MITNE-196
C. 3

NONLINEAR METHODS FOR SOLVING THE DIFFUSION EQUATION

by

R.A. Shober, A.F. Henry

November 1976

Massachusetts Institute of Technology
Department of Nuclear Engineering
Cambridge, Massachusetts 02139

Electric Power Research Institute Report

MITNE-196

NONLINEAR METHODS FOR SOLVING THE DIFFUSION EQUATION

by

R.A. Shober, A.F. Henry

November, 1976

Massachusetts Institute of Technology
Department of Nuclear Engineering
Cambridge, Massachusetts 02139

Electric Power Research Institute Report

NONLINEAR METHODS FOR SOLVING THE DIFFUSION EQUATION

by

Robert Anthony Shober

Submitted to the Department of Nuclear Engineering
on November 15, 1976, in partial fulfillment of the
requirements for theDegree of Doctor of Philosophy

ABSTRACT

This thesis is concerned with methods for the transient solution of the neutron diffusion equations in one or two energy groups. Initially, nonlinear methods for solving the static diffusion equations using the finite element method were investigated. By formulating a new eigenvalue equation, some improvement in the solution efficiency was obtained. However, the transient solution of the diffusion equation using the finite element method was considered to be overly expensive.

An analytic method for solving the one-dimensional diffusion equation was then developed. Numerical examples confirmed that this method is exact in one dimension. The method was extended to two dimensions, and results compared employing two different approximations for the transverse leakage. The method based on a flat approximation to the leakage was found to be superior, and it was extended to time-dependent problems. Results of time-dependent test problems show the procedure to be accurate and efficient. Comparisons with conventional finite difference techniques (such as TWIGL or MEKIN) indicate that the scheme can be an order of magnitude more cost effective.

Thesis Supervisor: Allan F. Henry

Title: Professor of Nuclear Engineering

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

The author wishes to thank his thesis supervisor, Professor Allan F. Henry, for his encouragement, assistance, and support throughout the course of this investigation. It has been an extreme pleasure to be associated with him during the author's experience at MIT. Thanks are also due to Professor Kent F. Hansen for his help at various times during this period.

The author expresses his thanks to his wife, Tika, who provided love and support during the course of this research. I thank God for the pleasure of sharing life with her.

This research was supported by the Electric Power Research Institute, whose assistance is gratefully acknowledged. All computation was performed on an IBM 370/168 at the MIT Information Processing Center.

Finally, thanks are due to Mrs. Esther Grande, who typed this thesis with considerable skill and care.

BIOGRAPHICAL NOTE

Robert Anthony Shober was born October 18, 1948 in St. Louis, Missouri. He attended elementary school in Kirkwood, Mo., and secondary school in Ft. Lauderdale, Florida. He graduated from Ft. Lauderdale High School in June, 1966.

He enrolled at the University of Florida, Gainesville, Florida in September, 1966. While an undergraduate he participated on the baseball team, and was supported by a partial Athletic Scholarship. In December, 1970, he received the degree Bachelor of Science in Nuclear Engineering Sciences. He entered Graduate School at the University of Florida in January, 1971, and received the degree Master of Science in Nuclear Engineering in August, 1972. His Master's Thesis dealt with Hybrid Computer methods for space-time kinetics.

From August, 1972 to January, 1974 he worked at Combustion Engineering in Windsor, Connecticut. He was active in development and analysis of methods for light-water reactor safety analysis.

In February, 1974, he entered Massachusetts Institute of Technology as a graduate student in the Department of Nuclear Engineering.

Mr. Shober is married to the former Titika Michaels of Westwood, Mass.

Chapter 1

INTRODUCTION

1.1 Introduction

The design and safety analysis of present-day light-water-moderated nuclear reactors requires very sophisticated mathematical and computational techniques. The complexity of safety analyses required for reactor licensing today far exceeds that which was needed seven to ten years ago. Although during this time period digital computer technology has improved dramatically, a realistic accident analysis of a large light-water reactor remains a very expensive calculation.

The objective of this thesis is to develop economical computational techniques for transient analysis of light-water reactors. In this chapter solution techniques in present use will be reviewed, and the scope of the present investigation will be described.

1.2 The Problem to Be Solved

The analysis of light-water-moderated nuclear reactors is most frequently performed by solving the diffusion equation in few energy-group form. This equation is a low-order approximation to the Boltzmann transport equation, a much more exact equation which may be derived from first principles. The diffusion equation is a parabolic partial differential equation with variable coefficients. Thus, for any practical configuration, the equations are not analytically solvable; and numerical schemes must be employed. Let us write the time-dependent, energy-group diffusion equations:[1]

$$\frac{1}{v_g}\frac{\partial}{\partial t}\phi_g(\underline{r},t) = \underline{\nabla}\cdot D_g(\underline{r},t)\,\underline{\nabla}\phi_g(\underline{r},t) + \sum_{g'=1}^{G}\Sigma_{gg'}(\underline{r},t)\,\phi_{g'}(\underline{r},t)$$

$$+ \sum_{i=1}^{I}\chi_{dgi}\lambda_i C_i(\underline{r},t) \qquad (1 \leq g \leq G)$$

$$(1.1)$$

$$\frac{\partial}{\partial t}C_i(\underline{r},t) = -\lambda_i C_i(\underline{r},t)$$

$$+ \sum_{g'=1}^{G}\beta_i \nu\Sigma_{fg'}(\underline{r},t)\,\phi_{g'}(\underline{r},t) \qquad (1 \leq i \leq I)$$

where

G = total number of neutron energy groups

I = total number of delayed precursor groups

ϕ_g = neutron flux in group g (n/cm^2 sec)

C_i = density of the i'th precursor (cm^{-3})

v_g = neutron speed in group g (cm/sec)

D_g = diffusion coefficient for group g (cm)

$\Sigma_{gg'}$ = macroscopic transfer cross section from group g' to g (cm^{-1}), where

$$\Sigma_{gg} = \chi_{pg}(1-\beta)\,\nu\Sigma_{fg} - \Sigma_{ag} - \sum_{g'\neq g}\Sigma_{sg'g}$$

χ_{pg} = prompt fission spectrum in group g

$\nu\Sigma_{fg}$ = nu, the number of neutrons per fission, times the macroscopic fission cross section in group g

Σ_{ag} = macroscopic absorption cross section in group g

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

β = total fractional yield of delayed neutron per fission

$\Sigma_{gg'} = \chi_{pg} \nu \Sigma_{fg'}(1-\beta) + \Sigma_{sgg'}, \quad g' \neq g$

λ_i = decay constant of the i'th group of delayed emitters

χ_{dgi} = fraction of group i delayed neutrons which appear in neutron group g

β_i = fraction of fissions which produce a delayed emitter of group i.

If the material constants are known as a function of $\underline{r}$ and t, and if an initial flux distribution in energy and space are known, then a unique solution to Eq. (1.1) may be obtained.

The solution of Eq. (1.1) can be divided into two stages. First, the time-independent version of Eq. (1.1), obtained by setting all the time derivative terms to zero, is solved to find the initial flux distribution in energy and space. Then, the time dependence is introduced; and the solution is advanced in time.

This thesis will not have the development of time integration methods as its principal objective. However, any method developed to solve the time-independent version of Eq. (1.1) should be extendable to time-dependent problems. Let us then write the time-independent version of Eq. (1.1):

$$-\underline{\nabla} \cdot D_g(\underline{r}, t) \, \underline{\nabla}\phi_g(\underline{r}, t) - \sum_{g'=1}^{G} \Sigma_{gg'}(\underline{r}, t) \, \phi_{g'}(\underline{r}, t) = 0 \qquad (1 \leq g \leq G) \qquad (1.2)$$

where now

$$\Sigma_{gg} = [\chi_{pg}(1-\beta) + \sum_i \chi_{dgi}\beta_i]\nu\Sigma_{fg} - \Sigma_{ag} - \sum_{g'\neq g} \Sigma_{sg'g}$$

$$\Sigma_{gg'} = [\chi_{pg}(1-\beta) + \sum_i \chi_{dgi}\beta_i]\nu\Sigma_{fg'} + \Sigma_{sgg'} \qquad g' \neq g.$$

Light water reactors are designed using a rectangular x-y-z coordinate system. Any x-y slice contains a great deal of geometric complexity, while the z direction is relatively homogeneous. In the x-y plane, the geometric detail is divided into several stages. The reactor is composed of a large number of "fuel elements," cylindrical rods which are 1/3" to 1/2" in diameter. The reactor is composed of a rectangular lattice of these fuel elements. The fuel elements are partitioned into rectangular "fuel assemblies," squares ~8" on a side. The fuel elements in each assembly are generally identical for the initial core loading (that is, they all have the same enrichment of U^{235}).

The first stage of a full core reactor analysis is the calculation of "equivalent, homogenized, diffusion theory constants" for each fuel element and surrounding water.[1] This is normally performed by solving an approximation to the transport equation more accurate than the diffusion approximation. Once these constants have been found, the solution of the diffusion equation (1.2) can commence.

In the x-y plane, the geometric detail needed to describe each fuel element yields roughly 250 × 250 mesh points. Thus, for most solution methods, a full core time-dependent reactor analysis would be extremely costly. To reduce the geometric detail required, equivalent homogenized constants for each fuel assembly are found, using the sets of equivalent constants for each fuel element, control rod, water hole, etc.[1] This procedure yields a regular square array of homogenized regions each ~8" wide. Equation (1.2) can then be solved using this lesser degree of geometric detail and the homogenized material constants associated with it. It is the solution of this "homogenized" problem to which this thesis will be addressed.

1.3 Review of Solution Techniques

A wide variety of techniques for solving Eq. (1.2) have been investigated. Some of the most successful of these methods are finite difference methods, synthesis methods, finite element methods, and nodal methods. These will now be described briefly.

The most common solution methods for Eq. (1.2) are the finite difference methods.[4, 5] These methods use a low-order finite difference approximation to the second derivative term $\underline{\nabla} \cdot D_g(\underline{r}, t)\ \underline{\nabla}\phi_g(\underline{r}, t)$. Several properties of finite difference methods are very advantageous:

i) The matrix equations which must be solved have a very simple structure.

ii) The matrices are easily formed.

iii) The method can be guaranteed to converge to the correct solution of the differential equation in the limit of small mesh spacings.

When applied to the "homogenized" problem, however, these techniques often require a large number of mesh points in order to obtain an accurate solution of the equations. Therefore, the number of unknowns required is high, not because of the geometric detail, but for accuracy considerations.

If the fuel elements within an assembly are not homogenized as described above, the geometric detail requires a very large number of mesh regions. A solution technique for which this is not a great drawback is the synthesis method.[2] This method finds an approximate solution to Eq. (1.2) by taking linear combinations of precomputed "trial functions." In the most successful of these methods, we expand the neutron flux in each energy group as follows:[3]

$$\phi_g(x, y, z) = \sum_{k=1}^{K} \psi_{gk}(x, y)\ T_{gk}(z)$$

where $\psi_{gk}(x, y)$ are precomputed, two-dimensional expansion functions. The accuracy of the method rests on how well the correct solution $\phi_g(x,y,z)$ can be approximated at each axial level by linear combinations of the $\psi_{gk}(x, y)$.

Synthesis methods achieve a drastic reduction in the number of unknowns; however, a poor choice of expansion functions can give an inaccurate answer. Moreover, there is no systematic way to estimate the magnitude of this error. This lack of a definite error bound has prevented synthesis schemes from being widely accepted.

Another method for solving Eq. (1.2) which has received considerable attention is the finite element method.[6,7] Some advantages of this method are:

i) The matrix equations to be solved, although more complex than those of finite differences, retain a generally simple structure.

ii) The solution can be guaranteed to converge to the correct solution of the differential equation in the limit of small mesh spacings.

iii) A high-order polynomial approximation may be used, thus accurate solutions are obtained for large mesh regions.

The major disadvantage of this method appears to be that the coefficient matrices become sufficiently complex that the time needed to solve the problem exceeds considerably that which would be expected from the number of unknowns. For that reason, a method of high accuracy which retains the simple matrix structure of finite differences is more desirable.

Nodal methods have been in existence for some time.[8] A nodal approximation results when the reactor is divided into a relatively small number of coupled regions, and the calculation is oriented toward obtaining the average flux or power level in each region. The "coupling coefficients" between adjacent nodes are generally not defined in a rigorous way.

More recently, successful attempts have been made to combine the "nodal" approach with more systematic means for calculating the couplings between nodes.[9, 10, 11] Most of these techniques utilize a local polynomial expansion to calculate couplings between two adjacent nodes. These techniques have the promise of providing accurate solutions to Eq. (1.2) while using a coarse spatial mesh.

1.4 Summary

This thesis will be concerned with both the finite element method and the nodal method. In Chapter 2, nonlinear methods will be applied in an effort to reduce the coupling complexity of the finite element matrices. An eigenvalue updating scheme will be developed which can improve the running time of Eq. (1.2) when the finite element method is used.

A two-group nodal method for solving Eq. (1.2) in one dimension will be derived in Chapter 3. In Chapter 4 this method will be extended to two-dimensional problems, and the scheme will be extended to time-dependent problems in Chapter 5. Numerical results for some realistic two-dimensional reactor transients will be presented. Chapter 6 will summarize the investigations.

Chapter 2

SOLUTION OF THE FINITE ELEMENT EQUATIONS

2.1 Introduction

The finite element method, created in order to solve complex problems in structural mechanics, has found applications to many other problems in mathematical physics. It has been applied with success to the solution of the multigroup diffusion equation by a number of investigators.[12, 13, 14, 15] For practical computations, the finite element method offers several advantages over other methods:

i) High-order polynomials can be used, yielding very accurate approximations with a relatively few number of unknowns.

ii) The method yields a continuous approximation to the variables of interest, rather than a discrete approximation.

iii) Boundary conditions are easily imposed.

iv) The system of linear equations is amenable to computer solution by well developed methods.

Full core solutions for light-water reactors generally involve core configurations with large homogeneous (homogenized) regions. For problems of this kind, the finite element method has been shown to yield acceptable accuracy while maintaining a relatively large mesh size, and hence relatively few unknowns.[13, 15] Because of this promise for use in full core light-water reactor problems, an investigation into appropriate solution techniques for the finite element equations was carried out.

2.2 Properties and Structure of the Equations

The "finite element method" is the use of piecewise polynomials as expansion functions for the Ritz-Galerkin method. Thus, the generation of an approximate solution begins with the definition of an appropriate mathematical space of functions, from which an approximate solution will be selected by application of the Ritz-Galerkin method.

Let us write the time-independent multigroup diffusion equation in matrix form:

$$-\underline{\nabla} \cdot [D(\underline{r})]\, \underline{\nabla}[\phi(\underline{r})] + [\Sigma_T(\underline{r})][\phi(\underline{r})]$$

$$= \frac{1}{\lambda}[\chi][\nu\Sigma_f(\underline{r})]^T\, [\phi(\underline{r})] \qquad (2.1)$$

where

$[\phi(\underline{r})]$ is a column vector of length G containing the neutron fluxes (n/cm^2 sec)

$[D(\underline{r})]$ is a diagonal G × G matrix containing the diffusion coefficients (cm)

$[\Sigma_T(\underline{r})]$ is a G × G matrix containing the absorption-minus-scattering cross sections (cm^{-1})

$[\nu\Sigma_f(\underline{r})]$ is a column vector of length G containing nu, the number of neutrons per fission, times the fission cross section (cm^{-1})

$[\chi]$ is a column vector of length G containing the fission neutron spectrum

λ is the critical eigenvalue of the problem.

Solutions to Eq. (2.1) have been obtained for several different spaces of functions, and a variety of orders of polynomial approximations. For example, linear Hermite polynomials,[12, 13] cubic Hermite polynomials,[13, 15] and quadratic Lagrange polynomials[14] all have been used. For the purposes of the present discussion, the discrete equations obtained for the linear Hermite scheme will be described, and relationships to other schemes will be discussed later. Light-water reactors are generally designed using an orthogonal x-y-z coordinate system. Thus, for purposes of illustration, the linear finite element method in one (x) and two (x-y) dimensions will be described. We shall limit our investigation to two-group problems.

A one-dimensional reactor configuration is defined over the region $R = [0, X]$. For approximation by linear polynomials, this region is divided into a partition as follows:

$$\pi: \quad 0 = x_1 < \ldots < x_{I+2} = X.$$

We define the linear function $u_i(x)$ as follows:

$$u_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leqslant x \leqslant x_i \\[2ex] \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & x_i \leqslant x \leqslant x_{i+1} \\[2ex] 0, & \text{otherwise} \end{cases}$$

An approximate solution for the flux in neutron energy group g is then

$$\phi_g(x) = \sum_{i=1}^{I+2} \phi_{gi} u_i(x). \tag{2.3}$$

The functions $u_1(x)$ and $u_{I+2}(x)$ are chosen so that the boundary conditions are properly accounted for.[16] If, for example, $u_1(x) = u_{I+2}(x) = 0$, there are I number of independent variables ϕ_{gi} for group g.

Inserting Eq. (2.2) into Eq. (2.1), weighting by Eq. (2.2) and integrating over $0 \leq x \leq X$ yields the following matrix equation:

$$\left\{ - \begin{pmatrix} [L_{11}] & 0 \\ 0 & [L_{22}] \end{pmatrix} + \begin{pmatrix} [A_{11}] & 0 \\ [A_{21}] & [A_{22}] \end{pmatrix} \right\} \begin{pmatrix} [\phi_1] \\ [\phi_2] \end{pmatrix}$$

$$= \frac{1}{\lambda} \begin{pmatrix} [\chi_1] \\ [\chi_2] \end{pmatrix} \begin{pmatrix} [M_1] \\ [M_2] \end{pmatrix}^T \begin{pmatrix} [\phi_1] \\ [\phi_2] \end{pmatrix} \qquad (2.4)$$

where

$-[L_{gg}]$ is a matrix of dimension I X I

$[A_{gg'}]$ is a matrix of dimension I X I

$[M_g]$ is a matrix of dimension I X I

$[\phi_g]$ is a column vector of length I

and

$$-L_{gg_{i,i'}} = \left(D_g \frac{d}{dx} u_{i'}(x), \frac{d}{dx} u_i(x)\right)$$

$$A_{gg_{i,i'}} = (\Sigma_{Tg} u_{i'}(x), u_i(x))$$

$$A_{21_{i,i'}} = (\Sigma_{r1} u_{i'}(x), u_i(x))$$

$$M_{g_{i,i'}} = (\nu \Sigma_{fg} u_{i'}(x), u_i(x))$$

where

$$(u, v) = \int_{\Omega} u(\underline{r})\ v(\underline{r})\ d\underline{r}.$$

Rewriting Eq. (2.4) in a more concise notation, we have

$$[-\mathcal{L} + \mathcal{A}][\Phi] = \frac{1}{\lambda}[\mathcal{M}][\Phi]. \tag{2.5}$$

Each of the matrices in Eq. (2.4) is tridiagonal. This is because of the local nature of the linear "tent functions." Thus, Eq. (2.5) can be solved using conventional methods. Specifically, a fission source iteration can be applied to converge the eigenvalue and fluxes.[18] For each fission source iteration, two matrix problems of the form

$$[B_g][\Phi_g] = [S_g] \tag{2.6}$$

must be solved. However, since each $[B_g]$ is tridiagonal, obtaining the solution is simple.

The above techniques are identical to those which would be used to solve one-dimensional finite difference equations. The distinction between the two methods lies in the fact that for finite differences the matrices $[A_{gg'}]$ and $[M_g]$ are diagonal rather than tridiagonal.

For two-dimensional (x-y) situations the equations become more complex. The region R is defined

$$R = [0, X] \times [0, Y]$$

with this region divided into a partition:

$$\pi: \quad 0 = x_1 < \dots < x_I = X$$

$$0 = y_1 < \dots < y_I = Y.$$

The approximation to the flux is defined by a product $u_i(x)\ u_j(y)$. Let us assume that I and J are the number of partitions in the x and y directions, respectively. The application of the method in two dimensions to Eq. (2.1) yields an equation of the same form of Eq. (2.4), with the exception that all matrices are of dimension IJ $\times$ IJ. We now have

$$\begin{aligned}
-L_{gg_{i,i'}} &= (D_g \underline{\nabla} u_{i'}(\underline{r}),\ \underline{\nabla} u_i(\underline{r})) \\
A_{gg_{i,i'}} &= (\Sigma_{Tg} u_{i'}(\underline{r}),\ u_i(\underline{r})) \\
A_{21_{i,i'}} &= (\Sigma_{r1} u_{i'}(\underline{r}),\ u_i(r)) \\
M_{g_{i,i'}} &= (\nu \Sigma_{fg} u_{i'}(\underline{r}),\ u_i(r)).
\end{aligned} \tag{2.7}$$

In two dimensions, the structure of each of the above matrices is block tridiagonal, with each block being tridiagonal. A nine-stripe structure results.

Let us draw a two-dimensional grid about point (i, j):

x (i-1, j+1)	x (i, j+1)	x (i+1, j+1)
x (i-1, j)	x (i, j)	x (i+1, j)
x (i-1, j-1)	x (i, j-1)	x (i+1, j-1)

We note that point (i, j) is coupled not only to itself, but also to the eight surrounding points. This nine-point coupling is present in the $[L_{gg}]$, $[A_{gg'}]$, and $[M_g]$ matrices. In finite differences, all the matrices $[A_{gg'}]$ and $[M_g]$ are diagonal and $[L_{gg}]$ is only five-stripe; point (i, j) being coupled to only its four immediate nearest neighbors. Since the matrices $[L_{gg}]$, $[A_{gg'}]$, and $[M_g]$ in Eq.(2.4) are symmetric and positive definite,[13, 16] and the equations are solved one group at a time as in Eq. (2.6), it can be proved that the matrices $[B_g]$ in Eq. (2.6) are positive definite. Therefore a block successive over-relaxation iteration scheme can be guaranteed to converge for all overrelaxation parameters ω such that $0 < \omega < 2$.[17]

The eigenvalue and eigenvector estimates can be found by a fission source iteration with Chebyshev acceleration, or by coarse mesh rebalancing.[18, 19] As mentioned above, the solution of the matrix equations (2.6) can be obtained by successive over-relaxation.

The equations obtained from the linear finite element method have a relatively simple structure since the unknowns represent the height of the "tents" at each mesh point. Although the use of higher order polynomials yields more accurate approximations, this accuracy is obtained at the expense of more complex coefficient matrices. The use of cubic Hermite polynomials in two dimensions (at nonsingular points) generally leads to four unknowns per group at each mesh point $(\phi, \phi_x, \phi_y, \phi_{xy})$. The nine-point structure found for linear elements is extended to 36-point coupling for the cubics (each of four unknowns coupled to 9 adjacent points). This structure holds for not only the $[L_{gg}]$ matrices, but also $[A_{gg'}]$ and $[M_g]$. Thus, the straightforward implementation of successive over-relaxation schemes will be complicated because of data-handling problems, and much denser matrices.

2.3 Nonlinear Iterative Schemes

The increased complexity of coefficient matrices in the finite element method increases execution times to more than would be expected in view of the number of unknowns. Thus it was desired to investigate schemes which would reduce the degree of coupling by nonlinear means. The two facets of this investigation were:

i) To reduce the coupling complexity of the matrices $[A_{21}]$ and $[M_g]$ from nine-stripe to diagonal (the same structure as finite difference).

ii) To reduce the coupling complexity of the $[B_g]$ matrices in Eq. (2.6) from nine-stripe to five-stripe (the same structure as finite differences).

Item (i) would help relieve data-handling problems, and speed the calculation of the vectors $[S_g]$ in Eq. (2.6). Item (ii) would speed the solution of Eq. (2.6) because of the sparseness of the matrix, and also by opening up the possibility of using the Cyclic Chebyshev method of iteration instead of successive over-relaxation. The Cyclic Chebyshev method is to be preferred because of its increased average rate of convergence.[17]

An investigation was carried out concerning the above objectives. A two-dimensional, two-group model of a PWR was used for eigenvalue calculations to test the various methods proposed. Results will be reported in a qualitative fashion only.

To illustrate the nonlinear reduction of a nine-stripe matrix to a diagonal matrix, let us write

$$[\Sigma][\phi] = [\Lambda][\phi] \tag{2.8}$$

where $[\Sigma]$ is a nine-stripe matrix, $[\Lambda]$ is a diagonal matrix, and $[\phi]$ is a column vector of fluxes. The diagonal elements of $[\Lambda]$ are given by $[1/\phi][\Sigma][\phi]$, where $[1/\phi]$ is a diagonal matrix whose entries are the inverse flux values, $(1/\phi_{gij})$. A nonlinear procedure would be to "diagonalize" the $[\Sigma]$ matrix by use of the most recent flux iterate. These nonlinear updates could be performed after a numuer of fission source (outer) iterations have been performed.

Numerical tests of the above scheme show that the method will converge to the correct solution of the difference equations using four or five nonlinear updates interspersed in the outer iterations. However, the nonlinear updates appear to degrade the rate of convergence that the Chebyshev polynomiaıs, used to accelerate the outer iterations, provide. Although less work is required to form the source terms, the increased number of outer iterations needed to solve the problem makes the overall solution time much longer. These results show that although the nonlinear updating procedure will converge, the potentiaı time savings for each outer iteration is insufficient to warrant auoption of the scheme.

It was felt that scheme (ii) held a much larger potential for time savings, especially since iterative schemes like Cyclic Chebyshev could then be used.

The collapsing of nine-stripe matrices to five-stripe matrices can be done in a variety of ways. The general transformation is

$$[L][\phi] = [L'][\phi] \qquad (2.9)$$

where $[L]$ is a nine-stripe matrix, and $[L']$ is a five-stripe matrix. We assume that the number of unknowns in the x and y directions are I and

J, respectively. Let us order the column vector [ϕ] first by rows, then by columns; such that the (i, j) element of [ϕ] is indexed k, where k = i + (k-1) * I. Then $L_{k,k'}$ represents the (k, k') element in the [L] matrix. We can write the k th equation of Eq. (2.9) as:

$$\begin{aligned} & L_{k,k-I-1}\phi_{k-I-1} + L_{k,k-I}\phi_{k-I} + L_{k,k-I+1}\phi_{k-I+1} \\ & \quad + L_{k,k-1}\phi_{k-1} + L_{k,k}\phi_k + L_{k,k+1}\phi_{k+1} \\ & \quad + L_{k,k+I-1}\phi_{k+I-1} + L_{k,k+I}\phi_{k+I} + L_{k,k+I+1}\phi_{k+I+1} \\ & = L'_{k,k-I}\phi_{k-I} + L'_{k,k-1}\phi_{k-1} + L'_{k,k}\phi_k \\ & \quad + L'_{k,k+1}\phi_{k+1} + L'_{k,k+I}\phi_{k+I} \end{aligned} \tag{2.10}$$

where the coefficients $L'_{k,k-I}$, $L'_{k,k-1}$, $L'_{k,k}$, $L'_{k,k+1}$, and $L'_{k,k+I}$ are unknown.

While Eq. (2.8) provides a unique solution for the diagonal elements of [Λ], there is no unique solution to Eq. (2.10). Three possible schemes used to obtain an [L'] matrix which satisfies Eq. (2.10) are:

a) Let the contribution of the four corner points be "folded" into the diagonal, as:

$$\begin{aligned} L'_{k,k}\phi_k = {} & L_{k,k-I-1}\phi_{k-I-1} + L_{k,k-I+1}\phi_{k-I+1} \\ & + L_{k,k}\phi_k + L_{k,k+I-1}\phi_{k+I-1} + L_{k,k+I+1}\phi_{k+I+1} \end{aligned} \tag{2.11}$$

with the other matrix elements $L_{i,j} = L'_{i,j}$.

b) Let the contribution of the four corner points be "folded" into the four adjacent off-diagonal points; for example, for point (k, k-I):

$$L'_{k,k-I}\phi_{k-I} = \frac{1}{2}(L_{k,k-I-1}\phi_{k-I-1} + L_{k,k-I+1}\phi_{k-I-1}) + L_{k-I}\phi_{k-I}. \tag{2.12}$$

c) Let part of the contribution of the four corner points be "folded" into the diagonal, and part be folded into the off-diagonal, as:

$$L'_{k,k}\phi_k = z(L_{k,k-I-1}\phi_{k-I-1} + L_{k,k-I+1}\phi_{k-I+1} + L_{k,k+I-1}\phi_{k+I-1} + L_{k,k+I+1}\phi_{k+I+1}) + L_{k,k}\phi_k \tag{2.13a}$$

and for the point (k, k-I):

$$L'_{k,k-I}\phi_{k-I} = \frac{(1-z)}{2}(L_{k,k-I-1}\phi_{k-I-1} + L_{k,k-I+1}\phi_{k-I+1}) + L_{k-I}\phi_{k-I} \tag{2.13b}$$

where z ranges from 0 (scheme b) to 1 (scheme a).

The above updating schemes were tested for a variety of z values. The nonlinear updates were inserted after a number (typically eight or ten) of outer iterations had been performed. The results obtained are summarized in Table 2.1.

The results in Table 2.1 show that the nonlinear process is convergent in only a small range of z values, and behaves very poorly when outside that range. Even though convergence was achieved for $z = .3$, the degrading effect of the updating procedure on the convergence rate of the outer iterations caused the running time to be much longer than that obtained by using standard linear iterative techniques.

Thus the nonlinear updating procedure of scheme (i), although computationally inefficient, did converge to the correct results. Whereas,

TABLE 2.1

Summary of Nonlinear Iterative Results

Value of z	Results Obtained
0 (scheme b)	Problem did not converge; eigenvalue oscillates with each update.
.2	Fluxes converge, although a slight eigenvalue oscillation with each update was noticed.
.3	Converged with no oscillations.
.5	Eigenvalue diverged farther with each update.
1.0 (scheme a)	Divergent results, negative fluxes obtained.

scheme (ii) generally failed to converge and proved to be very unstable.

To understand the causes of this behavior, let us rewrite Eq. (2.5):

$$\lambda[\Phi] = [-\mathcal{L} + \mathcal{A}]^{-1} [\mathcal{M}][\Phi]. \tag{2.14}$$

If $[\Phi]$ is the converged flux vector, the replacement of $[\mathcal{M}][\Phi]$ by $[\Lambda][\Phi]$ will not affect Eq. (2.14). However, the replacement of $[-\mathcal{L} + \mathcal{A}]$ by $[-\mathcal{L}' + \mathcal{A}']$ as above may give different results, since we have no guarantee that $[-\mathcal{L}' + \mathcal{A}']^{-1} = [-\mathcal{L} + \mathcal{A}]^{-1}$. In addition, when part of the contribution of the off-diagonal elements is added into the diagonal (schemes a or c) the potential exists for $L'_{k,k}$ to be either zero or negative. This situation may account for the oscillations observed, and the negative fluxes obtained.

In summary, attempts to improve the execution times of the linear finite element equations through nonlinear collapsing techniques proved to be either computationally inefficient or unstable. It appears that efforts to increase solution efficiencies must be directed elsewhere.

2.4 An Eigenvalue Updating Scheme

The methods introduced in the previous section to reduce the coupling complexity of the $[B_g]$ matrices in Eq. (2.6) were all based on keeping the full $[B_g]$ matrix on the left side of the equation, and then modifying its form. We shall now outline another possible scheme.

Let us begin with Eq. (2.5):

$$[-\mathcal{L} + \mathcal{Q}][\Phi] = \frac{1}{\lambda}[\mathcal{M}][\Phi]. \tag{2.5}$$

We split

$$[-\mathcal{L} + \mathcal{Q}] = [-\mathcal{L} + \mathcal{Q}]_1 + [-\mathcal{L} + \mathcal{Q}]_2. \tag{2.15}$$

Letting n be the outer iteration index and substituting Eq. (2.15) into Eq. (2.5), we have:

$$[-\mathcal{L} + \mathcal{Q}]_1 [\Phi]^{n+1} = \left(\frac{1}{\lambda}[\mathcal{M}] - [-\mathcal{L} + \mathcal{Q}]_2\right)[\Phi]^n.$$

Thus,

$$[\Phi]^{n+1} = [-\mathcal{L} + \mathcal{Q}]_1^{-1}\left(\frac{1}{\lambda}[\mathcal{M}] - [-\mathcal{L} + \mathcal{Q}]_2\right)[\Phi]^n \tag{2.16}$$

or

$$[\Phi]^{n+1} = [C(\lambda)][\Phi]^n.$$

One possible splitting of $[-\mathcal{L} + \mathcal{Q}]$ is to place the four corner point coefficients of $[-\mathcal{L} + \mathcal{Q}]$ in $[-\mathcal{L} + \mathcal{Q}]_2$, and the remaining coefficients in $[-\mathcal{L} + \mathcal{Q}]_1$. This splitting permits an application of the conventional fission source iteration.[20] However, attempting the procedure led to difficulties in obtaining a converged solution.

An examination of the mathematics involved suggests why the above scheme failed. The expression often used to estimate the eigenvalue for this outer iteration, $\|\Phi^{n+1}\| / \|\Phi^{n}\|$, is an estimate of the largest eigenvalue of $[C(\lambda)]$. However, this quantity is not the same as the number λ. Therefore, we should formulate a new eigenvalue problem, as:

$$\omega[\Phi] = [C(\lambda)][\Phi] \tag{2.17}$$

where ω is the eigenvalue of $[C(\lambda)]$. We note that if $\omega = 1$, then λ is the critical eigenvalue (k_{eff}).

We propose to solve the eigenvalue problem (2.17) instead of Eq. (2.5). The eigenvalue ω obtained will be used to "search" for a value of λ for which the spectral radius of $[C(\lambda)]$ is one. This reestimation of λ will be performed every eight or ten outer iterations of Eq. (2.17).

The formula used to calculate λ given ω was:

$$\lambda_{new} = \lambda_{old}\left[\frac{(\omega - 1)}{2} + 1\right].$$

Since the nonlinear updates may degrade the convergence of the outer iterations to some degree, it was desired to reduce the computational effort involved in evaluating $[C(\lambda)][\Phi]^n$. Thus, a different splitting from that previously used was attempted. Specifically, the diagonal elements of the matrices $(-[L_{11}]+[A_{11}])$ and $(-[L_{22}]+[A_{22}])$ and also the $[A_{21}]$ matrix (see Eq. (2.9)) were included in $[-\mathcal{L} + \mathcal{A}]_1$. Thus $[-\mathcal{L} + \mathcal{A}]_1$ was trivial to invert.

Table 2.2 presents a summary of the results obtained when this scheme was applied to the two-dimensional PWR problem referred to at the beginning of Section 2.3.

TABLE 2.2

Results of the Eigenvalue Updating Scheme

	Conventional Chebyshev and Successive Over-Relaxation	Eigenvalue Updating Scheme with Chebyshev Acceleration on Outers
Convergence Criteria	10^{-4}	10^{-4}
Converged λ	1.03105	1.03102
Number of Outers	29	49
Number of Inner Iterations per Outer (per group)	3	1
Run Time (sec)	4.06	3.26

2.5 Summary

The methods investigated in this chapter were designed to decrease the computational burden of solving matrix equations having the form of Eq. (2.6). The attempts to modify the coupling patterns in a nonlinear way were not successful. By forming a new eigenvalue equation, some improvement in execution time of the static finite element equations can be obtained. However, the improvement is not major.

It thus appears that the high degree of coupling in the finite element method is a major drawback of the method. The Hermite cubic elements are especially complex in this regard. The improvement in execution time obtained in Sec. 2.4 is not sufficient to make the finite element method significantly more attractive for production calculations. It appears that major improvements must come from the development of methods which retain the "nearest neighbor" coupling characteristic of finite difference methods.

Chapter 3

AN EXACT METHOD IN ONE DIMENSION

3.1 Introduction

The kinetic analysis of a nuclear reactor involves two different types of calculations:

i) Reactor physics calculations, where the power generated in every region of the reactor is found, and

ii) Thermal-hydraulics calculations, where the reactor temperatures coolant conditions are found.

A thermal and hydraulic reactor analysis, such as that performed by the computer code COBRA,[21] requires as part of its input the average power generated in each fuel assembly in the reactor for which core-wide thermal and hydraulic calculations are made. Thus, a method which directly calculates average fluxes over assembly sized regions would be very efficient.

In a recent thesis, Antonopoulos[22] developed one- and two-dimensional methods for solving Eq. (1.2) which began with a derivation of exact difference equations. However, Antonopoulos then made low-order series approximations to these exact difference equations. Numerical results using these approximations were somewhat improved over those of conventional finite differences; however, an extension to two groups and two dimensions failed to converge properly for mesh sizes larger than one centimeter.

In this chapter, a method for solving Eq. (1.2) which uses average fluxes as the quantities of interest will be derived from difference equations which are exact in one dimension. We shall show that this method is related to response matrix theory, and to other "nodal" techniques currently in the literature. In the final section, some numerical results for one-dimensional static problems will be presented.

3.2 Derivation of Exact Difference Equations

Let us rewrite Eq. (2.1) as a function of one independent variable:

$$-\frac{d}{dx}[D(x)]\frac{d}{dx}[\phi(x)] + [\Sigma_T(x)][\phi(x)] = \frac{1}{\lambda}[\chi][\nu\Sigma_f(x)]^T[\phi(x)]. \qquad (3.1)$$

We now divide the one-dimensional reactor configuration $R = [0, X]$ into a partition

$$\pi: \quad 0 < x_1 < \ldots < x_I = X,$$

with the restriction that any sub-region $x_i \leq x \leq x_{i+1}$ be nuclearly homogeneous. Let us integrate Eq. (3.1) over $x_i \leq x \leq x_{i+1}$:

$$[J(x_{i+1})] - [J(x_i)] + h_i[\Sigma_{Ti}][\bar{\phi}_i] = \frac{1}{\lambda}h_i[\chi][\nu\Sigma_{fi}]^T[\bar{\phi}_i] \qquad (3.2)$$

where

$$[J(x_i)] = -[D_i]\frac{d}{dx}[\phi(x)]\Big|_{x=x_{i+}}$$

$$[\bar{\phi}_i] = \frac{1}{h_i}\int_{x_i}^{x_{i+1}}[\phi(x)]\,dx$$

$$[\Sigma_{Ti}] = [\Sigma_T(x)]\Big|_{x \subset R_i}$$

$$R_i = (x_i, x_{i+1})$$

$$h_i = x_{i+1} - x_i.$$

Equation (3.2) shows that the average flux $[\bar{\phi}_i]$ in a region is dependent on the net current $[J(x_i)]$ on either side of the region. In order to obtain a solution to Eq. (3.2), another relationship is needed between $[J(x_i)]$ and the adjacent average fluxes $[\bar{\phi}_{i-1}]$ and $[\bar{\phi}_i]$. This relationship will be obtained by finding an analytic solution to a one-dimensional problem which spans two adjacent homogeneous regions.

To find this analytic solution, let us write Eq. (3.1) in P-1 form:

$$\frac{d}{dx}[J(x)] + [\Sigma_T(x)][\phi(x)] = \frac{1}{\lambda}[\chi][\nu\Sigma_f(x)]^T [\phi(x)]$$

$$\frac{d}{dx}[\phi(x)] + [D(x)]^{-1}[J(x)] = 0$$

and let us further define

$$[\Phi(x)] = \text{col}\left\{[\phi(x)], [J(x)]\right\} \tag{3.3a}$$

and

$$[N(x)] = \begin{bmatrix} [0] & [D(x)]^{-1} \\ [\Sigma_T(x)] - \frac{1}{\lambda}[\chi][\nu\Sigma_f(x)]^T & [0] \end{bmatrix} \tag{3.3b}$$

so that the above equation can be written

$$\frac{d}{dx}[\Phi(x)] + [N(x)][\Phi(x)] = 0.$$

This equation can be solved analytically over a homogeneous region R_i to give:[1]

$$[\Phi(x)] = e^{-[N_i](x-x_i)}[\Phi(x_i)]. \tag{3.4}$$

Since we wish to relate the current to the average flux over a region, we integrate Eq. (3.4) from x_i to x_{i+1} and divide by h_i; after rearrangement this gives

$$[N_i]\, h_i[\overline{\Phi}_i] = \left[I - e^{-[N_i]h_i}\right][\Phi(x_i)] \qquad (3.5)$$

where

$$[\overline{\Phi}_i] = \frac{1}{h_i}\int_{x_i}^{x_{i+1}} [\Phi(x)]\, dx.$$

Similarly, we can integrate Eq. (3.4) in the negative direction over homogeneous region R_{i-1} to obtain

$$-[N_{i-1}]\, h_{i-1}[\overline{\Phi}_{i-1}] = \left[I - e^{[N_{i-1}]h_{i-1}}\right][\Phi(x_i)]. \qquad (3.6)$$

We recall the trigonometric identities

$$\begin{aligned} 1 - e^{-x} &= 1 - \cosh x + \sinh x \\ 1 - e^{x} &= 1 - \cosh x - \sinh x. \end{aligned} \qquad (3.7)$$

Substituting Eq. (3.7) into Eq. (3.5) and Eq. (3.6), adding the resulting equations together, and rearranging gives:

$$\begin{aligned} &(\sinh^{-1}[N_i]\, h_i)([I] - \cosh\, [N_i]\, h_i)[\Phi(x_i)] + [\Phi(x_i)] \\ &\quad = (\sinh^{-1}[N_i]\, h_i)[N_i]\, h_i[\overline{\Phi}_i] \\ &(\sinh^{-1}[N_{i-1}]\, h_{i-1})([I] - \cosh\, [N_{i-1}]\, h_{i-1})[\Phi(x_i)] - [\Phi(x_i)] \\ &\quad = -(\sinh^{-1}[N_{i-1}]\, h_{i-1})[N_{i-1}]\, h_{i-1}[\overline{\Phi}_{i-1}]. \end{aligned} \qquad (3.8)$$

We now use the additional trigonometric identity

$$(\sinh^{-1} x)(1 - \cosh x) = -\tanh \frac{x}{2}$$

and add Eqs. (3.8) together to obtain:

$$\begin{aligned} -(\tanh\,[N_i]\,h_i/2 + \tanh\,[N_{i-1}]\,h_{i-1}/2)[\Phi(x_i)] \\ = (\sinh^{-1}[N_i]\,h_i)[N_i]\,h_i[\overline{\Phi}_i] \\ - (\sinh^{-1}[N_{i-1}]\,h_{i-1})[N_{i-1}]\,h_{i-1}[\overline{\Phi}_{i-1}]. \end{aligned} \qquad (3.9)$$

We recall that for two energy groups, Eq. (3.9) represents four equations. We define

$$\begin{aligned} [A^i] &= (\tanh\,[N_i]\,h_i/2) \\ [B^i] &= (\sinh^{-1}[N_i]\,h_i)[N_i]\,h_i \end{aligned} \qquad (3.10)$$

and block the matrices $[A^i]$ and $[B^i]$ into their four (2 X 2) elements, as $[A^i_{k,k'}]$ and $[B^i_{k,k'}]$. Then Eq. (3.9) becomes:

$$-\begin{bmatrix} [0] & [A^{i-1}_{1,2} + A^i_{1,2}] \\ [A^{i-1}_{2,1} + A^i_{2,1}] & [0] \end{bmatrix} \begin{bmatrix} [\phi(x_i)] \\ [J(x_i)] \end{bmatrix}$$

$$= \begin{bmatrix} [B^i_{1,1}] & [0] \\ [0] & [B^i_{2,2}] \end{bmatrix} \begin{bmatrix} [\overline{\phi}_i] \\ [\overline{J}_i] \end{bmatrix}$$

$$- \begin{bmatrix} [B^{i-1}_{1,1}] & [0] \\ [0] & [B^{i-1}_{2,2}] \end{bmatrix} \begin{bmatrix} [\overline{\phi}_{i-1}] \\ [\overline{J}_{i-1}] \end{bmatrix}. \qquad (3.11)$$

Writing only the top equation of Eq. (3.11) and rearranging, we have

$$[J(x_i)] = -\left[A^{i-1}_{1,2} + A^{i}_{1,2}\right]^{-1}\left[B^{i}_{1,1}\right][\bar{\phi}_i]$$

$$+\left[A^{i-1}_{1,2} + A^{i}_{1,2}\right]^{-1}\left[B^{i-1}_{1,1}\right][\bar{\phi}_{i-1}]. \qquad (3.12a)$$

Equation (3.12a) is a matrix equation relating the current values at point x_i to the average fluxes in the adjacent regions. Using Eq. (3.12a) and its counterpart at point x_{i+1}

$$[J(x_{i+1})] = -\left[A^{i}_{1,2} + A^{i+1}_{1,2}\right]^{-1}\left[B^{i+1}_{1,1}\right][\bar{\phi}_{i+1}]$$

$$+\left[A^{i}_{1,2} + A^{i+1}_{1,2}\right]^{-1}\left[B^{i}_{1,1}\right][\bar{\phi}_i] \qquad (3.12b)$$

and substituting them into Eq. (3.2), we have

$$[C^{i,i-1}][\bar{\phi}_{i-1}] + [C^{i,i}][\bar{\phi}_i] + [C^{i,i+1}][\bar{\phi}_{i+1}] + h_i[\Sigma_{Ti}][\bar{\phi}_i]$$

$$= \frac{1}{\lambda} h_i[\nu\Sigma_{fi}]^T[\bar{\phi}_i] \qquad (3.13)$$

where

$$[C^{i,i-1}] = -\left[A^{i-1}_{1,2} + A^{i}_{1,2}\right]^{-1}\left[B^{i-1}_{1,1}\right]$$

$$[C^{i,i}] = \left\{\left[A^{i-1}_{1,2} + A^{i}_{1,2}\right]^{-1} + \left[A^{i}_{1,2} + A^{i+1}_{1,2}\right]^{-1}\right\}\left[B^{i}_{1,1}\right]$$

$$[C^{i,i+1}] = -\left[A^{i}_{1,2} + A^{i+1}_{1,2}\right]^{-1}\left[B^{i+1}_{1,1}\right].$$

Equation (3.13) is a matrix equation which, when solved using conventional numerical methods, will give the exact values of the average fluxes for each group in each region i.

We have not discussed how the matrices $[A^i]$ and $[B^i]$ in Eq. (3.10) are calculated. This derivation, again based on an analytic solution in each region and insuring continuity of flux and current between each region, is performed in Appendix A.

From the definition of the matrix $[N(x)]$ in Eq. (3.3), we note that the eigenvalue λ is needed in order to calculate the matrix elements in Eq. (3.10). Thus, the solution of Eq. (3.13) must be iterative, since the matrices $[C^{i,j}]$ depend on λ. However, this dependence is not expected to be strong, and an iterative scheme where the matrices $[C^{i,j}]$ are recalculated during the static iterative process is expected to converge to the exact solution.

3.3 Relationship to Past Work

In the preceding section, an analytical solution to the diffusion equation was derived in order to obtain exact difference equations. In this sec tion, we shall show that in doing this, we have also derived the one-dimensional response matrix for a homogeneous region. By substituting this "response matrix" into the integrated diffusion equation, we obtain equations which involve only the average flux and are thus of lower order than the conventional response matrix equations.[1]

Let us define J^+_{gi} and J^-_{gi} as the partial currents in the +x and -x directions at x_i. The corresponding G-element column vectors of group partial currents are therefore $[J^+_i]$ and $[J^-_i]$. Then for region R_i in a vacuum, the transmission matrices $[T^\pm_i]$ are defined as

$$\left[T_i^+\right]\left[J_i^+\right] = \left[J_{i+1}^+\right]$$

$$\left[T_i^-\right]\left[J_i^-\right] = \left[J_{i-1}^-\right] \qquad (3.14)$$

and the reflection matrices $[R_i^\pm]$ as

$$[R_i^+][J_i^+] = [J_i^-]$$

$$[R_i^-][J_i^-] = [J_i^+]. \qquad (3.15)$$

For R_i imbedded in the reactor, we can then derive the following matrix equation:

$$\begin{bmatrix} [J_i^+] \\ \\ [J_i^-] \end{bmatrix} = [\mathcal{R}_i] \begin{bmatrix} [J_{i+1}^+] \\ \\ [J_{i+1}^-] \end{bmatrix} \qquad (3.16)$$

where

$$[\mathcal{R}_i] = \begin{bmatrix} [T_i^+]^{-1} & -[T_i^+]^{-1}\,[R_{i+1}^-] \\ \\ [R_i^+][T_i^+]^{-1} & -[R_i^+][T_i^+]^{-1}\,[R_{i+1}^-] + [T_{i+1}^-] \end{bmatrix}.$$

A P-1 expansion of the angular flux gives the following relationship:

$$\begin{bmatrix} [\phi_i] \\ \\ [J_i] \end{bmatrix} = \begin{bmatrix} 2[I] & 2[I] \\ \\ [I] & -[I] \end{bmatrix} \begin{bmatrix} [J_i^+] \\ \\ [J_i^-] \end{bmatrix}.$$

Eq. (3.16) is therefore equivalent to

$$\begin{bmatrix} [\phi_i] \\ \\ [J_i] \end{bmatrix} \quad \begin{bmatrix} 2[I] & 2[I] \\ \\ [I] & -[I] \end{bmatrix} [\mathcal{R}_i] \begin{bmatrix} \frac{1}{4}[I] & \frac{1}{2}[I] \\ \\ \frac{1}{4}[I] & -\frac{1}{2}[I] \end{bmatrix} \begin{bmatrix} [\phi_{i+1}] \\ \\ [J_{i+1}] \end{bmatrix} \qquad (3.17)$$

or

$$\begin{bmatrix} [\phi_i] \\ [J_i] \end{bmatrix} = [\mathcal{S}_i] \begin{bmatrix} [\phi_{i+1}] \\ [J_{i+1}] \end{bmatrix} .$$

By comparison with Eq. (3.4), we see that

$$[\mathcal{S}_i] = e^{[N_i](x_{i+1} - x_i)}$$

and therefore from Eq. (A.8) we have

$$[\mathcal{S}_i] = [E][G(-h)][E]^{-1}. \tag{3.18}$$

We have shown that the conventional response matrix $[R_i]$ for region i is related to the inverse of the expression shown in Eq. (A.8). Thus, the analytic technique used in Appendix A is, in effect, a calculation of an exact, one-dimensional response matrix. This method requires the use of the expressions $(\sinh^{-1}[N_i]h_i)[N_i]h_i$ and $\tanh[N_i]h_i/2$, which are calculated as shown above from the above expression for $\exp[N_i]h_i$. These expressions have a simpler form than that of $\exp[N_i]h_i$, in that several of the matrix elements cancel out (see Eq. (3.11)). This makes it simple to derive an analytic expression between the average fluxes and currents. The order of the equations to be solved is thereby reduced by a factor of two.

Several "nodal" methods current in the literature have shown promise as coarse mesh methods. One such method, called the Nodal Expansion Method (NEM), has been developed by Finnemann and Wagner.[9] This method begins with the integrated form of the diffusion equation, written in terms of the partial currents. A detailed flux distribution in one dimension is found by approximating the flux by a polynomial expansion, and using a weighted residual technique to determine the unknown coefficients. The unknowns used in the NEM are therefore the outgoing partial currents from each region; $[J_i^-]$ and $[J_{i+1}^+]$, and the average flux in the region, $[\phi_i]$.

Although the iteration strategy of the NEM is different from that of the method used here, the equations of the NEM can be written in the conventional response matrix form, Eq. (3.16). Thus, the NEM differs from the analytic method only in that the "response matrices" are calculated approximately by assuming a polynomial expansion for the flux. The NEM, however, requires three unknowns per region; while the analytic method used here requires only one.

It can thus be seen that the analytic method, as well as other "nodal" methods currently in the literature, have much similarity to response matrices.[23] In the next section, a discussion of the solution techniques used will be presented, as well as results for one-dimensional static problems.

3.4 One-Dimensional Calculations

This section will first discuss the details of implementating the method derived in Sec. 3.2 into a computer program for static one-dimensional diffusion rpoblems. Results obtained using this program, called 1DEX, will then be presented.

The complexity of the derivation in Appendix A demonstrates that it would be extremely difficult to extend this method beyond two groups.

For this reason, the remainder of this thesis will concentrate on problems which can be represented in one or two energy groups.

3.4.1 Boundary Conditions

1DEX is a two-group, one-dimensional static diffusion theory code. It allows two boundary condition options, symmetry and albedo. The form of the albedo used here is

$$\begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} J_1 \\ J_2 \end{bmatrix}. \tag{3.19}$$

This form is similar to that used by Kalambokas in his investigation of albedo boundary conditions.[24] Let us assume that an albedo boundary condition is to be applied at $x = x_{I+1}$. The presence of the albedo boundary will modify the coefficients $[C^{i,j}]$ in Eq. (3.13) as follows:

$$\begin{aligned} [C^{I,I-1}] &= -[A^{I-1}_{1,2} + A^{I}_{1,2}]^{-1}\,[B^{I-1}_{1,1}] \\ [C^{I,I}] &= \{[A^{I-1}_{1,2} + A^{I}_{1,2}]^{-1} + [[A^{I}_{1,2} + [\alpha]]^{-1}\}[B^{I}_{1,1}] \\ [C^{I,I+1}] &= [0]. \end{aligned} \tag{3.20}$$

We note from Eq. (3.19) that a zero flux boundary is imposed by setting $[\alpha]$ to the null matrix.

For a symmetry boundary condition, let us assume we have zero current at $x = x_o$. Then the coefficients $[C^{i,j}]$ in Eq. (3.13) become

$$[C^{1,0}] = [0]$$

$$[C^{1,1}] = [A^{1}_{1,2} + A^{2}_{1,2}]^{-1} [B^{1}_{1,1}] \qquad (3.21)$$

$$[C^{1,2}] = -[A^{1}_{1,2} + A^{2}_{1,2}][B^{2}_{1,1}].$$

3.4.2 Iteration Strategies in 1DEX

Equation (3.13) is an eigenvalue equation, the solution of which is the vector of nodal average neutron fluxes in each group. The matrices $[C^{i,j}]$ are, in general, full 2 X 2 matrices. Thus the 1DEX method is different from conventional finite differences, in which the matrices $[C^{i,j}]$ are diagonal. It was therefore convenient in 1DEX to solve for both energy groups simultaneously. Equation (3.13) is solved by a conventional power iteration where the right-hand side is replaced by a fission source vector. The resulting seven-stripe matrix can be inverted directly by an extension of the forward elimination, backward substitution method. Thus both group fluxes at all spatial mesh points are found simultaneously. This process is continued until the solution converges.

As discussed in Sec. 3.2, the eigenvalue λ is required to compute the matrices $[C^{i,j}]$. A nonlinear strategy was developed so that the matrices were determined three times during the static iteration process:

i) At the beginning of the problem (assuming $\lambda = 1$)

ii) When the degree of convergence of the problem became less than ten times the required convergence criteria

iii) After the problem reached the required convergence criteria; (then additional iterations were performed to reach that criteria again).

This strategy was found to be effective for all cases attempted.

3.4.3 Results Using 1DEX

Static results for three one-dimensional test cases will be presented here. The geometry and material constants for these test cases are given in Appendix B.

Test Case 3.1 is a two-region problem originally solved by Kang and Hansen.[7] Table 3.1 presents the inverse eigenvalues obtained by Kang and by the 1DEX code. The agreement between 1DEX, using a 20 cm mesh, and a fine mesh cubic finite element solution is excellent.

TABLE 3.1

Eigenvalues (λ^{-1}) for Test Case 3.1

Method	Mesh Spacing	Eigenvalue (λ^{-1})
Linear FEM	20 cm	.9757621
Cubic FEM	20 cm	.9789983
Cubic FEM	1 cm	.9795255
1DEX	20 cm	.9795257

Test Case 3.2 is the static portion of a one-dimensional time-dependent benchmark problem.[25] The geometry and material constants are given as Test Case 5.1. Results are presented in Table 3.2 using 1DEX for a 20 cm mesh, and finite differences using a 2 cm mesh. Since 1DEX is exact for any size mesh (as long as the regions are homogeneous), exact results with 1DEX could have been obtained using only three mesh regions;

TABLE 3.2

Results from Test Case 3.2

Quantity	RAUMZEIT*	1DEX
Eigenvalue	.9015507	.9015965
Power in Region 1** ($0 \leqslant x \leqslant 40$)	.2790	.27885
Power in Region 2** ($40 \leqslant x \leqslant 200$)	.4421	.44229

*RAUMZEIT (finite differences) using $\Delta x = 2$ cm.

**Powers normalized such that total power = 1.0.

TABLE 3.3

Results from Test Case 3.3

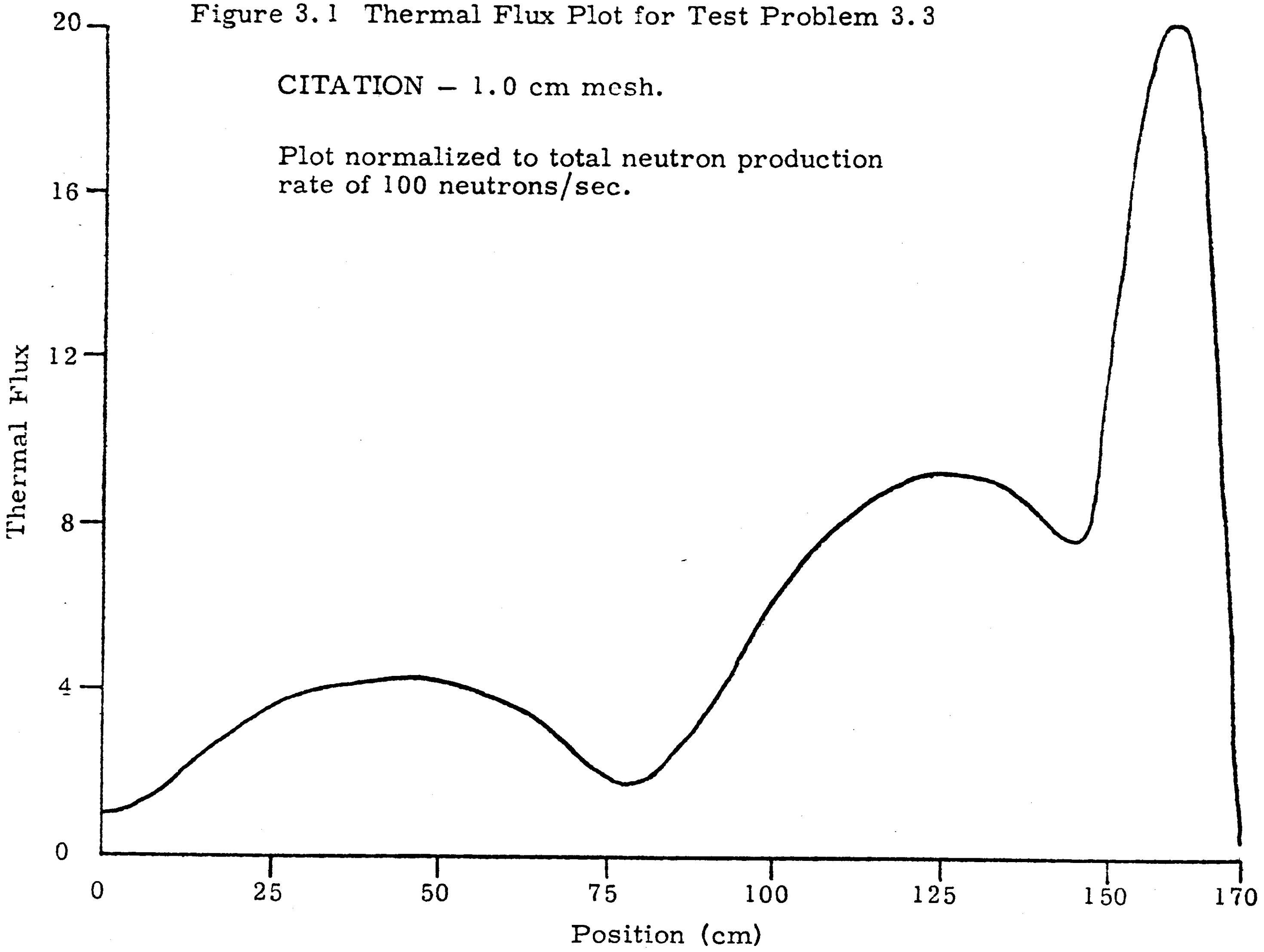
	CITATION		1DEX
	1 cm mesh, explicit reflector	.714 cm mesh, exact albedos	20 cm mesh
Eigenvalue	1.0044069	1.0045128	1.0045127
Region Powers			
Zone 1	.017473	.017295	.017249
2	.082829	.081929	.0816277
3	.110607	.109543	.109214
4	.094098	.093339	.0930729
5	.055290	.055237	.0552523
6	.168004	.168321	.1683945
7	.240006	.240929	.2412500
8	.231693	.233408	.2339394

spanning regions 1, 2, and 3. 1DEX runs using this mesh arrangement were made, and agree completely with the 20 cm results.

Test Case 3.3 is a one-dimensional slice through a model PWR core. The basic cross section sets are presented as Test Case 4.2 in Appendix B. This problem demonstrates the errors that can be obtained using finite difference methods where there are large flux peaks. Results are shown in Table 3.3 using the CITATION code with very small mesh spacings.[5] Figure 3.1 shows a plot of the thermal flux for the 1 cm mesh CITATION run. The large thermal flux peak in the reflector accounts for the inability of CITATION to give acceptable answers with 1 cm mesh spacings.

The test problems presented here verify that the 1DEX method produces accurate solutions using a coarse spatial mesh. The method shows enough promise to warrant extension to two dimensions. In Chapter 4 we shall examine the various methods of extending this scheme to two spatial dimensions.

Figure 3.1 Thermal Flux Plot for Test Problem 3.3



CITATION – 1.0 cm mcsh.

Plot normalized to total neutron production
rate of 100 neutrons/sec.

Chapter 4

SOLUTION METHODS FOR STATIC PROBLEMS IN TWO DIMENSIONS

4.1 Introduction

In Chapter 3 a method was derived which was based on an analytical solution of the diffusion equation. This derivation led to three-point difference relationships which are exact in one dimension. In two dimensions, it is not clear that exact difference equations exist, or what their matrix structure would be if they did exist. Nevertheless, Chapter 2 demonstrated the computational advantages of retaining a nearest-neighbor coupling relationship in two dimensions. Therefore, the exact method derived in Chapter 3 will be extended to two dimensions, with a nearest-neighbor coupling relationship retained.

4.2 Derivation in Two Dimensions

Let us begin in two-dimensional x-y geometry, with a region R defined as:

$$R = [0, X] \times [0, Y]$$

and with this region divided into a partition

$$\pi: \quad 0 = x_1 < \dots < x_I = X$$

$$0 = y_1 < \dots < y_J = Y.$$

We assume that any rectangle defined by the above partition is nuclearly homogeneous. The first step is to integrate Eq. (2.1) over (x_i, x_{i+1}) and (y_j, y_{j+1}) to obtain:

$$h_j([J_{x_{i+1,j}}] - [J_{x_{i,j}}]) + h_i([J_{y_{i,j+1}}] - [J_{y_{i,j}}])$$

$$+ h_i h_j [\Sigma_{Ti,j}][\bar{\phi}_{i,j}] = \frac{1}{\lambda} h_i h_j [\chi][\nu\Sigma_{fi,j}][\bar{\phi}_{i,j}] \tag{4.1}$$

where

$$h_i = x_{i+1} - x_i$$

$$h_j = y_{j+1} - y_j$$

$$[\bar{\phi}_{i,j}] = \frac{1}{h_i h_j} \int_{y_j}^{y_{j+1}} \int_{x_i}^{x_{i+1}} [\phi(x,y)]\, dxdy$$

$$[\Sigma_{Ti,j}] = [\Sigma_T(x,y)] \Big|_{\substack{x \subset (x_i, x_{i+1}) \\ y \subset (y_j, y_{j+1})}}$$

$$[J_{x_{i,j}}] = -\frac{1}{h_j} [D_{i,j}] \frac{\partial}{\partial x} \int_{y_j}^{y_{j+1}} [\phi(x_i, y)]\, dy$$

$$[J_{y_{i,j}}] = -\frac{1}{h_i} [D_{i,j}] \frac{\partial}{\partial y} \int_{x_i}^{x_{i+1}} [\phi(x, y_j)]\, dx.$$

The remaining step is to find relationships between the net currents $[J_{x_{i,j}}]$ and $[J_{y_{i,j}}]$, and the average fluxes $[\bar{\phi}_{i,j}]$. This will be accomplished by using the analytical procedure derived in Chapter 3.

Let us illustrate this procedure by finding a relationship between the x directed net currents and the adjacent average fluxes. To obtain the differential equation which must be solved analytically, we integrate Eq. (2.1) over (y_j, y_{j+1}) and divide by h_j. For $x \subset (x_i, x_{i+1})$ we obtain:

$$-[D_{i,j}]\frac{\partial^2}{\partial x^2}[\phi_j(x)] - [D_{i,j}]\int_{y_j}^{y_{j+1}} \frac{1}{h_j}\frac{\partial^2}{\partial y^2}[\phi(x,y)]\,dy$$

$$+ [\Sigma_{Ti,j}][\phi_j(x)] = \frac{1}{\lambda}[\chi][\nu\Sigma_{fi,j}]^T[\phi_j(x)] \qquad (4.2)$$

where

$$[\phi_j(x)] = \frac{1}{h_j}\int_{y_j}^{y_{j+1}} [\phi(x,y)]\,dy$$

$$[\Sigma_{Ti,j}] = [\Sigma_T(x,y)]\Big|_{\substack{x \subset (x_i, x_{i+1}) \\ y \subset (y_j, y_{j+1})}} .$$

Equation (4.2) has the same form as Eq. (3.1), with the exception of the additional integral representing leakage in the y direction. In order to solve Eq. (4.2) analytically, this integral must be approximated. Two possible approximations are:

i) Assume the leakage in the y direction is proportional to the flux in the x direction

ii) Assume the leakage in the y direction has a low-order polynomial representation.

We shall discuss both of the above approximations, and derive the appropriate difference equations.

4.2.1 The Buckling Approximation

Let us define the diagonal matrix $[B_{y_j}^{\,2}(x)]$ as follows:

$$[B_{y_j}^{\,2}(x)][\phi_j(x)] \equiv -[D_{i,j}]\int_{y_j}^{y_{j+1}} \frac{1}{h_i}\frac{\partial^2}{\partial y^2}[\phi(x,y)]\,dy. \qquad (4.3)$$

Although some function $[B_{y_j^2}(x)]$ exists such that Eq. (4.3) is exact, it is not obvious how to find this function in the general case.

However, one situation where the function $[B_{y_j^2}(x)]$ is easy to find is when the solution $[\phi(x,y)]$ is separable in x and y. For this case, we can replace $[B_{y_j^2}(x)]$ by a matrix $[B_{y_j^2}]$ which is constant over (x_i, x_{i+1}). Equation (4.3) then becomes:

$$[B_{y_j^2}][\phi_j(x)] = -[D_{i,j}] \int_{y_j}^{y_{j+1}} \frac{1}{h_i} \frac{\partial^2}{\partial y^2} [\phi(x,y)]\, dy. \tag{4.4}$$

The diagonal entries of $[B_{y_j^2}]$ are then related to the conventional "buckling" values. Substituting Eq. (4.4) into Eq. (4.2), we obtain:

$$-[D_{i,j}] \frac{\partial^2}{\partial x^2} [\phi_j(x)] + ([\Sigma_{Ti,j}] + [B_{y_j^2}])[\phi_j(x)] = \frac{1}{\lambda} [\chi][\nu\Sigma_{fi,j}][\phi_j(x)]. \tag{4.5}$$

Equation (4.5) can be solved analytically as in Chapter 3 to obtain a three-point difference relationship of the form (see Eqs. (3.12) and (3.13)):

$$[J_{x_{i+1,j}}] - [J_{x_{i,j}}] = [C_{xj}^{i,i-1}][\bar{\phi}_{i-1,j}] + [C_{xj}^{i,i}][\bar{\phi}_{i,j}]$$
$$+ [C_{xj}^{i,i+1}][\bar{\phi}_{i+1,j}]. \tag{4.6}$$

Equation (4.6) is substituted into Eq. (4.1) to eliminate the net currents in the x direction.

The y directed currents are eliminated in an analogous manner by solving analytically the diffusion equation integrated over (x_i, x_{i+1}). From this analytic solution we obtain the relationship:

$$[J_{y_{i,j+1}}] - [J_{y_{i,j}}] = [C_{yi}^{j,j-1}][\bar{\phi}_{i,j-1}] + [C_{yi}^{j,j}][\bar{\phi}_{i,j}] + [C_{yi}^{j,j+1}][\bar{\phi}_{i,j+1}]. \qquad (4.7)$$

To obtain difference equations, Eqs. (4.6) and (4.7) are substituted into Eq. (4.1). We thus have

$$h_i[C_{yi}^{j,j-1}][\bar{\phi}_{i,j-1}] + h_j[C_{xj}^{i,i-1}][\bar{\phi}_{i-1,j}] + h_j[C_{xj}^{i,i+1}][\bar{\phi}_{i+1,j}] + h_i[C_{yi}^{j,j+1}][\bar{\phi}_{i,j+1}] + (h_j[C_{xj}^{i,i}] + h_i[C_{yi}^{j,j}] + h_i h_j[\Sigma_{Ti,j}])[\bar{\phi}_{i,j}] = \frac{1}{\lambda} h_i h_j[\chi][\nu\Sigma_{fi,j}]^T[\bar{\phi}_{i,j}]. \qquad (4.8)$$

The calculation of the coefficient matrices $[C_{xj}^{i,i'}]$ and $[C_{yi}^{j,j'}]$ is made more complex since they are dependent on the matrices $[B_{y_j^2}]$ and $[B_{x_i^2}]$. The matrix $[B_{y_j^2}]$ is calculated by integrating Eq. (4.4) over (x_i, x_{i+1}) and dividing by h_i:

$$[B_{y_j^2}][\bar{\phi}_{i,j}] = \frac{1}{h_j}([J_{y_{i,j+1}}] - [J_{y_{i,j}}]) \qquad (4.9)$$

where

$$[J_{y_{i,j}}] = -[D_{i,j}]\frac{\partial}{\partial y}[\phi_i(y)]\Big|_{y=y_j}$$

$$[\phi_i(y)] = \frac{1}{h_i}\int_{x_i}^{x_{i+1}}[\phi(x,y)]\,dx.$$

Equation (4.6) is then substituted into Eq. (4.9) to obtain:

$$[B_{y_j^2}][\bar{\phi}_{i,j}] = \frac{1}{h_j}\left([C_{xj}^{i,i-1}][\bar{\phi}_{i-1,j}] + [C_{xj}^{i,i}][\bar{\phi}_{i,j}] + [C_{xj}^{i,i+1}][\bar{\phi}_{i+1,j}]\right). \tag{4.10}$$

The matrix $[B_{x_i^2}]$ is found in an analogous manner.

Since the coefficient matrices are dependent on the neutron fluxes, Eq. (4.8) is nonlinear. A fission source iteration with Chebyshev acceleration[18] was used to find the eigenvalue and eigenvector of Eq. (4.8). These iterations are called the "outer" iterations. At the beginning of the problem, the coefficient matrices were computed assuming all the bucklings to be zero. At various times during the static iterative process (generally every eight or ten fission source iterations), the coefficient matrices were recalculated using the bucklings found from the most recent flux iterate (using Eq. (4.10)). This process was repeated until convergence.

At each outer iteration, a matrix equation of the form

$$[P][\bar{\phi}] = [S] \tag{4.11}$$

must be solved. The matrix $[P]$ is block five-striped, where each entry is a 2 × 2 matrix. Equation (4.11) has been solved by using the Cyclic Chebyshev polynomial method of iteration.[17] These iterations are called the "inner" iterations.

4.2.2 Flat Leakage Approximation

Realistic reactor problems are not separable, and methods such as the "buckling" approximation which assume separability can be subject to significant errors.[26] Another approach would be to assume a low-order polynomial approximation to the leakage integral. Investigations reported in Refs. 26 and 27 show that excellent results can be obtained using large mesh sizes with a quadratic representation of the leakage integral.

If the leakage integral in Eq. (4.2) were replaced by a quadratic polynomial, the analytic solution would be extremely complicated; being composed of the solution to the homogeneous equation plus the particular solutions. It appears that a quadratic expansion is not practical using the analytic method. However, work on nodal methods by Sims^{28} indicates that accurate results can be obtained using a flat representation of the leakage. Therefore, we shall make this assumption.

Let us make the following approximation:

$$-\frac{1}{h_j}[L_{y_{i,j}}] = [D_{i,j}]\int_{y_j}^{y_{j+1}} \frac{1}{h_j}\frac{\partial^2}{\partial y^2}[\phi(x,y)]\,dy. \tag{4.12}$$

We integrate Eq. (4.12) over (x_i, x_{i+1}) and divide by h_i to obtain:

$$[L_{y_{i,j}}] = ([J_{y_{i,j+1}}] - [J_{y_{i,j}}]). \tag{4.13}$$

Substituting Eq. (4.12) into Eq. (4.2), we obtain

$$-[D_{i,j}]\frac{\partial^2}{\partial x^2}[\phi_j(x)] + [\Sigma_{Ti,j}][\phi_j(x)]$$

$$-\frac{1}{\lambda}[\chi][\nu\Sigma_{fi,j}]^T[\phi_j(x)] = -\frac{1}{h_j}[L_{y_{i,j}}]. \tag{4.14}$$

Equation (4.14) must be solved analytically to obtain coupling relationships between the currents and the average fluxes.

Using the notation of Eq. (3.3), we have

$$\frac{d}{dx}[\Phi(x)] + [N(x)][\Phi(x)] = [L] \tag{4.15}$$

where

$$[L] = \text{col}\left\{[0], -\frac{1}{h_j}[L_{y_{i,j}}]\right\}.$$

The analytical solution of Eq. (4.15) over (x_i, x_{i+1}) is

$$[\Phi(x)] = e^{-[N_i](x-x_i)}[\Phi(x_i)] + [N_i]^{-1}\left([I] - e^{-[N_i](x-x_i)}\right)[L]. \tag{4.16}$$

To relate the average fluxes to the net currents, we integrate Eq. (4.16) over (x_i, x_{i+1}) and divide by h_i to obtain:

$$[N_i]\, h_i[\bar{\Phi}_i] = \left([I] - e^{-[N_i]h_i}\right)[\Phi(x_i)]$$

$$+ \left(h_i[I] - [N_i]^{-1}\left([I] - e^{-[N_i]h_i}\right)\right)[L_i].$$

Similarly, for the region (x_{i-1}, x_i) we obtain:

$$-[N_{i-1}]\, h_{i-1}[\bar{\Phi}_{i-1}] = \left([I] - e^{[N_{i-1}]h_{i-1}}\right)[\Phi(x_i)]$$

$$- \left(h_i[I] + [N_{i-1}]^{-1}\left([I] - e^{[N_{i-1}]h_{i-1}}\right)\right)[L_{i-1}].$$

Using the trigonometric identities Eq. (3.7), adding the equations together, and then recognizing that

$$(\sinh^{-1} x)(1 - \cosh x) = -\tanh x/2,$$

we obtain:

$$
\begin{aligned}
&(\sinh^{-1}[N_i] h_i)[N_i]\, h_i[\bar{\Phi}_i] - (\sinh^{-1}[N_{i-1}] h_{i-1})[N_{i-1}]\, h_{i-1}[\bar{\Phi}_{i-1}] \\
&\quad - (\sinh^{-1}[N_i] h_i)\left(h_i[I] - [N_i]^{-1}\left([I] - e^{-[N_i] h_i}\right)\right)[L_i] \\
&\quad + (\sinh^{-1}[N_{i-1}] h_{i-1})\left(h_{i-1}[I] + [N_{i-1}]^{-1}\left([I] - e^{[N_{i-1}] h_{i-1}}\right)\right)[L_{i-1}] \\
&= -(\tanh [N_i]\, h_i/2 + \tanh [N_{i-1}]\, h_{i-1}/2)[\Phi(x_i)]. \qquad (4.17)
\end{aligned}
$$

As in Eq. (3.10), let us make the following definitions:

$$
\begin{aligned}
&[A^i] = (\tanh [N_i]\, h_i/2) \\
&[B^i] = (\sinh^{-1}[N_i] h_i)[N_i]\, h_i \qquad (4.18) \\
&[D^i] = -(\sinh^{-1}[N_i] h_i)\left(h_i[I] - [N_i]^{-1}\left([I] - e^{-[N_i] h_i}\right)\right).
\end{aligned}
$$

The above matrices are defined in Appendix A for both one- and two-group cases.

Let us partition each of the matrices in Eq. (4.18) into four blocks, each block being either a single element (for one group), or a 2 X 2 matrix (for two groups). Then writing only the top equation in Eq. (4.17) and rearranging, we obtain:

$$
\begin{aligned}
[J_{x_{i,j}}] &= -[A^{i-1}_{1,2} + A^i_{1,2}]^{-1}\,[B^i_{1,1}][\bar{\phi}_{i,j}] + [A^{i-1}_{1,2} + A^i_{1,2}]^{-1}\,[B^{i-1}_{1,1}][\bar{\phi}_{i-1,j}] \\
&\quad + \frac{1}{h_j}\,[A^{i-1}_{1,2} + A^i_{1,2}]^{-1}\,[D^i_{1,2}][L_{y_{i,j}}] - \frac{1}{h_j}\,[A^{i-1}_{1,2} + A^i_{1,2}]^{-1}\,[D^{i-1}_{1,2}][L_{y_{i-1,j}}].
\end{aligned}
$$

(4.19)

A similar expression for $[J_{x_{i+1,j}}]$ can be combined with the above equation to give:

$$[L_{x_{i,j}}] = [J_{x_{i+1,j}}] - [J_{x_{i,j}}] = -[C_{xj}^{i,i-1}][\bar{\phi}_{i-1,j}]$$

$$+ [C_{xj}^{i,i}][\bar{\phi}_{i,j}] - [C_{xj}^{i,i+1}][\bar{\phi}_{i+1,j}]$$

$$- \frac{1}{h_j}[E_{xj}^{i,i-1}][L_{y_{i-1,j}}] + \frac{1}{h_j}[E_{xj}^{i,i}][L_{y_{i,j}}]$$

$$- \frac{1}{h_j}[E_{xj}^{i,i+1}][L_{y_{i+1,j}}] \qquad (4.20)$$

where

$$[C_{xj}^{i,i-1}] = [A_{1,2}^{i-1} + A_{1,2}^{i}]^{-1}[B_{1,1}^{i-1}]$$

$$[C_{xj}^{i,i}] = \left([A_{1,2}^{i-1}+A_{1,2}^{i}]^{-1} + [A_{1,2}^{i}+A_{1,2}^{i+1}]^{-1}\right)[B_{1,1}^{i}]$$

$$[C_{xj}^{i,i+1}] = [A_{1,2}^{i}+A_{1,2}^{i+1}]^{-1}[B_{1,1}^{i+1}]$$

$$[E_{xj}^{i,i-1}] = [A_{1,2}^{i-1}+A_{1,2}^{i}]^{-1}[D_{1,2}^{i-1}]$$

$$[E_{xj}^{i,i}] = \left([A_{1,2}^{i-1}+A_{1,2}^{i}]^{-1} + [A_{1,2}^{i}+A_{1,2}^{i+1}]^{-1}\right)[D_{1,2}^{i}]$$

$$[E_{xj}^{i,i+1}] = [A_{1,2}^{i}+A_{1,2}^{i+1}]^{-1}[D_{1,2}^{i+1}].$$

At this point Eq. (4.20) should be contrasted with Eq. (4.6). In the "buckling" approximation, it was possible to relate the x directed leakage $[L_{x_{i,j}}]$ to the average fluxes in the nodes (i-1, j), (i, j), and (i+1, j). However, for the "flat source" scheme derived here, the x directed leakages are found to be related not only to the average fluxes but also to the y directed leakages in those three adjacent nodes. Therefore, the leakages cannot be completely eliminated from Eq. (4.1) as was done with the "buckling" approximation.

To obtain a matrix structure suitable for efficient computation, it will be convenient to rearrange some of the relevant equations. Accordingly, we assume that I * J is the total number of mesh regions in the reactor, and that G is the number of neutron groups. We then let K = I * J * G. Further, let us always order the unknowns such that the energy groups are indexed first, then the x direction, and finally the y direction. Then we define:

$[\bar{\phi}]$ = a column vector of length K containing the nodal average fluxes

$[L_x]$ = a column vector of length K containing the x directed leakages

$[L_y]$ = a column vector of length K containing the y directed leakages

$[C_x]$ = a matrix of order K × K containing the elements $[C_{xj}^{i,i'}]$ as defined in Eq. (4.19)

$[C_y]$ = a matrix of order K × K containing the elements $[C_{yi}^{j,j'}]$, elements defined analogously to Eq. (4.19)

$[E_x]$ = a matrix of order K × K containing the elements $[E_{xj}^{i,i'}]$ as defined in Eq. (4.19)

$[E_y]$ = a matrix of order K × K containing the elements $[E_{yi}^{j,j'}]$ defined analogously to Eq. (4.19)

$[\Sigma_T]$ = a matrix or order K × K containing the elements $h_i h_j [\Sigma_{Ti,j}]$ as in Eq. (4.1)

[M] = a matrix of order KXK containing the elements $h_i h_j[\chi][\nu\Sigma_{fi,j}]^T$ as in Eq. (4.1).

Then Eq. (4.1) can be rewritten

$$h_j[L_x] + h_i[L_y] + [\Sigma_T][\phi] = \frac{1}{\lambda}[M][\bar{\phi}]. \tag{4.21}$$

Equation (4.20) (and its counterpart in the y direction) can be written

$$[L_x] = [C_x][\bar{\phi}] + \frac{1}{h_j}[E_x][L_y] \tag{4.22}$$

$$[L_y] = [C_y][\bar{\phi}] + \frac{1}{h_i}[E_y][L_x]. \tag{4.23}$$

Equations (4.21, (4.22), and (4.23) comprise a linear set of three equations in the three unknowns. The objective of the analytic method was to develop schemes which involved only one unknown (the average flux) per mesh region and group. However, this seems impossible for the present case. Accordingly we attempt to shift the computational burden of the simultaneous solution of the above equations in such a way that the equation for the average fluxes will be "difficult" to solve (in the sense that iterative methods will be required), while the equations for the leakages will be relatively easy to solve. To accomplish this, Eqs. (4.22) and (4.23) were substituted into Eq. (4.21), to obtain the equations:

$$(h_j[C_x] + h_i[C_y] + [\Sigma_T])[\bar{\phi}] + [E_y][L_x] + [E_x][L_y] = \frac{1}{\lambda}[M][\bar{\phi}] \tag{4.24}$$

$$[L_x] = [C_x][\bar{\phi}] + \frac{1}{h_j}[E_x][L_y] \tag{4.22}$$

$$[L_y] = [C_y][\bar{\phi}] + \frac{1}{h_i}[E_y][L_y]. \tag{4.23}$$

Equations (4.24), (4.22), and (4.23) can be written in a more compact form:

$$[F][\psi] = \frac{1}{\lambda}[H][\psi] \tag{4.25}$$

where

$$[F] = \begin{bmatrix} (h_j[C_x] + h_i[C_y] + [\Sigma_T]) & [E_y] & [E_x] \\ [C_x] & -[I] & \frac{1}{h_j}[E_x] \\ [C_y] & \frac{1}{h_i}[E_y] & -[I] \end{bmatrix}$$

$$[H] = \begin{bmatrix} [M] & [0] & [0] \\ [0] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix}$$

$$\psi = \text{col}\left\{[\phi], [L_x], [L_y]\right\}.$$

Solving Eq. (4.25) requires care. The matrix [F] is not nonlinear; however, its elements depend on the eigenvalue λ. Therefore, an updating scheme identical to the one described in Sec. 3.4.2 is used in conjunction with a fission source iteration with Chebyshev acceleration as in Sec. 4.2.1. At each outer iteration, the matrix [F] is inverted by applying a Gauss-Seidel iteration to Eqs. (4.24), (4.22), and (4.23). The solution of Eq. (4.24) is obtained by using the Cyclic Chebyshev polynomial method of iteration[17] as described in Sec. 4.2.1.

4.3 Numerical Properties of the Equations

4.3.1 Nonlinear Iterations

Evaluation of the matrices [F] (in Eq. (4.25)) and [C] (in Eq. (4.8)) requires that the eigenvalue λ be known. An iterative updating procedure has been proposed in both the "buckling" and flat source methods to find a converged solution. The convergence properties of this iterative scheme are difficult to guarantee. However, in Sec. 2.4 a somewhat similar scheme was used to converge the fluxes and eigenvalue while simultaneously "searching" for the critical eigenvalue k_{eff}. The success of this method suggests that the matrix properties may not be strongly dependent on the eigenvalue λ. Therefore we suspect that setting $\lambda = 1$ will provide a good starting guess for most situations.

The matrix [C] in Eq. (4.8) is dependent not only on the eigenvalue λ, but also on the bucklings calculated from Eq. (4.10). Thus Eq. (4.8) is nonlinear. For the two-group case, the iteration necessitated by the inclusion of the bucklings failed to converge for Antonopoulos.[22] This was because the inclusion of the buckling terms destroyed the diagonal dominance of Antonopoulos' coefficient matrix. However, the diagonal dominance will not be significantly affected in the analytical method derived in Sec. 4.2.1. We are therefore hopeful that the iterations will converge, although it is difficult to guarantee such behavior.

4.3.2 Fission Source Iterations

The acceleration of the fission source iterations by the use of Chebyshev polynomials also requires certain assumptions about the iteration matrix.[18] Specifically, we assume all the eigenvalues are real and non-negative, and that the eigenvectors form a basis for the associated vector space. Under these conditions, the Chebyshev polynomials can be guaranteed to provide a specific error reduction. The above properties are difficult to show for the case of finite differences, and are also difficult to guarantee for the analytical method derived here. However, Chebyshev polynomials have been successfully applied to the outer iterations for both finite difference[18] and finite element[12] methods; thus it seems reasonable also to apply them to the analytic method.

4.3.3 The Inner Iterations

We shall examine the matrix equation which must be solved at each outer iteration to yield the nodal average fluxes.

Rewriting Eq. (4.24) as

$$(h_j[C_x] + h_i[C_y] + [\Sigma_T])[\bar{\phi}] = [S]$$

or (4.26)

$$[R][\bar{\phi}] = [S],$$

let us examine the numerical properties of the matrix [R].

i) We observe that even in one-group theory [R] is not symmetric. The coupling coefficients between mesh regions i and i+1 from Eq. (4.19) are

$$[C_{xj}^{i,i+1}] = [A_{1,2}^{i} + A_{1,2}^{i+1}]^{-1} [B_{1,1}^{i+1}]$$

$$[C_{xj}^{i+1,i}] = [A_{1,2}^{i} + A_{1,2}^{i+1}]^{-1} [B_{1,1}^{i}].$$

We notice that these coefficients would be equivalent only if $[B^{i}_{1,1}] = [B^{i+1}_{1,1}]$. Thus the matrix [R] is symmetric only in the homogeneous case.

ii) We also observe that even in one-group theory [R] cannot in general be guaranteed to be diagonally dominant. Referring to Eq. (4.19), we note that for the two-dimensional one-group case:

$$\text{Diagonal Sum} = h_j[C^{i,i}_{xj}] + h_i[C^{j,j}_{yi}] + h_i h_j[\Sigma_{Ti,j}]$$

$$\text{Off-Diagonal Sum} = h_j[C^{i,i-1}_{xj}] + h_j[C^{i,i+1}_{xj}] + h_i[C^{j,j-1}_{yi}] + h_i[C^{j,j+1}_{yi}].$$

If the problem is homogeneous, then

$$(\text{Diagonal Sum}) - (\text{Off-Diagonal Sum}) = h_i h_j[\Sigma_{Ti,j}]$$

and the matrix is diagonally dominant. However, for a heterogeneous problem, this property cannot be guaranteed. Nevertheless, for materials whose properties are not too dissimilar, and for large mesh spacings, the presence of the $h_i h_j[\Sigma_{Ti,j}]$ term on the diagonal will probably insure the diagonal dominance of the matrix.

In two-group theory, the situation is made more complex since the $[\Sigma_{Ti,j}]$ matrix includes scattering from group to group. However, since the energy groups are solved simultaneously, it is doubtful that this will severely hurt the matrix properties.

A diagonally dominant matrix is convergent for both the point Jacobi and Gauss-Seidel iteration schemes.[17] In general, the larger the diagonal elements are compared to the off-diagonal elements, the lower the spectral radius of the associated iteration matrix.

In view of the properties of the [R] matrix described above, we feel it is reasonable to apply iterative techniques to the solution of Eq. (4.25). Results for two-dimensional test problems for the "buckling" and flat source methods will be presented in Section 4.4.

4.3.4 Convergence to the Exact Solution

We define convergence to mean that the difference between the discrete approximation $[\bar{\phi}_{ij}]$ and the exact average fluxes can be made arbitrarily small for every mesh interval simply by choosing the mesh spacings h_i and h_j sufficiently small. For finite difference methods, proofs of convergence generally center around the fact that the difference equations are a low-order Taylor's series expansion of the correct solution. In the finite element method, the proofs often appeal to concepts in approximation theory; where any continuous function can be reproduced exactly by a series of ramps (in the linear case, for example) in the limit of small h.

In one dimension, the method is exact. Therefore, the method is convergent if and only if the approximation used for the transverse leakage integral becomes exact in the limit of small mesh sizes. The "buckling" approximation uses small sections of the function $[\phi_j(x)]$ to approximate the integral. The flat source method uses a staircase function to approximate the leakage integral. In the limit of samll mesh sizes, the leakage integral can be exactly represented using small sections of either of these functions. Therefore we expect both schemes to converge in the limit of small mesh spacings.

4.4 Results

4.4.1 Computer Programs Used

The program 2DEX was written to solve two-dimensional two-group static neutron diffusion problems using the "buckling" method derived in Sec. 4.2.1. The program 2DFS was written to solve the same problem using the flat source method described in Sec. 4.2.1. Both programs take advantage of the iterative strategies suggested in Sec. 4.2. Since the Cyclic Chebyshev method (used for the inner iterations) requires a knowledge of the spectral radius of the inner iteration matrix, both programs calculate this number before the outer iterations are begun. The coefficient matrices are calculated in subroutine MATRX. Both codes were written entirely in single precision.

4.4.2 Homogeneous Test Problem

Test Problem 4.1 is a homogeneous two-dimensional two-group test problem. The geometry and material constants are given in Appendix B. Table 4.1 summarizes the results of this test problem. Since this test problem is separable in x and y, the results from 2DEX are exact. In addition, both codes converge to a cosine-shaped eigenvector.

A very important aspect of the 2DFS method is the ease of solution of Eq. (4.25) at each outer iteration. This solution is performed by using a Gauss-Seidel iteration through the matrix [F]. Test cases with 2DFS were run which use one, two, and three Gauss-Seidel iterations

TABLE 4.1

Results of Test Problem 4.1

	2DEX	2DFS
Eigenvalue λ^*	1.03641	1.03652
Number of Outer Iterations	62	80
Number of Calls to MATRX	9	3
Number of Inner Iterations per Outer	3	3

*Exact eigenvalue: 1.03641.

through [F]. These runs show that one iteration per outer iteration is sufficient. Therefore our attempt to shift the difficult part of the solution of Eqs. (4.21), (4.22), and (4.23) to the flux equation (4.24) has succeeded. By substituting Eqs. (4.22) and (4.23) into (4.21), the fluxes have become sufficiently decoupled from the leakages such that one pass through the matrix [F] is sufficient.

We also observe that, as a result of the large mesh spacing (20 cm), very few "inner" iterations are required.

4.4.3 IAEA Benchmark Problem

The IAEA Benchmark Problem is a two- or three-dimensional two-group static problem, representative of a PWR.[29] Finite difference methods have been shown to be in error even when mesh sizes as small as one centimeter are used. The geometry and material constants for the two-dimensional problem are shown in Appendix B. Table 4.2

TABLE 4.2

Results of Test Problem 4.2

	2DEX	2DFS
Eigenvalue λ^*	1.03044	1.03001
Number of Outer Iterations	78	71
Number of Inner Iterations per Outer	4	2
Convergence Criteria (on pointwise flux)	10^{-5}	10^{-5}
Number of Calls to MATRX	8	3
Execution Time (sec)	2.3	2.6

*Reference eigenvalue: 1.02959.

Figure 4.1 Assembly Power Fractions and Power Fraction Errors for Test Problem 4.2

20 cm Results

xx	Reference Power Fraction
xx	2DEX Percent Error
xx	2DFS Percent Error

Convergence Criteria:
10^{-5} on pointwise flux.

All calculations done in quarter-core.

					.1322-1 10.8 .2		
				.1064-1 -.7 -1.4	.1548-1 6.2 2.8	.1351-1 8.9 1.9	
			.2694-1 -1.3 -.3	.2183-1 .5 .6	.2047-1 1.7 .7	.1914-1 4.7 .6	
		.3320-1 -3.3 -1.4	.3039-1 -2.1 -.7	.2664-1 .2 .9	.2420-1 -.1 -.2	.2206-1 1.1 -.1	.1564-1 3.5 -.4
	.3243-1 -2.7 -.6	.3342-1 -2.6 -.7	.2969-1 -2.1 -.5	.2418-1 -.9 .2	.2341-1 -.9 -.4	.2149-1 -.6 -.4	.1663-1 .3 .1
.1684-1 -5.2 -2.4	.2956-1 -2.0 .2	.3286-1 -1.3 .6	.2734-1 -1.5 .1	.1379-1 -4.1 -2.3	.2112-1 -.6 -.3	.2110-1 0 .5	.1706-1 -.1 .1

displays some of the results obtained for this test case. Figure 4.1 shows the assembly power fractions for the reference solution,[29] and the percent errors in assembly powers using 2DEX and 2DFS with a 20 cm mesh size. The 2DFS run required only one Gauss-Seidel iteration through Eq. (4.25) at each outer iteration.

This test problem shows the superiority in accuracy of the flat source method. Using a 20 cm mesh the 2DFS code increases the execution time by 20%. However, the eigenvalue error is cut in half and the worst assembly power error lowered from 10.8% to 2.8%. For a 10 cm mesh, 2DEX yields a maximum error of 2.3% in assembly power; however, the execution time is over seven seconds. Thus the 2DFS method appears to give the optimal combination of accuracy and execution time for this problem.

4.4.4 BWR Test Problem

The BWR test problem[30] is a two- or three-dimensional kinetics benchmark problem. We have investigated the two-dimensional steady state version of this problem. Geometric and material constants are shown in Appendix B as Test Problem 4.3. The problem was solved using 2DEX and 2DFS with a 15 cm mesh.

Table 4.3 summarizes the results of the runs. Figure 4.2 shows the reference assembly power fractions, as well as the percent error in assembly power fraction for 2DEX and 2DFS. The reference solution used here is from the 2DFS program with a 3.75 cm mesh. Comparison of this run with 5.0 cm and 7.5 cm runs shows that the results are essentially converged.

TABLE 4.3

Results of Test Problem 4.3

	2DEX	2DFS
Eigenvalue λ^*	.996863	.996933
Number of Outer Iterations	103	67
Number of Inner Iterations per Outer	4	3
Convergence Criteria	10^{-5}	10^{-5}
Number of Calls to MATRX	11	3
Execution Time (sec)	4.6	3.9

*Reference eigenvalue: .996361.

Figure 4.2 Assembly Power Fractions and Power Fraction Errors for Test Problem 4.3

15 cm Results

xx	Reference Power Fraction*
xx	2DEX Percent Error
xx	2DFS Percent Error

Convergence Criteria:
10^{-5} on pointwise flux.

All calculations done in quarter-core.

							.1660-1 5.9 2.7	
						.2773-1 2.1 2.8	.2089-1 2.5 2.3	.1080-1 5.1 1.5
					.2376-1 1.2 1.8	.2631-1 1.6 2.3	.2151-1 1.4 1.7	.1244-1 2.5 1.8
				.1109-1 0 .3	.1476-1 -.2 -.1	.1716-1 .3 .3	.1823-1 1.4 1.3	.1196-1 1.7 1.5
			.7089-2 -.5 0	.8698-2 -.7 -.3	.1081-1 -1.3 -1.3	.1310-1 -.7 -1.1	.1566-1 .5 -.1	.1094-1 1.1 .6
		.5442-2 -1.1 -.6	.6315-2 -.8 -.3	.7930-2 -1.2 -.8	.1003-1 -1.8 -1.8	.1239-1 -1.4 -1.8	.1504-1 -.3 -1.0	.1061-1 0 -.4
	.5128-2 -2.2 -1.8	.5218-2 -1.9 -1.6	.6292-2 -1.6 -1.2	.8606-2 -1.3 -.9	.1205-1 -1.7 -1.5	.1476-1 -1.7 -1.6	.1644-1 -1.0 -1.0	.1113-1 -1.0 -1.0
.7861-2 -2.0 -1.2	.5646-2 -2.9 -2.6	.5296-2 -3.1 -2.9	.6564-2 -3.4 -2.7	.1014-1 -2.0 -1.9	.1779-1 -.9 -.2	.2133-1 -1.0 -.1	.1900-1 -1.9 -1.4	.1185-1 -2.2 -1.9

*Reference is 2DFS with mesh size 3.75 cm.

The eigenvalue obtained using 2DFS for a 15 cm mesh is slightly less accurate than that obtained using 2DEX with the same mesh size. However, for this test problem the 2DEX code required 15% more computing time, and yields a maximum assembly power error of 5.9% as compared to 2.9% for 2DFS. For this test problem, the 2DFS method again appears to yield the optimal combination of accuracy and running time.

The 2DFS results were obtained using one Gauss-Seidel iteration through Eq. (4.25) per outer iteration. The strategy of using only one iteration per outer iteration has been successful for every static test problem attempted. It is therefore felt that this is a legitimate strategy for the general case.

4.4.5 Use of Albedos in Two Dimensions

The generalized albedo boundary condition described in Sec. 3.4.1 can be used in either 2DEX or 2DFS. The albedo required for either code is an average albedo over the edge of the mesh region to which it is being applied. Runs have been made using 2DFS for Test Problem 4.3 in an attempt to define appropriate albedo boundary conditions. It is apparent that more work is needed in this area. Albedos calculated using the one-dimensional formulas are considerably in error (assembly power errors in excess of 10%). We were unable to arrive at a simple prescription for calculating the albedos. However, if these albedos can be found, their implementation and use into either 2DEX or 2DFS is straightforward.

4.5 Summary

In this chapter, two possible methods of extending the method derived in Chapter 3 to two dimensions have been described. Results from three two-dimensional two-group test problems have been presented. For a homogeneous test problem, both methods gave exact results in space, while the 2DFS method had some eigenvalue error. For both realistic test problems, the 2DFS method was judged to provide a more effective combination of accuracy and running time. Therefore, the 2DFS method has been chosen to be extended to time-dependent problems. Since only one Gauss-Seidel iteration through the matrix $[F]$ of Eq. (4.25) is required per outer iteration, the running times of 2DFS compare well with 2DEX and with other methods of similar accuracy.

It is apparent that realistic reactor problems are not separable in space, and a method such as 2DEX which makes such an assumption can be subject to significant errors in assembly powers. Although the assumption of the transverse leakage being flat is very approximate, it appears to be more realistic than the assumption that the leakage is proportional to the flux. In Chapter 5 we will extend the 2DFS program to time-dependent problems.

Chapter 5

TIME-DEPENDENT ANALYSIS

5.1 One-Dimensional Time-Dependent Analysis

5.1.1 Development of the Equations

The time-dependent multigroup neutron diffusion equations are presented as Eq. (1.1). Let us rewrite these equations in the matrix form of Eq. (2.1):

$$\left[\frac{1}{v}\right] \frac{\partial}{\partial t}[\phi(x,t)] = \frac{\partial}{\partial x}[D(x,t)]\frac{\partial}{\partial x}[\phi(x,t)] - [\Sigma_T(x,t)][\phi(x,t)]$$

$$+ [\chi_p](1-\beta)[\nu\Sigma_f(x,t)]^T [\phi(x,t)] + \sum_{k=1}^{K} [\chi_{dk}]\lambda_k C_k(x,t) \tag{5.1}$$

$$\frac{\partial}{\partial t} C_k(x,t) = -\lambda_k C_k(x,t) + \beta_k[\nu\Sigma_f(x,t)]^T [\phi(x,t)] \qquad (1 \leq k \leq K) \tag{5.2}$$

where

$\left[\frac{1}{v}\right]$ is a diagonal G × G matrix containing the inverse neutron speeds

$[\chi_p]$ is a column vector of length G containing the prompt fission spectrum

$[\chi_{dk}]$ is a column vector of length G containing the neutron spectrum from delayed group k

$[\nu\Sigma_f(x,t)]$ is a column vector of length G containing the critical value of nu times the fission cross section.

All other terms are defined as in Eq. (2.1).

We now integrate Eqs. (5.1) and (5.2) over (x_i, x_{i+1}) to obtain:

$$h_i\left[\frac{1}{v}\right]\frac{\partial}{\partial t}[\bar{\phi}_i(t)] = -[J_{i+1}(t)] + [J_i(t)] - h_i[\Sigma_{Ti}(t)][\bar{\phi}_i(t)]$$

$$+ h_i(1-\beta)[\chi_p][\nu\Sigma_{fi}(t)]^T[\bar{\phi}_i(t)] + \sum_{k=1}^{K}[\chi_{dk}]\lambda_k\bar{C}_{ki}(t) \qquad (5.3)$$

$$\frac{\partial}{\partial t}\bar{C}_{ki}(t) = -\lambda_k\bar{C}_{ki}(t) + h_i\beta_k[\nu\Sigma_{fi}(t)]^T[\bar{\phi}_i(t)] \qquad (1 \leqslant k \leqslant K) \qquad (5.4)$$

where

$$[\bar{\phi}_i(t)] = \frac{1}{h_i}\int_{x_i}^{x_{i+1}}[\phi(x,t)]\,dx$$

$$[J_i(t)] = -[D(x,t)]\frac{\partial}{\partial x}[\phi(x,t)]\bigg|_{x=x_i}$$

$$[\Sigma_{Ti}(t)] = [\Sigma_T(x,t)]\bigg|_{x\subset R_i}$$

$$\bar{C}_{ki}(t) = \int_{x_i}^{x_{i+1}} C_k(x,t)\,dx$$

$$h_i = x_{i+1} - x_i.$$

To obtain a relationship between the net currents and the average fluxes, we must solve Eqs. (5.1) and (5.2) analytically over a two-region problem. The time derivative and delayed precursor terms complicate this solution, since they are not known in general as a function of x. To circumvent this difficulty, we make the following approximations.

For each region $\mathcal{R}_i \equiv (x_i, x_{i+1})$, we assume

$$\frac{\partial}{\partial t}[\phi(x,t)] = [\omega_{pi}][\phi(x,t)] \qquad (5.5a)$$

$$\frac{\partial}{\partial t} C_k(x,t) = \omega_{dki} C_k(x,t) \tag{5.5b}$$

where $[\omega_{pi}]$ is a diagonal G × G matrix. We insert Eq. (5.5) into Eqs. (5.1) and (5.2), then substitute Eq. (5.2) into Eq. (5.1) to obtain:

$$- \frac{\partial}{\partial x} [D(x,t)] \frac{\partial}{\partial x} [\phi(x,t)] + ([\Sigma_T(x,t)] + [\omega_{pi}]) [\phi(x,t)]$$

$$- \left\{ [\chi_p](1-\beta) + \sum_{k=1}^{K} [\chi_{dk}] \left(\frac{\lambda_k \beta_k}{\omega_{dki} + \lambda_k} \right) \right\} [\nu \Sigma_f(x,t)]^T [\phi(x,t)]. \tag{5.6}$$

If $[\omega_{pi}]$ and ω_{dki} are known for every region i over a time step interval, Eq. (5.6) can be solved analytically at a fixed time t as was done in Sec. 3.2. From this analytic solution, the following expression is obtained:

$$[J_{i+1}(t)] - [J_i(t)] = [C^{i,i-1}(t)][\bar{\phi}_{i-1}(t)] + [C^{i,i}(t)][\bar{\phi}_i(t)]$$

$$+ [C^{i,i+1}(t)][\bar{\phi}_{i+1}(t)] \tag{5.7}$$

where the terms $[C^{i,i'}(t)]$ are defined as in Eq. (3.13). Substituting Eq. (5.7) into Eq. (5.3) results in:

$$h_i \left[\frac{1}{v}\right] \frac{\partial}{\partial t} [\bar{\phi}_i(t)] = -[C^{i,i-1}(t)][\bar{\phi}_{i-1}(t)]$$

$$- ([C^{i,i}(t)] + h_i[\Sigma_{Ti}(t)]) [\bar{\phi}_i(t)] - [C^{i,i+1}(t)][\bar{\phi}_{i+1}(t)]$$

$$+ h_i(1-\beta)[\chi_p][\nu \Sigma_{fi}(t)]^T [\bar{\phi}_i(t)] + \sum_{k=1}^{K} [\chi_{dk}] \lambda_k \bar{C}_{ki}(t). \tag{5.8}$$

Equations (5.8) and (5.4) can be solved using any of the standard time integration methods.

The above derivation requires a knowledge of $[\omega_p]$ and ω_{dk} in Eq. (5.6). The $[\omega_p]$ terms are identical to the omegas which must be known at every time step for kinetics methods which use the so-called frequency transformation[31, 32] (although the methods referred to in Refs. 31 and 32 generally use the omega calculated from the thermal group as representative of all the groups). We therefore calculate the omegas as in the above references:

$$\begin{aligned} \omega_{pgi} &= \frac{1}{\Delta T} \ln \left(\bar{\phi}_{gi}^{n+1} / \bar{\phi}_{gi}^{n}\right) \\ \omega_{dki} &= \frac{1}{\Delta T} \ln \left(\bar{C}_{ki}^{n+1} / \bar{C}_{ki}^{n}\right) \end{aligned} \qquad (5.9)$$

where ΔT is the time step size, and n is the time step index.

We note that the approximations in Eq. (5.5) can be expressed as:

i) Each position-dependent quantity in a region has the same shape at the end of a time step as it did in the beginning of that time step.

ii) To calculate the coupling coefficients, this shape specified above grows (or decays) as a simple exponential over this time step.

The omegas evaluated above are used to calculate the matrices $[C^{i,i'}(t_n)]$ for every time step n. Since these matrices are somewhat costly to calculate, a time integration method should be chosen which allows a long time step ΔT to be taken without loss of accuracy. This criterion, of course, applies to any kinetics method. However, the considerable effort required to calculate the coefficient matrices makes it even more important here. Accordingly, a fully implicit time integration method[32] was chosen to solve Eqs. (5.8) and (5.4). The "exponential transformation" has been shown to significantly improve certain alternating-direction time integration methods.[31] However, when applied to the fully implicit method, the improvement is not major.[32] Therefore, we have not chosen to use the "exponential transformation" to speed the solution of these equations.

The accuracy of time integration methods such as the NSADE method[31] depends to a significant degree on the accuracy of the omegas. However, the omegas used in Eq. (5.6) affect only the calculation of the coupling coefficients, and as such have only a second-order effect on the transient solution. Therefore we anticipate that the solution of Eqs. (5.8) and (5.4) will be accurate even when large mesh regions are used.

5.1.2 Results of Test Problem 5.1

To examine further the method derived in Sec. 5.1.1, the computer program 1DEX was modified to perform time-dependent calculations. The new program, called 1DTD, solves the static and time-dependent one-dimensional, two-group diffusion equations. Up to six delayed neutron families are allowed.

Equations suitable for computation are obtained by first time differencing Eq. (5.4) to obtain an expression for $\overline{C}_{ki}^{n+1}$. This expression is then substituted into the time differenced form of Eq. (5.8). The calculational sequence is then:

i) The matrices $[C^{i,i'}]$ are calculated using the omegas from the previous time step.

ii) The new fluxes $[\overline{\phi}_i^{n+1}]$ are calculated. This calculation is performed using a matrix factorization technique.

iii) The new precursors $\overline{C}_{ki}^{n+1}$ are calculated.

iv) The new omegas are calculated from Eq. (5.9).

Test Problem 5.1 is the time-dependent version of Test Problem 3.2.[25] The geometric and material constants for this test problem are given in Appendix B. Table 5.1 shows the results of 1DTD runs using two different spatial mesh sizes. The reference solution is a finite difference solution using a 2 cm mesh. For 1DTD, a time step size of 10^{-2} sec is sufficiently small to insure temporal convergence. Table 5.2 shows the region powers versus time for the 1DTD run with a 20 cm mesh spacing, and with $\Delta T = 10^{-2}$ sec.

With the exception of the omega approximation, the 1DTD method provides exact spatial results. Since this approximation becomes more exact for smaller mesh spacings, the results in Table 5.1 indicate that 1DTD can provide highly accurate results using mesh sizes as large as 20 cm. The agreement of Runs A and B suggests that the finite difference results using a 2 cm mesh are not as accurate as the 20 cm 1DTD case.

5.2 Two-Dimensional Time-Dependent Analysis

5.2.1 Development of the Equations

The results presented in Sec. 4.4 demonstrated the superiority of the flat source method over the "buckling" method. In Section 5.1, approximations were made which allowed the one-dimensional method

TABLE 5.1

Total Power versus Time for Test Problem 5.1

Time	Reference	Run A	Run B
0.0	1.0	1.0	1.0
0.5	.7597	.7596	.7596
1.0	.6588	.6591	.6591
2.0	.6307	.6311	.6309

Reference: Finite Differences, $\Delta x = 2$ cm, $\Delta T = 10^{-3}$ sec.

Run A: IDTD, $\Delta T = 10^{-2}$ sec, $\Delta x = 2$ (20 cm), 4 (40 cm), 2 (20 cm).

Run B: IDTD, $\Delta T = 10^{-2}$ sec, $\Delta x = 20$ cm.

TABLE 5.2

Region Powers versus Time for Test Problem 5.1

			Region	
Time	Total Power	1	2	3
0.0	1.0	.27885	.44229	.27885
0.5	.7596	.14834	.34168	.26959
1.0	.6591	.09622	.29883	.26402
2.0	.6309	.08544	.28566	.25983

derived in Chapter 3 to be extended to two dimensions. With these approximations, the flat source method can be extended to time-dependent problems in a straightforward manner.

We shall first derive an expression relating the net leakage in the x direction to the adjacent average fluxes. From Eqs. (5.6) and (4.14), we see that the equation which is to be solved analytically can be written:

$$
\begin{aligned}
&-\frac{\partial}{\partial x}[D_{i,j}(t)]\frac{\partial}{\partial x}[\phi_j(x,t)] + ([\Sigma_{Ti,j}(t)]+[\omega_{pi,j}])[\phi_j(x,t)] \\
&\quad -\left\{[\chi_p](1-\beta) + \sum_{k=1}^{K}[\chi_{dk}]\left(\frac{\lambda_k\beta_k}{\omega_{dki,j}+\lambda_k}\right)\right\}[\nu\Sigma_{fi,j}(t)]^T[\phi_j(x,t)] \\
&= -\frac{1}{h_j}[L_{y_{i,j}}(t)]. \qquad (5.10)
\end{aligned}
$$

If the cross sections and the omegas are known, Eq. (5.10) can be solved analytically for a fixed time t as was done in Sec. 4.2.2. To insure initial criticality, ν has been adjusted by dividing it by the eigenvalue λ.

The analytic solution of Eq. (5.10) (and its counterpart in the y direction) leads to equations similar to Eqs. (4.22) and (4.23)

$$[L_x(t)] = [R_x(t)[\bar{\phi}(t)] + \frac{1}{h_j}[E_x(t)][L_y(t)] \qquad (4.22)$$

$$[L_y(t)] = [R_y(t)][\bar{\phi}(t)] + \frac{1}{h_i}[E_y(t)][L_x(t)] \qquad (4.23)$$

where the matrices $[C_x]$ and $[C_y]$ in Eqs. (4.22) and (4.23) have been renamed $[R_x(t)]$ and $[R_y(t)]$ to avoid confusion with the delayed neutron precursor concentrations. The unknowns in Eqs. (4.21), (4.22), and (4.23) are ordered such that the energy groups are indexed first, then the x direction, then the y direction. Let us write the spatially integrated, time-dependent diffusion equation in the form of Eq. (4.21)

$$\left[\frac{1}{v}\right] \frac{d}{dt}[\bar{\phi}(t)] = -h_j[L_x(t)] - h_i[L_y(t)] - [\Sigma_T(t)][\bar{\phi}(t)] + [M(t)][\bar{\phi}(t)] + \sum_{k=1}^{K} \lambda_k[\bar{C}_k(t)] \qquad (5.11)$$

where

$$[\bar{C}_k(t)] \equiv \mathrm{col}\left\{[\bar{C}_{k1,1}(t)], \ldots, [\bar{C}_{ki,j}(t)], \ldots, [\bar{C}_{kI,J}(t)]\right\}$$

and

$$[\bar{C}_{ki,j}] \equiv [\chi_{dk}]\, \bar{C}_{ki,j}$$

where

$[\chi_{dk}]$ is a column vector of length G containing the delayed neutron fission spectrum for family k

$\left[\frac{1}{v}\right]$ is a diagonal matrix containing the inverse neutron speeds for each group at each mesh point.

To insure criticality, the matrix [M] has been divided by the eigenvalue λ.

As in Sec. 4.2.2, Eqs. (4.22) and (4.23) are substituted into Eq. (5.11):

$$\left[\frac{1}{v}\right] \frac{d}{dt}[\bar{\phi}(t)] = (-h_j[R_x(t)] - h_i[R_y(t)] + [M(t)] - [\Sigma_T(t)])[\bar{\phi}(t)] - [E_y(t)][L_x(t)] - [E_x(t)][L_y(t)] + \sum_{k=1}^{K} \lambda_k[\bar{C}_k(t)]. \qquad (5.12)$$

The vector $[\psi(t)]$ is defined as

$$[\psi(t)] = \mathrm{col}\left\{[\overline{\phi}(t), [L_x(t)], [L_y(t)]\right\}$$

so that Eqs. (5.12), (4.22), and (4.23) can be written

$$[W]\frac{d}{dt}[\psi(t)] = [T(t)][\psi(t)] + \sum_{k=1}^{K} \lambda_k[Z_k(t)] \tag{5.13}$$

$$[W] = \begin{bmatrix} [1/v] & [0] & [0] \\ [0] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix}$$

$$[T(t)] = \begin{bmatrix} (-h_j[R_x(t)] - h_i[R_y(t)] - [\Sigma_T(t)] + [M]) & [E_y(t)] & [E_x(t)] \\ [R_x(t)] & -[I] & \frac{1}{h_j}[E_x(t)] \\ [R_y(t)] & \frac{1}{h_i}[E_y(t)] & -[I] \end{bmatrix}$$

$$[Z_k(t)] = \mathrm{col}\left\{[\overline{C}_k(t)], [0], [0]\right\}$$

$$\frac{d}{dt}[\overline{C}_k(t)] = -\lambda_k[\overline{C}_k(t)] + \beta_k[M(t)][\overline{\phi}(t)]. \tag{5.14}$$

If the omegas $[\omega_{pi,j}]$ and $\omega_{dki,j}$ are known, the matrices $[R_x(t)]$, $[R_y(t)]$, $[E_x(t)]$, and $[E_y(t)]$ can be calculated; and Eqs. (5.13) and (5.14) can be advanced over one time step. As in Sec. 5.1, a fully implicit time integration method was chosen to solve Eqs. (5.13) and (5.14). The "exponential transformation" was not used for this time integration.

We first time difference Eqs. (5.13) and (5.14) according to the fully implicit time integration method. From Eq. (5.14), an expression for $[\overline{C}_k^{n+1}]$ is substituted into Eq. (5.13). This results in a matrix equation of the form:

$$[A][\psi^{n+1}] = [S] \tag{5.15}$$

where

$$[\psi^{n+1}] \equiv \mathrm{col}\{[\phi^{n+1}],[L_x^{n+1}],[L_y^{n+1}]\}.$$

As in Sec. 4.2.2, a Gauss-Seidel iteration is used to invert the matrix [A]. The most difficult part of this iteration is the calculation of the average fluxes $[\bar{\phi}^{n+1}]$. The form of the matrix equation which must be solved is identical to that of Eq. (4.11). As in Sec. 4.2, the average fluxes were calculated by using the Cyclic Chebyshev method of iteration. After the new average fluxes and leakages have been calculated, the new precursors $\bar{C}_{ki,j}^{n+1}$ are calculated.

4.2.2 Results from a Homogeneous Test Problem

Test Problem 5.2 is a homogeneous test problem with two neutron groups and one delayed precursor family. Geometric and material constants are shown in Appendix B. Table 5.3 displays the total power versus time for a finite difference solution using a 20 cm mesh,[33] and 2DTD solutions using 20 and 10 cm mesh sizes. The 2DTD runs used a time step of 5 ms, which was determined to be adequate for temporal convergence.

Table 5.3 shows that a coarse finite difference spatial mesh can lead to errors in the transient solution. The 2DTD run using a 20 cm mesh is seen to be very accurate. Table 5.4 shows the initial power distribution using 2DTD (since 2DTD steady-state results are exact

TABLE 5.3

Total Powers versus Time for Test Problem 5.2

	Total Power		
Time (sec)	Finite Difference (20 cm)	2DTD (20 cm)	2DTD (10 cm)
0.0	1.0	1.0	1.0
.08	1.605	1.592	1.592
.16	2.136	2.103	2.105
.24	2.610	2.546	2.550
.32	3.031	2.934	2.940
.40	3.411	3.275	3.284

Finite Difference, $\Delta T = .5$ ms.

2DTD, $\Delta T = 5$ ms.

Initial Eigenvalues:	Analytic		1.01133
	2DTD	10 cm	1.01177
	2DTD	20 cm	1.01320

TABLE 5.4

Initial Region Powers for Test Problem 5.2

				Y Region		
		1	2	3	4	5
X Region	5	.1512 - 1	.1364 - 1	.1083 - 1	.6952 - 2	.2395 - 2
	4	.4389 - 1	.3960 - 1	.3142 - 1	.2018 - 1	.6952 - 2
	3	.6836 - 1	.6167 - 1	.4894 - 1	.3142 - 1	.1083 - 1
	2	.8614 - 1	.7771 - 1	.6167 - 1	.3960 - 1	.1364 - 1
	1	.9549 - 1	.8614 - 1	.6836 - 1	.4389 - 1	.1512 - 1

Region definitions based on a 20 cm mesh.

All powers normalized to unity total power.

for this case, 2DTD results using 20 or 10 cm mesh sizes are identical). Since the perturbation is homogeneous, the shape of the power distribution does not change with time.

A very important aspect of the 2DTD method is the degree of difficulty of solving Eq. (5.15) at each time step. The matrix [A] is inverted by using a Gauss-Seidel iteration. 2DTD runs have been made for this test problem using one, two, and three Gauss-Seidel iterations at each time step. The results from these runs showed no observable differences. Therefore, the general strategy used for steady state problems of performing only one Gauss-Seidel iteration appears also to be valid for the transient case.

In two dimensions, it is felt to be most efficient to calculate the average fluxes by an iterative technique. An important aspect of this iteration is the degree of convergence required to give accurate solutions. In 2DTD, the transient convergence criterion is expressed in the form of an error reduction, which is based on the asymptotic error reduction achievable using the Cyclic Chebyshev method. 2DTD runs were made using a variety of error reductions. It was found that for this problem an error reduction of .05 was sufficient. This error reduction corresponds to 8 Cyclic Chebyshev iterations per time step. We note that larger mesh sizes and smaller time steps will tend to make the iterations required to calculate the average fluxes easier to converge.

5.2.3 Results from TWIGL Test Problems

Test Problems 5.3 and 5.4 are test cases originally solved by the TWIGL program.[34] Problem 5.3 is a step insertion of positive reactivity, while Problem 5.4 is a ramp insertion of positive reactivity. Geometric and material properties are shown in Appendix B. To determine the appropriate spatial mesh sizes for 2DTD, several steady state runs were made. The eigenvalues so obtained are shown in Table 5.5. An examination of the eigenvalues and region powers showed that the 2DTD Fine Mesh solutions are very accurate. We shall therefore display transient results for 2DTD using the Fine Mesh results as the reference solution.

The TWIGL results given in Reference 5 do not include average powers within each homogeneous region, nor do they include the total power. Only the pointwise fluxes are shown, and these are not immediately comparable to 2DTD results. TWIGL execution times are available, however.

Tables 5.6 and 5.7 show the total powers versus time found from 2DTD for Test Problems 5.3 and 5.4, respectively. These results show that the 2DTD method using the "coarse" mesh structure yields very accurate results. The time-dependent assembly powers for both test problems are given in Appendix C. The time steps used by 2DTD were determined to provide converged temporal results. The time steps used compare favorably with the steps used to give accurate TWIGL solutions.

As was done for Test Problem 5.2, 2DTD runs were made to determine the number of Gauss-Seidel iterations required per time step in order to solve Eq. (5.15). Again, these results showed that one iteration per time step is quite adequate. The transient error reduction

TABLE 5.5

Eigenvalues Obtained for Test Problems 5.3 and 5.4

Method	Eigenvalue
2DTD Fine Mesh	.913363
2DTD Coarse Mesh	.913684
TWIGL (finite differences)	.914193

Most accurate eigenvalue avaliable – .91322.

2DTD Fine Mesh – Uniform 8 cm.

2DTD Coarse Mesh – 2 (12 cm), 2 (16 cm), 2 (12 cm).

TWIGL – Uniform 8 cm.

TABLE 5.6

Total Powers versus Time for Test Problem 5.3

	2DTD Total Power	
Time (sec)	Coarse Mesh	Fine Mesh
0.0	1.0	1.0
0.1	2.051	2.059
0.2	2.068	2.076
0.3	2.085	2.094
0.4	2.102	2.111
0.5	2.119	2.129

2DTD runs used $\Delta T = 10$ ms.

Transient Error Reduction: .05.

Number of Cyclic Chebyshev iterations per time step:
Coarse Mesh: 18
Fine Mesh: 33

Error Summary:

i) Maximum error in total power: -.5%.

ii) Error in perturbed region power at t = 0.0 sec: -.16%.

iii) Error in perturbed region power at t = 0.5 sec: -1.4%.

TABLE 5.7

Total Powers versus Time for Test Problem 5.4

	2DTD Total Power	
Time (sec)	Coarse Mesh	Fine Mesh
0.0	1.0	1.0
0.1	1.305	1.308
0.2	1.951	1.959
0.3	2.064	2.073
0.4	2.081	2.090
0.5	2.098	2.108

2DTD runs used $\Delta T = 5$ ms.

Transient Error Reduction: .05.

Number of Cyclic Chebyshev iterations per time step:
Coarse Mesh: 13
Fine Mesh: 24

Error Summary:

i) Maximum error in total power: -.5%.

ii) Error in perturbed region power at t = 0.0 sec: -.16%.

iii) Error in perturbed region power at t = 0.5 sec: -.57%.

criterion of .05 used in the previous test problem was found also to be adequate for Test Problems 5.3 and 5.4.

The execution times of 2DTD are of obvious interest. 2DTD execution times for the Coarse and Fine mesh sizes will be compared to the TWIGL execution times given in Reference 34. The TWIGL execution times are adjusted since TWIGL solves the full core problem rather than the quarter core problem. The digital computer speeds are comparable (IBM 370/168 - CDC 6600). The execution times are shown in Table 5.8. This table demonstrates the superiority of 2DTD for transient calculations. One very interesting aspect of Table 5.8 is the comparison between TWIGL and the 2DTD Fine Mesh case. Since the mesh sizes are identical, 2DTD involves three times the number of unknowns (a flux and two leakages) as TWIGL. However, strangely enough, it appears to run four times faster.

There appear to be two major reasons for the efficiency of 2DTD. First, the leakages do not require any iteration for their calculation; thus the majority of the computational effort is directed toward the calculation of the average fluxes. Second, TWIGL has two levels of iteration at each time step; an outer iteration (between the groups), and an inner iteration (to calculate the fluxes for this group). Since 2DTD solves both groups simultaneously, the extra level of iteration is eliminated.

TABLE 5.8

Execution Time Comparison for Test Problems 5.3 and 5.4

Test Problem	Method	Time Step	Execution Time (sec)
5.3	2DTD – Coarse	10 ms	5.28
5.3	2DTD – Fine	10 ms	18.78
5.3	TWIGL*	10 ms	86.5
5.4	2DTD – Coarse	5 ms	9.37
5.4	2DTD – Fine	5 ms	32.34
5.4	TWIGL*	5 ms	137.5

*All TWIGL times adjusted for quarter-core calculations.

5.2.4 Results from Test Problem 5.5

Test Problem 5.5 is a two-group problem based on a half-core configuration of the BWR Test Problem.[20] The thermal feedback model used in Ref. 20 was not used for this test problem. The geometric and material constants are shown in Appendix B. The half-core configuration was chosen so that the accuracy of the flat source method could be evaluated for test problems with large flux gradients.

This problem was designed to be representative of a rod ejection accident. As shown in Appendix B, the region labeled as Composition 6 is perturbed by a step change in Σ_a in both neutron groups. This perturbation represents a reactivity increase of 1.34 dollars. The transient is followed to .02 seconds, by which time the total power has risen by a factor of four and the perturbed region power by a factor of 20. This introduces a very severe spatial flux gradient.

Transient runs were made using the 2DTD probram with both 15 and 5 cm spatial mesh sizes. Steady state results showed that the assembly powers from the 5 cm mesh solution are accurate to within 0.2%. Therefore, the 5 cm results have been taken as the reference solution. To determine the appropriate time step size, transient runs using .05, .1, .2, and .4 ms time steps were made. The assembly powers and total power for the case using a .1 ms time step were determined to be accurate to within .5% as compared to the standard solution obtained by extrapolation. The five centimeter reference solution was also run using a .1 ms time step.

Table 5.9 shows some results from this test problem. For a

TABLE 5.9

Results from Test Problem 5.5

Time	Total Power	
	2DTD	2DTD
(ms)	(15 cm)*	(5 cm)†
0	1.0	1.0
2	1.1155	1.1160
4	1.2786	1.2786
6	1.4765	1.4745
8	1.7073	1.7011
10	1.9715	1.9586
12	2.2714	2.2488
14	2.6102	2.5741
16	2.9920	2.9379
18	3.4219	3.3443
20	3.9054	3.7980

*Time Step .1 ms, Execution Time 104 sec.

†Time Step .1 ms, Execution Time 1113 sec.

15 cm mesh, the maximum error in the total power is 2.8% (with the execution time under two minutes). Appendix C contains maps of the transient region powers for the reference solution, as well as percent errors in region powers for the coarse mesh 2DTD solution. An examination of these results shows that the maximum region power error for the 15 cm mesh solution occurs in one of the perturbed regions at T = .02 sec. The percentage error for this region at this time is 6.4%.

Runs were made for this test problem to determine the number of Gauss-Seidel iterations required per time step in order to solve Eq.(5.15). The results of these runs again showed that errors incurred from the use of only one iteration per time step are very small. Runs were also made to determine the appropriate transient error reduction for this test problem. Since the reactor is prompt critical, small time steps are required to follow the transient accurately. The use of small time steps tends to make the iterations which are needed to calculate the average fluxes converge more rapidly. To insure that the spatial iterations were fully converged, a transient error reduction of 10^{-4} was used for this test problem. For the 2DTD case with a time step of .1 ms and a 15 cm mesh size, this corresponded to the use of 8 Cyclic Chebyshev iterations per time step.

Numerical solutions to this test problem using finite difference techniques are not available. To obtain an estimate of the cost of solving this problem using finite difference methods, some static runs were made using the MEKIN code.[6] Our objective was to determine the spatial mesh size that is needed for MEKIN such that the maximum error in any assembly power was less than five percent. It was found that if a mesh size of 2.5 cm was used, the maximum assembly power was found to be 4.4%. The static solution to this half-core problem using MEKIN required 665 seconds of CPU time, as compared to 6 seconds for the 2DTD program using a 15 cm mesh.

For the transient problem, the CPU time per time step for the IBM 370/168 required by MEKIN is[6] $(3 \times 10^{-4}$ sec)(NPTX)(2+.3 * NDF) where NPTX is the number of spatial mesh points, and NDF is the number of delayed families. For this problem, NDF = 2 and NPTX = 8712; Therefore MEKIN would require 6.8 seconds per time step. The time integration method used by MEKIN requires smaller time step sizes to insure accurate solutions than does the fully implicit method. For many problems, five times more time steps are required.[1] Therefore, we estimate that a MEKIN solution of comparable accuracy to 2DTD would require 1000 time steps and thus 1.9 hours of CPU time. Table 5.9 shows that the 2DTD solution using a 15 cm mesh required less than two minutes to execute.

In the light of this discussion, we conclude that finite difference solutions cannot provide the combination of high accuracy and low computational effort that can be obtained from nodal schemes such as the flat source analytical method derived in this thesis.

5.2.5 Results from Test Problem 5.6

Test Problem 5.6 is a two-group transient problem which begins from the same initial conditions as does Test Problem 5.5. This test problem simulates the ramp insertion of several banks of control rods, with one of the banks "stuck" outside the core. The duration of the ramp is from 0 to .2 seconds, and the transient is followed to .5 seconds. The details of the perturbation are given in Appendix B.

Transient runs were made using the 2DTD program with a 15 cm mesh. The reference solution was calculated using 2DTD with a 5 cm mesh. To determine the appropriate time step size, transient runs were made using 5, 10, and 20 ms time steps. The results of these runs agreed to within .1% for the total power and assembly powers. The results using a 10 ms time step for both 15 and 5 cm mesh spacings will be presented.

Some results are shown in Table 5.10. Appendix C contains maps of the transient region powers for the reference solution, as well as percent errors in region powers for the coarse mesh 2DTD solution. An examination of these results shows that the region power errors for the coarse mesh solution are all less than 4.5%. The maximum error in the total power during the transient is 1.2%.

Transient runs were made to determine the number of Gauss-Seidel iterations required per time step in order to solve Eq. (5.15). Examination of the results using one, two, and three Gauss-Seidel iterations per time step showed no significant differences. The strategy of using one Gauss-Seidel iteration per time step has been successful for every transient problem attempted. Therefore, it is felt to be a good approximation for the general case.

The transient error reduction of .05 used in Sections 5.2.2 and 5.2.3 was found to be adequate for this test problem also. For the 15 cm 2DTD problem, this corresponded to the use of 12 Cyclic Chebyshev

TABLE 5.10

Results from Test Problem 5.6

Time	Total Power	
	2DTD	2DTD
(sec)	(15 cm)*	(5 cm)†
0.0	1.0	1.0
.05	.9182	.9112
.10	.8502	.8424
.15	.8020	.7936
.20	.7660	.7571
.25	.7604	.7526
.30	.7585	.7506
.35	.7566	.7487
.40	.7547	.7469
.45	.7531	.7451
.50	.7514	.7434

*Time Step 10 ms, Execution Time 28.4 sec.

†Time Step 10 ms, Execution Time 398 sec.

iterations per time step. The results in Table 5.10 and Appendix C show that the 15 cm 2DTD solution provides very accurate solutions in a small computing time. The fully implicit time integration method provides very accurate solutions for this problem.

In the calculation of the reference solution to this test problem, the 2DTD program predicted some negative fluxes in the reflector. This behavior was caused by the rapid decrease in power level of the perturbed assemblies. A scheme which was found to eliminate this problem will now be described.

In his investigation of albedo boundary conditions, Kalambokas[7] developed steady state and time-dependent expressions for the albedos. The time-dependent expressions involved omegas very similar to the ones used in this thesis. Kalambokas observed that unless the reactor was prompt supercritical, the omegas were rarely large enough to be of significant influence in the calculation of the albedos. Therefore, in 2DTD the prompt omegas (see Eq. (5.5a)) were set to zero for any region in which the fluxes were decreasing.

This scheme was found to eliminate completely the problem of negative fluxes for the 5 cm reference case. In addition, comparisons of this scheme with the normal omega scheme for the 15 cm solution to this test problem showed that the results agreed to within a few tenths of one percent. Therefore, it was felt that setting the prompt omegas equal to zero in any region where the fluxes are decreasing is a legitimate approximation for the general case. The results presented in this section are all based on the zero omega approximation.

5.2.6 Results from the BWR Kinetics Problem

Test Problem 5.7 is a two-dimensional quarter-core BWR kinetics problem.[30] This problem is designed to simulate a rod ejection accident from a low-power condition. The thermal feedback model used in this test problem incorporates adiabatic heatup with space-dependent Doppler feedback. The equations describing this process are

$$\alpha[\Sigma_{f1}(x,t)\ \phi_1(x,t) + \Sigma_{f2}(x,t)\ \phi_2(x,t)] = \frac{\partial}{\partial t} T(x,t) \tag{5.16}$$

$$\Sigma_{a1}(x,t) = \Sigma_{a1}(x,t{=}0)\{1 + \gamma[\sqrt{T(x,t)} - \sqrt{T_o}]\}. \tag{5.17}$$

The description of this test problem is given in Appendix B.

The 2DTD program was modified to include the above thermal feedback model, and several runs were made using a 15 cm spatial mesh. The following time step sizes were found to yield acceptably accurate results:

ΔT (sec)	Time (sec)
.01	$0 < t < 1.0$
.001	$1.0 < t < 1.3$
.0005	$1.3 < t < 1.6$
.002	$1.6 < t < 2.0$
.01	$2.0 < t < 3.0$

Table 5.11 presents a summary of the results obtained for this test problem, as well as results obtained by other investigators.[30, 37] Entries in Table 5.11 which have been left blank were either uncertain

TABLE 5.11

Results for Test Problem 5.7

	Method		
	Werner[37]	Finnemann[30]	2DTD
Number of time steps	1200		1300
Run time (sec)	120*		403
Initial λ	.99629	.99630	.99693
λ after rods out	1.01537	1.01531	1.01693
Time to first peak (sec)	1.455	1.4425	1.402
Time to second peak (sec)	2.0		2.0
P_{av} at first peak (w)	5712	5489	5627
P_{av} at second peak (w)	850		838
T_{max} at 3.0 sec (°K)		2979	3286
T_{av} at 3.0 sec (°K)		1096	1162

*Execution time on an IBM 360/91.

or unknown. Figure 5.1 shows a semilog plot of the total power during the interesting period of the transient. Detailed power distribution and temperature results are presented in Appendix C.

This test problem is very difficult since the neutron fluxes must increase by many orders of magnitude before the thermal feedback comes into play. For this particular problem, a time integration method which uses an exponential transformation would have been more effective than the fully implicit method used in 2DTD. The results in Table 5.11, however, show that 2DTD is capable of producing accurate results within a reasonable computing time.

5.3 Summary

One- and two-dimensional static methods for solving the diffusion equations were derived in Chapters 3 and 4. In this chapter, we have shown that accurate transient results can be obtained using spatial mesh sizes as large as 20 cm. The analytic method has also been shown to enjoy a significant cost savings over conventional finite difference methods.

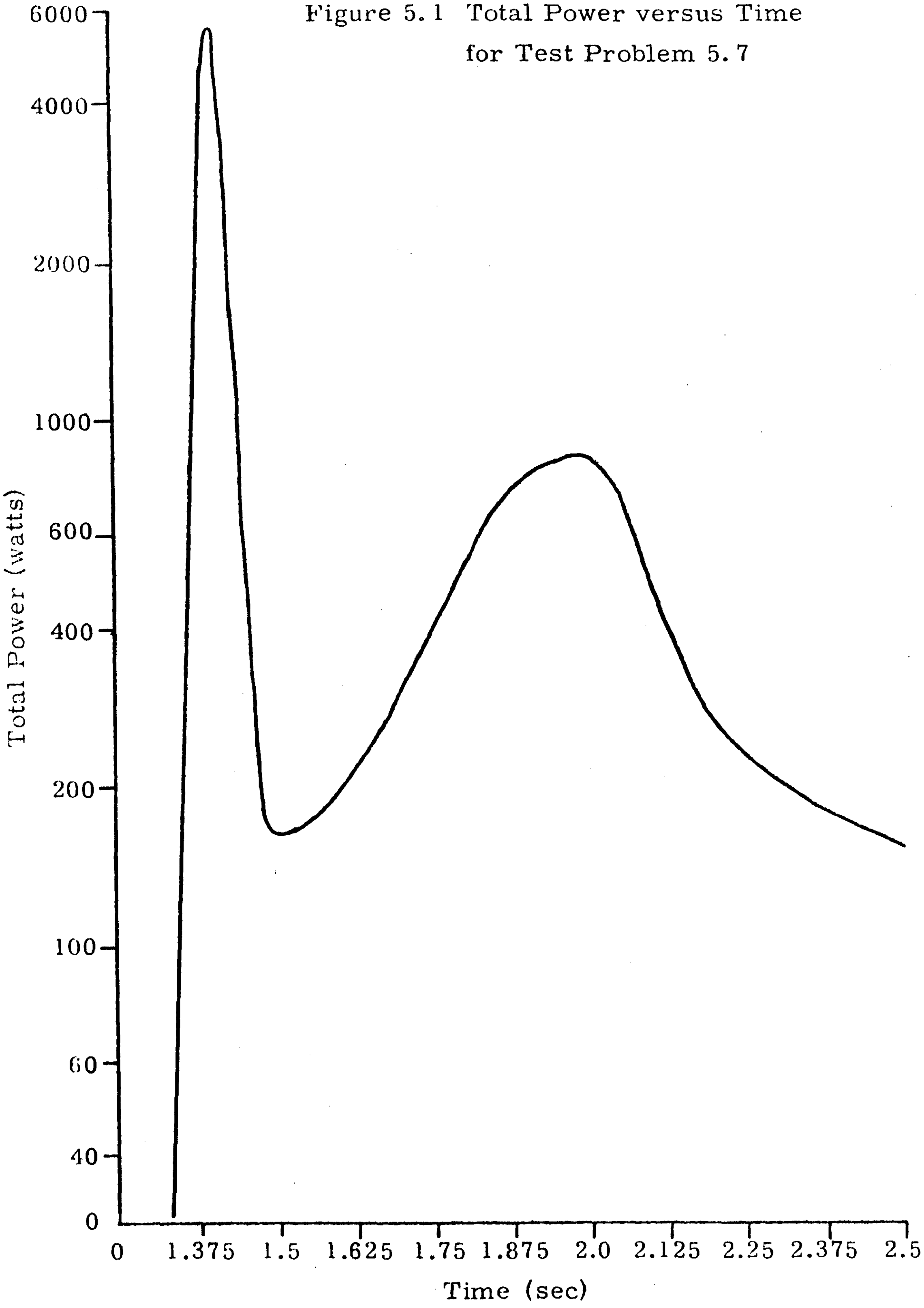


Figure 5.1 Total Power versus Time for Test Problem 5.7

Chapter 6
SUMMARY

6.1 Overview of Thesis Results

The objective of this thesis was to develop economical computational techniques for transient analysis of light-water reactors. The initial stages of the research consisted of an investigation of the finite element method.[12] This method has been shown to yield accurate results for static and transient reactor problems. In Chapter 2, the results of several attempts to speed the solution of the finite element equations by non-linear means were reported. Although some success was obtained by formulating a different eigenvalue problem, it was felt that a finite element solution to the space-dependent reactor kinetics equations was overly costly.

In Chapter 3 a one-dimensional analytic method for solving the static one- and two-group diffusion equations was derived. This method was shown to be exact in one dimension. The analytic method was successfully extended to two spatial dimensions in Chapter 4. Two procedures for accomplishing this extension were developed; the "buckling" method and the flat source method. These methods were tested for a variety of two-group, two-dimensional static problems. The results of these test problems demonstrated that the flat source method possessed the optimum combination of accuracy and execution time. The flat source method was therefore chosen to be extended to time-dependent problems.

In Chapter 5, the one-dimensional analytic method and the two-dimensional flat source method were both extended to time-dependent problems. The results for a variety of test problems showed that the approximations required to accomplish this extension were very accurate for mesh spacings as large as 20 cm. The two-dimensional time-dependent test problems demonstrated the accuracy and efficiency of the flat source method. For both of the TWIGL test problems, the 2DTD Coarse Mesh solution was shown to be more accurate than the TWIGL solution; however, its execution time was an order of magnitude smaller than that of TWIGL. Test Problems 5.5 and 5.6 showed that 2DTD retains high spatial accuracy even for problems with large flux tilts. Comparisons with conventional finite difference techniques (MEKIN) showed that for equivalent accuracy, 2DTD can be as much as 60 times more cost effective. Finally, Test Problem 5.7 showed that accurate transient results can be obtained for a very difficult test problem in which the flux changes by nine orders of magnitude.

The analytic method developed in this thesis has been shown to be accurate and economical for two-dimensional, two-group transient reactor calculations. Comparisons with finite difference methods show that the analytic method can be between 10 and 60 times less costly to run. The analytic method is therefore an attractive alternative to finite difference methods now in general use.

6.2 Extension to Three Dimensions

The results presented in this thesis have demonstrated that efficient computational techniques should possess a nearest-neighbor coupling relationship. The flat source method can be easily extended to three-dimensional problems, and will retain this nearest-neighbor coupling

TABLE 6.1

Execution Time Breakdown for Test Problems 5.5 and 5.6

	Test Problem 5.5	Test Problem 5.6
Matrix Generation	55%	51%
Iterative Solution of Average Fluxes	18%	24%
Leakage and Right-hand Side Calculation	27%	25%

relationship. In three dimensions, the result of each one-dimensional analytic solution is an equation in which the x directed leakages are coupled to the average fluxes, and also to the y and z directed leakages. This equation and its counterparts in the y and z directions are then substituted into the integrated diffusion equation. The numerical techniques used in the 2DTD code can be applied to the solution of the three-dimensional problem.

The numerical solution of the two-dimensional transient diffusion equations have been accomplished by the use of powerful iterative techniques, such as the Cyclic Chebyshev iterative method. It is doubtful that further improvements in execution time could be obtained by attempting a more efficient iterative solution.

The transient results for Test Problems 5.5 and 5.6 have been analyzed to determine where the bulk of the computational effort is being expended. Table 6.1 shows the breakdown of computational effort for the 15 cm solutions to Test Problems 5.5 and 5.6. This table shows that over 50% of the execution time during a transient is spent calculating the matrices at each time step. Thus far, no significant effort has been made to optimize the calculation of the matrices. It is apparent that the matrix calculation should be performed with considerable care so that the efficiency of the overall method is not seriously degraded.

6.3 Recommendations for Future Work

Future work is required in the following areas:

i) The method incorporated in the 2DTD program uses a flat representation of the transverse leakage. A higher order representation, such as a quadratic expansion, can lead to significant improvements in accuracy for some test problems. Although such a representation would appear to be very complicated, the accuracy improvement would suggest that this area should be explored further.

ii) The fully implicit time integration method used in 2DTD is a very simple solution technique. As discussed in Sec. 5.2.6, for some problems a time integration method which uses the exponential transformation could be superior to schemes which do not use it. The applicability of the exponential transformation to the method derived in this thesis should be investigated. In addition, other semi-implicit and alternating direction time integration methods should be investigated.

iii) Investigation of iteration strategies for three-dimensional problems.

iv) Efficient generation of the coefficient matrices. Also, investigations into whether approximations could be made to simplify the matrix elements.

v) Determination of average albedos at core-reflector and core-shroud interfaces.

vi) Efficient data management techniques for three-dimensional problems.

REFERENCES

1. A. F. Henry, Nuclear-Reactor Analysis (Cambridge, Mass.: MIT Press), 1975.

2. W. M. Stacey, Jr., Space-Time Nuclear Reactor Analysis (New York: Academic Press), 1969.

3. S. Kaplan, "Some New Methods of Flux Synthesis," NSE: 13, 22-35 (1962).

4. W. R. Cadwell, "PDQ-7 Reference Manual," WAPD-TM-678, Bettis Atomic Power Laboratory, 1967.

5. T. B. Fowler, D. R. Vondy, and G. W. Cunningham, "Nuclear Reactor Core Analysis Code: CITATION," ORNL-TM-2406, Oak Ridge National Laboratory, 1969.

6. G. Strang and G. J. Fix, An Analysis of the Finite Element Method (Englewood Cliffs, N.J.: Prentice-Hall), 1973.

7. C. M. Kang and K. F. Hansen, "Finite Element Methods for Reactor Analysis," NSE: 51, 456-495 (1973).

8. R. Avery, "Theory of Coupled Reactors," Proc. 2nd U.N. Conf. on Peaceful Uses of Atomic Energy, Geneva, 12, 182 (1958).

9. F. Bennewitz, H. Finnemann, and H. Modaschl, "Solution of the Multidimensional Neutron Diffusion Equation by Nodal Expansion," Proc. Conf. on Computational Methods in Nuclear Engineering, CONF-750413, I-99, 1975.

10. S. Langenbuch, W. Maurer, and W. Werner, "Simulation of Transients with Space-Dependent Feedback by Coarse Mesh Flux Expansion Method," CSNI/NEACRP Specialists' Meeting on New Develop-

ments in Three-Dimensional Neutron Kinetics and Review of Kinetics Benchmark Calculations, Garching (Munich), Jan. 22-24, 1975.

11. T. J. Burns and J. J. Dorning, "Multidimensional Applications of an Integral Balance Technique for Neutron Diffusion Calculations," Proc. Conf. on Computational Methods in Nuclear Engineering, CONF-750413, V-57, 1975

12. P. E. Rohan, C. M. Kang, R. A. Shober, and S. G. Wagner, "A Multi-dimensional Space-Time Kinetics Code for PWR Transients," CONF-750413, Charleston, S. C., p. VI-69.

13. C. M. Kang and K. F. Hansen, "Finite Element Methods for Reactor Analysis," NSE: 51, 456-495 (1973).

14. W. F. Walters and G. D. Miller, "Quadratic Finite Elements for X-Y and R-Z Geometrics," CONF-750413, Charleston, S. C., p. I-39.

15. L. O. Deppe and K. F. Hansen, "Application of the Finite Element Method to Two-Dimensional Diffusion Problems," NSE: 54, 456-465 (1974).

16. G. Strang and G. J. Fix, An Analysis of the Finite Element Method (Englewood Cliffs, N.J.: Prentice-Hall), 1973.

17. R. S. Varga, Matrix Iterative Analysis (Englewood Cliffs, N.J.: Prentice-Hall), 1962.

18. L. A. Hageman, "Numerical Methods and Techniques Used in the Two-Dimensional Neutron-Diffusion Program PDS-5," WAPD-TM-364, Bettis Atomic Power Laboratory, February 1963.

19. R. Froehlich, "A Theoretical Foundation for Coarse Mesh Variational Techniques," GA-7870, General Atomic Company, March 15, 1967.

20. J. W. Stewart, private communications (1976).

21. D. S. Rowe, "COBRA IIIC: A Digital Computer Program for Steady State and Transient Thermal-hydraulic Analysis of Rod Bundle Nuclear Fuel Elements," BNWL-1695 (March 1973).

22. P. Antonopoulos, "Large Mesh Model Development Study," Sc.D. Thesis, Department of Nuclear Engineering, MIT, Cambridge, Mass. (1972).

23. Z. Weiss, "Some Basic Properties of Response Matrix Equations," ANS Transactions, Vol. 23, June 1976, 198-199.

24. P. C. Kalambokas and A. F. Henry, "The Replacement of Reflectors by Albedo-Type Boundary Conditions," MITNE-183, MIT, Cambridge, Mass. (1975).

25. Argonne Code Center: Benchmark Problem Book, ANL-7416, Sept. 1, 1976.

26. H. Finnemann and M. R. Wagner, "A New Computational Technique for the Solution of Multidimensional Neutron Diffusion Problems," International Meeting of Specialists on "Methods of Neutron Transport Theory in Reactor Calculations," Bologna, Italy, Nov. 3-5, 1975.

27. F. Bennewitz, H. Finnemann, and M. R. Wagner, "Higher Order Corrections in Nodal Reactor Calculations," Trans. Am. Nucl. Soc. 22, 250 (1975).

28. R. Sims and A. F. Henry, "Coarse Mesh Nodal Methods Based on Response Matrix Considerations," Trans. Am. Nucl. Soc. 24 (1975).

29. G. K. Kristiansen, "Investigation of the Accuracy of Centerpoint-Cornerpoint, and Finite-Element Methods for Solution of the Neutron Diffusion Equation," RP-3-76/NEACRP-L-189.

30. W. Werner, H. Finnemann, and S. Langenbuch, "Two- and Three-Dimensional BWR Kinetics Problem," Trans. Am. Nucl. Soc. 23, 215-217 (1976).

31. D. R. Ferguson, "Solution of the Space-Dependent Reactor Kinetics Equations in Three Dimensions," MITNE-132, M.I.T. Department of Nuclear Engineering, Cambridge, Mass., August 1971.

32. M. M. Anderson, et al., "Three-Dimensional Coupled Neutronic and Engineering Calculations of Savannah River Reactors," CONF-750413, Charleston, S.C., p. VI-123 (1975).

33. W. H. Reed and K. F. Hansen, "Alternating Direction Methods for Reactor Kinetics Equations," NSE: 41, 431 (1970).

34. L. A. Hageman and J. B. Yasinsky, "Comparison of Alternating-Direction Time-Differencing Methods and Other Implicit Methods for the Solution of the Neutron Group Diffusion Equations," NSE: 38, 8-32 (1969).

35. R. W. Bowring, J. W. Stewart, R. A. Shober, and R. N. Sims, "MEKIN: MIT-EPRI Nuclear Reactor Core Kinetics Code," CCM-1, Research Project 227, Electric Power Research Institute, September 1975.

36. P. C. Kalambokas and A. F. Henry, "The Replacement of Reflectors by Albedo-Type Boundary Conditions," MITNE-183, MIT Department of Nuclear Engineering, Cambridge, Mass., November 1975.

37. W. Werner, private communications (1976).

38. R. A. Shober, "Description of the Two-Group Time-Dependent Program 2DTD," MIT Department of Nuclear Engineering Report, November 1976.

Appendix A

DERIVATION OF THE MATRICES $[A^i]$, $[B^i]$, AND $[D^i]$

We wish to calculate the matrices $[A^i]$, $[B^i]$, and $[D^i]$. The derivation will be carried out for the case of two neutron energy groups. However, the results for both one- and two-neutron energy groups will be shown.

We assume that region R_i $(x_i \leqslant x \leqslant x_{i+1})$ is nuclearly homogeneous. Thus, for the two-group case, Eq. (3.1) can be rewritten as:

$$\begin{bmatrix} D_1 \frac{d^2}{dx^2} - \Sigma_1 & \frac{1}{\lambda} \nu \Sigma_{f2} \\ \Sigma_{r1} & D_2 \frac{d^2}{dx^2} - \Sigma_2 \end{bmatrix} \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix} = 0 \qquad (A.1)$$

where

$$\Sigma_1 = \Sigma_{T1} - \frac{1}{\lambda} \nu \Sigma_{f1}.$$

We seek particular solutions such that

$$\begin{bmatrix} \frac{d^2}{dx^2} & 0 \\ 0 & \frac{d^2}{dx^2} \end{bmatrix} \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix} = \begin{bmatrix} -B^2 & 0 \\ 0 & -B^2 \end{bmatrix} \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix}. \qquad (A.2)$$

Substituting Eq. (A.2) into Eq. (A.1), we see that the numbers B^2 must be chosen so that

$$\begin{bmatrix} -D_1 B^2 - \Sigma_1 & \frac{1}{\lambda} \nu \Sigma_{f2} \\ \Sigma_{r1} & -D_2 B^2 - \Sigma_1 \end{bmatrix} \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix} = 0. \qquad (A.3)$$

Thus, the determinant of the coefficient matrix in Eq. (A. 3) must vanish. Equation (A. 2) is then valid only for certain values of B^2. We see that there are two and only two values of B^2 which satisfy Eq. (A. 2). These values, designated κ^2 and $-\mu^2$, are defined as:

$$\kappa^2 = -\frac{1}{2}\left(\frac{\Sigma_1}{D_1}+\frac{\Sigma_2}{D_2}\right) + \sqrt{\left(\frac{\Sigma_1}{2D_1}-\frac{\Sigma_2}{2D_2}\right)^2 + \frac{\nu\Sigma_{f2}\Sigma_{r1}}{\lambda D_1 D_2}}$$

$$\mu^2 = \frac{1}{2}\left(\frac{\Sigma_1}{D_1}+\frac{\Sigma_2}{D_2}\right) + \sqrt{\left(\frac{\Sigma_1}{2D_1}-\frac{\Sigma_2}{2D_2}\right)^2 + \frac{\nu\Sigma_{f2}\Sigma_{r1}}{\lambda D_1 D_2}} \qquad \text{(A. 4)}$$

where μ^2 is always positive, and κ^2 can be either negative or positive. Let us further define

$$R(B^2) = \frac{\phi_2}{\phi_1} = \frac{\Sigma_{r1}}{D_2 B^2 + \Sigma_2}$$

so we have $R(\kappa^2) \equiv r$ and $R(-\mu^2) \equiv s$.

The general solution to Eq. (A. 1) is then the linear combination

$$\begin{bmatrix}\phi_1(x)\\ \phi_2(x)\end{bmatrix} = a_1\begin{bmatrix}1\\ r\end{bmatrix}\sin\kappa x + a_2\begin{bmatrix}1\\ r\end{bmatrix}\cos\kappa x$$

$$+ a_3\begin{bmatrix}1\\ s\end{bmatrix}\sinh\mu x + a_4\begin{bmatrix}1\\ s\end{bmatrix}\cosh\mu x$$

or

$$\begin{bmatrix}\phi_1(x)\\ \phi_2(x)\end{bmatrix} = \begin{bmatrix}1 & 1\\ r & s\end{bmatrix}\begin{bmatrix}a_1\sin\kappa x + a_2\cos\kappa x\\ a_3\sinh\mu x + a_4\cosh\mu x\end{bmatrix} \qquad \text{(A. 5)}$$

and the corresponding current vector is

$$\begin{bmatrix} -D_1 \frac{d}{dx} \; \phi_1(x) \\ \\ -D_2 \frac{d}{dx} \; \phi_2(x) \end{bmatrix} = \begin{bmatrix} -D_1 & -D_1 \\ \\ -D_{2r} & -D_{2s} \end{bmatrix} \begin{bmatrix} a_1\kappa \cos \kappa x - a_2\kappa \sin \kappa x \\ \\ a_3\mu \cosh \mu\kappa + a_4\mu \sinh \mu x \end{bmatrix}$$

$$= \begin{bmatrix} J_1(x) \\ \\ J_2(x) \end{bmatrix}.$$

Let us write the total flux-current vector as

$$[\Phi(x)] = \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ J_1(x) \\ J_2(x) \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 1 & 0 & 0 \\ r & s & 0 & 0 \\ 0 & 0 & -D_1 & -D_1 \\ 0 & 0 & -D_{2r} & -D_{2s} \end{bmatrix} \begin{bmatrix} \sin \kappa x & \cos \kappa x & 0 & 0 \\ 0 & 0 & \sinh \mu x & \cosh \mu x \\ \kappa \cos \kappa x & -\kappa \sin \kappa x & 0 & 0 \\ 0 & 0 & \mu \cosh \mu x & \mu \sinh \mu x \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

$$= [E][F(x)][A]$$

where [A] is the column vector col $\{a_1, a_2, a_3, a_4\}$. Hence the above equation reduces to

$$[\Phi(x)] = [E][F(x)][A]. \qquad (A.6)$$

Both the matrices [E] and [F(x)] have inverses. They are given by

$$[E]^{-1} = \frac{1}{s - r}\begin{bmatrix} s & -1 & 0 & 0 \\ -r & 1 & 0 & 0 \\ 0 & 0 & -\frac{s}{D_1} & \frac{1}{D_2} \\ 0 & 0 & \frac{r}{D_1} & -\frac{1}{D_2} \end{bmatrix}$$

$$[F(x)]^{-1} = \begin{bmatrix} \sin \kappa x & 0 & \frac{1}{\kappa}\cos \kappa x & 0 \\ \cos \kappa x & 0 & -\frac{1}{\kappa}\sin \kappa x & 0 \\ 0 & -\sinh \mu x & 0 & \frac{1}{\mu}\cosh \mu x \\ 0 & \cosh \mu x & 0 & -\frac{1}{\mu}\sinh \mu x \end{bmatrix}.$$

Thus, the coefficients in the general solution are given by

$$[A] = [F(x)]^{-1}\,[E]^{-1}\,[\Phi(x)]. \tag{A.7}$$

If a homogeneous region extends from x_1 to x_2 we may find $[\Phi(x_1)]$ in terms of $[\Phi(x_2)]$ by applying Eq. (A.6) for $x = x_1$ and Eq. (A.7) for $x = x_2$. Thus

$$[\Phi(x_1)] = [E][F(x_1)][F(x_2)]^{-1}\,[E]^{-1}\,[\Phi(x_2)].$$

Defining $h = x_2 - x_1$, multiplying out and rearranging, we have

$$G(h) = [F(x_1)][F(x_2)]^{-1}$$

$$= \begin{bmatrix} \cos \kappa h & 0 & -\frac{1}{\kappa}\sin \kappa h & 0 \\ 0 & \cosh \mu h & 0 & -\frac{1}{\mu}\sinh \mu h \\ \kappa \sin \kappa h & 0 & \cos \kappa h & 0 \\ 0 & -\mu \sinh \kappa h & 0 & \cosh \mu h \end{bmatrix}.$$

Then we have

$$[\Phi(x_1)] = [E][G(h)][E]^{-1}[\Phi(x_2)]. \tag{A.8}$$

Comparing Eq. (A.8) to Eq. (3.4), we see that

$$\exp -[N_i]\, h = [E][G(h)][E]^{-1}.$$

Since we have an expression for $e^{-[N_i]h_i}$, we can obtain expressions for $\tanh [N_i]\, h_i/2$ and $\sinh [N_i]\, h_i$ by using the identities

$$\sinh x = \frac{1}{2}(e^x - e^{-x})$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

Performing the necessary algebra gives the final expressions

$$[A^i_{1,2}] = \left(\frac{1}{s_i - r_i}\right)\begin{bmatrix} \left(\frac{s_i \tan \kappa_i h_i/2}{\kappa_i D_{1i}} - \frac{r_i \tanh \mu_i h_i/2}{\mu_i D_{1i}}\right) & \left(-\frac{\tan \kappa_i h_i/2}{\kappa_i D_{2i}} + \frac{\tanh \mu_i h_i/2}{\mu_i D_{2i}}\right) \\ \left(\frac{r_i s_i \tan \kappa_i h_i/2}{\kappa_i D_{1i}} - \frac{r_i s_i \tanh \mu_i h_i/2}{\mu_i D_{1i}}\right) & \left(-\frac{r_i \tan \kappa_i h_i/2}{\kappa_i D_{2i}} + \frac{s_i \tanh \mu_i h_i/2}{\mu_i D_{2i}}\right) \end{bmatrix} \tag{A.9a}$$

$$[B^i_{1,1}] = \frac{h_i}{s_i - r_i} \begin{bmatrix} (\kappa_i s_i \csc \kappa_i h_i - \mu_i r_i \operatorname{csch} \mu_i h_i) & (-\kappa_i \csc \kappa_i h_i + \mu_i \operatorname{csch} \mu_i h_i) \\ (s_i r_i \kappa_i \csc \kappa_i h_i - s_i r_i \mu_i \operatorname{csch} \mu_i h_i) & (-\kappa_i r_i \csc \kappa_i h_i + \mu_i s_i \operatorname{csch} \mu_i h_i) \end{bmatrix} . \tag{A.9b}$$

To calculate the matrix $[D^i]$ defined in Eq. (4.18), we observe that the column vector [L] in Eq. (4.15) is defined

$$[L] = \mathrm{col}\left\{[0], -\frac{1}{h_j}[L_{y_{i,j}}]\right\},$$

and since only the top equation of Eq. (4.17) is used, we find that

$$[D^i] = -\left(\sinh^{-1}[N_i]\, h_i\right) h_i + [N_i]^{-1}.$$

Therefore it can be easily shown that

$$[D^i_{1,2}] = \frac{h_i}{s_i - r_i} \begin{bmatrix} -\dfrac{\csc \kappa_i h_i}{\kappa_i} & \dfrac{\operatorname{csch} \mu_i h_i}{\mu_i} \\ -\dfrac{r_i \csc \kappa_i h_i}{\kappa_i} & \dfrac{s_i \operatorname{csch} \mu_i h_i}{\mu_i} \end{bmatrix} \begin{bmatrix} -\dfrac{s_i}{D_{1i}} & \dfrac{1}{D_{2i}} \\ \dfrac{r_i}{D_{1i}} & -\dfrac{1}{D_{2i}} \end{bmatrix}$$

$$+ \frac{1}{\Sigma_{1i}\Sigma_{2i} - \frac{1}{\lambda}\nu\Sigma_{f2i}\Sigma_{r1i}} \begin{bmatrix} \Sigma_{2i} & \frac{1}{\lambda}\nu\Sigma_{f2i} \\ \Sigma_{r1i} & \Sigma_{1i} \end{bmatrix} . \tag{A.9c}$$

In a region where there is no fissioning, we note that s_i will tend to infinity. Thus, the matrices in Eq. (A.9) must be re-evaluated using l'Hopital's rule. We find that

$$[A^{i}_{1,2}] = \begin{bmatrix} \left(\dfrac{\tan \kappa_i h_i/2}{\kappa_i D_{1i}}\right) & 0 \\ \left(\dfrac{r_i \tan \kappa_i h_i/2}{\kappa_i D_{1i}} - \dfrac{r_i \tanh \mu_i h_i/2}{\mu_i D_{1i}}\right) & \left(\dfrac{\tanh \mu_i h_i/2}{\mu_i D_{2i}}\right) \end{bmatrix} \qquad \text{(A.10a)}$$

$$[B^{i}_{1,1}] = h_i \cdot \begin{bmatrix} (\kappa_i \csc \kappa_i h_i) & 0 \\ (r_i \kappa_i \csc \kappa_i h_i - r_i \mu_i \operatorname{csch} \mu_i h_i) & (\mu_i \operatorname{csch} \mu_i h_i) \end{bmatrix} \qquad \text{(A.10b)}$$

$$[D^{i}_{1,2}] = \begin{bmatrix} \left(\dfrac{h_i \csc \kappa_i h_i}{D_{1i} \kappa_i}\right) & 0 \\ \left(\dfrac{h_i r_i \csc \kappa_i h_i}{D_{1i} \kappa_i} + \dfrac{h_i r_i \operatorname{csch} \mu_i h_i}{D_{1i} \mu_i}\right) & \left(-\dfrac{h_i \operatorname{csch} \mu_i h_i}{D_{2i} \mu_i}\right) \end{bmatrix}$$

$$+ \frac{1}{\Sigma_{1i}\Sigma_{2i} - \frac{1}{\lambda}\nu\Sigma_{f2i}\Sigma_{r1i}} \begin{bmatrix} \Sigma_{2i} & \frac{1}{\lambda}\nu\Sigma_{f2i} \\ \Sigma_{r1i} & \Sigma_{1i} \end{bmatrix}. \qquad \text{(A.10c)}$$

For the case of one energy group, the following formulas are obtained:

Let $\kappa_i^2 = -\dfrac{\Sigma_i}{D_i}$

where

$$\Sigma_i = \Sigma_{Ti} - \frac{1}{\lambda}\nu\Sigma_{fi}.$$

Then

$$[A^i_{1,2}] = -\frac{\tan \kappa_i h_i/2}{D_i \kappa_i}$$

$$[B^i_{1,1}] = h_i \kappa_i \csc \kappa_i h_i \qquad (A.11)$$

$$[D^i_{1,2}] = \frac{h_i \csc \kappa_i h_i}{D_i \kappa_i} - \frac{1}{D_i \kappa_i^2} .$$

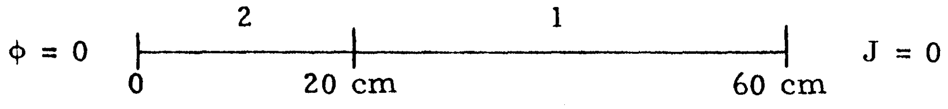
Appendix B

TEST PROBLEM DATA

1) Test Problem 3.1

Geometry:

Composition



Material Constants:

Group 1

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	1.5	.0623	.06	0
2	1.2	.101	.1	0

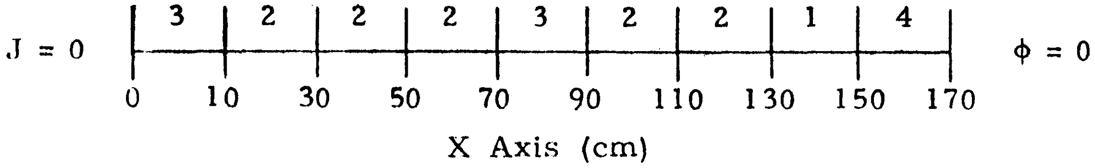
Group 2

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	.4	.2	0	.218
2	.15	.02	0	0

2) Test Problem 3.3

Geometry:

Composition



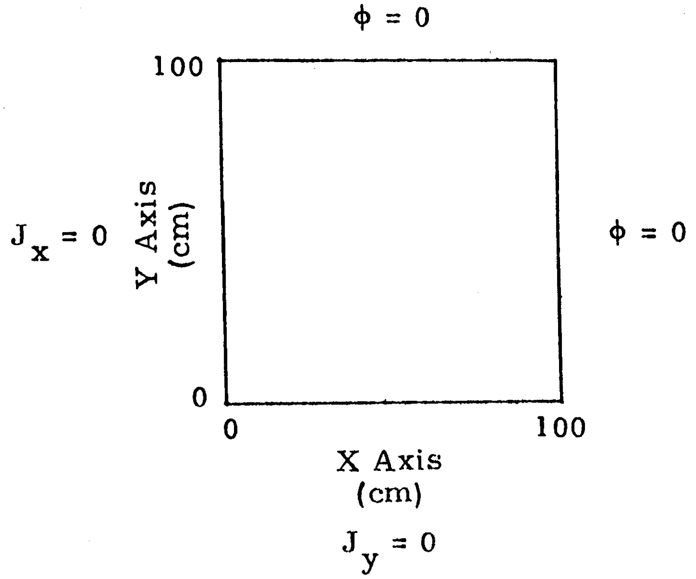
Material Constants:

Same as Test Problem 4.2. For the one-dimensional case, the buckling was not added.

3) Test Problem 4.1

Geometry:

A uniform square 100 cm on each side, with $\Delta x = \Delta y = 20$ cm.



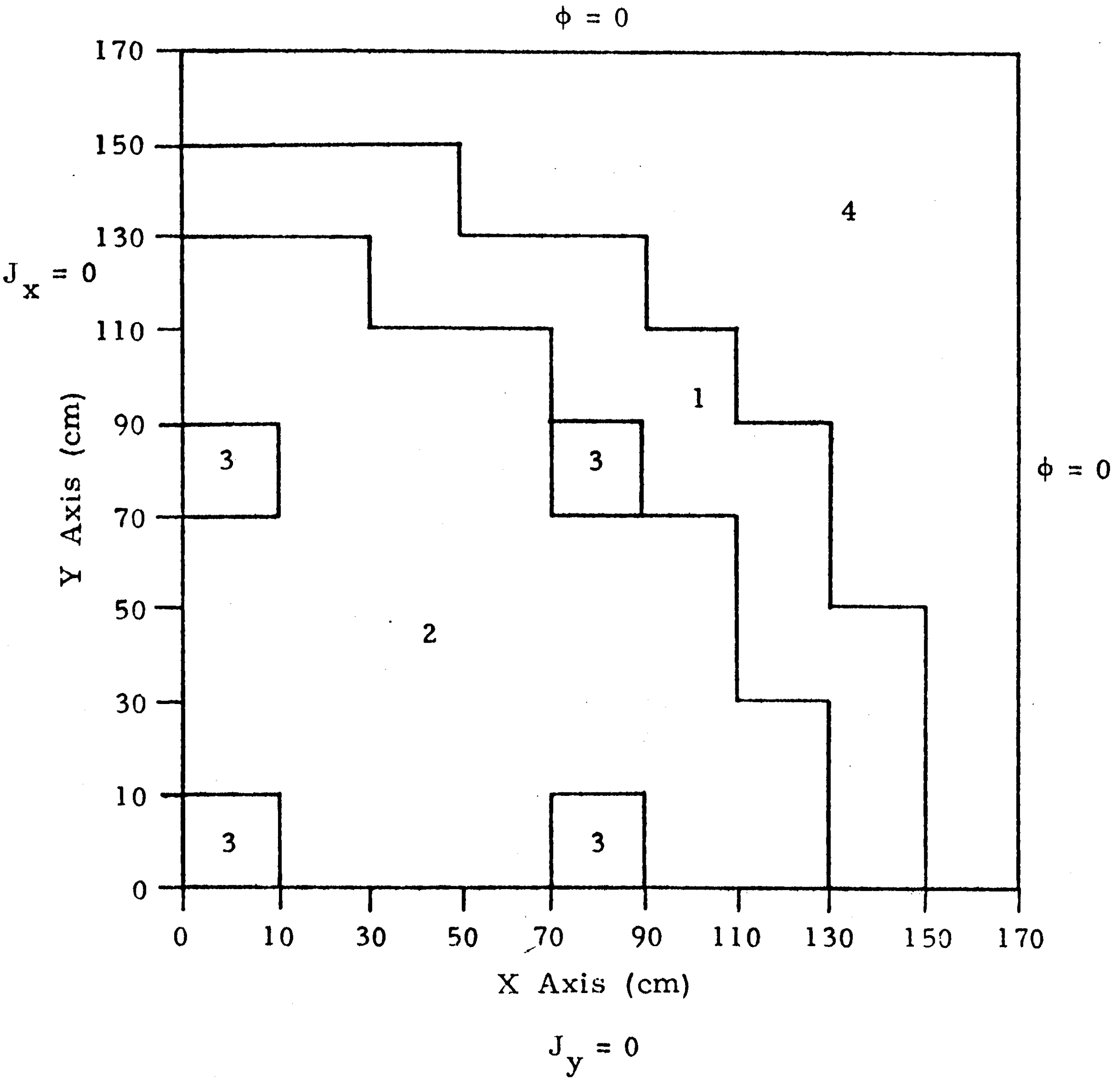
Material Constants

Group	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	1.5	.0623	.06	0
2	.4	.2	0	.218

4) Test Problem 4.2

Geometry:

This two-dimensional test problem is a square 170 cm on a side.



Material Constants:

Group 1

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	1.5	.03	.02	0
2	1.5	.03	.02	0
3	1.5	.03	.02	0
4	2.0	.04	.04	0

Group 2

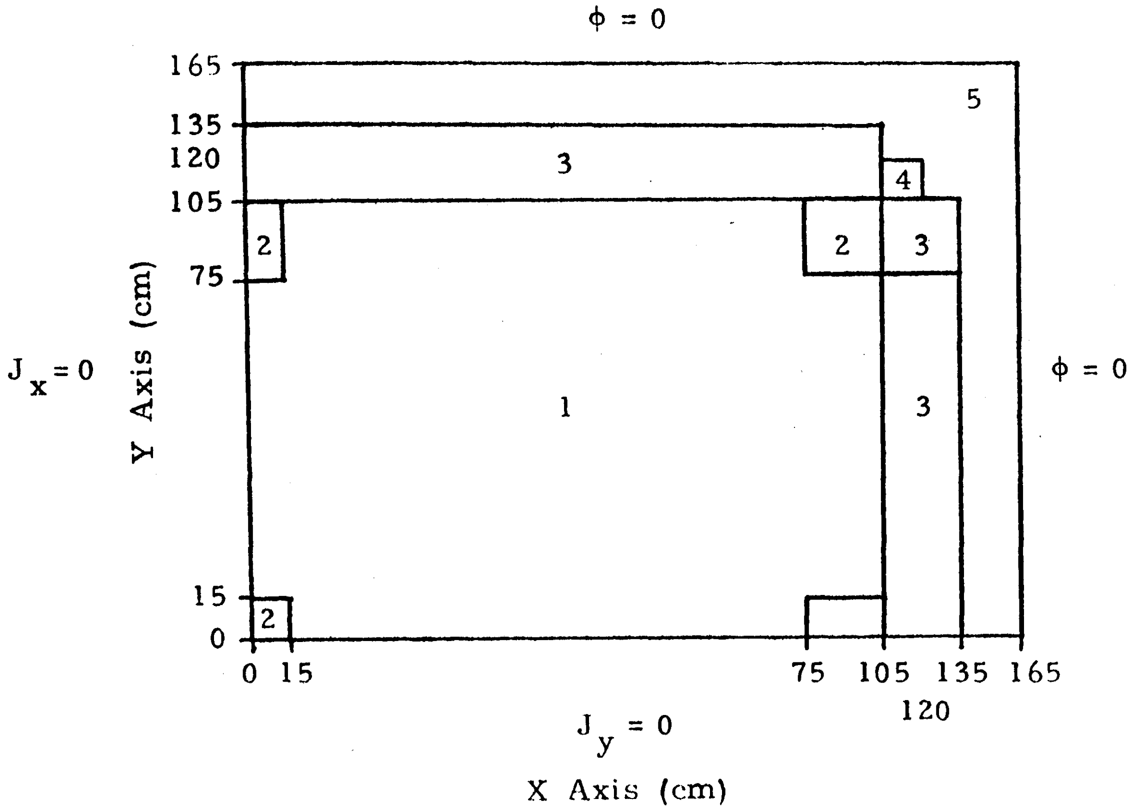
Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	.4	.08	0	.135
2	.4	.085	0	.135
3	.4	.13	0	.135
4	.3	.01	0	0

$B_z^2 = .8 \times 10^{-4}$ (in all compositions).

5) Test Problem 4.3

Geometry:

Test Problem 4.3 is a square 165 cm on a side.



Material Constants:

Group 1

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	1.255	.033582	.02533	.004602
2	1.268	.034851	.02767	.004609
3	1.259	.034172	.02617	.004663
4	1.234	.035172	.02805	.004668
5	1.257	.0481434	.04754	0

Group 2

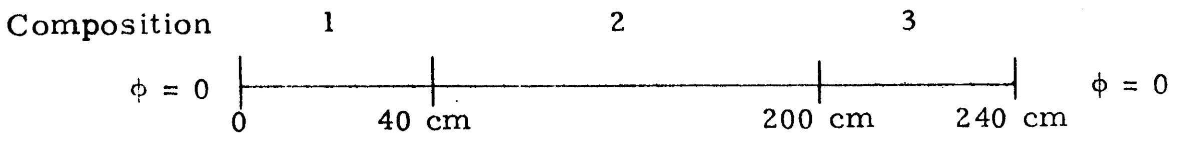
Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	.211	.1003	0	.1091
2	.1902	.07047	0	.08675
3	.2091	.08344	0	.1021
4	.1935	.06552	0	.08792
5	.1592	.01911	0	0

$B_z^2 = 1 \times 10^{-4}$ (all compositions).

6) Test Problem 5.1

Geometry:

Test Problem 5.1 is a one-dimensional two-group kinetics problem.



Material Constants:

Group 1

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	1.5	.026	.015	.01
2	1.0	.02	.015	.005
3	1.5	.026	.015	.01

Group 2

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	.5	.18	0	.2
2	.5	.08	0	.099
3	.5	.18	0	.2

Kinetic Parameters:

Delayed Family	Beta	Lambda
1	.00025	.0124
2	.00164	.0305
3	.00147	.111
4	.00296	.301
5	.00086	1.14
6	.00032	3.01

$v_1 = 10^7$

$v_2 = 3 \times 10^5$

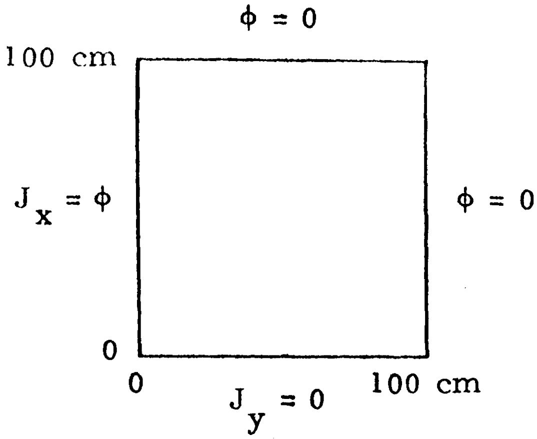
Perturbation:

Σ_{T2} in composition 1 is linearly increased by 3% over 1.0 seconds. Problem solved out to 2.0 sec.

7) Test Problem 5.2

Geometry:

Test Problem 5.2 is a two-dimensional two-group kinetics benchmark problem.



Material Constants:

Group	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	1.35	.003682	.0023	.00058322
2	1.08	.0056869	0	.0103148

Kinetic Parameters:

$\beta = .0064$, $\lambda = .08$

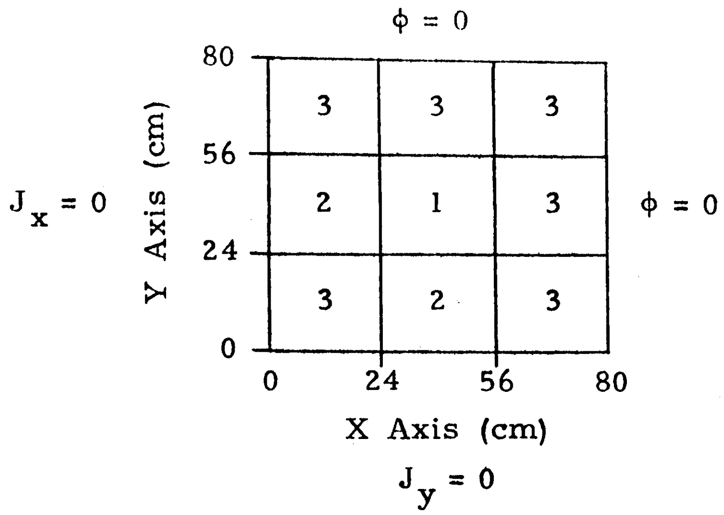
$v_1 = .3 \times 10^{-8}$, $v_2 = .22 \times 10^{-6}$

Step perturbation, $\Delta\Sigma_{T2} = -.369 \times 10^{-4}$.

8) Test Problems 5.3 and 5.4

Geometry:

Test Problems 5.3 and 5.4 are square test problems in quarter-core symmetry.



Composition 1 is the "seed" region.

Mesh spacings used:

TWIGL Mesh: Uniform 8 cm.

2DTD Coarse Mesh: 2 (12 cm), 2 (16 cm), 2 (12 cm).

2DTD Fine Mesh: Uniform 8 cm.

Material Constants:

Group 1

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	1.4	.02	.01	.007
2	1.4	.02	.01	.007
3	1.3	.018	.01	.003

Group 2

Composition	D_g	Σ_{Tg}	$\Sigma_{sgg'}$	$\nu\Sigma_{fg}$
1	.4	.15	0	.2
2	.4	.15	0	.2
3	.5	.05	0	.06

Kinetic Parameters:

$\beta = .0075$, $\lambda = .08$

$1/v_1 = 1 \times 10^{-7}$, $1/v_2 = 5 \times 10^{-6}$

Test Problem 5.3:

Step perturbation in Composition 1.

$\Delta\Sigma_{T2} = -.0035$. Problem duration $0 \leq t \leq .5$ sec.

Test Problem 5.4

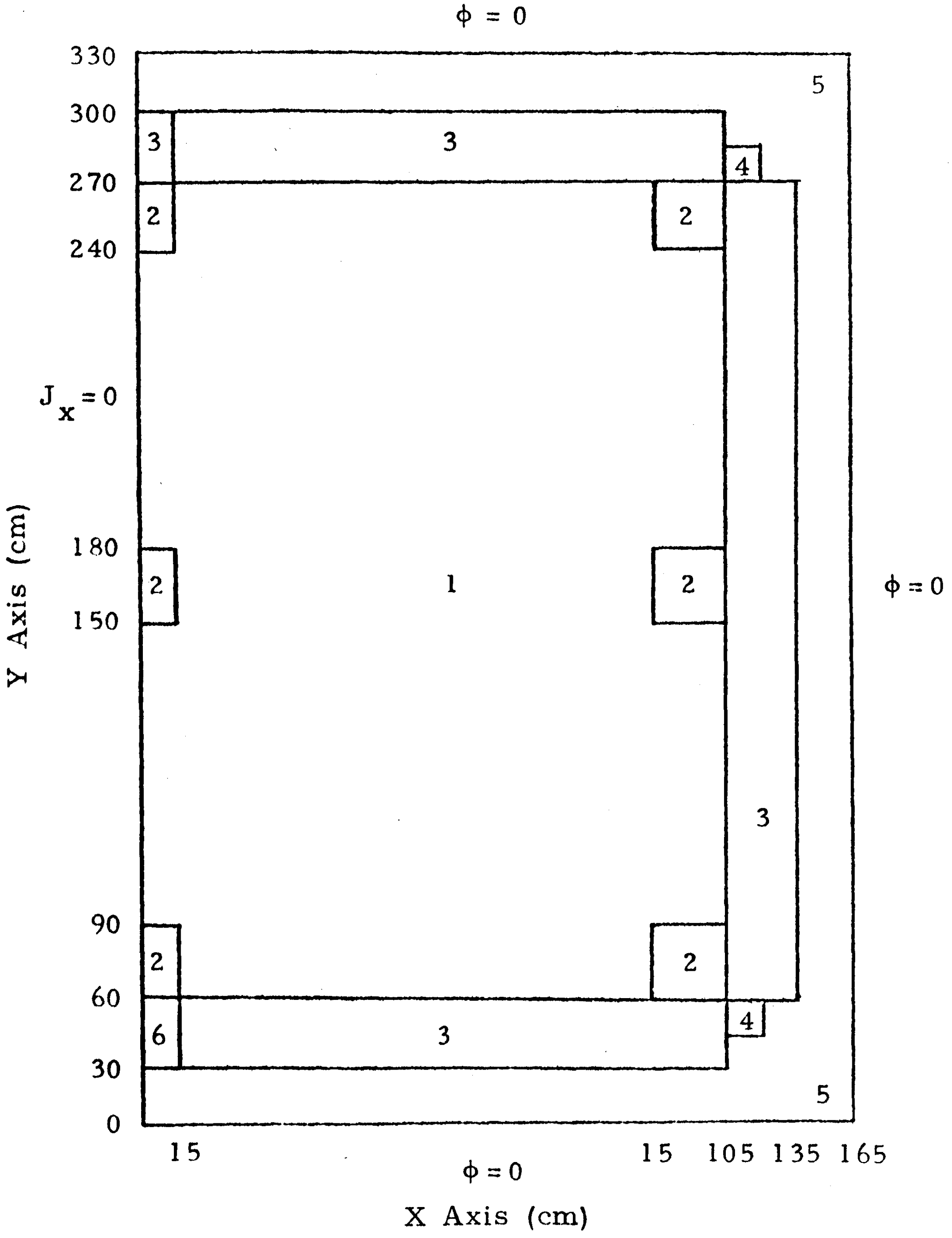
Ramp perturbation in Composition 1.

$\Delta\Sigma_{T2} = -.0035$, duration of ramp $0 \leq t \leq .2$ sec. Problem duration $0 \leq t \leq .5$ sec.

9) Test Problem 5. 5

Geometry:

Test Problem 5. 5 is a half-core BWR test Problem.



Material Constants:

Same as for Test Problem 4, 3, except for Composition 6; which is initially the same as Composition 3.

Kinetic Parameters:

Delayed Family	β	λ
1	.0054	.0654
2	.001087	1.35

$v_1 = 3 \times 10^7$, $v_2 = 3 \times 10^5$

Perturbation in Composition 6.

Step Change at $t = 0.0$

$\Delta\Sigma_{a1} = -.0006$

$\Delta\Sigma_{a2} = -.006$

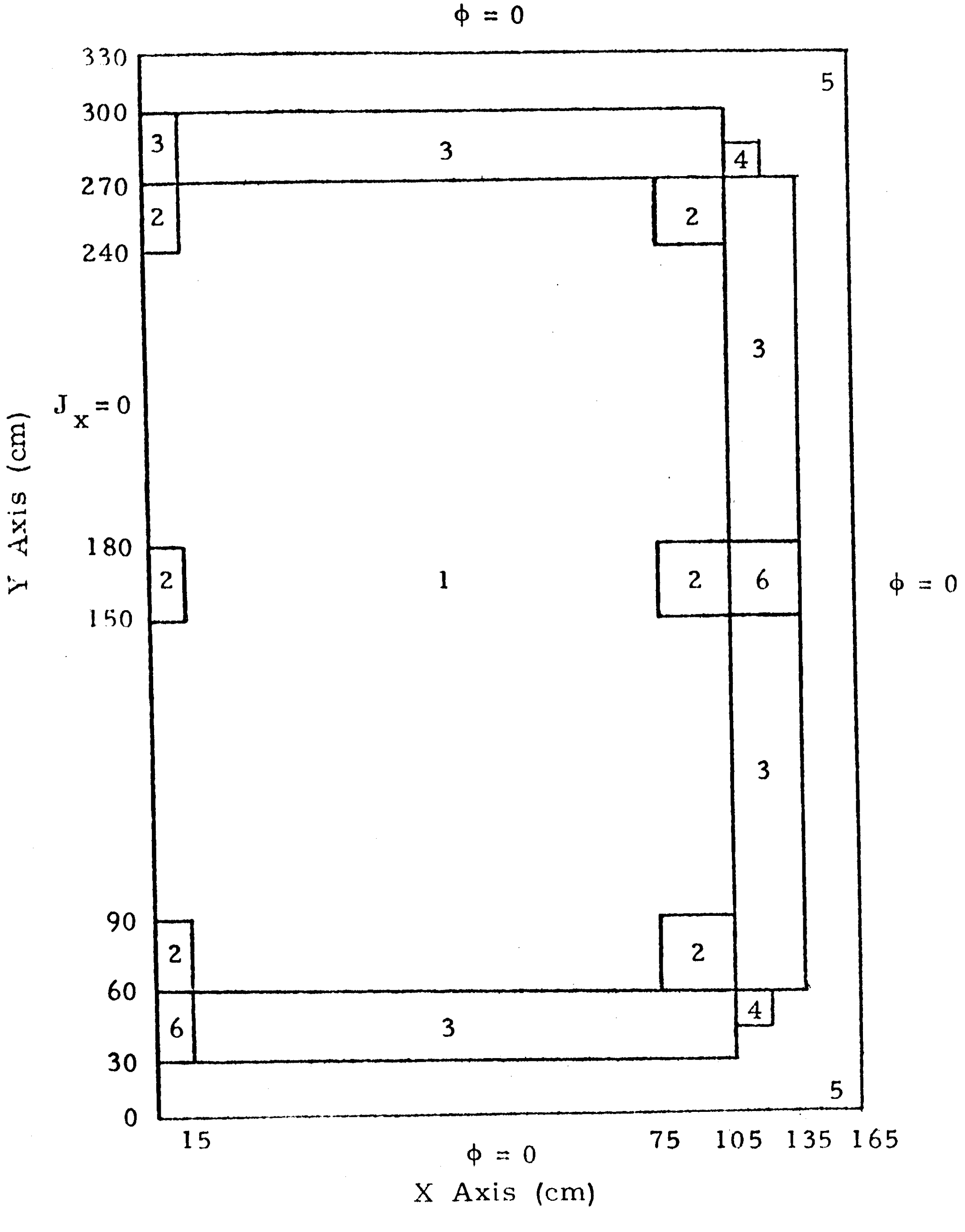
Duration of transient

$0 \leqslant t \leqslant .02$ sec.

10) Test Problem 5.6

Geometry:

Test Problem 5.6 is a half-core BWR test problem.



Material Constants:

Same as for Test Problem 4. 3, except Composition 6; which is initially the same as Composition 3.

Kinetic Parameters:

Same as Test Problem 5. 5.

Perturbation in Composition 6

Ramp change:

$$\Delta\Sigma_{a1} = 5. \times 10^{-4}$$

$$\Delta\Sigma_{a2} = 5 \times 10^{-3}$$

Duration of ramp

$0 \leq t \leq .2$ sec.

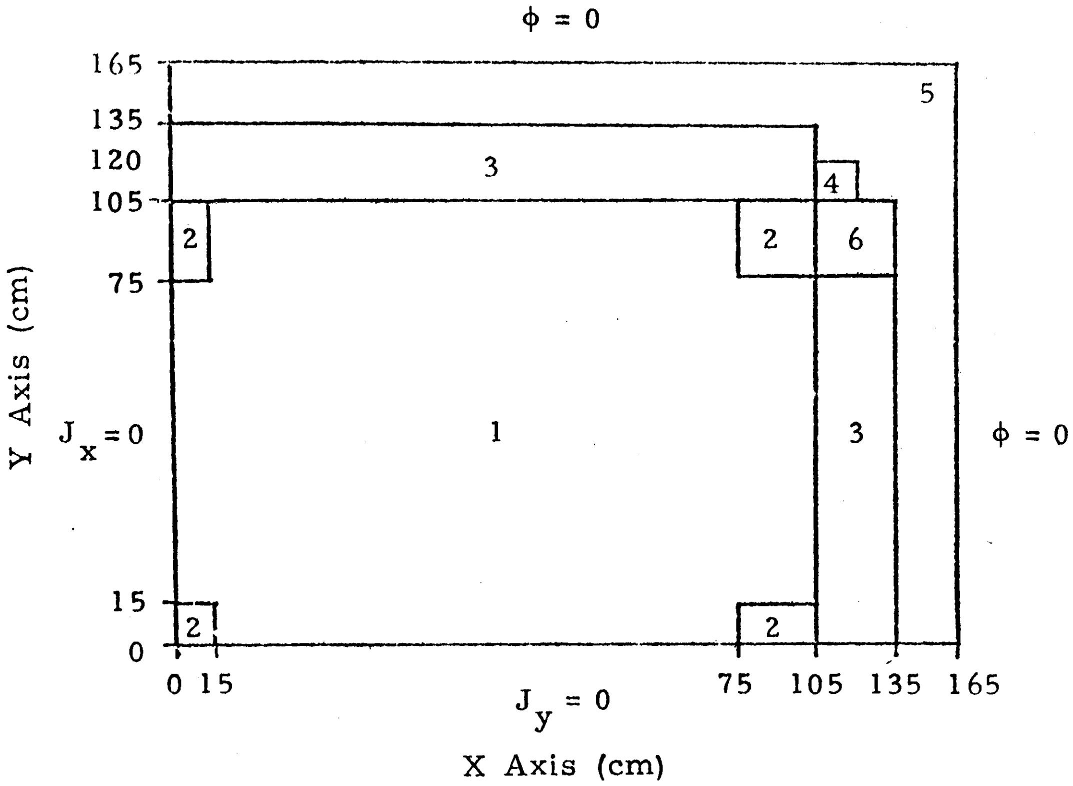
Duration of transient

$0 \leq t \leq .5$ sec.

11) Test Problem 5.7

Geometry:

Test Problem 5.7 is a quarter-core BWR test problem.



Material Constants:

Same as Test Problem 4.3, except for Composition 6, which is initially the same as Composition 3.

Kinetic Parameters:

Same as Test Problem 5.5.

Perturbation in Composition 6

Ramp change

$\Delta \Sigma_{a1} = 0$

$\Delta \Sigma_{a2} = .010116 \text{ cm}^{-1}$

Duration of ramp

$0 \leqslant t \leqslant 2.0$ sec.

Thermal Parameters:

Energy conversion parameter $\epsilon = .3204 \times 10^{-10}$ ωs/f

Mean power density at

$t = 0.0$ $\quad d = 1.0 \times 10^{-6}$ ω/cc

Conversion factor in feedback model

$\alpha = 3.83 \times 10^{-11}$ °K cc

Feedback constant

$\gamma = 3.034 \times 10^{-3}$ °K

Appendix C

RESULTS FROM TEST PROBLEMS

TABLE C.1

Region Powers at T = 0.0 and T = 0.5 sec for Test Problem 5.3

For both directions, let the regions be defined as

1: $0 \leq x \leq 24$ cm
2: $24 \leq x \leq 56$ cm
3: $56 \leq x \leq 80$ cm

$$\begin{pmatrix} \text{xxx} \\ \text{yyy} \end{pmatrix} = \begin{pmatrix} \text{2DTD Coarse Mesh} \\ \text{2DTD Fine Mesh} \end{pmatrix}$$

Region Power Map at T = 0.0 sec

	3	.03533 .03528	.03414 .03431	.008328 .008363
Y Region	2	.2438 .2433	.2518 .2522	.03414 .03431
	1	.1133 .1136	.2438 .2433	.03533 .03528
		1	2	3
			X Region	

Region Power Map at T = 0.5 sec

	3	.07360 .07442	.07305 .07435	.01797 .01825
Y Region	2	.5061 .5116	.5455 .5530	.07305 .07435
	1	.2331 .2366	.5061 .5116	.07360 .07442
		1	2	3
			X Region	

TABLE C.2

Region Powers at T = 0.5 sec for Test Problem 5.4

Region definitions given in Table C.2.

Region power map at T = 0.0 sec also given in Table C.2.

Region Power Map at T = 0.5 sec

	3	.07348 .07369	.07292 .07362	.01794 .01807
Y Region	2	.5052 .5066	.5445 .5476	.07292 .07362
	1	.2327 .2343	.5052 .5066	.07348 .07369
		1	2	3
			X Region	

Figure C.1 Initial Power Distribution for Test Problems 5.5 and 5.6: Reference Case

Y Axis (cm)	0–15	15–30	30–45	45–60	60–75	75–90	90–105	105–120	120–135
300–285	.5921-2	.5560-2	.5302-2	.5471-2	.5981-2	.6226-2	.5404-2		
285–270	.9493-2	.8218-2	.7516-2	.7827-2	.9121-2	.1076-1	.1045-1	.8308-2	
270–255	.1067-1	.7373-2	.6191-2	.6546-2	.8578-2	.1317-1	.1388-1	.1045-1	.5404-2
255–240	.8898-2	.6022-2	.5012-2	.5399-2	.7378-2	.1189-1	.1317-1	.1076-1	.6226-2
240–225	.5063-2	.4302-2	.3963-2	.4348-2	.5545-2	.7378-2	.8578-2	.9121-2	.5981-2
225–210	.3278-2	.3145-2	.3157-2	.3544-2	.4348-2	.5399-2	.6546-2	.7827-2	.5471-2
210–195	.2645-2	.2608-2	.2721-2	.3157-2	.3963-2	.5012-2	.6191-2	.7516-2	.5302-2
195–180	.2820-2	.2563-2	.2608-2	.3145-2	.4302-2	.6022-2	.7372-2	.8218-2	.5560-2
180–165	.3930-2	.2820-2	.2645-2	.3278-2	.5063-2	.8898-2	.1067-1	.9493-2	.5921-2
165–150	.3930-2	.2820-2	.2645-2	.3278-2	.5063-2	.8898-2	.1067-1	.9493-2	.5921-2
150–135	.2820-2	.2563-2	.2608-2	.3145-2	.4302-2	.6022-2	.7372-2	.8218-2	.5560-2
135–120	.2645-2	.2608-2	.2721-2	.3157-2	.3963-2	.5012-2	.6191-2	.7516-2	.5302-2
120–105	.3278-2	.3145-2	.3157-2	.3544-2	.4348-2	.5399-2	.6546-2	.7827-2	.5471-2
105–90	.5063-2	.4302-2	.3963-2	.4348-2	.5545-2	.7378-2	.8578-2	.9121-2	.5981-2
90–75	.8898-2	.6022-2	.5012-2	.5399-2	.7378-2	.1189-1	.1317-1	.1076-1	.6226-2
75–60	.1067-1	.7373-2	.6191-2	.6546-2	.8578-2	.1317-1	.1388-1	.1045-1	.5404-2
60–45	.9493-2	.8218-2	.7516-2	.7827-2	.9122-2	.1076-1	.1045-1	.8308-2	
45–30	.5921-2	.5560-2	.5302-2	.5471-2	.5982-2	.6226-2	.5404-2		

X Axis (cm)

Figure C. 2 Power Distribution at T = .005 for Test Problem 5. 5: Reference Case

Y Axis (cm)	0–15	15–30	30–45	45–60	60–75	75–90	90–105	105–120	120–135
300 / 285	.5944-2	.5582-2	.5322-2	.5490-2	.6001-2	.6246-2	.5421-2		
285 / 270	.9536-2	.8254-2	.7547-2	.7857-2	.9155-2	.1080-1	.1049-1	.8338-2	
270 / 255	.1073-1	.7413-2	.6223-2	.6578-2	.8616-2	.1322-1	.1394-1	.1050-1	.5428-2
255 / 240	.8968-2	.6072-2	.5052-2	.5439-2	.7424-2	.1195-1	.1324-1	.1082-1	.6264-2
240 / 225	.5137-2	.4368-2	.4022-2	.4403-2	.5603-2	.7445-2	.8653-2	.9204-2	.6036-2
225 / 210	.3391-2	.3249-2	.3250-2	.3631-2	.4435-2	.5493-2	.6651-2	.7948-2	.5554-2
210 / 195	.2850-2	.2789-2	.2875-2	.3298-2	.4106-2	.5167-2	.6363-2	.7708-2	.5431-2
195 / 180	.3229-2	.2887-2	.2863-2	.3371-2	.4541-2	.6305-2	.7680-2	.8528-2	.5758-2
180 / 165	.4780-2	.3375-2	.3050-2	.3628-2	.5445-2	.9438-2	.1124-1	.9967-2	.6202-2
165 / 150	.5135-2	.3643-2	.3264-2	.3802-2	.5594-2	.9594-2	.1140-1	.1010-1	.6291-2
150 / 135	.4215-2	.3751-2	.3572-2	.3937-2	.5004-2	.6717-2	.8076-2	.8944-2	.6035-2
135 / 120	.4886-2	.4573-2	.4301-2	.4401-2	.4986-2	.5923-2	.7080-2	.8467-2	.5934-2
120 / 105	.7465-2	.6658-2	.5811-2	.5524-2	.5926-2	.6784-2	.7858-2	.9176-2	.6342-2
105 / 90	.1364-1	.1068-1	.8322-2	.7416-2	.7994-2	.9684-2	.1071-1	.1106-1	.7146-2
90 / 75	.2718-1	.1700-1	.1170-1	.9875-2	.1101-1	.1597-2	.1684-1	.1337-1	.7600-2
75 / 60	.3789-1	.2343-1	.1569-1	.1270-1	.1331-1	.1811-2	.1807-1	.1320-1	.6693-2
60 / 45	.4291-1	.2892-1	.2005-1	.1581-1	.1474-1	.1524-1	.1386-1	.1062-1	
45 / 30	.2942-1	.2058-1	.1451-1	.1130-1	.9917-2	.9025-2	.7293-2		

0 15 30 45 60 75 90 105 120 135

X Axis (cm)

Figure C.3 Power Distribution at T = .01 for Test Problem 5.5: Reference Case

Y Axis (cm)	0	15	30	45	60	75	90	105	120	135
300	.6104-2	.5732-2	.5466-2	.5641-2	.6169-2	.6425-2	.5580-2			
285	.9806-2	.8487-2	.7761-2	.8084-2	.9425-2	.1113-1	.1081-1	.8601-2		
270	.1106-1	.7645-2	.6421-2	.6789-2	.8896-2	.1366-1	.1441-1	.1086-1	.5623-2	
255	.9298-2	.6303-2	.5251-2	.5653-2	.7711-2	.1241-1	.1375-1	.1126-1	.6527-2	
240	.5404-2	.4605-2	.4246-2	.4643-2	.5894-2	.7817-2	.9091-2	.9689-2	.6364-2	
225	.3704-2	.3547-2	.3539-2	.3935-2	.4787-2	.5914-2	.7155-2	.8553-2	.5978-2	
210	.3332-2	.3233-2	.3289-2	.3727-2	.4605-2	.5771-2	.7081-2	.8554-2	.6017-2	
195	.4100-2	.3599-2	.3475-2	.3993-2	.5301-2	.7312-2	.8853-2	.9773-2	.6575-2	
180	.6474-2	.4497-2	.3929-2	.4501-2	.6573-2	.1125-1	.1331-1	.1174-1	.7283-2	
165	.7356-2	.5169-2	.4480-2	.4968-2	.7003-2	.1174-1	.1382-1	.1222-1	.7603-2	
150	.6522-2	.5743-2	.5280-2	.5503-2	.6627-2	.8590-2	.1018-1	.1123-1	.7571-2	
135	.8241-2	.7584-2	.6860-2	.6630-2	.7096-2	.8094-2	.9454-2	.1119-1	.7810-2	
120	.1333-1	.1169-1	.9850-2	.8840-2	.8966-2	.9847-2	.1110-1	.1276-1	.8753-2	
105	.2509-1	.1938-1	.1460-1	.1232-1	.1255-1	.1461-1	.1575-1	.1599-1	.1022-1	
90	.5068-1	.3134-1	.2093-1	.1678-1	.1762-1	.2455-1	.2530-1	.1976-1	.1112-1	
75	.7109-1	.4354-1	.2841-1	.2191-1	.2164-1	.2822-1	.2750-1	.1979-1	.9926-2	
60	.8074-1	.5394-1	.3651-1	.2756-1	.2432-1	.2408-1	.2131-1	.1608-1		
45	.5549-1	.3849-1	.2651-1	.1981-1	.1652-1	.1440-1	.1133-1			
30										

X Axis (cm)

Figure C. 4 Power Distribution at T = .015 for Test Problem 5. 5: Reference Case

Y Axis (cm)

300	.6431-2	.6041-2	.5765-2	.5957-2	.6525-2	.6805-2	.5916-2		
285	.1035-1	.8959-2	.8200-2	.8552-2	.9987-2	.1181-1	.1149-1	.9149-2	
270	.1171-1	.8099-2	.6810-2	.7211-2	.9463-2	.1455-1	.1536-1	.1160-2	.6014-2
255	.9910-2	.6731-2	.5619-2	.6057-2	.8263-2	.1130-1	.1476-1	.1211-1	.7034-2
240	.5858-2	.5008-2	.4628-2	.5062-2	.6414-2	.8496-2	.9893-2	.1057-1	.6960-2
225	.4189-2	.4008-2	.3994-2	.4428-2	.5368-2	.6620-2	.8010-2	.9581-2	.6702-2
210	.4024-2	.3877-2	.3902-2	.4381-2	.5383-2	.6726-2	.8232-2	.9919-2	.6966-2
195	.5307-2	.4597-2	.4350-2	.4909-2	.6448-2	.8853-2	.1067-1	.1172-1	.7858-2
180	.8780-2	.6035-2	.5156-2	.5753-2	.8240-2	.1397-1	.1644-1	.1444-1	.8939-2
165	.1034-1	.7229-2	.6144-2	.6604-2	.9042-2	.1491-1	.1743-1	.1539-1	.9572-2
150	.9583-2	.8396-2	.7578-2	.7654-2	.8912-2	.1129-1	.1324-1	.1458-1	.9826-2
135	.1264-1	.1155-1	.1026-1	.9636-2	.9999-2	.1113-1	.1282-1	.1508-1	.1050-1
120	.2098-1	.1828-1	.1512-1	.1326-1	.1309-1	.1407-1	.1562-1	.1780-1	.1215-1
105	.3999-1	.3071-1	.2282-1	.1883-1	.1868-1	.2134-1	.2271-1	.2283-1	.1452-1
90	.8121-1	.5000-1	.3301-1	.2590-1	.2649-1	.3623-1	.3693-1	.2860-1	.1600-1
75	.1142-0	.6969-1	.4501-1	.3406-1	.3279-1	.4195-1	.4041-1	.2885-1	.1440-1
60	.1298-0	.8646-1	.5799-1	.4302-1	.3711-1	.3602-1	.3148-1	.2357-1	
45	.8931-1	.6175-1	.4217-1	.3101-1	.2530-1	.2166-1	.1682-1		
30									

0 15 30 45 60 75 90 105 120 135

X Axis (cm)

Figure C.5 Power Distribution at T = .02 for Test Problem 5. 5: Reference Case

Y Axis (cm) \ X Axis (cm)	0–15	15–30	30–45	45–60	60–75	75–90	90–105	105–120	120–135
285–300	.6921-2	.6504-2	.6213-2	.6431-2	.7057-2	.7371-2	.6417-2		
270–285	.1116-1	.9662-2	.8853-2	.9250-2	.1082-1	.1282-1	.1249-1	.9960-2	
255–270	.1267-1	.8771-2	.7385-2	.7834-2	.1030-1	.1586-1	.1677-1	.1268-1	.6588-2
240–255	.1080-1	.7353-2	.6155-2	.6644-2	.9067-2	.1459-1	.1622-1	.1335-1	.7769-2
225–240	.6505-2	.5579-2	.5172-2	.5658-2	.7157-2	.9466-2	.1104-1	.1184-1	.7087-2
210–225	.4849-2	.4645-2	.4622-2	.5110-2	.6177-2	.7605-2	.9203-2	.1102-1	.7712-2
195–210	.4956-2	.4747-2	.4734-2	.5271-2	.6446-2	.8034-2	.9809-2	.1179-1	.8272-2
180–195	.6913-2	.5927-2	.5522-2	.6141-2	.7998-2	.1094-1	.1313-1	.1435-1	.9603-2
165–180	.1183-1	.8075-2	.6789-2	.7428-2	.1048-1	.1763-1	.2066-1	.1809-1	.1118-1
150–165	.1429-1	.9952-2	.8347-2	.8781-2	.1177-1	.1915-1	.2229-1	.1966-1	.1222-1
135–150	.1361-1	.1189-1	.1061-1	.1050-1	.1195-1	.1488-1	.1733-1	.1905-1	.1284-1
120–135	.1842-1	.1676-1	.1473-1	.1360-1	.1384-1	.1517-1	.1730-1	.2026-1	.1408-1
105–120	.3101-1	.2691-1	.2208-1	.1908-1	.1852-1	.1964-1	.2161-1	.2447-1	.1666-1
90–105	.5950-1	.4556-1	.3361-1	.2738-1	.2675-1	.3022-1	.3190-1	.3187-1	.2020-1
75–90	.1212-0	.7446-1	.4886-1	.3789-1	.3817-1	.5164-1	.5227-1	.4027-1	.2244-1
60–75	.1706-0	.1039-0	.6678-1	.5001-1	.4746-1	.6003-1	.5744-1	.4082-1	.2030-1
45–60	.1942-0	.1291-0	.8616-1	.6332-1	.5391-1	.5175-1	.4489-1	.3345-1	
30–45	.1336-0	.9222-1	.6270-1	.4570-1	.3685-1	.3120-1	.2405-1		

Figure C.6 Percent Errors for 15 cm 2DTD Solution at T = 0.0 for Test Problems 5.5 and 5.6

-1.9	-.9	0.0	.6	1.5	1.7	1.3		
-1.2	-.9	-.9	0.0	1.2	1.5	2.2	.2	
-.1	-1.4	-1.6	-1.0	.2	2.1	2.6	2.2	1.3
-.2	-1.4	-1.7	-1.2	0.0	1.7	2.1	1.5	1.7
-1.7	-.8	-.7	-.3	.2	0.0	.2	1.2	1.5
-2.5	-1.2	-.2	0.0	-.3	-1.2	-1.0	0.0	.6
-2.7	-1.5	0.0	-.2	-.7	-1.7	-1.6	-.9	0.0
-2.4	-1.7	-1.5	-1.2	-.8	-1.4	-1.4	-.9	-.9
-1.1	-2.4	-2.7	-2.5	-1.7	-.2	-.2	-.1	-1.9
-1.1	-2.4	-2.7	-2.5	-1.7	-.2	-.2	-.1	-1.9
-2.4	-1.7	-1.5	-1.2	-.8	-1.4	-1.4	-.9	-.9
-2.7	-1.5	0.0	-.2	-.7	-1.7	-1.6	-.9	0.0
-2.5	-1.2	-.2	0.0	-.3	-1.2	-1.0	0.0	.6
-1.7	-.8	-.7	-.3	.2	0.0	.2	1.2	1.5
-.2	-1.4	-1.7	-1.2	0.0	1.7	2.1	1.5	1.7
-.1	-1.4	-1.6	-1.0	.2	2.1	2.6	2.2	1.3
-1.2	-.9	-.9	0.0	1.2	1.5	2.2	.2	
-1.9	-.9	0.0	.6	1.5	1.7	1.3		

Y Axis (cm): 300 285 270 255 240 225 210 195 180 165 150 135 120 105 90 75 60 45 30

X Axis (cm): 0 15 30 45 60 75 90 105 120 135

Figure C.7 Percent Errors for 15 cm 2DTD Solution at T = .01 for Test Problem 5.5

-2.0	-1.0	- .4	.5	1.4	1.6	1.2		
-1.4	-1.1	-1.0	-.2	1.1	1.3	2.0	2.4	
-.3	-1.6	-1.8	-1.1	.1	2.0	2.4	2.0	1.1
-.3	-1.5	-1.8	-1.4	-.2	1.5	2.0	1.3	1.5
-1.9	-1.0	-.9	-.5	.1	-.3	0	.9	1.2
-2.7	-1.4	-.4	-.1	-.5	-1.4	-1.3	-.4	.3
-2.9	-1.6	-.7	-.4	-.9	-1.9	-1.9	-1.2	-.6
-2.3	-1.6	-1.5	-1.2	-.9	-1.6	-1.6	-1.2	-1.1
-.8	-2.1	-2.5	-2.4	-1.7	-.3	-.3	-1.5	-2.1
-.7	-1.9	-2.1	-2.0	-1.5	-.3	-.3	-1.4	-2.1
-1.6	-.8	-.5	-.3	-.5	-1.3	-1.5	-1.2	-1.1
-1.4	-.2	.8	.8	-.1	-1.4	-1.6	-1.1	-.4
-.8	.8	1.6	1.3	.3	-1.1	-1.2	-.2	.5
.6	1.7	1.5	.8	.4	-.3	-.1	.8	1.2
2.5	1.6	.7	-.2	-.2	1.0	1.3	.9	1.0
3.3	1.9	.5	-.5	-.4	1.1	1.5	1.1	.4
3.5	2.4	.7	-.2	0	.1	.8	1.2	
3.1	2.4	1.2	.4	.1	.1	-.2		

Y Axis (cm): 300 285 270 255 240 225 210 195 180 165 150 135 120 105 90 75 60 45 30

X Axis (cm): 0 15 30 45 60 75 90 105 120 135

Figure C.8 Percent Errors for 15 cm 2DTD Solution at T = .02 for Test Problem 5.5

Y Axis (cm) \ X Axis (cm)	0–15	15–30	30–45	45–60	60–75	75–90	90–105	105–120	120–135
285–300	-2.1	-1.2	-.5	.3	1.1	1.3	.8		
270–285	-1.5	-1.2	-1.2	-.4	.8	1.1	1.7	2.0	
255–270	-.3	-1.7	-1.9	-1.3	-.1	1.6	2.0	1.6	.7
240–255	-.4	-1.6	-1.9	-1.5	-.4	1.2	1.5	.9	1.1
225–240	-1.8	-.9	-.8	-.4	0	-.4	-.3	.5	.8
210–225	-2.2	-.9	0	.2	-.3	-1.3	-1.3	-.5	.2
195–210	-1.9	-.5	.2	.4	-.3	-1.4	-1.5	-.8	-.2
180–195	-.7	0	-.1	0	.1	-.6	-.8	-.3	-.4
165–180	1.2	-.2	-.7	-.8	-.4	.9	.8	-.4	-1.2
150–165	1.4	.3	0	-.1	0	1.1	1.0	.3	-.9
135–150	.7	1.5	1.9	1.8	1.3	.3	-.1	.3	.3
120–135	1.2	2.4	3.4	3.2	1.9	.5	.1	.6	1.3
105–120	1.9	3.6	4.3	3.8	2.5	.9	.7	1.7	2.4
90–105	3.3	4.5	4.2	3.4	2.7	1.6	1.7	2.8	3.0
75–90	5.4	4.4	3.4	2.3	1.6	3.0	3.2	2.7	2.9
60–75	6.3	4.7	3.2	2.1	1.8	3.0	3.3	2.9	2.1
45–60	6.4	5.1	3.6	2.2	2.1	2.0	2.6	3.0	
30–45	6.0	5.3	3.9	2.8	2.3	2.1	1.6		

Figure C.9 Power Distribution at T = .2 for Test Problem 5.6: Reference Case

Y Axis (cm)	0–15	15–30	30–45	45–60	60–75	75–90	90–105	105–120	120–135
285–300	.5393-2	.5052-2	.4791-2	.4908-2	.5328-2	.5516-2	.4768-2		
270–285	.8641-2	.7461-2	.6782-2	.7006-2	.8099-2	.9498-2	.9186-2	.7275-2	
255–270	.9693-2	.6679-2	.5568-2	.5829-2	.7568-2	.1154-1	.1212-1	.9086-2	.4676-2
240–255	.8504-2	.5428-2	.4471-2	.4759-2	.6438-2	.1031-1	.1135-1	.9218-2	.5302-2
225–240	.4541-2	.3831-2	.3480-2	.3757-2	.4732-2	.6247-2	.7193-2	.7568-2	.4927-2
210–225	.2877-2	.2733-2	.2692-2	.2952-2	.3544-2	.4322-2	.5153-2	.6084-2	.4220-2
195–210	.2240-2	.2183-2	.2221-2	.2488-2	.3006-2	.3671-2	.4403-2	.5245-2	.3663-2
180–195	.2296-2	.2062-2	.2033-2	.2336-2	.3024-2	.4016-2	.4660-2	.4943-2	.3262-2
165–180	.3112-2	.2203-2	.1992-2	.2336-2	.3393-2	.5634-2	.6250-2	.4750-2	.2779-2
150–165	.3049-2	.2155-2	.1951-2	.2299-2	.3357-2	.5589-2	.6201-2	.4706-2	.2750-2
135–150	.2124-2	.1909-2	.1900-2	.2217-2	.2913-2	.3899-2	.4533-2	.4800-2	.3162-2
120–135	.1902-2	.1879-2	.1962-2	.2263-2	.2798-2	.3460-2	.4175-2	.4983-2	.3481-2
105–120	.2231-2	.2179-2	.2247-2	.2580-2	.3198-2	.3968-2	.4776-2	.5667-2	.3939-2
90–105	.3256-2	.2852-2	.2767-2	.3186-2	.4184-2	.5637-2	.6561-2	.6950-2	.4540-2
75–90	.5422-2	.3813-2	.3417-2	.3945-2	.5628-2	.9226-2	.1027-1	.8391-2	.4845-2
60–75	.6020-2	.4436-2	.4121-2	.4744-2	.6544-2	.1026-1	.1089-1	.8226-2	.4251-2
45–60	.4573-2	.4703-2	.4920-2	.5628-2	.6927-2	.8379-2	.8221-2	.6560-2	
30–45	.2676-2	.3102-2	.3437-2	.3911-2	.4523-2	.4837-2	.4247-2		

X Axis (cm)

Figure C.10 Power Distribution at T = .3 for Test Problem 5.6: Reference Case

.5364-2	.5024-2	.4764-2	.4879-2	.5296-2	.5481-2	.4738-2		
.8594-2	.7420-2	.6744-2	.6964-2	.8049-2	.9439-2	.9128-2	.7229-2	
.9641-2	.6642-2	.5536-2	.5794-2	.7521-2	.1148-1	.1204-1	.9027-2	.4646-2
.8009-2	.5397-2	.4446-2	.4729-2	.6397-2	.1024-1	.1128-1	.9157-2	.5266-2
.4514-2	.3809-2	.3459-2	.3733-2	.4701-2	.6204-2	.7144-2	.7514-2	.4891-2
.2859-2	.2716-2	.2674-2	.2931-2	.3519-2	.4290-2	.5114-2	.6038-2	.4187-2
.2224-2	.2168-2	.2204-2	.2468-2	.2981-2	.3640-2	.4366-2	.5200-2	.3631-2
.2277-2	.2044-2	.2016-2	.2314-2	.2997-2	.3977-2	.4614-2	.4894-2	.3230-2
.3084-2	.2184-2	.1973-2	.2313-2	.3360-2	.5574-2	.6182-2	.4699-2	.2749-2
.3020-2	.2134-2	.1932-2	.2276-2	.3320-2	.5529-2	.6132-2	.4653-2	.2720-2
.2103-2	.1890-2	.1880-2	.2194-2	.2882-2	.3858-2	.4484-2	.4747-2	.3128-2
.1882-2	.1860-2	.1942-2	.2240-2	.2768-2	.3424-2	.4131-2	.4930-2	.3444-2
.2207-2	.2154-2	.2223-2	.2553-2	.3164-2	.3928-2	.4728-2	.5609-2	.3899-2
.3219-2	.2820-2	.2737-2	.3152-2	.4142-2	.5581-2	.6496-2	.6880-2	.4494-2
.5358-2	.3770-2	.3379-2	.3904-2	.5571-2	.9133-2	.1017-1	.8308-2	.4798-2
.5948-2	.4384-2	.4076-2	.4696-2	.6479-2	.1016-1	.1079-1	.8144-2	.4209-2
.4518-2	.4649-2	.4866-2	.5569-2	.6857-2	.8296-2	.8140-2	.6496-2	
.2643-2	.3066-2	.3400-2	.3870-2	.4477-2	.4789-2	.4206-2		

Y Axis (cm)

300 285 270 255 240 225 210 195 180 165 150 135 120 105 90 75 60 45 30

0 15 30 45 60 75 90 105 120 135

X Axis (cm)

Figure C.11 Power Distribution at T = .5 for Test Problem 5.6: Reference Case

Y Axis (cm) \ X Axis (cm)	0–15	15–30	30–45	45–60	60–75	75–90	90–105	105–120	120–135
285–300	.5332-2	.4993-2	.4734-2	.4848-2	.5260-2	.5444-2	.4706-2		
270–285	.8541-2	.7374-2	.6702-2	.6919-2	.7995-2	.9373-2	.9064-2	.7179-2	
255–270	.9581-2	.6601-2	.5500-2	.5755-2	.7470-2	.1139-1	.1196-1	.8962-2	.4612-2
240–255	.7959-2	.5362-2	.4417-2	.4697-2	.6352-2	.1017-1	.1120-1	.9090-2	.5227-2
225–240	.4486-2	.3783-2	.3435-2	.3707-2	.4667-2	.6158-2	.7090-2	.7457-2	.4853-2
210–225	.2839-2	.2696-2	.2654-2	.2909-2	.3490-2	.4254-2	.5071-2	.5986-2	.4151-2
195–210	.2207-2	.2150-2	.2187-2	.2447-2	.2953-2	.3605-2	.4323-2	.5150-2	.3596-2
180–195	.2256-2	.2027-2	.1997-2	.2292-2	.2965-2	.3934-2	.4564-2	.4841-2	.3194-2
165–180	.3054-2	.2162-2	.1954-2	.2289-2	.3322-2	.5511-2	.6110-2	.4643-2	.2717-2
150–165	.2990-2	.2112-2	.1911-2	.2250-2	.3282-2	.5511-2	.6059-2	.4597-2	.2687-2
135–150	.2081-2	.1869-2	.1859-2	.2169-2	.2848-2	.3811-2	.4430-2	.4689-2	.3090-2
120–135	.1860-2	.1838-2	.1920-2	.2213-2	.2735-2	.3384-2	.4082-2	.4871-2	.3403-2
105–120	.2179-2	.2129-2	.2197-2	.2523-2	.3127-2	.3882-2	.4673-2	.5543-2	.3854-2
90–105	.3177-2	.2784-2	.2702-2	.3114-2	.4094-2	.5518-2	.6423-2	.6802-2	.4443-2
75–90	.5286-2	.3721-2	.3337-2	.3859-2	.5508-2	.9031-2	.1005-2	.8215-2	.4744-2
60–75	.5862-2	.4327-2	.4024-2	.4640-2	.6404-2	.1004-1	.1067-1	.8054-2	.4162-2
45–60	.4457-2	.4587-2	.4804-2	.5502-2	.6778-2	.8202-2	.8050-2	.6423-2	
30–45	.2607-2	.3026-2	.3358-2	.3824-2	.4426-2	.4734-2	.4159-2		

Figure C.12 Percent Errors for 15 cm 2DTD Solution at T = .2 for Test Problem 5:6

Y Axis (cm)

300	-1.3	-.4	0	1.3	2.2	2.6	1.9		
285	-.7	-.4	-.3	.6	2.0	2.4	3.1	3.6	
270	.4	-.9	-1.1	-.3	1.1	3.1	3.6	3.3	2.5
255	.4	-.9	-1.1	-.5	.8	2.7	3.3	2.7	3.0
240	-1.2	-.3	-.1	.4	1.2	.9	1.4	2.5	2.8
225	-2.0	-.5	.4	.8	.6	-.3	.1	1.3	2.1
210	-2.1	-.8	.2	.5	0	-.9	-.6	.4	1.1
195	-1.7	-.9	-.7	-.3	0	-.6	-.7	0	0
180	-.2	-1.5	-1.8	-1.6	-.7	.9	.8	-1.1	-1.8
165	-.1	-1.4	-1.8	-1.6	-.7	.9	.8	-1.1	-1.8
150	-1.4	-.7	-.5	-.2	.1	-.5	-.6	.1	.1
135	-1.7	-.5	.5	.8	.3	-.7	-.5	.6	-1.3
120	-1.5	-.1	.8	1.2	.9	.1	.4	1.6	2.4
105	-.6	.2	.3	.9	1.7	1.4	1.8	2.9	3.3
90	1.1	-.4	-.6	.1	1.4	3.3	3.8	3.3	3.5
75	.9	-.5	-.4	.5	1.8	3.8	4.4	3.9	3.1
60	-.9	4.2	.6	1.6	3.0	3.3	3.9	4.3	
45	-1.7	.3	1.4	2.4	3.3	3.5	3.1		

30

0 15 30 45 60 75 90 105 120 135

X Axis (cm)

Figure C.13 Percent Errors for 15 cm 2DTD Solution at T = .5 for Test Problem 5.6

Y Axis (cm)										
300	-1.4	-.4	.3	1.2	2.2	2.5	2.2			
285	-.8	-.4	-.4	.5	1.9	2.4	3.1	3.5		
270	.4	-.9	-1.1	-.4	1.0	3.0	3.5	3.2	2.4	
255	.3	-.9	-1.2	-.6	.7	2.6	3.1	2.7	2.9	
240	-1.2	-.3	-.2	.3	1.1	.9	1.3	2.4	2.8	
225	-2.1	-.6	.3	.7	.5	-.3	.1	1.2	2.0	
210	-2.2	-2.6	0	.4	0	-.9	-.7	.3	1.0	
195	-1.7	-1.0	-.7	-.4	-.1	-.7	-.8	-.2	-.1	
180	-.3	-1.6	-1.9	-1.7	-.8	.7	.6	-1.3	-2.0	
165	-.3	-1.5	-1.9	-1.6	-.8	-.1	.6	-1.3	-2.0	
150	-1.5	-.8	-.6	-.3	0	-.6	-.7	.1	-.1	
135	-1.7	-.6	.3	.7	.1	-.8	-.5	.5	1.2	
120	-1.6	-.3	.6	1.1	.8	0	.3	1.5	2.3	
105	-.7	0	.2	.9	1.6	1.3	1.7	2.8	3.2	
90	.9	-.6	-.7	0	1.3	3.2	3.7	3.1	3.4	
75	.8	-.6	-.5	.4	1.7	3.7	4.1	3.7	2.9	
60	-1.2	0	.5	1.6	2.9	3.1	3.7	4.1		
45	-1.9	0	1.2	2.3	3.2	3.4	2.9			
30	0	15	30	45	60	75	90	105	120	135

X Axis (cm)

Figure C.14 Region Powers for Test Problem 5.7

T=0.0
T=1.405

.9060-6 .3103+4	.8592-6 .3137+4	.8248-6 .3420+4	.8589-6 .4168+4	.9472-6 .5303+4	.9885-6 .6227+4	.8546-6 .5971+4		
.1462-5 .4970+4	.1269-5 .4599+4	.1162-5 .4811+4	.1220-5 .5970+4	.1441-5 .8206+4	.1706-5 .1103+5	.1667-5 .1231+5	.1330-5 .1143+5	
.1662-5 .5624+4	.1133-5 .4081+4	.9496-6 .3946+4	.1011-5 .5012+4	.1342-5 .7823+4	.2099-5 .1390+5	.2223-5 .1721+5	.1667-5 .1708+5	.8546-6 .1020+5
.1385-5 .4695+4	.9261-6 .3339+4	.7683-6 .3204+4	.8321-6 .4150+4	.1151-5 .6745+4	.1887-5 .1254+5	.2099-5 .1625+5	.1706-5 .1726+5	.9885-6 .1125+5
.7761-6 .2655+4	.6652-6 .2423+4	.6134-6 .2552+4	.6761-6 .3314+4	.8676-6 .4922+4	.1151-5 .7356+4	.1342-5 .9643+4	.1441-5 .1153+5	.9472-6 .8065+4
.4981-6 .1718+4	.4847-6 .1772+4	.4911-6 .2006+4	.5531-6 .2578+4	.6761-6 .3536+4	.8321-6 .4770+4	.1011-5 .6226+4	.1220-5 .7861+4	.8589-6 .5697+4
.4011-6 .1371+4	.4005-6 .1439+4	.4218-6 .1656+4	.4911-6 .2119+4	.6134-6 .2852+4	.7683-6 .3771+4	.9496-6 .4854+4	.1162-5 .6093+4	.8248-6 .4407+4
.4290-6 .1431+4	.3927-6 .1366+4	.4005-6 .1500+4	.4847-6 .1947+4	.6652-6 .2787+4	.9260-6 .3967+4	.1133-5 .4958+4	.1269-5 .5673+4	.8592-6 .3907+4
.6058-6 .1985+4	.4290-6 .1458+4	.4011-6 .1460+4	.4981-6 .1924+4	.7761-6 .3095+4	.1385-5 .5595+4	.1662-5 .6790+4	.1462-5 .6057+4	.9060-6 .3807+4

Y Axis (cm): 135 120 105 90 75 60 45 30 15 0

0 15 30 45 60 75 90 105 120 135

X Axis (cm)

Figure C.15 Region Powers for Test Problem 5.7

T=2.0
T=3.0

Y Axis (cm)

135	.3685+3 .4792+2	.3824+3 .4891+2	.4369+3 .5425+2	.5613+3 .6757+2	.7495+3 .8816+2	.9217+3 .1069+3	.9266+3 .1066+3		
120	.5877+3 .7649+2	.5582+3 .7146+2	.6136+3 .7611+2	.8051+3 .9659+2	.1165+4 .1364+3	.1646+4 .1899+3	.1960+4 .2241+3	.1962+4 .2236+3	
105	.6648+3 .8672+2	.4949+3 .6346+2	.5050+3 .6258+2	.6804+3 .8139+2	.1121+4 .1307+3	.2099+4 .2414+3	.2804+4 .3197+3	.3178+4 .3606+3	.2034+4 .2309+3
90	.5575+3 .7287+2	.4070+3 .5230+2	.4127+3 .5122+2	.5675+3 .6794+2	.9734+3 .1135+3	.1903+4 .2190+3	.2657+4 .3031+3	.3202+4 .3635+3	.2207+4 .2507+3
75	.3189+3 .4174+2	.2987+3 .3848+2	.3308+3 .4127+2	.4538+3 .5471+2	.7075+3 .8319+2	.1109+4 .1283+3	.1538+4 .1764+3	.1942+4 .2220+3	.1408+4 .1610+3
60	.2095+3 .2760+2	.2211+3 .2866+2	.2605+3 .3283+2	.3495+3 .4274+2	.4984+3 .5956+2	.6970+3 .8203+2	.9408+3 .1097+3	.1217+4 .1415+3	.8985+3 .1045+3
45	.1690+3 .2254+2	.1804+3 .2370+2	.2135+3 .2734+2	.2811+3 .3508+2	.3871+3 .4738+2	.5222+3 .6308+2	.6840+3 .8196+2	.8700+3 .1039+3	.6368+3 .7598+2
30	.1769+3 .2394+2	.1709+3 .2279+2	.1911+3 .2486+2	.2518+3 .3200+2	.3630+3 .4545+2	.5188+3 .6437+2	.6531+3 .8051+2	.7548+3 .9269+2	.5253+3 .6437+2
15	.2451+3 .3348+2	.1816+3 .2445+2	.1843+3 .2420+2	.2449+3 .3144+2	.3940+3 .4987+2	.7111+3 .8944+2	.8649+3 .1083+3	.7764+3 .9680+2	.4922+3 .6127+2

0

0 15 30 45 60 75 90 105 120 135

X Axis (cm)

Figure C.16 Average Temperatures for Test Problem 5.7

T=2.0
T=3.0

Y Axis (cm) \ X Axis (cm)	0–15	15–30	30–45	45–60	60–75	75–90	90–105	105–120	120–135
120–135	606 731	612 740	645 789	727 910	851 1093	957 1253	940 1237		
105–120	790 989	757 944	785 987	912 1174	1154 1530	1466 1993	1630 2256	1568 2196	
90–105	855 1080	705 872	698 865	814 1036	1116 1477	1776 2448	2174 3068	2249 3262	1494 2143
75–90	764 953	632 769	624 760	727 911	1006 1318	1634 2242	2071 2918	2267 3286	1609 2313
60–75	563 671	542 643	558 668	641 789	814 1043	1080 1436	1342 1833	1570 2189	1199 1648
45–60	471 542	478 552	503 590	565 679	667 829	801 1026	961 1263	1141 1531	913 1202
30–45	437 495	445 506	468 539	516 609	593 720	689 860	804 1026	935 1218	761 968
15–30	443 504	437 496	451 516	497 581	583 703	703 874	804 1019	879 1127	699 872
0–15	499 584	446 509	447 510	494 576	612 743	863 1100	984 1271	911 1168	685 847