of meshes in the x and y directions in the model (including boundary meshes outside the region of solution).

8.4 BASIC FUEL MAP MODIFICATIONS

Any modifications to the basic fuel map are printed as they are read. There are five columns given the designations BFT, IL, IH, JL, and JH to represent the basic fuel type number, starting mesh space in the x direction, final mesh space in the x direction, and initial and final y mesh spaces respectively. The "overlay" method of respecification is employed.

8.5 DEVICE MAP MODIFICATIONS

Any modifications to the device map are printed as read. The six columns of numbers identified as DN, DT, IL, IH, JL, and JH correspond respectively to the device number, device type, initial and final mesh spaces in the x direction, and initial and final mesh spaces in the y direction. All devices are rectangular in shape.

8.6 FINAL DEVICE LOCATIONS

Any modifications to the device map (Section 8.5) are combined with the existing device map and the final locations of all devices are printed using the same six columns as in Section 8.5.

8.7 BASIC FUEL MAP

If PRT(1)=2, a basic fuel map is printed. The map is a rectangular array of basic fuel type numbers. The x direction extends from left to right across the page. The y direction extends from the top to the bottom of the page. The axes are identified with the mesh space numbers. Since mesh spaces 1 and NI in the x direction and mesh spaces 1 and NJ in the y direction are outside the region of solution (basic fuel type numbers are zero), they are not included in the map.

8.8 MATERIALS MAP

If PRT(1)≠1, a materials map is printed. The code combines basic fuel types with the corresponding devices as is necessary to specify the full model configuration. Each distinct combination is given a unique material number. These numbers are printed in the materials map. The particular combination of basic fuel type and devices for a given material number are printed later in the output (q.v. Section 8.16).

8.9 MEMORY REQUIREMENTS

The number of words of unlabelled (blank) common required and available are printed.

8.10 EXTERNAL BOUNDARY MARKERS

IS, IR, JS, JR and KEXTR from Section 5.9 are printed beside the respective titles: "LOW X" "HIGH X", "LOW Y", "HIGH Y", and "EXTRAPOLATION OPTION".

8.11 GLOBAL MATERIAL PROPERTIES

The fast and thermal diffusion coefficients and axial bucklings are printed. Diffusion coefficient units are cm. Units of buckling are cm^{-2}.

8.12 XENON DATA

The xenon-related data from Section 5.10 are printed. Units are as in Section 5.10.

8.13 DELAYED NEUTRON DATA

The number of delayed neutron groups, the fast and thermal neutron velocities (cm/s) and the total delayed netron fractions are printed. This is followed by a list of the specific delayed group numbers, fractions, and decay constants (s^{-1}). These data are spatially and temporally invariant.

8.14 BASIC FUEL TYPE PROPERTIES

The basic fuel type properties input in Section 5.14 are printed with appropriate headings.

8.15 INCREMENTAL DEVICE TYPE PROPERTIES

The device type incremental properties input in Section 5.15 are printed with appropriate headings.

8.16 MATERIAL PROPERTIES

All unique basic fuel type-device combinations required in the model specification are printed. Under the heading "MAT", are the material numbers (the map corresponding to these material numbers was described in Section 8.8). Under the heading "BFT", are the basic fuel types corresponding to the particular material numbers. Under the heading "COMPONENT DEVICE TYPES" are the device type numbers (up to a maximum of six) that are components of the particular material numbers. The material properties are also printed under appropriate headings. All properties except the "H-FACTORS" are in units of cm^{-1}. The H-FACTORS are relative numbers (q.v. Section 5.14). The thermal absorption cross sections printed do not include a component due to xenon if $IXOP \neq 1$.

8.17 ITERATION-RELATED DATA

The values of the parameters controlling the iterations are printed as input in Section 5.16.

8.18 STEADY-STATE DIRECT AND ADJOINT FLUX CALCULATIONS

If NCASE=1 or 2, the steady-state direct or adjoint flux iteration history is printed. The iteration, eigenvalue, convergence level, and fast and thermal fluxes at the central mesh point are printed for each iteration under the respective headings: "ITERATION NUMBER", "KE", "OCON", "FAST FLUX", and "SLOW FLUX". OCON is defined as:

$$\frac{1}{\mathrm{TOT}} \sum_{i=1}^{\mathrm{TOT}} \left| \frac{\phi_1(\text{present iteration}) + \phi_2(\text{present iteration})}{\phi_1(\text{previous iteration}) + \phi_2(\text{previous iteration})} - 1.0 \right|$$

where TOT is the number of mesh cells in the region of solution and the sum is taken over all mesh cells in the region of solution.

The iterative calculation will stop for one of three reasons:

1) Covergence of the flux shape has been attained (q.v. Section 5.16).

2) The maximum number of iterations specified in Section 5.16 has been reached.

3) The SRU count indicates that insufficient resources are remaining to carry out further iterations. The printing of maps and integrals and the writing of data tapes will take priority.

If the calculation ends due to reasons 2) or 3) above, a warning message to that effect is printed.

At the end of the iterations the maximum thermal and fast fluxes and their mesh locations are printed. Throughout this section, the units of fluxes are neutrons/cm^2/s.

The SRU count is given before the start of the iterations, and the SRUs expended during the calculation are printed at the end of the iterations.

8.19 DYNAMIC FLUX CALCULATION

If NCASE $\geq$3, the dynamic flux iteration history is printed. The initial value of the amplitude and the steady-state eigenvalue are printed. The iteration number, convergence level, flux integral ratio, and fast and thermal flux shape values (in n/cm^2/s) at the central mesh point are printed for each iteration under the respective headings: "ITERATION", "OCON", "ARMN", "PF", and "PS". ARMN is the ratio of the integral in Equation 2.1-6 calculated at the steady-state to its value calculated at this particular iteration.

After each amplitude calculation the values of the amplitude, reactivity, neutron generation time ($\bar{\ell}$) and time sub-interval chosen by DGEAR are printed. IER, an error flag from DGEAR, is also printed. If IER >132, the amplitude calculation could not be performed correctly. Under normal circumstances this will not occur, but if it does, the user should consult Reference 6.

The iterations will stop if any of the following criteria are met:

1) Convergence of the flux shape, amplitude and reactivity has been attained.
2) The maximum number of iterations specified in Section 5.16 has been reached.
3) The SRU count indicates that there is insufficient resources remaining to properly terminate the case.

If the calculation ends due to items 2) or 3) above, a warning message to that effect is printed.

At the end of the iterations, the maximum thermal and fast flux shape values and their mesh locations are printed. The units of the fluxes are neutrons/cm^2/s. The SRU count is given before the start of the iterations, and the SRUs expended during the calculation are printed at the end of the iterations.

8.20 THERMAL FLUX SHAPE MAP

If PRT(2)=0, the thermal neutron flux shape distribution is printed. The units are neutrons/cm^2/s. To facilitate ease of interpretation, the fluxes are multiplied by a scaling factor prior to being printed. This scaling factor is printed at the top of the map.

8.21 FAST FLUX SHAPE MAP

If PRT(3)=0, the fast neutron flux shape distribution is printed. The units are neutrons/cm^2/s. To facilitate ease of interpretation, the fluxes are multiplied by a scaling factor prior to being printed. This scaling factor is printed at the top of the map.

8.22 XENON MAP

If PRT(6)=0, the xenon distribution is printed. The units are atoms/cm^3. The numbers in the map are multiplied by a scaling factor which is printed at the top of the map.

8.23 IODINE MAP

If PRT(7)=0, the iodine distribution is printed. The units are atoms/cm^3. The numbers in the map are multiplied by a scaling factor which is printed at the top of the map.

8.24 FLUX-WEIGHTED INTEGRALS

For each material and the reactor as a whole, flux-weighted reaction rates are printed. They are (in order),

$\int \Sigma_{a,1}\Phi_1 dV$, $\int \Sigma_{a,2}\Phi_2 dV$, $\int \nabla \cdot D_1 \nabla \Phi_1 dV$, $\int \nabla \cdot D_2 \nabla \Phi_2 dV$, $\int \Sigma_m \Phi_1 dV$, $\int \nu \Sigma_{f,2}\Phi_2 dV$, and $\int dV$.

The average fast and thermal flux values for each material and the reactor as a whole are also printed. The value and location of the maximum fast and thermal fluxes in the model are printed. The integrals are printed for all except the adjoint case.

8.25 DIRECT AND ADJOINT FLUX-WEIGHTED INTEGRALS

Direct and adjoint flux-weighted integrals are printed after the flux calculation, for all but the steady-state direct flux case. The following integrals are printed for each material type and the reactor as a whole:

$\int \phi_1^* \Sigma_{a,1} \Phi_1 dV$, $\int \phi_2^* \Sigma_{a,2} \Phi_2 dV$, $\int \phi_1^* \nabla \cdot D_1 \nabla \Phi_1 dV$, $\int \phi_2^* \nabla \cdot D_2 \nabla \Phi_2 dV$,

$\int \phi_2^* \Sigma_m \Phi_1 dV$, $\int \phi_1^* \nu \Sigma_{f,2} \Phi_2 dV$.

Below each table, the integrals summed over all materials are also printed. If delayed neutrons are being treated explicitly, for each group, $\lambda_i \int C_i \ \phi_1^* dV$ is printed under the title "ADJOINT WEIGHTED INTEGRATED PRECURSOR DENSITIES".

In the adjoint case, the constant, K, in Equation 2.1-6, which is known internally to the code as ATFO is printed.

8.26 POWER-RELATED DATA

The steady-state power, the normalization multiplier (S from Equation 3.6-2) and the amplitude are printed. The maximum cell power and its location in terms of mesh spaces is printed. The total power and average power in the power-producing mesh cells and the radial form factor (RFF) are then printed. The RFF is defined as:

$$\mathrm{RFF} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{POWER(i)} \ / \ \mathrm{MAXPOW}$$

where POWER(i) is the power in mesh space "i", N is the number of power-producing mesh cells, MAXPOW is the maximum mesh cell power in the reactor, and the summation is taken over all power-producing mesh cells.

8.27 POWER MAP

If PRT(4)=0, the power distribution is printed. The units are MW(th)/mesh cell. The numbers in the map are multiplied by a scaling factor which is printed at the top of the map.

8.28 ENERGY MAP

If PRT(5)=0, the energy distribution is printed. The units are MJ(th)/mesh cell. The numbers in the map are multiplied by a scaling factor which is printed at the top of the map.

8.29 DATA TAPE READS AND WRITES

Any time through the calculation, if a data tape is read or written, a message to that effect is printed on the output file.

8.30 FINAL PAGE

The final page of any run is similar to the first page, though this time with the final resource expenditures and the words "NORMAL TERMINATION OF TM-2 RUN".

9.0 PROGRAMMED ERROR STOPS

The TM-2 code has 18 programmed error stops in addition to the normal FORTRAN error stops. These mainly represent situations in which inconsistencies in the input or errors in the calculations have been encountered and it is not safe to allow execution to continue.

The error stops cause program execution to end following the printing of an error number and keyword.

9.1 LIST OF PROGRAMMED ERROR STOPS

ERROR NUMBER	LOCATION	KEYWORD	EXPLANATION
1	TMTWO	MAPP=0	NCASE≠1 and MAPP=0
2	MATMAP1	BFT MAP	In Section 5.7, M<0 or M>NBFT
3	MATMAP1	BFT LOCATION	In Section 4.7, IH<IL, or JH<JL.
4	TMTWO	MODEL OVFL	Preliminary check on model size indicates insufficient blank common area.
5	MATMAP1	DEV NUM	In Section 5.8, N>ND, or M>NDT.
6	MARMAP1	DEV TP NUM	In Section 5.8, M<0, or M>NDT.
7	MATMAP1	DEV MAP	In Section 5.8, IL<2, or IH>IL, or JL<2, or JH>JJ, or JL>JH or IL>IH.
8	NAARRAY	MODEL OVFL	Check on model size indicates insufficient blank common area.
9	NAARRAY	DEV OVERFL	More than six devices were placed in one mesh cell.
10	NAARRAY	MODEL OVFL	As for Error Number 8.
11	TMTWO	MODEL OVFL	As for Error Number 8.
12	MATMP	MODEL OVFL	As for Error Number 8.
13	MATPRP1	LAMDA NEG	LAMDA(I) <0
14	MATPRP1	BETA NEG	BETA(I) <0

ERROR NUMBER	LOCATION	KEYWORD	EXPLANATION
15	MATPRP1	BFT NUMBER	In Section 5.14, I<0, or I>NBFT.
16	MATPRP1	DEV TP NUM	In Section 5.15, I<0, or I>NDT.
17	FLUX3	DGEAR IER	Error in DGEAR subroutine. See Reference 6.
18	WRITPOW	WRITPOW	Tape from which powers are read has a different number of mesh spacings than expected.

10.0 NOMENCLATURE

10.1 VARIABLES

A	-	Amplitude component of flux.
B^2	-	Axial Buckling.
C	-	Precursor concentration.
$\bar{C}$	-	Weighted precursor concentration.
D	-	Diffusion coefficient.
H	-	Relative power per unit volume per unit flux.
I	-	Iodine Concentration
k-eff-		Effective multiplication constant (steady-state eigenvalue).
$\bar{\ell}$	-	Weighted neutron generation time.
N	-	Number of delayed neutron groups.
r	-	Space variable.
S	-	Power normalization factor.
T	-	Time interval.
t	-	Time variable.
V	-	Volume.
Δx	-	Mesh space size in the x direction.
Δy	-	Mesh space size in the y direction.
X	-	Xenon concentration.
β	-	Delayed neutron fraction.
λ	-	Decay constant or neutron mean free path.
ν	-	Neutrons born per fission.

ρ - Reactivity.

σ - Xenon microscopic absorption cross section.

Σ - Neutron macroscopic cross section.

υ - Neutron velocity.

Φ - Neutron flux.

ϕ - Flux shape.

10.2 SUBSCRIPTS

a - Absorption.

f - Fission.

I - Iodine.

i - Group index.

m - Moderator.

tr - Transport.

X - Xenon.

1 - Fast neutron energy group.

2 - Thermal neutron energy group.

11.0 REFERENCES

1. Henry, A.F., *Nuclear Reactor Analysis*. Cambridge: The MIT Press, 1980.

2. Ott, K.O. and Meneley, D.A., "Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics", *Nuclear Science and Engineering, 36*. pp. 402-411, 1969.

3. Dastur, A.R., and Buss, D.B., *Space-Time Kinetics of CANDU Systems*, AECL-5181. Mississauga: Atomic Energy of Canada Limited, 1975.

4. Dastur, A.R., Private Communication.

5. Henry, A.F., "The Application of Reactor Kinetics to the Analysis of Experiments", *Nuclear Science and Engineering, 36*. pp. 52-70, 1958.

6. *IMSL Library Reference Manual*, IMSL LIB-0008. Houston: IMSL Inc., 1980.