SPACE-DEPENDENT NUCLEAR REACTOR TRANSIENT ANALYSES BY MULTIPLE TEMPORAL-MODE TRANSFORMATION

by

KELVIN F.T. TSE, B. Eng.

PART A: OFF-CAMPUS PROJECT*

A Report Submitted to the School of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Engineering

Department of Engineering Physics
McMaster University
Hamilton, Ontario, Canada
October 1975

*One of two project reports: The other part is designated PART B: MCMASTER (ON-CAMPUS) PROJECT.

MASTER OF ENGINEERING (1975) MCMASTER UNIVERSITY

Department of Engineering Physics Hamilton, Ontario

TITLE: Space-Dependent Nuclear Reactor Transient Analyses by Multiple Temporal-Mode Transformation

AUTHOR: Kelvin F.T. Tse, B. Eng. (McMaster University)

SUPERVISOR: Dr. F.N. McDonnell

NUMBER OF PAGES: vi, 107

ABSTRACT

A multiple temporal-mode transformation is applied to the finite difference form of the space-dependent reactor kinetics equations with the aim of reducing the truncation error. The transformation method is incorporated into an existing alternating-direction explicit code and is tested on three homogeneous problems in one and two dimensions as well as on a one-dimensional space-dependent CANDU-type problem. The numerical results have confirmed some characteristics of this solution formalism with respect to accuracy and stability. In addition, this study has identified some areas for improvement to the multiple temporal-mode transformation technique.

ACKNOWLEDGEMENTS

I wish to thank my supervisor, Dr. Frank McDonnell, for his valuable guidance and advice during the course of this project. I also wish to thank Andre Baudouin for his helpful suggestions on many occasions and for his assistance in some of the programming work. The Atomic Energy of Canada Limited through its Industrial Internship Program has been most generous for making this project possible. Finally, I would like to thank Mrs. Sharon Baker for the typing of the report.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Purpose Of The Study

Nuclear reactor transient analysis forms an integral part of safety studies related to fast or short term effects in a nuclear reactor. Fast transients may result from temperature changes, prescribed reactivity changes introduced by control rod motion, or accidental changes such as those induced by a loss of coolant. Safety analyses of large nuclear reactors require the accurate prediction of the neutron flux as a function of position and time. In past years, much effort has gone into developing solution techniques for solving the time-dependent multigroup reactor kinetics equations in one or more spatial dimensions. The so-called indirect methods, which include space-time synthesis[1,2] and quasistatic methods[3], have been applied successfully to several types of reactor problems and can have small computing time requirements. However, these indirect methods suffer from a lack of a definite error estimator. The direct finite differencemethods[4,5,6] can be applied in a simple manner to obtain an accurate solution of the space-time kinetics equations and have fairly definite error bounds. These methods generally require large amounts of computer time for multidimensional calculations. Recently, it has been demonstrated by Ferguson[6] that one class of finite difference methods, the alternating-direction explicit method, is capable of solving some types of three-dimensional calculations at reasonable cost. An exponential transformation of the flux is employed in this method to reduce the truncation error and this allows the use of larger time steps to advance the solution.

An alternative form of frequency transformation, the multiple temporal-mode transformation, has been applied by Garland et al[7]

to a two-group two-dimensional analysis of a CANDU-BLW nuclear reactor cell, and by Harms et al[8] to a two-group three-dimensional analysis of a bare cylindrical reactor, in both cases, using the alternating-direction implicit method. The purpose of this project is to investigate the applicability of the multiple temporal-mode transformation with an alternating-direction explicit method to transient flux calculations of some CANDU type problems in one and two spatial dimensions. A comparison of the results with those obtained by an existing code at CRNL, ADEP[9], will also be given.

1.2 The Space-Dependent Reactor Kinetics Equations

We consider the following reactor kinetics equations written in multigroup diffusion approximation form[4]:

$$\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\bar{r},t) = \bar{\nabla}\cdot D_g(\bar{r},t)\bar{\nabla}\phi_g(\bar{r},t) + \sum_{g'=1}^{G}\Sigma_{gg'}(\bar{r},t)\phi_{g'}(\bar{r},t) + \sum_{i=1}^{I} f_{gi}C_i(\bar{r},t) \qquad (1\leq g\leq G) \tag{1.1}$$

$$\frac{\partial}{\partial t}C_i(\bar{r},t) = -\lambda_i C_i(\bar{r},t) + \sum_{g'=1}^{G} p_{ig'}(\bar{r},t)\phi_{g'}(\bar{r},t) \qquad (1\leq i\leq I)$$

Parameters appearing in the above equations have the following meanings:

- g = index number of the energy group
- i = index number of the delayed neutron precursor group
- G = total number of energy groups
- I = total number of delayed neutron precursor groups
- ϕ_g = scalar neutron flux [n/(cm^2-sec)] in energy group g
- C_i = concentration (cm^{-3}) of i^{th} precursor
- D_g = diffusion coefficient (cm) for neutrons in energy group g

v_g = speed (cm/sec) of the neutrons in energy group g

$\Sigma_{gg'}$ = intergroup macroscopic transfer cross section (cm^{-1}) from group g' to group g with the following structure:

$\Sigma_{gg} = \chi_g(1-\beta)\nu_g\Sigma_{fg} - \Sigma_{ag} - \sum_{g' \neq g} \Sigma_{g \to g'}$

χ_g = fission spectrum yield in group g

ν_g = average number of neutrons per fission in group g

Σ_{fg} = macroscopic fission cross section (cm^{-1}) in group g

Σ_{ag} = macroscopic absorption cross section (cm^{-1}) in group g

$\Sigma_{g \to g'}$ = macroscopic scattering cross section from group g to group g'

β = total fractional yield of delayed neutrons per fission

$\Sigma_{gg'} = \chi_g(1-\beta)\nu_{g'}\Sigma_{fg'} + \Sigma_{g' \to g}$

$f_{gi} = \lambda_i \chi_{gi}$ = probability (sec^{-1}) that the i^{th} precursor will yield a neutron in group g where λ_i is the decay constant and χ_{gi} the fraction of decays in delayed group i which yields neutrons in group g

$p_{ig'} = \beta_i \nu_{g'} \Sigma_{fg'}$ = production factor (cm^{-1}) for the i^{th} precursor having fractional yield β_i by fissions in group g'

Boundary conditions for Equations (1.1) will be of the homogeneous Neumann or Dirichlet type. At internal interfaces, continuity of the flux and the normal component of the neutron current, $\bar{n} \cdot D\bar{\nabla}\phi$, will be required. The initial flux and precursor distributions in space and energy must be specified.

For convenience, the G+I equations, Equations (1.1), can be represented in the following matrix form:

$$\frac{d}{dt}\bar{\theta}(\bar{r},t) = \underline{M}(\bar{r},t)\bar{\theta}(\bar{r},t) \qquad (1.2)$$

where

$$\bar{\theta}(\bar{r},t) = \begin{bmatrix} \phi_1(\bar{r},t) \\ \phi_2(\bar{r},t) \\ \vdots \\ \phi_G(\bar{r},t) \\ C_1(\bar{r},t) \\ \vdots \\ C_I(\bar{r},t) \end{bmatrix} \qquad (1.3)$$

and

$$\underline{M}(\bar{r},t) = \left[\begin{array}{cccc|ccc} v_1(\bar{\nabla}\cdot D_1\bar{\nabla}+\Sigma_{11}) & v_1\Sigma_{12} & \cdots & v_1\Sigma_{1G} & v_1f_{11} & \cdots & v_1f_{1I} \\ v_2\Sigma_{21} & v_2(\bar{\nabla}\cdot D_2\bar{\nabla}+\Sigma_{22}) & \cdots & v_2\Sigma_{2G} & v_2f_{21} & \cdots & v_2f_{2I} \\ & \cdots & & & & & \\ v_G\Sigma_{G1} & v_G\Sigma_{G2} & \cdots & v_G(\bar{\nabla}\cdot D_G\bar{\nabla}+\Sigma_{GG}) & v_Gf_{G1} & \cdots & v_Gf_{GI} \\ \hline P_{11} & P_{12} & \cdots & P_{1G} & -\lambda_1 & & 0 \\ & \cdots & & & & \ddots & \\ P_{I1} & P_{I2} & \cdots & P_{IG} & 0 & & -\lambda_I \end{array}\right] \qquad (1.4)$$

Writing various equations in matrix form will simplify the discussions in later chapters. The matrix $\underline{M}(\bar{r},t)$ will be referred to in Section 2.2, which contains the analytical development of the multiple temporal-mode transformation technique.

CHAPTER 2

BASIS OF SOLUTION FORMALISM

2.1 Multiple Temporal-Mode Solution Representation

The use of the multiple temporal-mode transformation in the solution representation of the reactor kinetics equations is motivated by the following observations. First, for many transients the behaviour of the flux is basically exponential in nature. Second, the neutron groups and delayed neutron precursors are coupled, as shown in Equations (1.1), implying that a transformation which retains this coupling explicitly is desirable. Thus the following Ansatz is chosen.

$$\begin{aligned}\theta_\ell(\bar{r},t) &= \psi_\ell(\bar{r},t)\sum_{\ell'=1}^{L}\beta_{\ell\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t] \\ &= \sum_{\ell'=1}^{L}\psi'_{\ell\ell'}(\bar{r},t)\exp[\alpha_{\ell'}(\bar{r})t] \qquad (1\leq\ell\leq L)\end{aligned} \tag{2.1}$$

where

$$\psi'_{\ell\ell'}(\bar{r},t) = \psi_\ell(\bar{r},t)\beta_{\ell\ell'}(\bar{r}) \tag{2.2}$$

$L=G+I$ = total number of energy groups and precursor groups

$\theta_\ell(\bar{r},t)$ = group g neutron flux $\phi_g(\bar{r},t)$ for $\ell=g$, $(1\leq g\leq G)$; i^{th} precursor concentration $C_i(\bar{r},t)$ for $\ell=i$, $(1\leq i\leq I)$

$\psi_\ell(r,t)$ = correction function of the energy group or precursor group

$\alpha_{\ell'}(\bar{r})$ = the ℓ'^{th} frequency

$\beta_{\ell\ell'}(\bar{r})$ = the corresponding ℓ'^{th} spatial moment of the ℓ^{th} group

Equation (2.1) shows that the solution of the neutron flux or precursor concentration is represented by a sum of exponential terms weighted by a time-dependent correction function $\psi_\ell(\bar{r},t)$. The

multi-mode Ansatz, when viewed in historical perspective, does resemble the Ansatz previously used by Reed et al[4], Wight et al[5] and Denning[9],

$$\phi_g(\bar{r},t) = \psi_g(\bar{r},t)\exp[\alpha_g(\bar{r},t)] \tag{2.3}$$

and that used by Ferguson[6],

$$\phi_g(\bar{r},t) = \psi_g(\bar{r},t)\exp[\alpha(\bar{r})t] \tag{2.4}$$

The difference in these three solution models is found primarily with the influence given to the time dependence of the various groups.

2.2 Analytical Development

Consider an arbitrary time interval, Δt, for which the system parameters, $D_g(\bar{r})$, $\Sigma_{gg'}(\bar{r})$ and $p_{ig'}(\bar{r})$, are assumed to be given as a function of position only. Performing the transformation by substituting Equation (2.1) into Equations (1.1) gives at time t

$$\begin{aligned}
&\frac{1}{v_g}\sum_{\ell'=1}^{L}\beta_{g\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\left[\alpha_{\ell'}(\bar{r})\psi_g(\bar{r},t) + \frac{\partial}{\partial t}\psi_g(\bar{r},t)\right] \\
&= \sum_{\ell'=1}^{L}\beta_{g\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\ [\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\psi_g(\bar{r},t)] \\
&+ \psi_g(\bar{r},t)\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\left\{\sum_{\ell'=1}^{L}\beta_{g\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\right\} \\
&+ \sum_{\ell'=1}^{L}\exp[\alpha_{\ell'}(\bar{r})t]\left\{\sum_{g'=1}^{G}\Sigma_{gg'}(\bar{r})\beta_{g'\ell'}(\bar{r})\psi_{g'}(\bar{r},t)\right\} \\
&+ \sum_{\ell'=1}^{L}\exp[\alpha_{\ell'}(\bar{r})t]\left\{\sum_{i=1}^{I}f_{gi}\beta_{G+i,\ell'}(\bar{r})\psi_{G+i}(\bar{r},t)\right\} \quad (1\le g\le G)
\end{aligned} \tag{2.5a}$$

$$\sum_{\ell'=1}^{L} \beta_{G+i,\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\left[\alpha_{\ell'}(\bar{r})\psi_{G+i}(\bar{r},t) + \frac{\partial}{\partial t}\psi_{G+i}(\bar{r},t)\right]$$

$$= -\sum_{\ell'=1}^{L} \beta_{G+i,\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\,\lambda_i\psi_{G+i}(\bar{r},t)$$

$$+\sum_{\ell'=1}^{L} \exp[\alpha_{\ell'}(\bar{r})t]\sum_{g'=1}^{G} p_{ig'}(\bar{r})\beta_{g'\ell'}(\bar{r})\psi_{g'}(\bar{r},t) \qquad (1\leq i\leq I) \qquad (2.5b)$$

Equation (2.5a) and Equation (2.5b) are valid at all times within the interval under consideration.

Writing the transformed equations in this form identifies the common multiplier containing the ℓ'^{th} order frequency, $\exp[\alpha_{\ell'}(\bar{r})t]$, in each but the second term on the right hand side of Equation (2.5a). If a normalization of the following type is imposed upon the spatial moments,

$$\sum_{\ell'=1}^{L} \beta_{\ell\ell'}(\bar{r}) = 1 \qquad \text{for all } \bar{r} \text{ and } \ell,\ 1\leq\ell\leq L, \qquad (2.6)$$

then at t=o, the following is obtained

$$\psi_g(\bar{r},t)\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\sum_{\ell'=1}^{L}\beta_{g\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\,\Big|_{t=o}$$

$$= \psi_g(\bar{r},o)\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}(1)$$

$$= o \qquad (1\leq g\leq G) \qquad (2.7)$$

The initial condition is given by

$$\theta_\ell(\bar{r},o) = \psi_\ell(\bar{r},o)$$

$$= \sum_{\ell'=1}^{L} \psi_\ell(\bar{r},o)\beta_{\ell\ell'}(\bar{r}) \qquad (1\leq\ell\leq L) \qquad (2.8)$$

Hence, by defining the initial time t=o as the beginning of each time interval under consideration, the second term on the right hand side of Equation (2.5a) is eliminated. Furthermore, at the beginning of

each time interval, the initial condition, Equation (2.8), is obtained, equating the neutron flux, $\phi_g(\bar{r},o)$, or precursor concentration, $C_i(\bar{r},o)$, to the corresponding correction function at t=o.

Since the temporal transformation has extracted the dominant time dependence of the neutron flux or the precursor concentration, the correction function, $\psi_\ell(\bar{r},t)$, is now slowly varying in time. Thus, within an arbitrary time interval Δt, the following condition is assumed:

$$\frac{\partial}{\partial t}\ \psi_\ell(\bar{r},t)\ \Big|_{t\epsilon\ \Delta t} = o \qquad (2.9)$$

Using the normalization condition, Equation (2.6), and the above assumption, Equation (2.9), at t=o, the beginning of the time interval, Equation (2.5a) and Equation (2.5b) can be written as follows:

$$\frac{1}{v_g}\sum_{\ell'=1}^{L}\beta_{g\ell'}(\bar{r})\alpha_{\ell'}(\bar{r})\psi_g(\bar{r},o)$$

$$= \bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\psi_g(\bar{r},o) + \sum_{\ell'=1}^{L}\sum_{g'=1}^{G}\Sigma_{gg'}(\bar{r})\beta_{g'\ell'}(\bar{r})\psi_{g'}(\bar{r},o)$$

$$+ \sum_{\ell'=1}^{L}\sum_{i=1}^{I} f_{gi}\beta_{G+i,\ell'}(\bar{r})\psi_{G+i}(\bar{r},o) \qquad (1\leq g\leq G) \qquad (2.10a)$$

$$\sum_{\ell'=1}^{L}\beta_{G+i,\ell'}(\bar{r})\alpha_{\ell'}(\bar{r})\psi_{G+i}(\bar{r},o)$$

$$= -\sum_{\ell'=1}^{L}\beta_{G+i,\ell'}(\bar{r})\lambda_i\psi_{G+i}(\bar{r},o) + \sum_{\ell'=1}^{L}\sum_{g'=1}^{G} p_{ig'}(\bar{r})\beta_{g'\ell'}(\bar{r})\psi_{g'}(\bar{r},o)$$

$$(1\leq i\leq I) \qquad (2.10b)$$

In Equation (2.10a) and Equation (2.10b), all expressions except the first one on the right hand side of Equation (2.10a) contain a summation of terms over ℓ', $1\leq\ell'\leq L$.

Because of the initial condition, Equation (2.8), the term $\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\psi_g(\bar{r},o)$ is simply given by the neutron leakage at the beginning of the time interval. Thus, with the following definition

$$E_g(\bar{r},o) = [\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\psi_g(\bar{r},o)]/\psi_g(\bar{r},o)$$

$$= [\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\phi_g(\bar{r},o)]/\phi_g(\bar{r},o)$$

$$= [\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\phi_g(\bar{r},o)]/[\sum_{\ell'=1}^{L}\psi_g(\bar{r},o)\beta_{g\ell'}(\bar{r})]$$

$$(1\leq g\leq G) \qquad (2.11)$$

and by equating the ℓ'^{th} terms in each summation in Equation (2.10a) and Equation (2.10b), the following equations are derived:

$$\frac{1}{v_g}\alpha_{\ell'}(\bar{r})\psi_g(\bar{r},o)\beta_{g\ell'}(\bar{r})$$

$$= E_g(\bar{r},o)\psi_g(\bar{r},o)\beta_{g\ell'}(\bar{r}) + \sum_{g'=1}^{G}\Sigma_{gg'}(\bar{r})\psi_{g'}(\bar{r},o)\beta_{g'\ell'}(\bar{r})$$

$$+\sum_{i=1}^{I} f_{gi}\psi_{G+i}(\bar{r},o)\beta_{G+i,\ell'}(\bar{r}) \qquad (1\leq g\leq G,\ 1\leq\ell'\leq L)$$

(2.12a)

$$\alpha_{\ell'}(\bar{r})\psi_{G+i}(\bar{r},o)\beta_{G+i,\ell'}(\bar{r})$$

$$= -\lambda_i\psi_{G+i}(\bar{r},o)\beta_{G+i,\ell'}(\bar{r}) + \sum_{g'=1}^{G} p_{ig'}(\bar{r})\psi_{g'}(\bar{r},o)\beta_{g'\ell'}(\bar{r})$$

$$(1\leq i\leq I,\ 1\leq\ell'\leq L)$$

(2.12b)

Equation (2.12a) and Equation (2.12b) can be combined into the following matrix form:

$$[\underline{M}'(\bar{r},o)-\alpha_{\ell'}(\bar{r})\underline{I}]\,\bar{\theta}'_{\ell'}(\bar{r},o) = 0 \qquad (2.13)$$

where

$$\bar{\theta}'_{\ell'}(\bar{r},o) = \begin{bmatrix} \psi'_{1\ell'}(\bar{r},o) \\ \psi'_{2\ell'}(r,o) \\ \vdots \\ \psi'_{G\ell'}(\bar{r},o) \\ \psi'_{G+1,\ell'}(r,o) \\ \vdots \\ \psi'_{L\ell'}(\bar{r},o) \end{bmatrix} = \begin{bmatrix} \psi_1(\bar{r},o)\beta_{1\ell'}(\bar{r}) \\ \psi_2(\bar{r},o)\beta_{2\ell'}(\bar{r}) \\ \vdots \\ \psi_G(\bar{r},o)\beta_{G\ell'}(\bar{r}) \\ \psi_{G+1}(\bar{r},o)\beta_{G+1,\ell'}(\bar{r}) \\ \vdots \\ \psi_L(\bar{r},o)\beta_{L\ell'}(r) \end{bmatrix} \qquad (2.14)$$

and

$$\underline{M}'(\bar{r},o) = \left[\begin{array}{cccc|ccc} \nu_1(E_1+\Sigma_{11}) & \nu_1\Sigma_{12} & \cdots & \nu_1\Sigma_{1G} & \nu_1 f_{11} & \cdots & \nu_1 f_{1I} \\ \nu_2\Sigma_{21} & \nu_2(E_2+\Sigma_{22}) & \cdots & \nu_2\Sigma_{2G} & \nu_2 f_{21} & \cdots & \nu_2 f_{2I} \\ & \cdots & & & & & \\ \nu_G\Sigma_{G1} & \nu_G\Sigma_{G2} & \cdots & \nu_G(E_G+\Sigma_{GG}) & \nu_G f_{G1} & \vdots\ \vdots & \nu_G f_{GI} \\ \hline p_{11} & p_{12} & \cdots & p_{1G} & -\lambda_i & & 0 \\ & \cdots & & & & \ddots & \\ p_{I1} & p_{I2} & \cdots & p_{IG} & 0 & & -\lambda_I \end{array}\right] \qquad (2.15)$$

Here $\underline{M}'(\bar{r},o)$ differs from $\underline{M}(\bar{r},o)$ of Eq. (1.4) in that the group g neutron leakage, $\bar{\nabla}\cdot D_g(\bar{r})\bar{\nabla}\phi_g(\bar{r},t)$, is evaluated specifically at the beginning of the time interval (i.e. t=o). The group g neutron flux, $\phi_g(\bar{r},o)$, is known at the beginning of the time interval, because either the initial flux distribution is specified or it has been calculated from the previous time interval. Thus the term $Eg(\bar{r},o)$ can be calculated easily at t=o according to Eq. (2.11). The elements of the coefficient matrix $\underline{M}'(\bar{r},o)$ in Eq. (2.15) are also all known at the beginning of the time interval. Eq. (2.13) is in fact an eigenvalue equation. There are only certain values of $\alpha_{\ell'}(\bar{r})$ for which non-trivial solutions in $\bar{\theta}_{\ell'}(\bar{r},o)$ exist. Therefore, it can be seen that the Ansatz in Eq. (2.1) is consistent, with the solution expanded in the eigenfunctions of Eq. (2.15), together with their corresponding exponentials to account for the dominant time-dependence. It is then possible to evaluate the frequencies, $\alpha_{\ell'}(\bar{r})$, and the spatial moments, $\beta_{\ell\ell'}(\bar{r})$, for all $\bar{r}$ at each interval based on the most recent values of $Eg(\bar{r},o)$, $\Sigma_{gg'}(\bar{r})$ and $p_{ig}(\bar{r})$ by solving for the eigenvalues and the corresponding eigenvectors of Eq. (2.13). The computed eigenvalues are the frequencies, but to obtain the spatial moments, the eigenvectors must be properly normalized, subject to the normalization condition in Eq. (2.6). This procedure is described in detail in Appendix B. The values of $Eg(\bar{r},o)$, $\Sigma_{gg'}(\bar{r})$ and $p_{ig'}(\bar{r})$ are updated at the beginning of each iteration.

It should be pointed out that the spatial moments and frequencies are calculated at the beginning of each time interval from Eq. (2.13) which is derived from Eq. (2.5a) and Eq. (2.5b) by setting t=o and by utilizing the assumption, Eq. (2.9). Once they are calculated, they can be substituted back into the time-dependent transformed equations, Eq. (2.5a) and Eq. (2.5b), leaving equations in the correction functions, $\psi_\ell(\bar{r},t)$ as follows:

$$\frac{1}{v_g}\sum_{\ell'=1}^{L}\beta_{g\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\,\frac{\partial}{\partial t}\psi_g(\bar{r},t)$$

$$= \bar{\nabla} \cdot D_g(\bar{r})\bar{\nabla}\psi_g(\bar{r},t)\sum_{\ell'=1}^{L} \beta_{g\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]$$

$$+\sum_{g'=1}^{G} \Sigma_{gg'}(\bar{r})\psi_{g'}(\bar{r},t)\sum_{\ell'=1}^{L} \beta_{g'\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]$$

$$+\sum_{i}^{I} f_{gi}\psi_{G+i}(\bar{r},t)\sum_{\ell'=1}^{L} \beta_{G+i,\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]$$

$$- \frac{1}{v_g} \psi_g(\bar{r},t)\sum_{\ell'=1}^{L} \beta_{g\ell'}(\bar{r})\alpha_{\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t] \qquad (1\leq g\leq G) \qquad (2.16a)$$

$$\sum_{\ell'=1}^{L} \beta_{G+i,\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]\ \frac{\partial}{\partial t}\ \psi_{G+i}(\bar{r},t)$$

$$= -\lambda_i\psi_{G+i}(\bar{r},t)\sum_{\ell'=1}^{L} \beta_{G+i,\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]$$

$$+\sum_{g'=1}^{G} p_{ig'}(\bar{r})\psi_{g'}(\bar{r},t)\sum_{\ell'=1}^{L} \beta_{g'\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t]$$

$$- \psi_{G+i}(\bar{r},t)\sum_{\ell'=1}^{L} \beta_{G+i,\ell'}(\bar{r})\alpha_{\ell'}(r)\exp[\alpha_{\ell'}(r)t] \qquad (1\leq i\leq I) \qquad (2.16b)$$

The correction functions, $\psi_\ell(\bar{r},t)$, can then be solved by standard numerical techniques such as finite difference methods.

2.3 Finite Difference Analysis

The equations presented in the previous sections are continuous

in both spatial and temporal variables. In order to discretize the spatial variable, a spatial mesh is superimposed upon the reactor of interest. The semi-discrete form of the reactor kinetics equations, shown in Appendix A.2, are obtained when Equations (1.1) are integrated over the volume of each mesh cell using the finite difference approximation for the spatial derivative operator.

The semi-discrete equations derived in Appendix A.2 for the group g neutron flux, $\bar{\phi}_g$, and the i^{th} precursor concentration, $\bar{C}_i$, at all mesh points are written as

$$\frac{d}{dt}\bar{\phi}_g = \underline{D}_g\bar{\phi}_g + \sum_{g'=1}^{G} \underline{T}_{gg'}\bar{\phi}_{g'} + \sum_{i=1}^{I} \underline{F}_{gi}\bar{C}_i \qquad (1 \leq g \leq G)$$

$$\frac{d}{dt}\bar{C}_i = -\underline{\Lambda}_i\bar{C}_i + \sum_{g'=1}^{G} \underline{P}_{ig'}\bar{\phi}_{g'} \qquad (1 \leq i \leq I) \qquad (2.17)$$

When the neutron flux and the precursor concentration are arranged in a sequential manner, group by group and, within a group, mesh point by mesh point, Equations (2.17) can be combined into the matrix form

$$\frac{d}{dt}\bar{\Phi} = \underline{A}\,\bar{\Phi} \qquad (2.18)$$

where $\bar{\Phi}$ is a column vector of length (JxK)x(G+I)

$$\bar{\Phi} = \begin{bmatrix} \bar{\phi}_1 \\ \bar{\phi}_2 \\ \cdot \\ \cdot \\ \cdot \\ \bar{\phi}_G \\ \bar{C}_1 \\ \cdot \\ \cdot \\ \cdot \\ \bar{C}_I \end{bmatrix} \tag{2.19}$$

and $\underline{A}$ is a square matrix of order (JxK)x(G+I)

$$\underline{A} = \left[\begin{array}{cccc|ccc} \underline{D}_1+\underline{T}_{11} & \underline{T}_{12} & \cdots & \underline{T}_{1G} & \underline{F}_{11} & \cdots & \underline{F}_{1I} \\ \underline{T}_{21} & \underline{D}_2+\underline{T}_{22} & \cdots & \underline{T}_{2G} & \underline{F}_{21} & \cdots & \underline{F}_{2I} \\ \underline{T}_{G1} & \underline{T}_{G2} & \cdots & \underline{D}_G+\underline{T}_{GG} & \underline{F}_{G1} & \cdots & \underline{F}_{GI} \\ \hline \underline{P}_{11} & \underline{P}_{12} & \cdots & \underline{P}_{1G} & -\underline{\Lambda}_1 & & \underline{0} \\ & \cdots & & & & \ddots & \\ \underline{P}_{I1} & \underline{P}_{I2} & \cdots & \underline{P}_{IG} & \underline{0} & & -\underline{\Lambda}_I \end{array}\right] \tag{2.20}$$

where (JxK) is the total number of mesh points.

Here $\underline{T}_{gg'}$ is a diagonal matrix containing terms representing intergroup transfer processes. $\underline{P}_{ig'}$ is a diagonal matrix representing the production of delayed precursor i due to fissions in group g'.

$\underline{F}_{gi}$ is a diagonal matrix representing the transfer of delayed neutron into group g due to decays in precursor group i. $\underline{D}_g$ is a stripe matrix containing terms which represent the averaged diffusion coefficients. In one- and two-spatial dimensions, $\underline{D}_g$ is a three- and five-stripe matrix respectively.

For any period of time, Δt, during which all terms in $\underline{A}$ are constant, the solution to Equation (2.18) is given by

$$\bar{\Phi}(\Delta t) = e^{\underline{A}\Delta t}\bar{\Phi}(o) \tag{2.21}$$

In practice, it is time consuming to calculate $\bar{\Phi}(\Delta t)$ from Equation (2.21). Nevertheless, Equation (2.21) is significant and will be referred to later in Section 3.2 to establish an error estimator for the finite difference method.

The transformation, Equation (2.1), can be written as

$$\theta_\ell(\bar{r},t) = \Omega_\ell(\bar{r},t)\psi_\ell(\bar{r},t) \qquad (1\leqslant\ell\leqslant L) \tag{2.22}$$

where

$$\Omega_\ell(\bar{r},t) = \sum_{\ell'=1}^{L} \beta_{\ell\ell'}(\bar{r})\exp[\alpha_{\ell'}(\bar{r})t] \qquad (1\leqslant\ell\leqslant L) \tag{2.23}$$

With the following definitions

$$\Omega_{\ell,j,k} = \sum_{\ell'=1}^{L} \beta_{\ell\ell',j,k}\exp[\alpha_{\ell',j,k}\cdot t] \tag{2.24a}$$

$$\underline{\Omega}_\ell = \text{diag}[\Omega_{\ell,j,k}] \tag{2.24b}$$

the semi-discrete form of the transformed equations for the correction functions at all points for group g, $\bar{\psi}_g$, and for the i^{th} precursor group, $\bar{\psi}_{G+i}$, can be written, as shown in Appendix A.3, as follows

$$\underline{\Omega}_g \frac{d}{dt} \bar{\psi}_g = \underline{D}_g \underline{\Omega}_g \bar{\psi}_g + \sum_{g'=1}^{G} \underline{T}_{gg'} \underline{\Omega}_{g'} \bar{\psi}_{g'}$$

$$+ \sum_{i}^{I} \underline{F}_{gi} \underline{\Omega}_{G+i} \bar{\psi}_{G+i} - \dot{\underline{\Omega}}_g \bar{\psi}_g \qquad (1 \leq g \leq G) \qquad (2.25a)$$

$$\underline{\Omega}_{G+i} \frac{d}{dt} \bar{\psi}_{G+i} = -\underline{\Lambda}_i \underline{\Omega}_{G+i} \bar{\psi}_{G+i} + \sum_{g'=1}^{G} \underline{P}_{ig'} \underline{\Omega}_{g'} \bar{\psi}_{g'}$$

$$- \dot{\underline{\Omega}}_{G+i} \bar{\psi}_{G+i} \qquad (1 \leq i \leq I) \qquad (2.25b)$$

where $\dot{\underline{\Omega}}_\ell$ is the time derivative of $\underline{\Omega}_\ell$, $1 \leq \ell \leq L$.

Thus when the correction functions for energy groups and precursors at all mesh points are arranged in a sequential manner, the transformation

$$\bar{\Phi} = \underline{\Omega}\, \bar{\psi} \qquad (2.26)$$

leads to the following compact representation of the transformed equations

$$\underline{\Omega} \frac{d}{dt} \bar{\psi} = (\underline{A} - \dot{\underline{\Omega}}\, \underline{\Omega}^{-1}) \underline{\Omega}\, \bar{\psi} \qquad (2.27a)$$

or

$$\frac{d}{dt}\bar{\psi} = \underline{\Omega}^{-1}(\underline{A}-\dot{\underline{\Omega}}\,\underline{\Omega}^{-1})\underline{\Omega}\,\bar{\psi} \qquad (2.27b)$$

where $\bar{\psi}$ is a column vector of length (JxK)x(G+I) containing the correction functions

$$\bar{\psi} = \begin{bmatrix} \bar{\psi}_1 \\ \bar{\psi}_2 \\ \vdots \\ \bar{\psi}_G \\ \bar{\psi}_{G+1} \\ \vdots \\ \bar{\psi}_L \end{bmatrix} \qquad (2.28)$$

and $\underline{\Omega}$ is a diagonal matrix of order (JxK)x(G+I) containing the effective exponential terms at all mesh points

$$\underline{\Omega} = \text{diag}[\Omega_\ell] \qquad (1\leqslant\ell\leqslant L) \qquad (2.29)$$

$\underline{\Omega}_\ell$ has previously been defined in Equation (2.24b). $\underline{A}$ is the same as defined previously in Equation (2.20).

Equation (2.27a) or Equation (2.27b) is further discretized in the time variable by approximating the time derivative with a finite difference expression. This will result in a system of algebraic equations which can be solved to obtain the correction functions as will be shown in Chapter 3.

CHAPTER 3

ALTERNATING-DIRECTION EXPLICIT METHOD

The solution to the kinetics equations given by Eq. (2.21) involves evaluating $e^{\underline{A}\Delta t}$. However, since $\underline{A}$ is not diagonal, it is time consuming to calculate $e^{\underline{A}\Delta t}$ directly. Both the alternating-direction explicit methods (ADE)[6,9] and the alternating-direction implicit methods (ADI)[5], using a simple frequency transformation, can generate a good approximation of $e^{\underline{A}\Delta t}$ and thus can avoid the above difficulty. In the ADE and ADI methods, the time derivative in Eq. (2.18) is approximated by a finite difference. The time interval under consideration, called a time step, is divided into two equal half-steps. Within each half-step, the $\underline{A}$ matrix is split into two composite parts. Thus the resulting approximation to $e^{\underline{A}\Delta t}$ involves inverting only part of the $\underline{A}$ matrix over each half-step.

The $\underline{A}$ matrix is real, square and irreducible with non-negative off diagonal elements and negative diagonal elements. These elements span 6 to 8 orders of magnitude. From a mathematical point of view, Eq. (2.18) represents a stiff system of equations in which the eigenvalues of the $\underline{A}$ matrix also span 6 to 8 orders of magnitude.

Using a finite difference to replace the time derivative in Eq. (2.18) will necessarily introduce truncation error difficulties and for transients with rapidly varying spatial changes sufficiently small time steps must be taken to obtain an accurate solution and to maintain numerical stability. It has been found that the use of a simple frequency transformation[5,6,9] with the ADE or ADI methods permits larger time steps to be taken because of the smaller truncation error in the subsequent application of the finite difference approximation to the time derivative of the correction function $\overline{\psi}$. The purpose of this study is to assess the multiple temporal-mode transformation as applied to the ADE method as an alternative method of reducing the truncation error.

3.1 Method Of Evaluating The Multiple Temporal-Mode Transformation

It was decided at the beginning of the project that a general kinetics computer program would not need to be developed for this study. Instead, two subroutines have been written which calculate the spatial moments and frequencies and these have been incorporated into the existing CRNL code ADEP. ADEP, which is an alternating-direction explicit code, has been modified to utilize

the two subroutines at the beginning of each time step to calculate the effective exponential terms and the diagonal elements of the $\underline{\Omega}$ matrix, which are based on the spatial moments and frequencies. We have used the multiple temporal-mode transformation technique with an alternating-direction explicit method in this study, however it could equally well be applied to an implicit method.

It is noted that in ADEP the delayed neutron source is approximated by the precursor concentrations at the previous time level to avoid the added complexity of solving a larger number of equations simultaneously. Although the multi-mode temporal transformation, Eq. (2.1), has been assumed for all energy groups and precursors in order to set up the equations for eigenvalues and eigenvectors at the beginning of the time step, it is not applied to the precursor groups during the time step.

In effect, Eq. (2.26) and Eq. (2.27a) have been modified in the following manner:

$$\overline{\Phi} = \underline{\Omega}\overline{\psi} \tag{3.1}$$

$$\underline{\Omega}\frac{d\overline{\psi}}{dt} = (\underline{A} - \dot{\underline{\Omega}}\underline{\Omega}^{-1})\underline{\Omega}\overline{\psi} \tag{3.2}$$

where the vector $\overline{\psi}$ has been redefined as

$$\overline{\Psi} = \begin{bmatrix} \overline{\psi}_1 \\ \overline{\psi}_2 \\ \vdots \\ \overline{\psi}_G \\ \overline{C}_1 \\ \vdots \\ \overline{C}_I \end{bmatrix} \tag{3.3}$$

with $\overline{\psi}_g$ the correction function of energy group g and $\overline{C}_i$ the concentration of precursor group i. Likewise, the $\underline{\Omega}$ and $\dot{\underline{\Omega}}$ matrices have been redefined as follows:

$$\underline{\Omega} = \begin{bmatrix} \underline{\Omega}_1 & & & & & & \\ & \underline{\Omega}_2 & & & \underline{0} & & \\ & & \ddots & & & & \\ & & & \underline{\Omega}_G & \underline{I} & & \\ & \underline{0} & & & & \ddots & \\ & & & & & & \underline{I} \end{bmatrix} \tag{3.4}$$

$$\dot{\underline{\Omega}} = \begin{bmatrix} \dot{\underline{\Omega}}_1 & & & & & & \\ & \dot{\underline{\Omega}}_2 & & & \underline{0} & & \\ & & \ddots & & & & \\ & & & \dot{\underline{\Omega}}_G & \underline{0} & & \\ & \underline{0} & & & & \ddots & \\ & & & & & & \underline{0} \end{bmatrix} \tag{3.5}$$

where $\underline{\Omega}_g$ is a diagonal matrix containing the effective exponential terms of energy group g at all mesh points and $\dot{\underline{\Omega}}_g$ is the time derivative of $\underline{\Omega}_g$.

For each half-step, the precursor concentrations are calculated initially according to the following equation:

$$\frac{\overline{C}_i(t_2) - \overline{C}_i(t_1)}{\Delta t} = -0.5\underline{\Lambda}_i\overline{C}_i(t_2) - 0.5\underline{\Lambda}_i\overline{C}_i(t_1) + \sum_{g'=1}^{G} \underline{P}_{ig'}\overline{\phi}_{g'}(t_1) \quad (1 \leq i \leq I) \quad (3.6)$$

where t_1 and t_2 are the times at the beginning and the end respectively of the half-step under consideration. Here $\overline{C}_i$ and $\overline{\phi}_g$ represent the concentrations of precursor group i and group g neutron fluxes respectively at all mesh points. $\underline{\Lambda}_i$ is a diagonal matrix which has as its diagonal elements the decay constants of precursor group i at all mesh points. Likewise, $\underline{P}_{ig'}$ is a diagonal matrix whose elements represent the production of delayed precursor i due to fissions in group g'.

The precursor concentrations at the end of the half-step, $t = t_2$, are obtained from the following equation:

$$C_i(t_2) = \left[\underline{I} + \frac{\Delta t}{2}\underline{\Lambda}_i\right]^{-1} \left\{\left[\underline{I} - \frac{\Delta t}{2}\underline{\Lambda}_i\right]\overline{C}_i(t_1) + \Delta t \sum_{g'=1}^{G} \underline{P}_{ig'}\overline{\phi}_{g'}(t_1)\right\} \quad (1 \leq i \leq I) \qquad (3.7)$$

The matrix inversion, $[\underline{I} + \frac{\Delta t}{2}\underline{\Lambda}_i]^{-1}$, is easily performed since $\underline{I}$ and $\underline{\Lambda}_i$ are diagonal matrices. The precursor concentrations obtained are then utilized in Eq. (3.2) to calculate the neutron fluxes which are given as a product of the correction functions, $\overline{\psi}$, and the amplitude functions, $\underline{\Omega}$. The calculation of the neutron fluxes will be described in the following section.

3.2 The Advancement Matrix And Temporal Truncation Error

As mentioned previously, in ADE and ADI methods, the $\underline{A}$ matrix is split into two composite parts within each half-step. In this study, a time step is represented by $2\Delta t$. In general, the splitting of the $\underline{A}$ matrix can be represented as follows:

$$\begin{aligned} \underline{A} &= \underline{A}_1 + \underline{A}_2 \qquad (0 \leq t \leq \Delta t) \\ \underline{A} &= \underline{A}_3 + \underline{A}_4 \qquad (\Delta t \leq t \leq 2\Delta t) \end{aligned} \qquad (3.8)$$

The ADEP method of splitting the $\underline{A}$ matrix is employed in this study. The components of the matrices $\underline{A}_1$, $\underline{A}_2$, $\underline{A}_3$ and $\underline{A}_4$ are not described here but they can be obtained by referring to the ADEP manual[9].

For each half-step, the time derivative in Eq. (3.2) is approximated by a simple forward difference. There are various ways of approximating the other terms in Eq. (3.2) which do not involve space or time derivatives, but a central average of the fluxes in time has been shown by Denning[9] to give the smallest truncation error. Thus Eq. (3.2) is written in the discretized form as follows:

$$\underline{\Omega}(t)\,\frac{\overline{\Psi}(\Delta t)-\overline{\Psi}(o)}{\Delta t} = \left[\underline{A}_2 - 0.5\dot{\underline{\Omega}}(t)\underline{\Omega}^{-1}(t)\right]\underline{\Omega}(t)\overline{\Psi}(\Delta t) + \left[\underline{A}_1 - 0.5\dot{\underline{\Omega}}(t)\underline{\Omega}^{-1}(t)\right]\underline{\Omega}(t)\overline{\Psi}(o) \qquad (o \leq t \leq \Delta t) \qquad (3.9a)$$

$$\underline{\Omega}(t)\,\frac{\overline{\Psi}(2\Delta t)-\overline{\Psi}(\Delta t)}{\Delta t} = \left[\underline{A}_4 - 0.5\dot{\underline{\Omega}}(t)\underline{\Omega}^{-1}(t)\right]\underline{\Omega}(t)\overline{\Psi}(2\Delta t) + \left[\underline{A}_3 - 0.5\dot{\underline{\Omega}}(t)\underline{\Omega}^{-1}(t)\right]\underline{\Omega}(t)\overline{\Psi}(\Delta t) \qquad (\Delta t \leq t \leq 2\Delta t) \qquad (3.9b)$$

Here the time dependence of the matrices $\underline{\Omega}(t)$ and $\dot{\underline{\Omega}}(t)$ are explicitly indicated. Their values depend upon the specific point in time within each half-step at which they are to be calculated.

During the course of this project, three different ways of advancing the solution in time have evolved. In all three cases, the spatial moments and frequencies are computed once for each time step and are assumed

constant within that time step. For convenience, the three numerical algorithms considered are labelled MTT1, MTT2 and MTT3. The following sections describe each numerical algorithm. In addition, the temporal truncation error estimator for each case is also given.

3.2a MTT1

According to the formulation of the multiple temporal-mode transformation technique as described in Chapter 2, the time consuming evaluations of the exponentials in $\Omega_{g,j,k}$ have to be performed once for each half-step. To minimize computer time, the matrix $\underline{\Omega}(t)$ and $\dot{\underline{\Omega}}(t)$ for both half-steps are evaluated at $t=\Delta t$ and the solution formalism is modified with the following assumptions for the second half-step:

$$\bar{\Psi}(\Delta t) = \bar{\Phi}(\Delta t) \tag{3.10a}$$

$$\bar{\Phi}(2\Delta t) = \underline{\Omega}(\Delta t)\bar{\Psi}(2\Delta t) \tag{3.10b}$$

Thus Eq. (3.9a) and Eq. (3.9b) can be combined to eliminate $\bar{\Psi}(\Delta t)$ and the following equation is obtained:

$$\begin{aligned}\underline{\Omega}\bar{\Psi}(2\Delta t) = {} & [\underline{I}-\Delta t(\underline{A}_4-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]^{-1}\ [\underline{I}+\Delta t(\underline{A}_3-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]\underline{\Omega} \\ & \cdot [\underline{I}-\Delta t(\underline{A}_2-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]^{-1}\ [\underline{I}+\Delta t(\underline{A}_1-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]\underline{\Omega}\bar{\Psi}(o)\end{aligned} \tag{3.11}$$

where $\underline{\Omega}(\Delta t)$ and $\dot{\underline{\Omega}}(\Delta t)$ are simply represented by $\underline{\Omega}$ and $\dot{\underline{\Omega}}$ respectively. Since

$$\bar{\Phi}(o) = \bar{\psi}(o)$$

and with the assumption in Eq. (3.10b),

$$\bar{\Phi}(2\Delta t) = \underline{\Omega}(\Delta t)\bar{\psi}(2\Delta t)$$

Eq. (3.11) can be written as

$$\bar{\Phi}(2\Delta t) = \underline{B}_1 \bar{\Phi}(o) \tag{3.12a}$$

where

$$\underline{B}_1 = [\underline{I}-\Delta t(\underline{A}_4-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]^{-1}\ [\underline{I}+\Delta t(\underline{A}_3-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]\underline{\Omega}$$
$$\cdot\ [\underline{I}-\Delta t(\underline{A}_2-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]^{-1}\ [\underline{I}+\Delta t(\underline{A}_1-0.5\dot{\underline{\Omega}}\underline{\Omega}^{-1})]\underline{\Omega} \tag{3.12b}$$

The matrix $\underline{B}_1$ then is an operator which advances the solution over a time step. It may be called the advancement matrix for MTT1.

In order to establish an estimate of the error bound for the numerical algorithm, the matrix $\underline{B}_1$ is first expanded in a Taylor series as follows:

$$\underline{B}_1 = \underline{I}+2\Delta t(\underline{A}-\dot{\underline{\Omega}}\underline{\Omega}^{-1}+\underline{C}) + \ O(\Delta t^2) \tag{3.13a}$$

where $\underline{C}$ is a diagonal matrix given by

$$e^{\Delta t \underline{C}} = \underline{\Omega}(\Delta t) \qquad (3.13b)$$

As mentioned previously in Section 2.3, if all elements in the $\underline{A}$ matrix are constant over a time step, $2\Delta t$, the solution to the kinetics equations, Eq. (2.18), is given by

$$\bar{\Phi}(2\Delta t) = e^{2\Delta t \underline{A}}\, \Phi(o)$$

The operator $e^{2\Delta t \underline{A}}$ can also be expanded in a Taylor series to yield

$$e^{2\Delta t \underline{A}} = \underline{I} + 2\Delta t \underline{A} + 2\Delta t^2 \underline{A}^2 + \ldots \qquad (3.14)$$

By defining the temporal truncation error over one time step as

$$T.E. = e^{2\Delta t \underline{A}} - \underline{B}_1 \qquad (3.15)$$

it is found that the temporal truncation error for MTT1 is proportional to Δt.

$$T.E._1 = 2\Delta t(\dot{\underline{\Omega}}\underline{\Omega}^{-1} - \underline{C}) + O(\Delta t^2) \qquad (3.16)$$

It is assumed that the terms proportional to Δt are the dominant source of truncation error.

3.2b MTT2

After some initial test runs using the MTT1 algorithm, it was observed that calculations of the spatial moments

and frequencies required more computer time than the evaluations of the exponentials in $\Omega_{g,j,k}$. So the MTT2 algorithm based on the basic solution formalism as described in Chapter 2 is attempted. In this case, the matrices $\underline{\Omega}(t)$ and $\dot{\underline{\Omega}}(t)$ are evaluated once for each half step, resulting in a slight increase in computer time as compared with MTT1. Thus Eq. (3.9a) and Eq. (3.9b) can be combined to eliminate $\bar{\Psi}(\Delta t)$ and the following equation is obtained:

$$\begin{aligned}\Omega(2\Delta t)\bar{\Psi}(2\Delta t) = {} & [\underline{I}-\Delta t(\underline{A}_4-0.5\dot{\underline{\Omega}}(2\Delta t)\underline{\Omega}^{-1}(2\Delta t))]^{-1} \\ & \cdot [\underline{I}+\Delta t(\underline{A}_3-0.5\dot{\underline{\Omega}}(2\Delta t)\Omega^{-1}(2\Delta t))]\underline{\Omega}(2\Delta t)\underline{\Omega}^{-1}(\Delta t) \\ & \cdot [\underline{I}-\Delta t(\underline{A}_2-0.5\dot{\underline{\Omega}}(\Delta t)\underline{\Omega}^{-1}(\Delta t))]^{-1} \\ & \cdot [\underline{I}+\Delta t(\underline{A}_1-0.5\dot{\underline{\Omega}}(\Delta t)\underline{\Omega}^{-1}(\Delta t))]\underline{\Omega}(\Delta t)\bar{\Psi}(o)\end{aligned} \tag{3.17}$$

where the diagonal matrices $\underline{\Omega}(\Delta t)$ and $\underline{\Omega}(2\Delta t)$ are evaluated at $t=\Delta t$ and $t=2\Delta t$ respectively. Likewise, their time derivatives $\dot{\underline{\Omega}}(\Delta t)$ and $\dot{\underline{\Omega}}(2\Delta t)$ are also evaluated at $t=\Delta t$ and $t=2\Delta t$ respectively. It should be pointed out that in MTT2, in contrast to MTT1, no further assumptions are invoked other than the ones previously mentioned in Chapter 2. Since

$$\bar{\Phi}(2\Delta t) = \underline{\Omega}(2\Delta t)\bar{\Psi}(2\Delta t)$$

and

$$\bar{\Phi}(o) = \bar{\Psi}(o)$$

Eq. (3.17) can be written as

$$\bar{\Phi}(2\Delta t)= \underline{B}_2\bar{\Phi}(o) \tag{3.18a}$$

where the advancement matrix, $\underline{B}_2$, for MTT2 is given by

$$\begin{aligned}\underline{B}_2 = {} & [\underline{I}-\Delta t(\underline{A}_4-0.5\dot{\underline{\Omega}}(2\Delta t)\underline{\Omega}^{-1}(2\Delta t))]^{-1}\ [\underline{I}+\Delta t(\underline{A}_3-0.5\dot{\underline{\Omega}}(2\Delta t)\underline{\Omega}^{-1}(2\Delta t))] \\ & \cdot\ \underline{\Omega}(2\Delta t)\underline{\Omega}^{-1}(\Delta t)[\underline{I}-\Delta t(\underline{A}_2-0.5\dot{\underline{\Omega}}(\Delta t)\underline{\Omega}^{-1}(\Delta t))]^{-1} \\ & \cdot\ [\underline{I}+\Delta t(\underline{A}_1-0.5\dot{\underline{\Omega}}(\Delta t)\underline{\Omega}^{-1}(\Delta t))]\underline{\Omega}(\Delta t)\end{aligned} \tag{3.18b}$$

The advancement matrix for MTT2 can also be expanded in a Taylor series to yield

$$\underline{B}_2 = \underline{I}+\Delta t(2\underline{A}-\underline{W}_1-\underline{W}_2+\underline{C}_1+\underline{C}_2) + O(\Delta t^2) \tag{3.19}$$

where

$$\begin{aligned}\underline{W}_1 &= \dot{\underline{\Omega}}(\Delta t)\underline{\Omega}^{-1}(\Delta t) \\ \underline{W}_2 &= \dot{\underline{\Omega}}(2\Delta t)\underline{\Omega}^{-1}(2\Delta t) \\ e^{\Delta t\underline{C}_1} &= \underline{\Omega}(\Delta t)\end{aligned} \tag{3.20}$$

and

$$e^{\Delta t\underline{C}_2} = \underline{\Omega}(2\Delta t)$$

Comparing Eq. (3.19) with Eq. (3.14), it can be seen that the temporal truncation error for MTT2 is also proportional to Δt.

$$T.E._2 = \Delta t(\underline{W}_1+\underline{W}_2-\underline{C}_1-\underline{C}_2) + O(\Delta t^2) \tag{3.21}$$

3.2c MTT3

The previous two cases, MTT1 and MTT2, have been shown to have temporal truncation errors over one time step proportional to Δt. The numerical results of some test problems, to be presented in Chapter 4, indicated that their performance, as regards stability, was different from that of the ADEP methods, with or without the simple frequency transformation. The temporal truncation error in the standard ADEP method has been shown by Denning[9] to be proportional to Δt^2. Similarly, the ADEP method with the simple frequency transformation option also has an truncation error proportional to Δt^2.

In the MTT3 algorithm an attempt is made to minimize the temporal order of the truncation error with the following assumptions:

$$\underline{\Omega}(2\Delta t)\underline{\Omega}^{-1}(\Delta t) = \underline{\Omega}(\Delta t)\underline{\Omega}^{-1}(o) = e^{\Delta t \underline{W}} \qquad (3.22a)$$

where the matrix $\underline{W}$ is defined as

$$\underline{W} = \dot{\underline{\Omega}}(\Delta t)\underline{\Omega}^{-1}(\Delta t) = \dot{\underline{\Omega}}(2\Delta t)\underline{\Omega}^{-1}(2\Delta t) \qquad (3.22b)$$

Here the diagonal elements of $\dot{\underline{\Omega}}(\Delta t)$ and $\underline{\Omega}(\Delta t)$ are evaluated at $t=\Delta t$, but their values are used for both half-steps. Thus it is assumed that the effective frequencies, $\underline{W}$, are the

same for both half-steps. Based on these assumptions, Eq. (3.9a) and Eq. (3.9b) are combined to yield

$$\underline{\Omega}(2\Delta t)\bar{\psi}(2\Delta t) = [\underline{I}-\Delta t(\underline{A}_4-0.5\underline{W}]^{-1}\ [\underline{I}+\Delta t(\underline{A}_3-0.5\underline{W})]e^{\Delta t\underline{W}}$$
$$[\underline{I}-\Delta t(\underline{A}_2-0.5\underline{W}]^{-1}\ [\underline{I}+\Delta t(\underline{A}_1-0.5\underline{W})]e^{\Delta t\underline{W}}\bar{\psi}(o) \qquad (3.23)$$

Eq. (3.23) can be represented as

$$\bar{\Phi}(2\Delta t) = \underline{B}_3\ \bar{\Phi}(o) \qquad (3.24a)$$

where the advancement matrix, $\underline{B}_3$, for MTT3 is defined as

$$\underline{B}_3 = [\underline{I}-\Delta t(A_4-0.5\underline{W})]^{-1}\ [\underline{I}+\Delta t(\underline{A}_3-0.5\underline{W})]e^{\Delta t\underline{W}}$$
$$\cdot\ [\underline{I}-\Delta t(\underline{A}_2-0.5\underline{W})]^{-1}\ [I+\Delta t(\underline{A}_1-0.5\underline{W})]e^{\Delta t\underline{W}} \qquad (3.24b)$$

This advancement matrix can be expanded in a Taylor series to yield

$$\underline{B}_3 = \underline{I}+2\Delta t\underline{A}+\Delta t^2(\underline{A}^2+\underline{A}_2\underline{A}+\underline{A}_4\underline{A}-\underline{W}\underline{A}+\underline{A}\underline{W})+O(\Delta t^3) \qquad (3.25)$$

Thus the temporal error for MTT3 is proportional to Δt^2.

$$T.E._3 = \Delta t^2(\underline{A}^2-\underline{A}_2\underline{A}-\underline{A}_4\underline{A} + \underline{W}\underline{A}-\underline{A}\underline{W}) + O(\Delta t^3) \qquad (3.26)$$

It should be pointed out that the assumptions, Eq. (3.22a) and Eq. (3.22b), are somewhat arbitrary and have modified the basic formalism of the multi-mode

temporal transformation technique. The numerical algorithm MTT3 is mainly intended to provide more information on the relation between the temporal truncation error and numerical stability.

Though it has been mentioned that the multiple temporal-mode transformation technique is applied in this study with an ADE method, the temporal truncation error analyses just described for MTT1 and MTT2 are also applicable to an ADI method. As can be seen from Eq. (3.16) and Eq. (3.21), the expressions involving Δt are independent of the splitting of the $\underline{A}$ matrix and hence the equations hold for both ADE and ADI methods.

To summarize then, for all three algorithms, the pointwise spatial moments and frequencies are computed at the beginning of each time step as described in Appendix B. These are assumed constant within the time step. The correction functions are assumed to be updated at the end of each half-step according to the finite differencing scheme, Eqs. (3.9a) and (3.9b), though they are not directly calculated in the code ADEP. Thus the solution at time $t+\Delta t$ can be found, based on the known flux and precursor concentration distributions at time t. The three algorithms differ in the way the amplitude function is represented.

CHAPTER 4

NUMERICAL RESULTS

As stated in the introduction, the purpose of this study is to assess the applicability of the multiple temporal-mode transformation method to the analysis of transient behavior in CANDU-type reactors. It was found in the early stages of this project that the method was numerically stable only at time steps considerably smaller than those required for accurate solutions with the existing ADEP code which uses a simple exponential transformation. Much effort has been devoted to improving the numerical methods in an attempt to avoid the problem of instability at larger time steps, with limited success. This chapter presents the numerical results obtained with the method for four test cases. The ADEP code was used to generate solutions for comparison purposes. For convenience in later discussions, the standard method in ADEP with no frequency transformation is referred to hereafter as Standard ADEP. ADEP-Exponential refers to the option in ADEP which uses a simple group-dependent frequency transformation. All computer programs were run on the CDC 6600.

The first three cases considered are homogeneous problems with no spatial change in the neutron flux. They

served as simplified tests of the multiple temporal-mode transformation technique. The last test case is a space-dependent problem with properties based on a CANDU-type reactor. All four test cases are specified in Cartesian coordinates. The boundary condition for all test cases is of the homogeneous Dirichlet type, with zero flux on the outer reactor boundary. It should also be noted that in the notation employed, one time step is represented by $2\Delta t$.

At time steps for which the methods MTT1, MTT2 or MTT3 were unstable, the numerical difficulty occurred in one of the following forms:

a) The argument in the exponential $\exp(\alpha\Delta t)$ is too large for the machine to handle.

b) A singular matrix during normalization of the eigenvectors (see Appendix B).

c) Very large positive or negative fluxes, followed by (b).

4.1 Test Case 1, Homogeneous Slab

Geometry and composition: Appendix C.1

This test case represents a bare homogeneous slab reactor, 240 cm in width, with two neutron groups and one precursor group. A 10 cm uniform spatial mesh was used.

The initial core was supercritical by ~ 9.5 mk and no further perturbation was applied.

The initial cosine flux distribution was normalized to unity at the reactor centre. The exact solution to the semi-discrete equations of this problem was generated by an eigenvector expansion technique. By finite differencing in space, the exact leakage was approximated by replacing $\nabla^2\bar{\phi}$ with $-B^2\bar{\phi}$ where the buckling, B^2, is given by[4]

$$B^2=\frac{2}{\Delta x^2}\left(1 - \cos\frac{\pi}{K}\right)$$

In the above expression, Δx is the mesh spacing and K is the number of mesh intervals in the x direction.

Figure 1 shows the time behavior of the centre point thermal flux. For this homogeneous problem with no spatial change, it also represents the relative increase in flux, $\phi_g(t)/\phi_g(o)$, for both energy groups at each mesh point.

It has been shown in Section 3.2 that the MTT1 and MTT2 methods have temporal truncation errors proportional to Δt whereas with MTT3 it is proportional to Δt^2. The numerical results of this test case indicated that the performance of MTT1 and MTT2 was clearly different from that of MTT3.

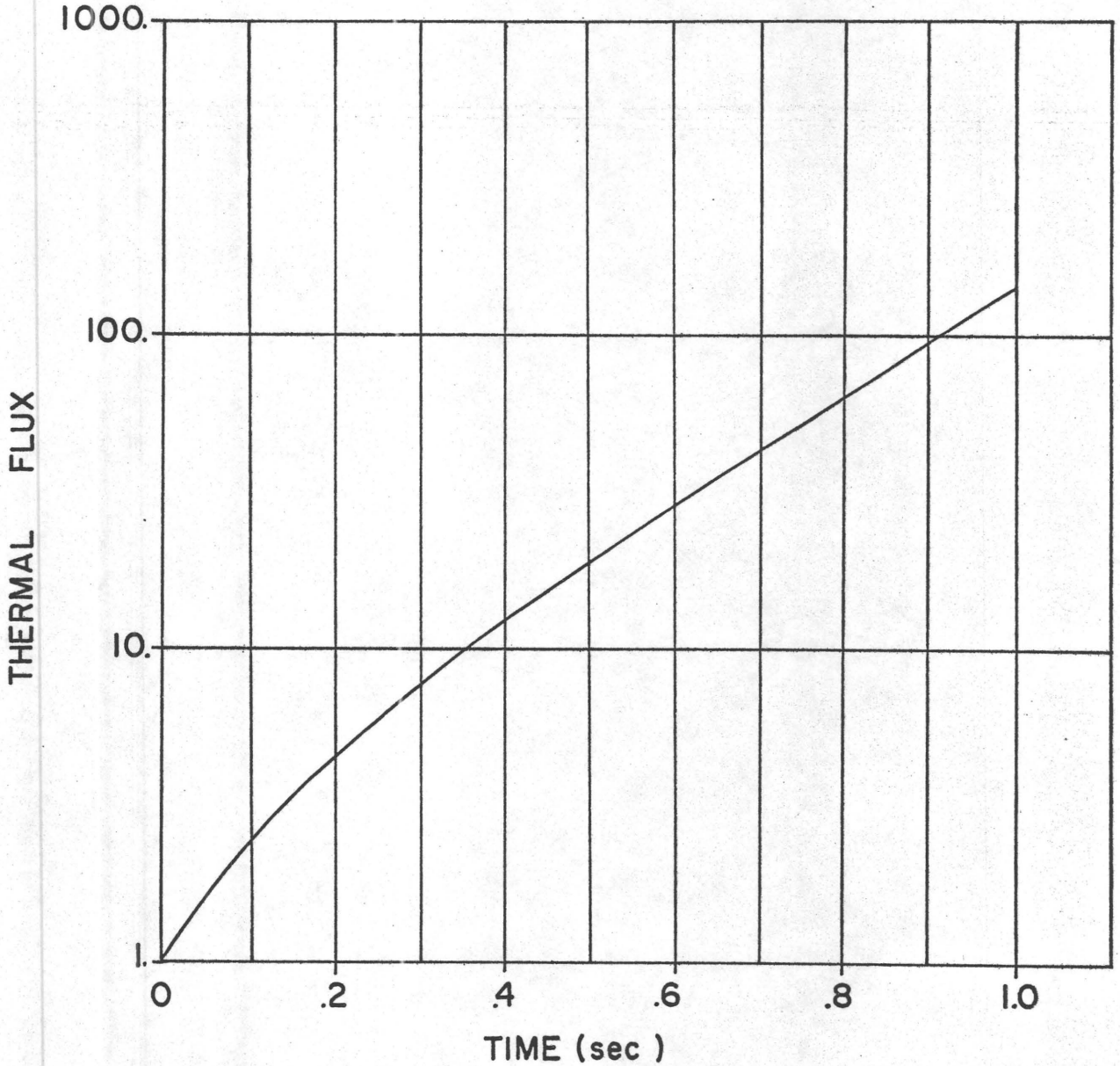


FIG. 1: TIME BEHAVIOR OF THERMAL FLUX AT REACTOR CENTRE POINT FOR $\rho = 9.5$ mk

Table 1.1 shows the centre point thermal flux obtained by MTT1 and MTT2 at time steps for which they were stable. At Δt = 5 μsec and at Δt = 10 μsec, the neutron fluxes obtained by MTT2 at all mesh points agreed with the exact solutions to seven significant digits. Likewise, at Δt = 5 μsec, MTT1 gave solutions with the same accuracy at all mesh points. At Δt = 8 μsec, the fast and thermal fluxes obtained by MTT1 at points near the boundary had large errors at the start of the transient. At t = 0.4 msec, the fast flux at mesh point 24 differed from the reference by ~ 40% (see Table 1.2a and Table 1.2b). After 1000 time steps, at 0.016 sec into the transient, the group fluxes approached the exact solution with less than 1.0% error at all mesh points.

Table 1.2a and Table 1.2b summarize the time-dependent behavior of the fast flux at mesh points 2 and 24 respectively. These points are symmetric about the reactor centre and should exhibit an identical transient response. For this test case, MTT1 was unstable at Δt = 10 μsec and MTT2 unstable at Δt = 20 μsec. At these time steps, the transient could be calculated accurately for a few steps, but after some 20 to 50 steps, large errors accumulated causing the runs to abort. The results obtained by MTT1 at Δt = 8 μsec became more symmetric after 1000 time

Table 1.1 Centre Point Thermal Flux

Time (sec)	Exact	Δt (sec)			
		MTT1		MTT2	
		5×10^{-6}	8×10^{-6}	5×10^{-6}	1×10^{-5}
0	1.0	1.0	1.0	1.0	1.0
.004	1.047541	1.047541	1.054157	1.047541	1.047541
.008	1.095943	1.095943	1.105021	1.095943	1.095943
.012	1.145079	1.145079	1.154554	1.145079	1.145079
.016	1.194961	1.194961	1.204600	1.194961	1.194961
.020	1.245599	1.245599	--------	1.245599	1.245599

Table 1.2a Fast Flux (x 10) at Mesh Point 2

Time (sec)	Exact	Δt (sec) MTT2 2×10^{-5}	MTT1 8×10^{-6}	MTT1 1×10^{-5}	MTT3 2×10^{-4}
0	.7374471	.7374471	.7374471	.7374471	.7374471
.0004	.7478335	.7478335	.7478205	.7477986	.7424215
.0008	.7513417	.7513417	.7513350	(a)	.7468733
.0012	.7548552	.7548546	.7548566		.7508999
.0016	.7583739	.7581164	.7583912		.7546847
.002	.7618980	.6233239	.7619445		.7584131
.006	.7974313	(a)	.7992457		.7957506
.01	.8335033		.8379445		(a)
.016	.8886397		.8950558		

Table 1.2b Fast Flux (x 10) at Mesh Point 24

Time (sec)	Exact	Δt (sec) MTT2 2×10^{-5}	MTT1 8×10^{-6}	MTT1 1×10^{-5}	MTT3 2×10^{-4}
0	.7374471	.7374471	.7374471	.7374471	.7374471
.0004	.7478335	.7478335	.4352222	1.689272	.7442919
.0008	.7513417	.7513417	.8972262	(a)	.7493673
.0012	.7548552	.7548552	.8485059		.7532372
.0016	.7583739	.7583727	.8259797		.7565637
.002	.7618980	.7616590	.8141915		.7599961
.006	.7974313	(a)	.8130789		.7992400
.010	.8335033		.8426998		(a)
.016	.8886397		.8961805		

(a) Program aborted because of numerical difficulty.

steps but still errors of ~ 1% were present.

Tables 1.3a, 1.3b and 1.3c show the centre point thermal flux obtained by MTT3, ADEP-Exponential and Standard ADEP. MTT3 was stable at $\Delta t \leq 0.1$ msec with less than 0.1% error at 0.1 sec into the transient. Standard ADEP gave solutions with the same accuracy but at a smaller time step, $\Delta t = 10$ μsec. The performance of ADEP-Exponential was superior to both methods. At $\Delta t = 0.2$ msec, ADEP-Exponential gave solutions comparable in accuracy to those of MTT3 at $\Delta t = 0.1$ msec. The increase in computer CPU time was ~ 10% when the frequency transformation was used in ADEP whereas application of the multiple temporal-mode transformation required considerably more time (see Table 1.5).

Figure 2 plots the percentage error in the centre point thermal flux as a function of time step size for the MTT3, Standard ADEP and ADEP-Exponential methods at 0.02 sec and 0.1 sec into the transient. The plots show that for an identical time step size, Standard ADEP has the largest error when compared to MTT3 and ADEP-Exponential. In addition, the plots for Standard ADEP have a slope of ~ 2. This clearly indicates that the rate of convergence is proportional to Δt^2. The slopes of the plots for MTT3 and ADEP-Exponential cannot be so well defined. It is believed that the frequency transformation

Table 1.3a Centre Point Thermal Flux by MTT3

Time (sec)	Exact	Δt (sec) 2×10^{-5}	4×10^{-5}	5×10^{-5}	1×10^{-4}
0	1.0	1.0	1.0	1.0	1.0
.01	1.12042	1.12040	1.12036	1.12032	1.12007
.02	1.24560	1.24558	1.24552	1.24547	1.24517
.04	1.51058	1.51054	1.51046	1.51040	1.51003
.06	1.79637	1.79631	1.79621	1.79614	1.79571
.08	2.10466	2.10459	2.10447	2.10439	2.10389
.10	2.43731	2.43722	2.43708	2.43699	2.43642

Table 1.3b Centre Point Thermal Flux by ADEP-Exponential

Time (sec)	Exact	Δt (sec) 5×10^{-5}	1×10^{-4}	2×10^{-4}	5×10^{-4}	1×10^{-3}
0	1.0	1.0	1.0	1.0	1.0	1.0
.01	1.12042	1.12038	1.12028	1.11961	-------	1.08734
.02	1.24560	1.24554	1.24542	1.24472	1.23251	1.19058
.04	1.51058	1.51048	1.51032	1.50957	1.49860	1.42647
.06	1.79637	1.79622	1.79603	1.79523	1.78476	1.69702
.08	2.10466	2.10448	2.10424	2.10340	2.09330	1.99911
.10	2.43731	2.43709	2.43681	2.43590	2.42613	2.33092

Table 1.3c Centre Point Thermal Flux by Standard ADEP

Time (sec)	Exact	Δt (sec) 1×10^{-5}	2×10^{-5}	5×10^{-5}	1×10^{-4}	2×10^{-4}
0	1.0	1.0	1.0	1.0	1.0	1.0
.01	1.12042	1.12036	1.12020	1.11907	1.11530	1.10348
.02	1.24560	1.24549	1.24517	1.24292	1.23533	1.21043
.04	1.51058	1.51035	1.50966	1.50493	1.48885	1.43501
.06	1.79637	1.79600	1.79490	1.78731	1.76148	1.67471
.08	2.10466	2.10414	2.10257	2.09170	2.05472	1.93057
.10	2.43731	2.43661	2.43449	2.41988	2.37018	2.20375

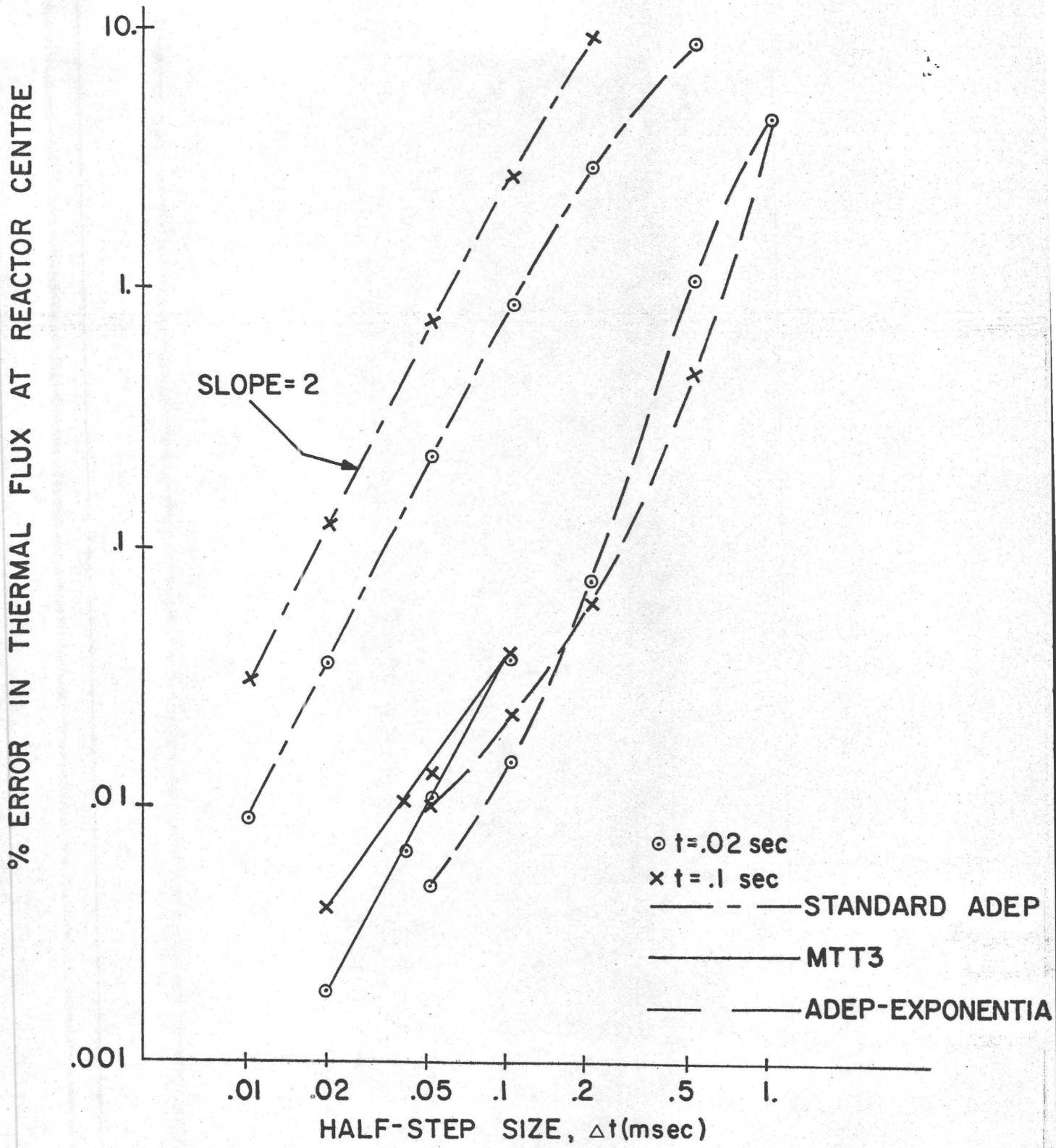


FIG. 2: PERCENTAGE ERROR VERSUS TIME STEP SIZE AT t = .02 sec AND AT t = .1 sec

has complicated the accumulation of the truncation error[6]. Both MTT1 and MTT2 were very accurate with stable time step sizes and hence the percentage error has not been plotted. A small increase in time step size from stable values has caused the MTT1 and MTT2 methods to become unstable.

One reason for numerical instability is the so-called feedback effect of the frequencies[6]. Recall that the spatial moments and frequencies are computed at the beginning of each time step by solving for the eigenvectors and eigenvalues of the coefficient matrix $\underline{M}'(\bar{r},o)$ in Eq. (2.13). The diagonal elements of the matrix $\underline{M}'(\bar{r},o)$ are of the form

$$v_g(E_g + \Sigma_{gg})$$

where $E_g\phi_g$ is the group g neutron leakage at the beginning of the time step. For a one-dimensional problem, E_g at a mesh point j is defined as

$$E_{g,j} = \left(D_{g,j+1} \frac{\phi_{g,j+1} - \phi_{g,j}}{\Delta x_{j+1}^2} - D_{g,j} \frac{\phi_{g,j} - \phi_{g,j-1}}{\Delta x_j^2} \right) \Big/ \phi_{g,j}$$

where Δx_j and $D_{g,j}$ etc. are the appropriate mesh spacings and diffusion coefficients associated with the mesh point j. Thus, the percentage errors in the computed $E_{g,j}$

can be large, depending on the magnitudes and distribution of the errors in the neutron fluxes. Table 1.4 compares the exact values of E_g with the cross sections Σ_{gg}. It can be seen that because E_g and Σ_{gg} are comparable, large errors in E_g can introduce large errors in the diagonal elements, $E_g + \Sigma_{gg}$. This may lead to inaccurately computed frequencies which in turn affect the accuracy of the solutions at the end of the time step. This point is demonstrated by the following observations. By replacing the terms $E_{g,j}$ with their exact values, $-D_g B^2$, the time step size required for stable and accurate solutions could be increased from 5 μsec to 0.5 msec for MTT1 and from 10 μsec to 0.01 sec for MTT2. Only at small enough time steps will the frequency feedback effect be insignificant.

Table 1.4 Removal Cross Sections and Exact E_g Terms

Group	Σ_{gg}	Exact E_g
1	-.8154 E-2	-.2241 E-3
2	.7344 E-3	-.1488 E-3

Table 1.5 shows the computer CPU time for this test case. The times quoted were based on runs of 1000 time steps on the CDC 6600. The computer time requirements for the

different methods will be discussed in detail in Section 4.6.

Table 1.5 CPU Time for 1000 Time Steps

	Standard ADEP	ADEP-Exponential	MTT1	MTT2	MTT3
CPU Time (sec)	22.2	24.9	241.9	252.4	246.8

4.2 Test Case 2, Homogeneous Slab

Geometry and composition: Appendix C.2

The multiple temporal-mode transformation technique has previosly been applied by Garland et al,[7] Harms et al,[8] and Harding[10] with an implicit method to problems involving two energy groups and no delayed neutrons. In their works, small time steps in the range of microseconds were used. This test case is intended primarily to demonstrate that the instability problem encountered in test case 1 is not peculiar to the reactor model used in test case 1, which included one delayed precursor group.

The geometry of this test case is identical to that of test case 1. It differs only in that no precursor group is considered. The remaining lattice parameters are the same as in case 1. Only MTT1, MTT3 and ADEP-Exponential

were tested on this problem. The exact solution to the semi-discrete equations was also obtained by an eigen-vector expansion technique as in case 1.

The numerical results with respect to accuracy and stability indicated that the performance of each method was similar to that in test case 1. It is expected that MTT2 and Standard ADEP will exhibit the same characteristics as they did in case 1.

The centre point thermal flux is summarized in Table 2.1. Tables 2.2a and 2.2b give the fast flux at two points symmetric about the reactor centre. At 0.1 sec into the transient, the solutions obtained by MTT3 or ADEP-Exponential at the symmetric points were identical and differed from the exact solution by less than 0.05%. At the same step size, ADEP-Exponential results were more accurate than the MTT3 results. This agreed with the observation in test case 1. MTT1 at Δt = 8 μsec again gave poor results. Though the fast fluxes at points near the boundary converged, they showed a 1.0% error at 0.016 sec into the transient. The centre point thermal flux at this time also showed a 1.0% error.

At Δt = 5 μsec, all three methods gave solutions which agreed with the exact solution to seven significant

Table 2.1 Centre Point Thermal Flux

Time (sec)	Exact	Δt(sec) MTT1	MTT3		ADEP-Exponential	
		8×10^{-6}	1×10^{-5}	1×10^{-4}	5×10^{-5}	1×10^{-4}
0.	1.0	1.0	1.0	1.0	1.0	1.0
.004	1.04831	1.05513	1.04830	1.04798	1.04830	1.04821
.008	1.09911	1.10876	1.09910	1.09871	1.09910	1.09901
.016	1.20821	1.21914	1.20820	1.20783	1.20820	1.20810
.02	1.26676	-------	1.26675	1.26636	1.26674	1.26664
.10	3.26371	-------	-------	3.26267	3.26367	3.26340

Table 2.2a Fast Flux (x 10) at Mesh Point 2

Time (sec)	Exact	Δt (sec) MTT1	MTT3		ADEP-Exponential	
		8×10^{-6}	1×10^{-5}	1×10^{-4}	5×10^{-5}	1×10^{-4}
0.	.737447	.737447	.737447	.737447	.737447	.737447
.004	.780398	.781008	.780392	.779753	.780270	.779784
.008	.818214	.821687	.818208	.817775	.818154	.817924
.016	.899433	.906762	.899424	.899112	.899413	.899308
.02	.943017	-------	.943007	.942702	.943002	.942910
.10	2.42961	-------	-------	2.42884	2.42959	2.42938

Table 2.2b Fast Flux (x 10) at Mesh Point 24

Time (sec)	Exact	Δt (sec) MTT1	MTT3		ADED-Exponential	
		8×10^{-6}	1×10^{-5}	1×10^{-4}	5×10^{-5}	1×10^{-4}
0.	.737447	.737447	.737447	.737447	.737447	.737447
.004	.780398	.805834	.780374	.780609	.780506	.780792
.008	.818214	.830293	.818201	.820910	.818254	.818344
.016	.899433	.908064	.899422	.897728	.899431	.899385
.02	.943017	-------	.943007	.942843	.943010	.942944
.10	2.42961	-------	-------	2.42884	2.42959	2.42938

digits. MTT1 was unstable for a $\Delta t \geq 10$ μsec and MTT3 unstable for $\Delta t \geq 0.2$ msec.

Table 2.3 shows the computer time for this test case, based on runs of 1000 time steps on the CDC 6600. Numerically, this test case was adapted from case 1 by simply setting the delayed neutron fraction, β, to zero. For ADEP-Exponential, the same amount of computational effort was involved in both cases and therefore the computer time required was the same (see Table 1.5 and Table 2.3). In contrast, the computer time required for MTT1 or MTT3 in this case was greatly reduced from that in case 1.

Appendix B describes the procedure for determining the eigenvalues and eigenvectors of the coefficient matrix at a mesh point. In a system with no delayed neutrons, the number of spatial moments and frequencies are reduced. The order of the coefficient matrix is then G instead of G + I, G and I being the number of energy groups and precursor groups respectively. Likewise, fewer equations are involved in normalizing the eigenvectors. For MTT1 or MTT3, there is a large difference in computer time between test cases 1 and 2. This indicates that the calculation of the spatial moments

and frequencies accounts for a significant part of the computer time.

Table 2.3 CPU Time for 1000 Time Steps

CPU Time (sec)	ADEP-Exponential	MTT1	MTT3
	24.9	130.4	131.3

4.3 Test Case 3, Homogeneous Square

Geometry and composition: Appendix C.3

Perturbation: Step change $\Delta \Sigma_2 = -.369 \times 10^{-4}$ cm^{-1}

This test case represents a bare homogeneous square reactor, 200 cm on a side, with two neutron groups and one precursor group. A uniform mesh spacing of 20 cm resulted in an 11 x 11 spatial mesh. The test case was adapted from a three-dimensional test case used by Ferguson[6], with the buckling $D_g B^2$ added to the absorption cross sections to account for neutron leakage in the z-direction. The initial configuration was made critical by dividing the production cross sections by the critical k_{eff}. The perturbation was introduced by a step decrease in the thermal group absorption cross section. It had a reactivity worth of ~ 5 mk.

Only a short transient was simulated in this test case. Since numerical results for such a short transient were not given by Ferguson, a comparison of results between the multi-mode temporal transformation method and the 3DKIN method[6] was not attempted. It should be noted that the initial reactor power was normalized to a different value in this test case. The solutions obtained by ADEP-Exponential at a very small time step, Δt = 1 μsec, were used as reference solutions. The Standard ADEP method was not tested in this case.

Tables 3.1a to 3.1c summarize the centre point thermal flux obtained by the different methods. MTT1 was found to be unstable at Δt = 8 μsec in this case and MTT2 was unstable at Δt = 15 μsec. As can be seen from Table 3.1a, the results for MTT1 and MTT2 were very accurate at stable time steps. MTT3 was found to be unstable at Δt = 0.2 msec. At Δt = 0.1 msec, both ADEP-Exponential and MTT3 were accurate, with less than 0.1% error in the centre point thermal flux at 0.04 sec into the transient. The results produced by MTT3 were slightly more accurate in this case.

At time steps for which MTT1, MTT2 or MTT3 were unstable, the numerical results showed an oscillation

Table 3.1a Centre Point Thermal Flux

Time (sec)	Reference (*)	MTT1 $\Delta t = 5 \times 10^{-6}$	MTT2 $\Delta t = 1 \times 10^{-5}$
0.	.6600856	.6600856	.6600856
.002	.6706579	.6706579	.6706579
.004	.6811847	.6811848	.6811848
.006	.6916669	.6916671	.6916671
.008	.7021047	.7021049	.7021049
.01	.7124983	.7124986	.7124986

(*) Reference Solution obtained by ADEP-Exponential at Δt = 1μsec

Table 3.1b Centre Point Thermal Flux

Time (sec)	MTT3, Δt (sec) 1×10^{-5}	1×10^{-4}
0.	.6600856	.6600856
.002	.6706574	.6706592
.004	.6811838	.6811812
.006	.6916655	.6916590
.008	.7021029	.7020924
.01	.7124961	.7124818

Table 3.1c Centre Point Thermal Flux

Time (sec)	ADEP-Exponential, Δt (sec)		
	1×10^{-5}	1×10^{-4}	2×10^{-4}
0.	.6600856	.6600856	.6600856
.002	.6706572	.6704483	.6697463
.004	.6811836	.6809919	.6799072
.006	.6916654	.6914786	.6902214
.008	.7021027	.7019144	.7006170
.01	.7124959	.7123071	.7110284

Table 3.2 Fast Flux Distribution Along Horizontal Mesh Line k=6 for MTT2 at $\Delta t = 2 \times 10^{-5}$ sec(**)

X (cm)	t = .0002 sec		t = .0004 sec		t = .0006 sec	
	Ref.(*)	MTT2	Ref.(*)	MTT2	Ref.(*)	MTT2
0	0.	0.	0.	0.	0.	0.
20	.55803	.55803	.55891	.55886	.55980	.54813
40	1.0614	1.0614	1.0631	1.0631	1.0648	1.0971
60	1.4609	1.4609	1.4633	1.4634	1.4656	1.4067
80	1.7174	1.7174	1.7202	1.7200	1.7229	1.7881
100	1.8058	1.8058	1.8087	1.8089	1.8116	1.7605
120	1.7174	1.7174	1.7202	1.7201	1.7229	1.7290
140	1.4609	1.4609	1.4633	1.4631	1.4656	1.4782
160	1.0614	1.0614	1.0631	1.0633	1.0648	1.0392
180	.55803	.55803	.55891	.55884	.55980	.57137
200	0.	0.	0.	0.	0.	0.

(**) MTT2 Unstable at $\Delta t = 2 \times 10^{-5}$ sec.

(*) Reference Solution obtained by ADEP-Exponential at $\Delta t = 1 \times 10^{-6}$ sec.

about the reference solutions from mesh point to mesh point. Mesh points close to the reactor boundary were the first to show signs of instabilities. The errors at these points then propagated towards the inner mesh points. After some 20 to 50 steps, numerical difficulty was encountered and caused the run to abort. To demonstrate this point, the fast flux distribution obtained by MTT2 at Δt = 20 μsec is shown in Table 3.2. This run aborted after ~ 15 time steps. It can be seen that just before the run terminated, the magnitude of the oscillation in the neutron flux was still small. The error at each mesh point was less than 5%, but such errors were large enough to cause numerical difficulty in the next few time steps. In this example, a singularity in the matrix occurred at t = 0.00064 sec.

Table 3.3 lists the computer times required in this test case for runs of 500 time steps on the CDC 6600.

Table 3.3 CPU Time for 500 Time Steps

CPU Time (sec)	ADEP-Exponential	MTT1	MTT2	MTT3
	43.8	424.7	441.3	424.2

4.4 Test Case 4, 1-D Space-Dependent Problem

Geometry and composition: Appendix C.4

Perturbation: Regions 2 and 3

$$\frac{\partial \Sigma_2}{\partial t} = -1.0 \times 10^{-4} \ (cm^{-1}sec^{-1}), \ t \leq 0.4 \text{ sec}$$

This test case is a one-dimensional space-dependent problem, with two energy groups and two precursor groups. The problem represents an idealized configuration of a CANDU-type reactor. Non-uniform mesh spacings of 15 and 30 cm form a 33 point spatial mesh (see Appendix C.4). Regions 2, 3, 4 and 5 represent the core and regions 1 and 6 the reflector (see Fig. 3).

The test case was adapted from a two-dimensional CANDU benchmark problem. The 2-D benchmark problem simulates voiding in one half of the core, regions 2 and 3, for 0.6 sec, followed by subsequent shut off rod insertion in the central part of the core, regions 3 and 4, from 0.6 to 1.5 sec. Because of the large amount of computer time required by the multiple temporal-mode transformation technique to follow the complete transient, only the initial part of the transient was calculated in this test case.

The initial reactor configuration was made critical by dividing the production cross sections by the critical k_{eff}. The initial flux distribution is shown in Fig. 3. The perturbation was introduced by a ramp decrease in the thermal group absorption cross section in regions 2 and 3. The transient was followed for 0.04 sec.

Tables 4.1a and 4.1b show the thermal flux at the interface between region 2 and region 3. At Δt = 5 μsec, all four methods gave solutions which agreed to six significant digits. The results obtained at this time step can be regarded as the reference solutions. ADEP-Exponential performed well for this problem. For a Δt = 0.001 sec, the peak thermal flux obtained by ADEP-Exponential had an error of less than 0.1% at 0.04 sec into the transient. For Δt = 10 μsec, MTT3 had an error of ~ 0.01% in the peak thermal flux. In contrast, at Δt = 10 μsec, MTT1 and MTT2 had errors of ~ 0.3% at t = 0.04 sec. At this time step, Δt = 10 μsec, all three methods, MTT1, MTT2 and MTT3 had large percentage errors in the neutron flux at points in the reflector, region 1. As an example, Table 4.2 shows the fast flux distribution at 0.04 sec obtained by MTT1, MTT2 and MTT3. The percentage error in the fast flux at mesh point 2 was ~ 7%. The thermal flux, not shown here, had a smaller percentage error of

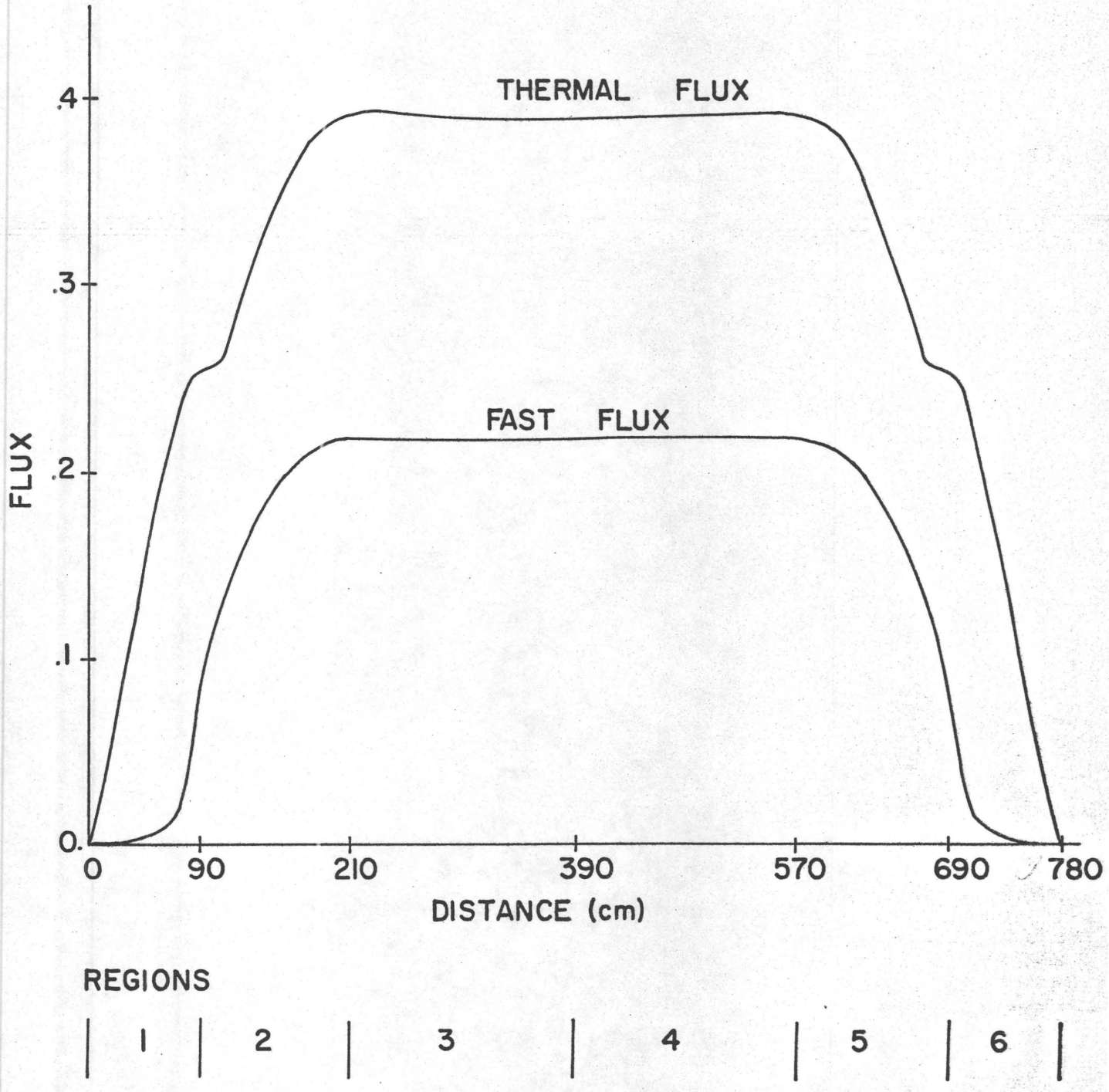


FIG. 3: INITIAL FLUX DISTRIBUTION

Table 4.1a Thermal Flux at x = 210 cm

Time (sec)	ADEP-Exponential, Δt (sec)				
	5×10^{-6}	1×10^{-5}	1×10^{-4}	2×10^{-4}	1×10^{-3}
0.	.392043	.392043	.392043	.392043	.392043
.01	.392598	.392599	.392608	.392618	.392685
.02	.394144	.394145	.394161	.394179	.394286
.03	.396533	.396534	.396556	.396580	.396715
.04	.399659	.399660	.399687	.399716	.399873

Table 4.1b Thermal Flux at x = 210 cm

Time (sec)	MTT1, Δt (sec)		MTT2, Δt (sec)		MTT3
	5×10^{-6}	1×10^{-5}	5×10^{-6}	1×10^{-5}	1×10^{-5}
0.	.392643	.392043	.392643	.392043	.392043
.01	.392598	.392623	.392598	.392654	.392598
.02	.394144	.394551	.394144	.394632	.394055
.03	.396533	.397402	.396533	.397496	.396312
.04	.399659	.400942	.399659	.401043	.399317

Table 4.2 Fast Flux Distribution at t = .04 sec ($\Delta t = 1 \times 10^{-5}$ sec for MTT1, MTT2 and MTT3)

Mesh Point j	Reference(*)	MTT1	MTT2	MTT3
1	0.	0.	0.	0.
2	.53143E-3	.56920E-3	.57183E-3	.55934E-3
3	.47906E-2	.46904E-2	.47121E-2	.46091E-2
4	.19522E-1	.21238E-1	.21337E-1	.20870E-1
5	.68487E-1	.68266E-1	.68582E-1	.67083E-1
6	.12676	.12905	.12831	.12711
7	.15682	.15756	.15815	.15615
8	.17788	.17967	.17922	.17776
9	.19434	.19517	.19556	.19393
10	.21611	.21709	.21711	.21586
11	.22258	.22329	.22336	.22239
12	.22130	.22182	.22187	.22116
13	.22015	.22053	.22057	.22006
14	.21915	.21942	.21944	.21908
15	.21826	.21845	.21847	.21821
16	.21745	.21758	.21759	.21742
17	.21670	.21678	.21679	.21668
18	.21616	.21622	.21623	.21615
19	.21601	.21605	.21605	.21600
20	.21621	.21624	.21624	.21621
21	.21674	.21675	.21675	.21674
22	.21756	.21757	.21757	.21756
23	.21864	.21864	.21864	.21863
24	.21219	.21219	.21219	.21219

(*) Reference Solution obtained by all three methods at $\Delta t = 5 \times 10^{-6}$ sec.

~ 1 %. The runs for MTT1, MTT2 and MTT3 were followed for 2000 time steps at Δt = 10 μsec and were stable.

In this test case, all the methods using the multiple temporal-mode transformation technique were found to be numerically unstable at $\Delta t \geq 15$ μsec. For such time steps, MTT1, MTT2 and MTT3 followed the transient well immediately after initiation of the perturbation. However, after some 80 time steps, the mesh points in the right reflector, region 6, showed large percentage errors and eventually caused the runs to abort.

This test case differed in configuration from the previous cases in that reflector regions were included. The numerical results indicate that the reflector region is more susceptible to the instability problem. The fast flux gradients are large in the reflector region and hence the terms $E_{l,j,k}$, defined previously, and the absorption cross sections $\Sigma_{l,j,k}$ at mesh points (j,k) in the reflector region are comparable in magnitude. Thus, errors in the terms $E_{l,j,k}$ could have significant effects on the frequencies computed.

Table 4.3 compares the computer time required by the different methods to traverse 2000 time steps on the CDC 6600.

Table 4.3 CPU Time for 2000 Time Steps

CPU Time (sec)	ADEP-Exponential	MTT1	MTT2	MTT3
	73.5	1017.7	1038.0	1020.1

4.5 Computer Storage

As previously mentioned, the spatial moments and frequencies at each mesh point for all groups are computed at the beginning of each time step. In this study, the multiple temporal-mode transformation was applied only to the neutron groups and not to the precursor groups. Hence, the spatial moments for the energy groups, as well as all the frequencies, were stored. For a problem with G energy groups, I precursor groups and N mesh points, the frequencies require (G + I) x N Storage locations and the spatial moments (G + I) x G x N Storage locations. Computer memory must also be allocated for the matrix elements used in computing the eigenvalues and eigenvectors and in normalizing the eigenvectors (see Appendix B). The matrix $\underline{F}'$ in Eq. (B.11) requires $(G + I)^2 \cdot [2(G + I) - 1]$ storage locations. Thus, for the numerical algorithms in this study, the minimum additional computer storage required was

$$(G + I) \times (G + 1) \times N + (G + I)^2 \times [2(G + I) - 1]$$

A realistic three-dimensional reactor kinetics problem may include two energy groups and six precursor groups. A spatial mesh of 33 x 33 x 24 points may be required for a detailed representation of the reactor configuration and composition. The additional computer storage for such a problem will then be 628224 locations. The approach used in this study is obviously inadequate for a realistic problem. However, it is believed that with some programming effort, the storage requirement for the spatial moments and frequencies can be eliminated. Only the effective exponential terms, the effective frequencies and the elements of the coefficient matrices would then need be stored. The storage for the exponential terms and the effective frequencies is normally required in a kinetics code which employs a simple frequency transformation. Thus the additional computer storage required for the multi-mode temporal transformation method would be given by the size of the matrices used in computing the eigenvalues and eigenvectors and in normalizing the eigenvectors. For the realistic kinetics problem previously mentioned, this will amount to ~ 1000 locations. Hence, it can be concluded that the computer storage requirement is not a limiting factor in applying the multiple temporal-mode

transformation technique.

4.6 Computer Time

The computer time required for each method for the four test cases is summarized in Table 5. The MTT1, MTT2 and MTT3 methods require approximately the same computer time. As previously mentioned in Section 4.1, with ADEP the increase in computer CPU time is ~ 10% when the simple frequency transformation is applied.

Table 5 CPU sec per Time Step on the CDC 6600

Test Case	Energy Group, G	Precursor Group, I	Mesh Point, N	MTT1	MTT2	MTT3	ADEP-Exp.
1	2	1	25	.242	.252	.247	.0249
2	2	0	25	.130	---	.131	.0249
3	2	1	121	.849	.883	.848	.0876
4	2	2	33	.509	.519	.510	.0368

To provide more information on the efficiency of a numerical method, an empirical formula relating the computer CPU time to the number of variables in a problem is often useful. The variables here refer to the number of energy groups, G, precursor groups, I, and mesh points, N. From the data in Table 5, the following expressions were obtained.

For ADEP-Exponential,

$$\text{CPU sec/step} = 5.4 \times 10^{-4} \times N' \times (G + 0.05 \times I)$$

For MTT1,

$$\text{CPU sec/step} = 1.94 \times 10^{-3} \times N' \times (G + I)^{1.54}$$

Here N' is the number of interior mesh points. Use of N' instead of N will give a better estimate of the computer time required for each step since the time consuming task of calculating the spatial moments and frequencies for each time step is done only for the interior mesh points.

As in ADEP-Exponential, the time required by the multiple temporal-mode transformation technique depends on the total number of energy and precursor groups. Furthermore, the relation between computer time per step and (G + I) is not linear. This is a direct result of solving the eigenvectors and eigenvalues of a matrix in order to obtain the spatial moments and frequencies. Increasing the number of either energy groups or precursor groups will significantly increase the computer time required.

Though the computer time requirements listed in Table 5 were based on one- or two-dimensional problems

with a small number of mesh points, the results can be used to give a rough estimate of the computer time required in problems of a larger size. For a realistic problem with two energy groups, six precursor groups and 20 x 20 x 12 interior mesh points, ADEP would require ~ 6 CPU sec/step while MTT1 would require ~ 230 CPU sec/step. The numerical algorithm used in this study compares poorly with the ADEP method with respect to computer time for the types of transient investigated.

4.7 Further Numerical Results

In the four test cases just presented, the mesh structure was held fixed while the effects on accuracy and stability of various time step sizes were examined. It was found that the errors in calculating the neutron leakage at each mesh point at the beginning of each time step resulted in instability problems if the step size was increased beyond a certain value. Since the term $E_{g,j,k}$ is inversely proportional to the square of the mesh spacing, Δx^2, the percentage errors in the diagonal elements $(E_g + \Sigma_{gg})$ would be reduced by using a larger mesh spacing. The errors in the computed frequencies would in turn be reduced and the frequency feedback effect could be minimized. To verify this

point, the mesh structure in test case 1 was modified.

The uniform mesh spacing in test case 1 was increased from 10 cm to 20 cm. Several time steps were tested using the MTT2 method and the results were compared with those of a previous run at Δt = 10 μsec and Δx = 10 cm, as shown in Tables 6a and 6b. Using a larger mesh spacing, Δx = 20 cm, MTT2 was found to be stable for $\Delta t \leq$ 0.1 msec. For the stable time steps, at Δx = 20 cm, the results agreed with one another to at least six significant digits at all mesh points.

In a finite difference method, the number of mesh points used dictates the level of accuracy that can be attained in the solution to the semi-discrete reactor kinetics equations. The additional runs on test case 1 again indicate that the multiple temporal-mode transformation method for stable time steps is very accurate, within the limits of attainable accuracy with the larger mesh spacing. However, there still exists a threshold time step size beyond which the method becomes numerically unstable.

Table 6a Centre Point Thermal Flux (by MTT2)

Time (sec)	Δx = 10 cm*	Δx = 20 cm**
0	1.0	1.0
.01	1.120419	1.123985
.02	1.245599	1.253298

Table 6b Fast Flux at x = 20 cm (by MTT2)

Time (sec)	Δx = 10 cm*	Δx = 20 cm**
0	.1462276	.1462053
.01	.1652745	.1658159
.02	.1836346	.1847841

* Results obtained by MTT2 at Δt = 10 μsec

** Results obtained by MTT2 at Δt = 10 μsec, 20 μsec and 0.1 msec.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

Three different numerical algorithms have been examined in this study to evaluate the merits of the multiple temporal-mode transformation technique. The following discussions summarize the experimental results of four test cases with respect to accuracy, numerical stability and computer requirements. The results have also identified certain areas which could improve the performance of the technique.

5.1 Accuracy

The numerical algorithms, MTT1 and MTT2, are very accurate for time step sizes which yield numerically stable results. The temporal truncation errors of MTT1 and MTT2 have been shown theoretically to be of order Δt. The order of the temporal truncation error could not be confirmed experimentally. The percentage errors at extremely small time step sizes cannot be determined accurately without carrying an unreasonably large number of digits in the computer output. The numerical results obtained by MTT1 and MTT2 have indicated that the multiple temporal-mode tranformation technique can be used to generate very accurate solutions to the semi-discrete reactor kinetics equations, but small time steps have to be used in order to maintain numerical stability if a

reasonable spatial mesh size is used.

The numerical algorithm MTT3 involves a somewhat arbitrary assumption related to the effective frequencies; this results in a truncation error of order Δt^2, the same as in ADEP. MTT3 and ADEP with the exponential option have comparable accuracies. The standard method in ADEP has relatively large errors as compared with MTT3 and ADEP with the exponential option. Using the simple frequency transformation in ADEP greatly improves the accuracy with an increase in computer time of ~10%. In the first three test cases, for a Δt = 0.2 msec, ADEP-Exponential gave solutions with less than 1% error. The MTT3 method was numerically unstable at this time step. In test case 4, ADEP-Exponential again performed much better than MTT3. The poor performance of MTT3 at larger time steps agreed with what was observed in MTT1 and MTT2. The three algorithms displayed what appears to be a significant disadvantage of the multiple temporal-mode transformation technique, that is the limit on the time step size to obtain numerically stable results. This will be discussed in detail in the following section.

5.2 Numerical Stability

The multiple temporal-mode transformation technique

is susceptible to numerical instability problems. Test case 1 has identified one important factor which contributes to instability. This factor is related to the method of computing the spatial moments and frequencies at each mesh point at the beginning of each time step. This instability is caused by the so-called frequency feedback effect.

Truncation errors are inherent in finite difference techniques. The first observable errors in the cases being investigated here usually showed up at points near the reactor boundary. This is thought to be a characteristic of the explicit code, where the fluxes are solved point by point.[6] Thus, as the spatial mesh is swept in alternating directions from one boundary of the mesh to the other boundary, the points near the boundary have accumulated proportionately larger errors than the inner mesh points. The errors at points near the boundary then propagate at succeeding time steps towards the centre of the plane because of the coupling in the finite difference approximations. Depending upon the magnitudes and distribution of such errors and upon the system parameters, the truncation errors may be sufficiently large to cause errors in the computed frequencies which in turn give rise to errors in the computed neutron flux.

Eventually these errors will result in numerical difficulty and the computer run will abort. Stability will be maintained only at small time steps when the frequency feedback effect is minimal. Small time steps, in the range of microseconds, were also necessary in the previous works of Garland et al[7,8] and Harding[10] when using the multiple temporal-mode transformation technique with, in these cases, implicit codes. Hence it is felt that the instability problem encountered here is not a direct result of the use of an explicit code, but rather it is a result of the magnification of the truncation errors associated with the finite difference approximations. The effect of truncation errors is significant in the multiple temporal-mode tranformation technique because the errors which result from approximating the neutron leakage at a point $\bar{\nabla}.D_g\bar{\nabla}\phi_g$, with $E_g\phi_g$ can be large. The problem of numerical instability is more severe in reflector regions where the fast flux gradients are large. For test case 4, large errors in the computed frequencies appeared first in the reflector regions.

A formal proof of the stability criteria has not been established in this study. A brief discussion on the stability condition of the multiple temporal-mode transformation technique has been given by Garland et al.[7]

It can be seen that the proof of convergence and stability depends on the assumption that the correction function is slowly varying in the time interval $2\Delta t$, such that

$$\frac{\partial \psi(\bar{r},t)}{\partial t} = 0 \qquad (o \leq t \leq 2\Delta t)$$

This study has shown that at small time steps, the technique is numerically stable. The maximum time step size for stable and accurate results, which is problem-dependent, has proven to be a severe restriction for the cases investigated.

5.3 Computer Requirement

The limiting factor in applying the multiple temporal-mode transformation technique is computer time. Because of the large amount of computer time required to calculate the spatial moments and frequencies, combined with the limit on time step size, the technique is not a suitable method for thermal reactor transient analyses. A typical space-dependent transient considered for CANDU-type reactors may last two to three seconds. Since the computer time required for each step is large, this technique is very expensive to apply to such a transient because of the small step size required for numerical stability.

5.4 Recommendations for Future Work

The multiple temporal-mode tranformation technique, though expensive for the type of thermal reactor transients investigated here, may be used as an accurate method to predict the neutron fluxes in a short time interval (i.e. $t \lesssim 1$ msec) immediately following the onset of a perturbation. However, experience at CRNL has indicated that transients of such a short duration are not of much practical interest in linked reactor-physics-thermal-hydraulic safety calculations in thermal reactors.[11]

Previous work at CRNL with bench mark calculations has found that small time steps must be used with the ADEP code for fast reactor transient problems. This may well suggest a possible area of application for the multiple temporal-mode transformation technique. It should be noted however, that finite difference methods are generally not considered to be the best method for fast reactor transient analyses where there is little spatial change in the neutron flux.

This study has identified several areas for possible future improvement to the multiple temporal-mode transformation technique. They are listed as below:

(a) An efficient method for solving the eigenvalues and

eigenvectors of the system coefficient matrix.

(b) A better method of normalizing the eigenvectors to obtain the spatial moments.

(c) An improved method to minimize the errors in computing the leakage term at a mesh point at the beginning of each time step.

(d) A closer examination of the relationship between numerical stability and mesh spacing. Increasing the mesh spacing will minimize the contribution of errors in the leakage terms which are inversely proportional to the square of the mesh spacing. This, of course, increases the spatial truncation errors. An optimum mesh spacing is required so that the time step size can be increased without causing numerical instability problems or sacrificing the desirable accuracy. This optimum condition is most likely problem-dependent.

(3) Optimization of the computer code to minimize computer time and storage.

REFERENCES

1. J.B. Yasinsky and S. Kaplan, "Synthesis of Three-Dimensional Flux Shapes Using Discontinuous Sets of Trial Functions," Nucl. Sci. Eng., 28, 426 (1967).
2. W.M. Stacey, Jr., "A Variational Multichannel Space-Time Synthesis Method for Nonseparable Reactor Transients," Nucl. Sci. Eng., 34, 45 (1968).
3. K. Ott, "Quasistatic Treatment of Spatial Phenomena in Reactor Dynamics," Nucl. Sci. Eng., 26, 563 (1966).
4. W.H. Reed and K.F. Hansen, "Finite Difference Techniques for the Reactor Kinetics Equations," MIT-3903-2, Massachusetts Institute of Technology (May, 1969).
5. A.L. Wight, K.F. Hansen, and D.R. Ferguson, "Application of Alternating-Direction Implicit Methods to the Space-Dependent Kinetics Equations," Nucl. Sci. Eng., 44, 239 (1971).
6. D.R. Ferguson, "Solution of the Space-Dependent Reactor Kinetics Equations in Three Dimensions," PhD Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (August, 1971).
7. W.J. Garland, A.A. Harms, and J. Vlachopoulos, "Space-Time Nuclear Reactor Analysis by Temporal Transformation," Nucl. Sci. Eng., 55, 119 (1974).

8. A.A. Harms, W.J. Garland, W.A. Pearce, M.F. Harding, O.A. Trojan, and J. Vlachopoulos, "Multiple Temporal-Mode Analysis for Three-Dimensional Reactor Dynamics," Proceedings of the Joint NEACRP/CSNI Specialists' Meetings on New Developments in Three-Dimensional Neutron Kinetics and Review of Kinetics Benchmark Calculations, P. 73 (March, 1975).

9. R.S. Denning, "ADEP, One - and Two-Dimensional Few-Group Kinetics Code," BMI-1911 (July, 1971).

10. M.F. Harding, "Three-Dimensional Reactor Dynamics by Multiple Mode Temporal Transformation," McMaster Project Report, Department of Engineering Physics, McMaster University (April, 1975).

11. Private communication, Dr. Frank McDonnell, CRNL (October, 1975).

12. IMSL Library 3, Edition 4, International Mathematical and Statistical Libraries, Inc., November, 1974.

APPENDIX A

A.1 Mesh Structure

The space-time reactor kinetics equations are discretized in the spatial variable by superimposing a mesh structure upon the reactor model. The spatial mesh is described here for x-y geometry only. It can easily be extended to include other geometries in both two and three spatial dimensions. The mesh structure described is that used in the ADEP code.[9]

The reactor core is divided into regions in which the nuclear properties are homogeneous. A cell-edge or interstitial x-y mesh is used such that the mesh lines are parallel to the two axes. The nodes, the points of intersection of the mesh lines, occur at the physical boundaries of the medium and at region boundaries as well as at points interior to the regions. A rectangular area is associated with each node. The sides of each rectangle are parallel to the mesh and occur halfway between nodes. The spatial mesh has a total of J columns and K rows. Mesh points are numbered sequentially from left to right and by rows from top to bottom. The rectangular area associated with mesh point (j,k) may be called cell (j,k). Nodes exterior to the reactor are

ignored. Zero flux boundary conditions are applied at all exterior boundaries.

Figure (A.1) shows a node (j,k) with its associated cell divided into four quadrants.

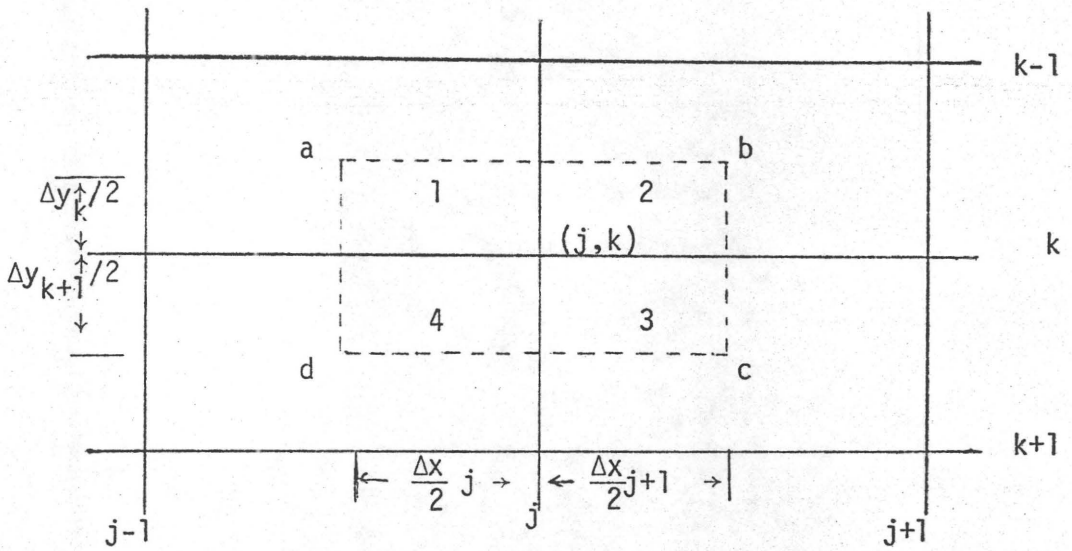


FIGURE A.1: Cell (j,k)

A.2 The Semi-Discrete Form of the Reactor Kinetics Equations

The derivation of the semi-discrete form of the space-dependent kinetics equations is carried out here for x-y geometry only, in accordance with the mesh structure described in Section A.1.

The discrete equations for mesh point (j,k) are obtained by integrating the basic kinetics equations, Eqs. (1.1), over the volume $V_{j,k}$ of cell (j,k).

$$\int_{V_{j,k}} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\bar{r},t)dV = \int_{V_{j,k}} \bar{\nabla}.D_g(\bar{r})\bar{\nabla}\phi_g(\bar{r},t)dV$$

$$+\sum_{g'=1}^{G} \int_{V_{j,k}} \Sigma_{gg'}(\bar{r})\phi_{g'}(\bar{r},t)dV+\sum_{i=1}^{I} f_{gi} \int_{V_{j,k}} C_i(r,t)dV \quad (1\leq g\leq G)$$

(A.1)

$$\int_{V_{j,k}} \frac{\partial}{\partial t} C_i(\bar{r},t)dV=-\lambda_i \int_{V_{j,k}} C_i(\bar{r},t)dV+\sum_{g'=1}^{G} \int_{V_{j,k}} P_{ig'}(\bar{r},t)\phi_{g'}(\bar{r},t)dV \quad (1\leq i\leq I)$$

By Gauss' Theorem, the volume integral for the diffusion terms is changed to a surface integral as follows:

$$\int_{V_{j,k}} \bar{\nabla}.D_g(\bar{r})\bar{\nabla}\phi_g(\bar{r},t)dV= \int_{B_{j,k}} D_g(\bar{r})\bar{\nabla}\phi_g(\bar{r},t).\hat{n}dS \qquad (A.2)$$

where $B_{j,k}$ denotes the boundary of the cell (j,k) and $\hat{n}$ is a unit normal to this boundary.

The current normal to the boundary represented by ab (see Fig. A.1) for example, may be approximated by a finite difference expression as follows:

$$\bar{\nabla}\phi_g(x,y).\hat{n}_y \Big|_{y=y_k-\frac{\Delta Y_k}{2}} = \frac{\phi_{g,j,k} - \phi_{g,j,k-1}}{\Delta Y_k} \qquad (A.3)$$

With the following definitions,

$$V_{j,k} = \tfrac{1}{4}(\Delta x_j+\Delta x_{j+1})\ (\Delta Y_k+\Delta Y_{k+1}) \qquad (A.4)$$

$$\frac{1}{v_{g,j,k}} \frac{d}{dt} \phi_{g,j,k} = \frac{1}{V_{j,k}} \int_{V_{j,k}} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\bar{r},t)dV \quad (A.5)$$

$$\Sigma_{gg',j,k}\phi_{g',j,k} = \frac{1}{V_{j,k}} \int_{V_{j,k}} \Sigma_{gg'}(\bar{r})\ \phi_{g'}(\bar{r},t)dV \quad (A.6)$$

$$C_{i,j,k} = \frac{1}{V_{j,k}} \int_{V_{j,k}} C_i(\bar{r},t)dV \quad (A.7)$$

$$P_{ig',j,k}\ \phi_{g',j,k} = \frac{1}{V_{j,k}} \int_{V_{j,k}} P_{ig'}(\bar{r})\phi_{g'}(\bar{r},t)dV \quad (A.8)$$

$$ADL_{g,j,k}(\phi_{g,j-1,k} - \phi_{g,j,k}) + ADR_{g,j,k}(\phi_{g,j+1,k} - \phi_{g,j,k})$$
$$+ADU_{g,j,k}(\phi_{g,j,k-1} - \phi_{g,j,k}) + ADD_{g,j,k}(\phi_{g,j,k+1} - \phi_{g,j,k})$$
$$= \frac{1}{V_{j,k}} \int_{B_{j,k}} D_g(\bar{r})\bar{\nabla}\phi_g(\bar{r},t).\hat{n}dS \quad (A.9)$$

where the terms $ADL_{g,j,k}$, $ADR_{g,j,k}$, $ADU_{g,j,k}$ and $ADD_{g,j,k}$ contain the appropriately weighted diffusion coefficients of energy group g across the boundaries of cell (j,k), Eqs. (A.1) can then be written in the following semi-discrete form:

$$\frac{d}{dt} \phi_{g,j,k} = v_{g,j,k} \left\{ ADR_{g,j,k}(\phi_{g,j+1,k} - \phi_{g,j,k}) \right.$$
$$+ADL_{g,j,k}(\phi_{g,j-1,k} - \phi_{g,j,k}) + ADD_{g,j,k}(\phi_{g,j,k+1} - \phi_{g,j,k})$$
$$+ADU_{g,j,k}(\phi_{g,j,k-1} - \phi_{g,j,k}) + \sum_{g'=1}^{G} \Sigma_{gg',j,k}\ \phi_{g',j,k}$$
$$\left. + \sum_{i=1}^{I} f_{gi}\ C_{i,j,k} \right\} \qquad (1 \leq g \leq G)$$

(A.10)

$$\frac{d}{dt} C_{i,j,k} = -\lambda_i C_{i,j,k} + \sum_{g'=1}^{G} P_{ig',j,k} \phi_{g',j,k} \quad (1 \leq i \leq I)$$

The factor Δz has been left out of the volume term since it occurs in each term of the equations. The terms representing the neutron speed, diffusion coefficients and cross sections associated with the point (j,k) are to be explained later.

Furthermore, by defining the following NXN square matrices

$$\underline{T}_{gg'} = \text{diag} \left\{ v_{g,j,k} \Sigma_{gg',j,k} \right\} \quad (A.11)$$

$$\underline{F}_{gi} = \text{diag} \left\{ v_{g,j,k} f_{gi} \right\} \quad (A.12)$$

$$\underline{\Lambda}_i = \lambda_i \underline{I} \quad (A.13)$$

$$\underline{P}_{ig'} = \text{diag} \left\{ P_{ig',j,k} \right\} \quad (A.14)$$

and $\underline{D}_g$ such that

$$\underline{D}_g \bar{\phi}_g = \text{col} \Big\{ v_{g,j,k} [ADR_{g,j,k}(\phi_{g,j+1,k} - \phi_{g,j,k}) + ADL_{g,j,k}(\phi_{g,j-1,k} - \phi_{g,j,k}) + ADD_{g,j,k}(\phi_{g,j,k+1} - \phi_{g,j,k}) + ADU_{g,j,k}(\phi_{g,j,k-1} - \phi_{g,j,k})] \Big\} \quad (A.15)$$

the semi-discrete form of the kinetics equations for all mesh points can be written as follows:

$$\frac{d}{dt}\bar{\phi}_g = \underline{D}_g\,\bar{\phi}_g + \sum_{g'=1}^{G} \underline{T}_{gg'}\bar{\phi}_{g'} + \sum_{i=1}^{I} \underline{F}_{gi}\bar{C}_i \qquad (1\leq g\leq G)$$

(A.16)

$$\frac{d}{dt}\bar{C}_i = -\underline{A}_i\,\bar{C}_i + \sum_{g'=1}^{G} \underline{P}_{ig'}\,\bar{\phi}_{g'} \qquad (1\leq i\leq I)$$

In the ADEP program, flux calculations are performed for all mesh points except those lying on the exterior boundary of the core. Hence, we can consider three cases in the approximations of the terms involving reactor parameters, $v_{g,j,k}$, $\Sigma_{gg',j,k}$, $P_{ig',j,k}$, $ADR_{g,j,k}$, $ADL_{g,j,k}$ $ADD_{g,j,k}$ and $ADU_{g,j,k}$:

(1) Mesh point (j,k) is a node within a region so that material properties of the four quadrants 1,2,3,4 (Fig. A.1) are homogeneous.

(2) Mesh point (j,k) lies on the interface of two regions so that material properties in one region, represented by quadrants 1 and 4 say, are different from those of the other region, quadrants 2 and 3.

(3) Mesh point (j,k) is a corner node with different material properties in adjacent regions.

When a node intersects a region boundary, average parameters for the energy groups are obtained by area weighting the cross sections associated with the node. Diffusion coefficients are averaged by weighting with the

mesh intervals in the adjacent regions. The previously mentioned parameters for each case can easily be identified in the ADEP program and hence are not listed here.

A.3 The Semi-Discrete Form of the Transformed Equations

The semi-discrete form of the transformed equations can be derived in a similar manner as in Section A.2. Hence the details are not given here and only the final equations are quoted below:

$$\underline{\Omega}_g \frac{d}{dt} \bar{\psi}_g = \underline{D}_g \underline{\Omega}_g \bar{\psi}_g + \sum_{g'=1}^{G} \underline{T}_{gg'} \underline{\Omega}_{g'} \bar{\psi}_{g'} + \sum_{i=1}^{I} \underline{F}_{gi} \underline{\Omega}_{G+i} \bar{\psi}_{G+i} - \dot{\underline{\Omega}}_g \bar{\psi}_g \qquad (1 \leq g \leq G) \tag{A.17}$$

$$\underline{\Omega}_{G+i} \frac{d}{dt} \bar{\psi}_{G+i} = -\underline{\Lambda}_i \underline{\Omega}_{G+i} \bar{\psi}_{G+i} + \sum_{g'=1}^{G} \underline{P}_{ig'} \underline{\Omega}_{g'} \bar{\psi}_{g'} - \dot{\underline{\Omega}}_{G+i} \bar{\psi}_{G+i} \qquad (1 \leq i \leq I)$$

where

$$\underline{\Omega}_\ell = \text{diag} \left\{ \underline{\Omega}_{\ell,j,k} \right\} \qquad (1 \leq \ell \leq L,\ L=G+I)$$

$$\underline{\Omega}_{\ell,j,k} = \sum_{\ell'=1}^{L} \beta_{\ell\ell',j,k} \exp \left[\alpha_{\ell',j,k} t \right] \qquad (1 \leq \ell \leq L)$$

Here, $\beta_{\ell\ell',j,k}$ and $\alpha_{\ell',j,k}$ are the ℓ'^{th} spatial moment and frequency for group ℓ at point (j,k). The diagonal matrix $\dot{\underline{\Omega}}_\ell$ is the time derivative of $\underline{\Omega}_\ell$. The vector $\bar{\psi}_\ell$ contains the correction functions for group ℓ. The other matrices have previously been defined in Section A.2.

APPENDIX B

THE CALCULATION OF THE SPATIAL MOMENTS AND FREQUENCIES

This section describes how the matrices are set up numerically to calculate the spatial moments and frequencies. As an example, a one-dimensional system with two energy groups and one precursor group is considered. In Chapter 2 it was noted that the spatial moments and frequencies are obtained by solving for the eigenvalues and eigenvectors of the coefficient matrix $\underline{M}'(\bar{r},o)$ in Eq. (2.13). The spatial variable, $\bar{r}$, in this system of equations is discretized by the box integration method described in Appendix A. Hence, for this sample problem, the discretized form of Eq. (2.13) can be written for each mesh point as follows

$$\begin{bmatrix} v_1(E_1+\Sigma_{11}) & v_1\Sigma_{12} & v_1 f_{11} \\ v_2\Sigma_{21} & v_2(E_2+\Sigma_{22}) & v_2 f_{21} \\ p_{11} & p_{12} & -\lambda \end{bmatrix} \begin{bmatrix} \psi_1\beta_{1\ell} \\ \psi_2\beta_{2\ell} \\ \psi_3\beta_{3\ell} \end{bmatrix} = \alpha_\ell \begin{bmatrix} \psi_1\beta_{1\ell} \\ \psi_2\beta_{2\ell} \\ \psi_3\beta_{3\ell} \end{bmatrix}$$

$$\ell=1,2,3 \qquad (B.1)$$

where the subscripts denoting the mesh point have been omitted to simplify the equations. Here, $E_{g,j}\ \phi_{g,j}$ represents the group g neutron leakage at point j. The term $E_{g,j}$ is defined in one dimension as

$$E_{g,j} = \left(D_{g,j+1} \frac{\phi_{g,j+1} - \phi_{g,j}}{\Delta x_{j+1}^2} - D_{g,j} \frac{\phi_{g,j} - \phi_{g,j-1}}{\Delta x_j^2} \right) \Big/ \phi_{g,j} \qquad (B.2)$$

where $D_{g,j}$ and Δx_j etc. are the appropriate diffusion coefficients and mesh spacings associated with the mesh point j.

It is understood that, at each mesh point, the spatial moments and frequencies are to be calculated for each time step. The spatial moments at each mesh point j for each energy group or precursor group are required to satisfy the following normalization condition,

$$\sum_{\ell'=1}^{3} \beta_{\ell\ell',j} = 1 \quad \text{for all } \ell,\ \ell=1,2,3. \qquad (B.3)$$

By defining time zero as the beginning of each time step, the following initial conditions are obtained,

$$\phi_{g,j}^{o} = \psi_{g,j}^{o} \qquad g=1,2.$$
$$C_{1,j}^{o} = \psi_{3,j}^{o} \qquad (B.4)$$

where the superscript denotes time zero. Here $\phi_{g,j}^{o}$ and $C_{1,j}^{o}$ are the group g neutron flux and the precursor concentration respectively at the beginning of each time step. These are known quantities because they are either given

as the initial values for a problem or they have been calculated in the previous time step. Thus $E_{g,j}$ can easily be computed at the beginning of each time step from the following equation:

$$E_{g,j}=\left(D_{g,j+1}\frac{\phi^{o}_{g,j+1}-\phi^{o}_{g,j}}{\Delta x^{2}_{j+1}}-D_{g,j}\frac{\phi^{o}_{g,j}-\phi^{o}_{g,j-1}}{\Delta x^{2}_{j}}\right)\Big/\phi^{o}_{g,j} \qquad (B.5)$$

In this study, Eq. (B.1) is then modified to the following form:

$$\begin{bmatrix} v_1(E_1+\Sigma_{11}) & v_1\Sigma_{12}\frac{\phi^{o}_2}{\phi^{o}_1} & v_1 f_{11}\frac{C^{o}_1}{\phi^{o}_1} \\ v_2\Sigma_{21}\frac{\phi^{o}_1}{\phi^{o}_2} & v_2(E_2+\Sigma_{22}) & v_2 f_{21}\frac{C^{o}_1}{\phi^{o}_2} \\ p_{11}\frac{\phi^{o}_1}{C^{o}_1} & p_{12}\frac{\phi^{o}_2}{C^{o}_1} & -\lambda \end{bmatrix} \begin{bmatrix} \beta_{1\ell} \\ \beta_{2\ell} \\ \beta_{3\ell} \end{bmatrix} = \alpha_\ell \begin{bmatrix} \beta_{1\ell} \\ \beta_{2\ell} \\ \beta_{3\ell} \end{bmatrix} \qquad \ell=1,2,3 \qquad (B.6)$$

where the subscripts denoting a mesh point are again omitted.

All elements of the coefficient matrix in Eq. (B.6) are known. An IMSL library subroutine, EIGRF[12] is used to compute the eigenvalues and eigenvectors of the coefficient matrix. The eigenvalues thus obtained are the frequencies at that mesh point, however, the eigenvectors computed are

not normalized. In order to obtain the spatial moments, the eigenvectors are normalized subject to the condition set in Eq. (B.3).

If the computed eigenvectors are denoted by $\underline{Z}$ and their elements by z, then for the sample problem, $\underline{Z}$ is a 3x3 matrix. With the following definition,

$$\underline{B} = \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} \\ \beta_{21} & \beta_{22} & \beta_{23} \\ \beta_{31} & \beta_{32} & \beta_{33} \end{bmatrix} \tag{B.7}$$

it is clear that

$$\underline{Z} \propto \underline{B} \tag{B.8}$$

From Eq. (B.3) and the relation (B.8), a set of algebraic equations for this sample problem can be written as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ z_{21} & 0 & 0 & -z_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & z_{22} & 0 & 0 & -z_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & z_{23} & 0 & 0 & -z_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & z_{31} & 0 & 0 & -z_{21} & 0 & 0 \\ 0 & 0 & 0 & 0 & z_{32} & 0 & 0 & -z_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & z_{33} & 0 & 0 & -z_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \beta_{13} \\ \beta_{21} \\ \beta_{22} \\ \beta_{23} \\ \beta_{31} \\ \beta_{32} \\ \beta_{33} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{B.9}$$

This set of equations can be represented in matrix form as

$$\underline{F}\,\bar{X} = \bar{C} \tag{B.10}$$

where $\underline{F}$ is the coefficient matrix, $\bar{X}$ the unknown vector containing the spatial moments at a mesh point and $\bar{C}$ the vector in the right hand side of Eq. (B.9). Hence, the spatial moments for all groups at a mesh point can be obtained by solving Eq. (B.9).

For a general problem with G energy groups and I precursor groups, the order of the coefficient matrix $\underline{F}$ is $(G+I)^2$. For G=2, I=6, the number of elements in $\underline{F}$ is 4096. Since $\underline{F}$ is a band matrix, it can be stored in band storage mode to save on computer memory as shown below:

$$\underline{F}' = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & z_{21} & 0 & 0 & -z_{11} \\ 0 & z_{22} & 0 & 0 & -z_{12} \\ 0 & z_{23} & 0 & 0 & -z_{13} \\ 0 & 1 & 1 & 1 & 0 \\ z_{31} & 0 & 0 & -z_{21} & 0 \\ z_{32} & 0 & 0 & -z_{22} & 0 \\ z_{33} & 0 & 0 & -z_{23} & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{B.11}$$

In general, the size of the coefficient matrix in the band storage mode is

$$(G+I)^2 \times [(G+I-1) + (G+I-1) + 1]$$

where (G+I-1) is the number of upper or lower codiagonals in the original matrix $\underline{F}$. For G=2 and I=6, the size of the matrix $\underline{F}'$ is 960, significally reduced from 4096 in full storage mode. An IMSL library subroutine, LEQT1B[12], is used to solve the system of equations, Eq. (B.10), for the unknown spatial moments, with the coefficient matrix $\underline{F}$ in band storage mode, $\underline{F}'$. The elements of the matrix $\underline{F}'$ contribute to the additional computer storage required to implement the multiple temporal-mode transformation technique.

In a reactor model without delayed neutrons or in reflector regions where the precursor concentration is zero, Eq. (B.6) is set up for the energy groups only. Thus the terms involving ϕ_g^o/C_i^o are not included and the problem of division by zero can be avoided. The set of equations to normalize the eigenvectors, Eq. (B.9), are adjusted accordingly. The coefficient matrix in Eq. (B.6) then has order G and the matrix $\underline{F}$ in Eq. (B.10) has order G^2.

APPENDIX C

TEST PROBLEM DATA

Four test cases have been investigated in this study. The reactor configurations and parameters are listed here. All symbols have previously been defined in Section 1.2. An additional symbol, Σ_g, is used here. Σ_g is the sum of the absorption cross section, Σ_{ag}, and scattering cross section, $\Sigma_{g\rightarrow g'}$.

The boundary condition for all test cases is zero flux on the reactor outer boundary.

C.1 Test Case 1

Number of energy groups=2

Number of precursor groups=1

Geometry: Bare homogeneous slab, 240 cm in width.

Mesh Structure: Uniform mesh spacing Δx=10 cm

Number of mesh points=25

Precursor Constants:

λ=0.08 sec^{-1}, β=0.0065, χ_{11}=1.0, χ_{21}=0.

Material Properties:

roup g	v_g(cm/sec)	D_g(cm)	$\nu\Sigma_{fg}$(cm^{-1})	Σ_g(cm^{-1})	$\Sigma_{g\rightarrow g+1}$($cm^{-1}$)	χ_g
1	1.0×10^7	1.31	.0	.008154	.007368	1.0
2	3.0×10^5	.8695	.004779	.004014	0.	0.

Initial Conditions:

Initial Spatial Shape: cosine

Critical k_{eff} = 1.0096154

Initial precursor concentrations are in equilibrium with the initial neutron flux distribution.

C.2 Test Case 2

Number of energy groups=2

Number of precursor groups=0

This test case differs from case 1 in that no precursor group is considered. The geometry and the parameters pertaining to the energy groups are the same as in case 1.

C.3 Test Case 3

Number of energy groups=2

Number of precursor groups=1

Geometry: Bare homogeneous square, 200 cm on a side.

Mesh Structure: Uniform mesh spacing $\Delta x=\Delta y=10$cm forms an 11 x 11 grid.

Precursor Constants:

$\lambda=.08$, $\beta=.0064$, $\chi_{11}=1.0$, $\chi_{21}=0.$

Material Properties:

Group g	v_g	D_g	$\nu\Sigma_{fg}$	Σ_g	$\Sigma_{g\rightarrow g+1}$	χ_g
1	3.0×10^7	1.3506	.00058322	.0040152	.0023	1.0
2	2.2×10^5	1.0808	.0098328	.0057536	0.	0.

Initial Conditions:

Initial Spatial Shape: cosine

Critical k_{eff} = 0.89450788

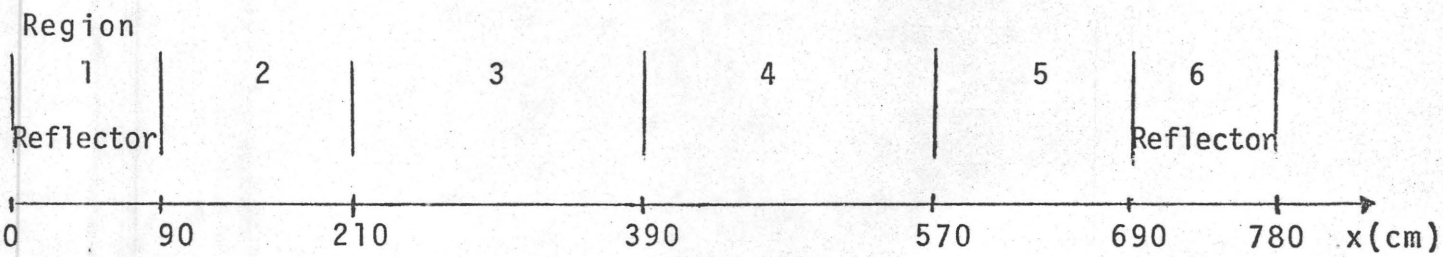
Initial configuration is made critical by dividing the production cross sections by the critical k_{eff}. Initial precursor concentrations are in equilibrium with the critical neutron flux distribution.

C.4 Test Case 4

Number of energy groups=2

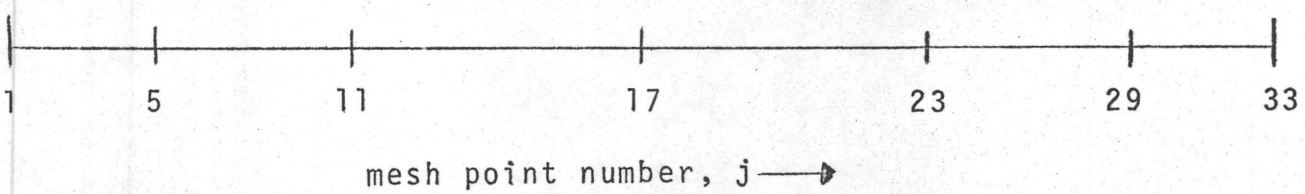
Number of precursor groups=2

Geometry: One dimensional, six-region reactor, 780 cm in width.



Mesh Structure: Non-uniform mesh spacings of 15 and 30 cm.

Number of mesh points=33.



Precursor Constants:

λ_1=.06297, β_1=.003213, χ_{11}=1.0, χ_{21}=0.

λ_2=.6871, β_2=.004556, χ_{12}=1.0, χ_{22}=0.

Material Properties:

Region	Group g	D_g	$\nu\Sigma_{fg}$	Σ_g	$\Sigma_{g\rightarrow g+1}$
1,6	1	1.310	0.	.01021	.01018
	2	.8695	0.	.0002335	0.
2,5	1	1.264	.0002247	.008177	.007368
	2	.9328	.004523	.004031	0.
3,4	1	1.264	.0002217	.008163	.007368
	2	.9328	.004462	.004106	0.

Additional parameters for all regions are:

v_1=1.x10^7 cm/sec, v_2=3.x10^5 cm/sec, χ_1=1.0, χ_2=0.

Initial Conditions:

Critical k_{eff} = 1.0084550

Initial configuration is made critical by dividing the production cross sections by the critical k_{eff}. The initial precursor concentrations are in equilibrium with the initial critical flux distribution.

APPENDIX D

THE "MTT" COMPUTER SUBROUTINES AND MODIFICATIONS TO THE ADEP CODE

Two computer subroutines which calculate the spatial moments and frequencies have been written. A listing of the two subroutines, SETMF and SPMF, is given in the following pages. In addition, the update instructions for modifying the ADEP code to utilize the above two subroutines are also listed here.

```
      SUBROUTINE SETMF(PHI,C,ASF,V,D,SIG,SIGR,W,WD,FREQ,SPMMT,IMJKM,NDJK
     1M,IMLTHN,LTH,JKMAX,IMLTH,IM1LTH,NW,NWD,IMAX,IMAXND,JKL,JKLG)
      COMMON AL(6),BET(6),C1(6),C2(6),DYW(50),JI(50),JF(50),INTY(21),O(2
     10),INTX(41),BM(10),COF(10),COC(10),DTNEW(5),DXP(40),DYP(20),F(10),
     2IDF(10),IDC(10),IPR(10),INN(5),TAU(5),FD(10),IDT(10),ALFC,AAT,BETA
     3,BBT,CP,CF,CC,DT,G,HEFF,IEXP,IR,INIT,IRST,ISS,IPRM,IMIM1,IM1,IT,IT
     4PR,IP1,JIND,JM1,JP1,KP1,KM1,LTHN,MM,NPC,ND,NLX,NLY,NFF,NCF,NCT,R,R
     5HOF,TP,TMAX,T,TR,VC,VF,XA,XC,XD1,XD,XE,NY,BETM1,XC2,LTH2,LTHIM1,A(
     6110),Y(1)
      DIMENSION PHI(IMJKM),ASF(IMLTHN),V(IMLTH),D(IMLTH),SIG(IMLTH),SIGR
     1(IM1LTH),W(NW),WD(NWD),C(NDJKM)
      DIMENSION FREQ(JKL),SPMMT(JKLG)
      DIMENSION AV1(4),AS1(4),ADL1(4),ADR1(4),ADU1(4),ADD1(4),ASR1N(4),A
     1SR2N(4)
      INTEGER DI3,DI4
C   SETMF DEFINES AV,AS,ADL,ADR,ADU,ADD,ASR1 AND ASR1 FOR A MESH POINT
C   AS IN FORCK AND CALLS SPMF TO SET UP THE MATRICES FOR EIGENVALUES
C   AND EIGENVECTORS
      DO 5 I=1,4
      ADL1(I)=0.
      ADR1(I)=0.
      ADU1(I)=0.
      ADD1(I)=0.
      ASR1N(I)=0.
 5    ASR2N(I)=0.

C   SET UP INTERIOR MESH POINTS
      DI4=1
      DI3=1
      LL1=1+LTH
      L=0
      DO 500 I1=1,NLY
      DO 500 I2=1,NLX
      L=L+1
      L0=0
      DO 65 I=1,IMAX
      L1=L0+L
      I4=DI4
      ADL1(I)=D(L1)*WD(I4)
      I4=I4+1
      ADR1(I)=ADL1(I)
      IF (JIND.EQ.0) GO TO 50
      ADU1(I)=D(L1)*WD(I4)
      ADD1(I)=ADU1(I)
      I4=I4+1
 50   AV1(I)=V(L1)
      AS1(I)=SIG(L1)
      IF (I-2) 65,60,55
 55   ASR2N(I)=SIGR(L1-LTH2)
 60   ASR1N(I)=SIGR(L1-LTH)
```

```
 65   L0=L0+LTH
      DI4=I4
      K1=INTY(I1)+1
      K2=INTY(I1+1)-1
      J1=INTX(I2)+1
      J2=INTX(I2+1)-1
      IL0=(K1-2)*JIND
      DO 200 K=K1,K2
      IL0=IL0+JIND
      DO 200 J=J1,J2
      IL=IL0+J
      CALL SPMF(PHI,C,ASF,AV1,AS1,ASR1N,ASR2N,ADL1,ADR1,ADU1,ADD1,FREQ,S
     1PMMT,IMJKM,NDJKM,IMLTHN,LTH,JKMAX,IMLTH,IM1LTH,IMAX,IMAXND,JKL,JKL
     2G,IL,L)
 200  CONTINUE

C  SET UP RIGHT BOUNDARY POINTS
      IF(I2.EQ.NLX) GO TO 300
      L0=0
      DO 220 I=1,IMAX
      I3=DI3
      L1=L0+L
      L2=L1+1
      I4=DI4
      ADL1(I)=D(L1)*WD(I4)
      ADR1(I)=D(L2)*WD(I4+1)
      I4=I4+2
      IF (JIND.EQ.0) GOTO 205
      ADU1(I)=WD(I4)*D(L1)+WD(I4+1)*D(L2)
      ADD1(I)=ADU1(I)
      I4=I4+2
 205  X0=1.-W(I3)
      AV1(I)=1./(W(I3)/V(L1)+X0/V(L2))
      AS1(I)=W(I3)*SIG(L1)+X0*SIG(L2)
      IF(I-2) 220,215,210
 210  ASR2N(I)=W(I3)*SIGR(L1-LTH2)+X0*SIGR(L2-LTH2)
 215  ASR1N(I)=W(I3)*SIGR(L1-LTH)+X0*SIGR(L2-LTH)
 220  L0=L0+LTH
      DI4=I4
      DI3=I3+1
      DO 290 K=K1,K2
      IL=(K-1)*JIND+J2+1
      CALL SPMF(PHI,C,ASF,AV1,AS1,ASR1N,ASR2N,ADL1,ADR1,ADU1,ADD1,FREQ,S
     1PMMT,IMJKM,NDJKM,IMLTHN,LTH,JKMAX,IMLTH,IM1LTH,IMAX,IMAXND,JKL,JKL
     2G,IL,LL1)
 290  CONTINUE
      LL1=LL1+1

C  SET UP BOTTOM BOUNDARY POINTS
 300  IF (I1.EQ.NLY) GO TO 500
```

```
      L0=0
      DO 320 I=1,IMAX
      L1=L+L0
      L2=L1+NLX
      I4=DI4
      I3=DI3
      ADL1(I)=D(L1)*WD(I4)+D(L2)*WD(I4+1)
      ADR1(I)=ADL1(I)
      I4=I4+2
      IF (JIND.EQ.0) GO TO 305
      ADU1(I)=D(L1)*WD(I4)
      ADD1(I)=D(L2)*WD(I4+1)
      I4=I4+2
  305 X0=1.-W(I3)
      AV1(I)=1./(W(I3)/V(L1)+X0/V(L2))
      AS1(I)=W(I3)*SIG(L1)+X0*SIG(L2)
      IF (I-2) 320,315,310
  310 ASR2N(I)=W(I3)*SIGR(L1-LTH2)+X0*SIGR(L2-LTH2)
  315 ASR1N(I)=W(I3)*SIGR(L1-LTH)+X0*SIGR(L2-LTH)
  320 L0=L1+LTH
      DI4=I4
      DI3=I3+1
      IL0=K2*JIND
      DO 390 J=J1,J2
      IL=IL0+J
      CALL SPME(PHI,0,ASE,AV1,AS1,ASR1N,ASR2N,ADL1,ADR1,ADU1,ADD1,FREQ,S
     1PMNT,IMJKM,NDJKM,IMLTHN,LTH,JKMAX,IMLTH,IM1LTH,IMAX,IMAXND,JKL,JKL
     2G,TL,LL1)
  390 CONTINUE
      LL1=LL1+1

C  SET UP CORNER POINTS
  400 IF (I2.EQ.NLX) GO TO 500
      L0=0
      DO 430 I=1,IMAX
      L1=L+L0
      L2=L1+1
      L5=L1+NLX
      L6=L1+NLX+1
      I3=DI3
      I4=DI4
      I7=I3+1
      I8=I3+2
      I9=I3+3
      AV1(I)=1./(W(I3)/V(L1)+W(I7)/V(L2)+W(I8)/V(L5)+W(I9)/V(L6))
      AS1(I)=W(I3)*SIG(L1)+W(I7)*SIG(L2)+W(I8)*SIG(L5)+W(I9)*SIG(L6)
      IF (I-2) 420,415,410
  410 ASR2N(I)=W(I3)*SIGR(L1-LTH2)+W(I7)*SIGR(L2-LTH2)+W(I8)*SIGR(L5-LTH
     12)+W(I9)*SIGR(L6-LTH2)
  415 ASR1N(I)=W(I3)*SIGR(L1-LTH)+W(I7)*SIGR(L2-LTH)+W(I8)*SIGR(L5-LTH)+
```

```
     1W(I9)*SIGR(L6-LTH)
420   ADL1(I)=WD(I4)*D(L1)+WD(I4+1)*D(L5)
      ADR1(I)=WD(I4+2)*D(L2)+WD(I4+3)*D(L6)
      I4=I4+4
      IF (JIND.EQ.0) GO TO 430
      ADU1(I)=WD(I4)*D(L1)+WD(I4+1)*D(L2)
      ADD1(I)=WD(I4+2)*D(L5)+WD(I4+3)*D(L6)
      I4=I4+4
430   L0=L0+LTH
      DI4=I4
      DI3=I3
      IL=K2*JIND+J2+1
      CALL SPMF(PHI,C,ASF,AV1,AS1,ASR1N,ASR2N,ADL1,ADR1,ADU1,ADD1,FREQ,S
     1PMMT,IMJKM,NDJKM,IMLTHN,LTH,JKMAX,IMLTH,IM1LTH,IMAX,IMAXND,JKL,JKL
     2G,IL,LL1)
      LL1=LL1+1
500   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SPME(PHI,C,ASF,AV1,AS1,ASR1N,ASR2N,ADL1,ADR1,ADU1,ADD1,
     1FREQ,SPNMT,IMJKM,NDJKM,IMLTHN,LTH,JKMAX,IMLTH,IM1LTH,IMAXND,I
     2KL,JKLG,TL,L)
      COMMON AL(6),BET(6),C1(6),C2(6),DYW(50),JI(50),JF(50),INTY(21),O(2
     10),INTX(41),BM(10),COF(10),COC(10),DTHEW(5),DXP(40),DYP(20),F(10),
     2IDF(10),IDC(10),IFR(10),INN(5),TAU(5),FD(10),IDT(10),ALFC,AAT,BETA
     3,RRT,CP,CF,CC,DT,G,HEFF,IEXP,IR,INIT,IRST,ISS,IPRM,IMIN1,IM1,IT,IT
     4PR,IP1,JIND,JM1,JP1,KP1,KM1,LTH1,MM,NPC,ND,NLX,NLY,NFF,NCF,NCT,R,R
     5HOF,TP,TMAX,T,TR,VC,VF,XA,XC,XD1,XD,XF,MX,BETM1,XC2,LTH2,LTHIM1,A(
     6110),Y(1)
      DIMENSION PHI(IMJKM),C(NDJKM),ASF(IMLTHN)
      DIMENSION AV1(IMAX),AS1(IMAX),ASR1N(IMAX),ASR2N(IMAX),ADL1(IMAX),A
     1DR1(IMAX),ADU1(IMAX),ADD1(IMAX)
      DIMENSION FREQ(JKL),SPNMT(JKLG)
      DIMENSION F(4)
      DIMENSION AA(10,10),Z(10,10),WK(120),WW(10)
      DIMENSION AAA(100,10),B(100,1),XL(100)
      COMPLEX Z,WW
C   SPME SETS UP THE MATRICES
C   FIXED DIMENSIONS PROVIDE FOR 4 ENERGY GROUPS AND 6 PRECURSOR GROUPS
C   AA IS COEF. MATRIX FOR FINDING SPATIAL MOMENTS AND FREQUENCIES
C   AAA IN BAND STORAGE MODE FOR NORMALIZING SPATIAL MOMENTS
C   NN=ORDER OF MATRIX AA
C   IMXND2=ORDER OF MATRIX AAA
      IA=10
      IZ=10
      IJOB=2
      IA1=100
      IZ1=100
      IJOB1=0
      NN=IMAXND
      DO 10 I=1,IMAXND
      DO 10 II=1,IMAXND
 10   AA(I,II)=0.
      FLAG=0.
      L0=0
      L2=0
      DO 100 I=1,IMAX
      IL1=IL+L2
      L1=L+L0
      F(I)=(ADL1(I)*PHI(IL1-1)+ADR1(I)*PHI(IL1+1)+ADU1(I)*PHI(IL1-JIND)+
     1ADD1(I)*PHI(IL1+JIND)-(ADL1(I)+ADR1(I)+ADU1(I)+ADD1(I))*PHI(IL1))/
     2PHI(IL1)
      AA(I,I)=AV1(I)*(F(I)+BETM1*F(I)*ASF(L1)-AS1(I))
      L3=L-LTHN
      IL2=IL-JKMAX
      DO 60 IX=1,IMAY
      L3=L3+LTHN
      IL2=IL2+JKMAX
      IF (IX.EQ.1) GO TO 60
```

```
      X=PHI(IL2)/PHI(IL1)
      AA(I,IX)=AV1(I)*BETM1*F(I)*ASF(L3)*X
      I0=IX+1
      IF (I0.EQ.I) AA(I,IX)=AA(I,IX)+AV1(I)*ASP1N(I)*X
      I0=IX+2
      IF (I0.EQ.I) AA(I,IX)=AA(I,IX)+AV1(I)*ASP2N(I)*X
 60   CONTINUE
      IL2=IL
      DO 70 N=1,ND
      IX=N+IMAX
      AA(IX,IX)=-AL(N)
      X=C(IL2)/PHI(IL1)
      AA(I,IX)=AV1(I)*FD(I)*AL(N)*X
C  TEST FOR THE CASE WITHOUT DELAYED NEUTRONS
      IF (BETA.EQ.0.) GO TO 70
      IF (ASF(L1) .EQ. 0.) GO TO 70
      AA(IX,I)=BET(N)*ASF(L1)/X
 70    IL2=IL2+JKMAX
      FLAG=FLAG+ASF(L1)
      L0=L0+LTHN
 100  L2=L2+JKMAX
      IF (FLAG.EQ.0.) NN=IMAX
      CALL EIGRF(AA,NN,JA,JOB,WW,Z,IZ,WK,IER)
      DO 120 I=1,NN
      I0=(I-1)*JKMAX+IL
 120  FREQ(I0)=WW(I)
      IMXND2=NN*NN
      IMNDP1=NN+1
      NLC=NN-1
      NUC=NLC
      NS=NLC+NUC+1
      DO 161 I=1,IMXND2
 161  B(I)=0.
      DO 163 I=1,IMXND2
      DO 163 II=1,NS
 163  AAA(I,II)=0.
      ILL=IMXND2-IMNDP1
      L0=NLC
      IX1=0
      DO 170 I=1,ILL,IMNDP1
      B(I)=1.
      L1=L0+1
      L2=L0+NN
      DO 165 II=L1,L2
 165  AAA(I,II)=1.
      IX1=IX1+1
      IX2=IX1+1
      DO 168 IX0=1,NN
      IX=IX0+I
      AAA(IX,L0)=Z(IX2,IX0)
```

```
168   AAA(IX,L2)=-7(IX1,IX0)
      L0=L0-1
170   CONTINUE
      IX=IX+1
      B(IX)=1.
      DO 172 IX0=1,NN
172   AAA(IX,IX0)=1.
      CALL LEQT1B(AAA,IMXND2,NLC,NUC,IA1,B,1,I71,IJ081,XL,IER)
      IF (IER.EQ.129) STOP
      DO 180 I=1,IMAX
      DO 180 IX=1,NN
      I0=(I-1)*JKL+(IX-1)*JKMAX+IL
      II=(I-1)*NN+IX
      SPMMT(I0)=B(II)
180   CONTINUE
      RETURN
      END
```

```
*IDENT EXP2
*/ FOR CHANGING STORAGE TO INCLUDE SPMMT AND FREQ
*/ DEFINE IN ADEP PARAM.FOR VARIABLE DIMENSION
*INSERT ADEP.46
      JKL=IMJKM+NDJKM
      JKLG=JKL*IMAX
*DELETE ADEP.59
     1X+4*IMJKM+JKL+JKLG
*DELETE ADEP.128
      N47=N46+JKL
      N48=N47+JKLG
      PRINT 40,N48
*DELETE ADEP.133,ADEP.134
     438),Y(N39),Y(N40),Y(N41),Y(N42),Y(N43),Y(N44),Y(N45),Y(N46),Y(N47)
     5,IMJKM,NDJKM,IMLTHN,LTH,JKMAX,IMLTH,IM1LTH,NW,NWD,JMAX,IILN,NC,IMA
     6X,KMAX,JKL,JKLG)
*DELETE FLUX.4,FLUX.5
     2IDL,V,D,SIGF,SIG,SIGR,W,WD,RL,RR,DXW,COET,FREQ,SPMMT,IMJKM,NDJKM,I
     3MLTHN,LTH,JKMAX,IMLTH,IM1LTH,NW,NWD,JMAX,IILN,NC,IMAX,KMAX,JKL,JKL
     4G)
*INSERT FLUX.20
      DIMENSION FREQ(JKL),SPMMT(JKLG)
*INSERT FLUX.39

C   BETA=0. FOR THE CASE WITHOUT DELAYED NEUTRONS
      IMAXND=IMAX+ND
      IF (BETA .EQ. 0.) IMAXND=IMAX

*DELETE FEEDBAK.40
 10   IF (IEXP.NE.1) GO TO 20
*/ THIS SET OF UPDATE CORRECTS FOR ERRORS IN ADEP CODE
*DELETE FLUX.153,FLUX.156
 70   X0=1.-A1(L0)+X1+A3(L0)+OMDT2(IL1)
      A(I1)=((1+A1(L0)-X2-A5(L0)-OMDT2(IL1))*PHI(IL1)+X2*PHI(IL1+1)+A5(L
     10)*PHI(IL1+JIND)+X1*PHI(IL1-1)+A3(L0)*PHI(IL1-JIND)+A6(L0)*ALC+SM(
     2I))/X0
*DELETE FLUX.164,FLUX.167
 85   X0=1.-A1(L0)+X2+A5(L0)+OMDT2(IL1)
      A(I1)=((1.+A1(L0)-X1-A3(L0)-OMDT2(IL1))*PHI(IL1)+X1*PHI(IL1-1)+A3(
     1L0)*PHI(IL1-JIND)+X2*PHI(IL1+1)+A5(L0)*PHI(IL1+JIND)+A6(L0)*ALC+SM
     2(I))/X0
```

```
*IDENT MTT1
*/ SAME OMEGA AND P FOR BOTH SWEEPS, BASED ON FLUXES AT START OF PRESENT S
*INSERT FLUX.72
      CALL SETMF(PHI,C,ASF,V,D,SIG,SIGR,W,WD,FREQ,SPMMT,IMJKM,NDJKM,IMLT
     1TN,LTH,JKMAX,IMLTH,IM1LTH,NW,NWD,IMAX,IMAXND,JKL,JKLG)
      DO 425 I=1,IMAX
      IL2=(I-1)*JKMAX
      IL0=0
      I1=IL2*IMAXND
      DO 425 K=2,KM1
      IL0=IL0+JIND
      J1=JI(K)+IL0+1
      J2=JF(K)+IL0-1
      DO 425 IL=J1,J2
      IL1=IL+IL2
      I2=-JKMAX+IL
      X1=0.
      X2=0.
      DO 418 IX=1,IMAXND
      I2=I2+JKMAX
      I3=I1+I2
      X3=SPMMT(I3)*EXP(DT*FREQ(I2))
      X1=X1+X3
      X2=X2+X3*FREQ(I2)
  418 CONTINUE
      OMDT2(IL1)=.5*DT*X2/X1
      P(IL1)=X1
  425 CONTINUE
*DELETE FLUX.212
  135 IF (IEXP.NE.1) GO TO 145
```

```
*IDENT MTT2
*/ DIFFERENT OMEGA AND EXPONENTIAL TERMS IN ALTERNATE SWEEPS
*INSERT FLUX.71
      CALL SETMF(PHI,C,ASF,V,D,SIG,SIGR,W,WD,FREQ,SPMMT,IMJKM,NDJKM,IMLT
     1HN,LTH,JKMAX,IMLTH,IM1LTH,NW,NWD,IMAX,IMAXND,JKL,JKLG)
      DT1=DT
*INSERT FLUX.76
      I1=IL2*IMAXND
*INSERT FLUX.82
      IF (IEXP.EQ.1) GO TO 20
      I2=-JKMAX+IL
      X1=0.
      X2=0.
      DO 18 IX=1,IMAXND
      I2=I2+JKMAX
      I3=I1+I2
      X3=SPMMT(I3)*EXP(DT1*FREQ(I2))
      X1=X1+X3
      X2=X2+X3*FREQ(I2)
 18   CONTINUE
      OMDT2(IL1)=0.5*DT*X2/X1
      P(IL1)=Y1/PTN(IL1)
      PTN(IL1)=X1
      IF (IP .EQ. 2) PTN(IL1)=1.
*INSERT FLUX.210
      IF (IEXP.EQ.2) DT1=DT1+DT1
*DELETE FLUX.212
 135  IF (IEXP.NE.1) GO TO 145
*DELETE SETCON.242
      PTN(IL)=PHI(IL)
      IF (IEXP.EQ.2) PTN(IL)=1.
 120  CONTINUE
```

```
*IDENT MTT3
*/ SAME OMEGA AND P=EXP(DT*OMEGA) FOR BOTH SWEEPS
*INSERT FLUX.72
      CALL SETMF(PHI,C,ASF,V,D,SIG,SIGR,W,WD,FREQ,SPMMT,IMJKM,NDJKM,IMLT
     1TN,LTH,JKMAX,IMLTH,IM1LTH,NW,NWD,IMAX,IMAXND,JKL,JKLG)
      DO 425 I=1,IMAX
      IL2=(I-1)*JKMAX
      IL0=0
      I1=IL2*IMAXND
      DO 425 K=2,KM1
      IL0=IL0+JIND
      J1=JI(K)+IL0+1
      J2=JF(K)+IL0-1
      DO 425 IL=J1,J2
      IL1=IL+IL2
      I2=-JKMAX+IL
      X1=0.
      X2=0.
      DO 418 IX=1,IMAXND
      I2=I2+JKMAX
      I3=I1+I2
      X3=SPMMT(I3)*EXP(DT*FREQ(I2))
      X1=X1+X3
      X2=X2+X3*FREQ(I2)
  418 CONTINUE
      OMDT=DT*X2/X1
      OMDT2(IL1)=.5*OMDT
      P(IL1)=EXP(OMDT)
  425 CONTINUE
*DELETE FLUX.212
  135 IF (IEXP.NE.1) GO TO 145
```