A Transient, Quadratic Nodal Method for Triangular-Z Geometry

by

Thomas F. DeLorey

M.S., Energy Engineering (Nuclear Option), University of Lowell
(1990)

B.S., Nuclear Engineering, University of Lowell
(1988)

Submitted to the Department of Nuclear Engineering
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1993

Author Thomas F. D
Department of Nuclear Engineering
May 12, 1993

Certified by
Allan F. Henry
Professor, Department of Nuclear Engineering
Thesis Supervisor

Accepted by
Allan F. Henry
Chairman, Department Committee on Graduate Students

A Transient, Quadratic Nodal Method for Triangular-Z Geometry

by

Thomas F. DeLorey

Submitted to the Department of Nuclear Engineering
on May 12, 1993, in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Abstract

Many systematically–derived nodal methods have been developed for Cartesian geometry due to the extensive interest in Light Water Reactors. These methods typically model the transverse–integrated flux as either an analytic or low order polynomial function of position within the node. Recently, quadratic nodal methods have been developed for R-Z and hexagonal geometry. In this work, a static and transient quadratic nodal method is developed for triangular-Z geometry. This development is particularly challenging because the quadratic expansion in each node must be performed between the node faces and the triangular points. As a consequence, in the 2-D plane, the flux and current at the points of the triangles must be treated.

The quadratic nodal equations are solved using a non–linear iteration scheme, which utilizes the corrected, mesh–centered finite difference equations, and forces these equations to match the quadratic equations by computing discontinuity factors during the solution. The transient nodal equations are solved using the improved quasi–static method, which has been shown to be a very efficient solution method for transient problems. A simple thermal feedback model is used to demonstrate feedback response in static and transient problems.

Several static problems are used to compare the quadratic nodal method to the Coarse Mesh Finite Difference (CMFD) method. For select cases, the quadratic method is shown to give significant benefit over the CMFD method. In all cases tested, the quadratic method is shown to give more accurate node–averaged fluxes. However, it appears that the method has difficulty predicting node leakages near reactor boundaries and severe material interfaces. The consequence is that the eigenvalue may be poorly predicted for certain reactor configurations.

The transient methods are tested using a simple analytic test problem, a heterogeneous heavy water reactor benchmark problem, and three thermal hydraulic test problems. The results indicate that the transient methods have been implemented correctly.

Thesis Supervisor: Allan F. Henry
Title: Professor, Department of Nuclear Engineering

Acknowledgments

I would like to thank my advisor, Professor Allan F. Henry, for his patience, his insight, and his sense of humor. It has truly been a pleasure and an honor working with Professor Henry throughout my stay at M.I.T. Thanks also to my Thesis Reader, Professor David D. Lanning. Professor Lanning's continuing interest in my work has always been encouraging.

There have been many students which I have had the pleasure of knowing during my studies at M.I.T. Thanks, of course, to Jonathan Witter (Mmmmmm, Dooooonuuuuts), Santiago Parra, Mark Byers, Jess Gehin, and the whole third floor NW-12 crew. I wish them all success in the future.

My family has always supported me throughout my education. I must extend my greatest thanks to my wife Karen, who has sacrificed and worked so hard over the years. Thanks also to my parents, Claire and Len DeLorey, my sister Karen, my brother Kevin and his wife Brenda. Thank you to Nancy and Roy Robichaud and family. Thank you to Phil Delmolino, for being a good friend through the years.

Finally, thank you to the newest, young nuclear engineer in the DeLorey Family, my godson Ryan!

This research was performed under appointment to the Nuclear Engineering & Health Physics Fellowship Program administered by the Oak Ridge Institute for Science and Education for the U.S. Department of Energy.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Table of Contents

List of Figures

List of Tables

Number **Title** **Page**

Chapter 1

Introduction

1.1 Motivation and Background

Modern, systematically derived nodal methods have been used for the efficient and accurate modeling of nuclear reactor cores for over fifteen years. The focus of research in this area has primarily been directed towards the development of tools for the analysis of Light Water Reactors (LWRs). Typical LWRs have Cartesian geometry, and thus many different static and transient nodal methods have emerged suited to the study of reactors with Cartesian geometry. Recently, a quadratic nodal method was developed and implemented for R-Z geometry [B-1], which can be used for High Temperature Gas Reactors (HTGR). A quadratic method was also developed for hexagonal geometry [S-3], which is ideally suited to many liquid metal fast breeder reactors, modular high temperature gas cooled reactors, and heavy water reactors. Recent interest in space reactors [M-1,P-3] has produced new designs in hexagonal geometry such as the Particle Bed Reactor, and re-produced old designs such as the NERVA [A-1] rocket. These designs are small, highly coupled cores in which diffusion theory is not generally valid.

The research reactor at the Massachusetts Institute of Technology, called MITR-II, is a unique design which is ideally suited for a triangular nodal mesh [R-1]. This reactor has been modeled in the past by the finite difference diffusion theory code CITATION [F-1], and also the Monte Carlo code MCNP [B-3]. Figure 1.1 shows the geometry of the MITR-II reactor. Recent interest in upgrading the MITR-II has created interest in a triangular nodal code for static and transient calculations.

Modern nodal methods have demonstrated that the accuracy of fine mesh finite difference methods can be achieved at much lower computational cost. A reduction in the number of unknowns is achieved by using relatively large, homogeneous nodes. In order to retain the fine-mesh, heterogeneous accuracy, correction factors are introduced which force the nodal solution to match the results of independent, detailed, fine mesh calculations. It is desirable to restrict the independent, detailed calculations to a few, representative assemblies (if the nodes are assembly-sized) with a zero current (or, if known, an albedo) boundary condition. The goal is to obtain acceptable correction factors for every node in the nodal model without having to model the whole core in detailed calculations. The independent, detailed calculations are ideally transport models with a fine mesh, and heterogeneity. Thus, the correction factors correct for heterogeneity, the use of a finite difference approximation with large node sizes, and transport effects. Clearly, it is difficult to adequately correct for these three factors with "one degree of freedom" if one or more of the approximations is severe. Nodal methods which model the flux as a low-order polynomial or an analytic function within the node somewhat "relieve" the correction factors from the need to correct for the large node sizes by replacing the finite difference expression by a higher order one. Another way to "relieve" the correction factors from the finite difference approximation is to decrease the node size. In LWRs, a strong desire exists to maintain assembly-sized nodes, so polynomial representations as high as fourth order have been developed [Z-1]. Nodal methods in hexagonal geometry are forced to model one node per assembly because hexagons cannot be subdivided.

A motivation to produce a triangular nodal code exists for two reasons. First, the unique geometry of the MITR-II requires a triangular mesh. The extensive calculations which will be required for an upgrade of the MITR-II should be performed using state-of-the-art static and transient solution methods. Second, hexagonal reactors with severe heterogeneity or transport effects may not be adequately modeled by hexagonal nodal codes which require one node per assembly. A triangular nodal code has the capability of modeling 6,24,96,384, etc. nodes per assembly, as well as some degree of heterogeneity. The ability to model the spatial mesh and heterogeneity using a triangular mesh "relieves" the correction factors from these approximations and allows transport effects to be corrected. A triangular nodal code will therefore be much more flexible than a hexagonal nodal code, and may be able to model cores (such as space reactor cores) which hexagonal codes cannot.

1.2 Research Objectives

The objectives of this research are to develop a quadratic nodal method in triangular-z geometry for both static and transient applications. The goal is to provide both accurate and efficient computational tools.

For the solution of static problems, a corrected, three dimensional mesh-centered finite difference model will be developed. To improve upon the spatial treatment, a higher order expression for the surface-averaged current will be derived by assuming the transverse-averaged flux has the form of a quadratic polynomial within the triangular-z node. The non-linear iteration method will be used to solve the quadratic equations. In addition, acceleration procedures will be implemented to improve convergence rates of the iterative methods.

The transient nodal equations will be cast into fully implicit form to ensure unconditional stability for all time step sizes. The delayed neutron precursor equations will be treated using a direct integration procedure. The resulting equations will be solved using the improved quasi-static method. The improved quasi-static method is a very efficient solution scheme for many types of transient problems, and has been implemented successfully in other nodal methods.

A simple thermal hydraulic feedback model will also be incorporated to verify the thermal hydraulic response of the static and transient methods.

1.3 Thesis Organization

In Chapter 2, the time dependent nodal balance equations are derived for a single, homogenized node, *l*, and energy group, g. In Chapter 3, these equations are cast into steady state form, and discretized using Fick's Law and the finite difference approximation. The Coarse Mesh Finite Difference (CMFD) equations are presented for fully internal nodes and nodes on the surface of the reactor. The quadratic equations are then derived by assuming that the transverse-averaged flux can be represented by a quadratic polynomial within each node.

Chapter 4 returns to the transient nodal equations. The time dependent CMFD and quadratic equations are derived in this chapter. Chapter 5 discusses the solution of the static and transient equations. Iterative methods are presented, along with the non-linear iteration procedure for the quadratic equations. The improved quasi-static method is presented for the solution of the transient equations. A thermal feedback method is also presented.

Chapter 6 presents test results for both the static and transient methods. Chapter 7 presents conclusions and recommendations for future research.

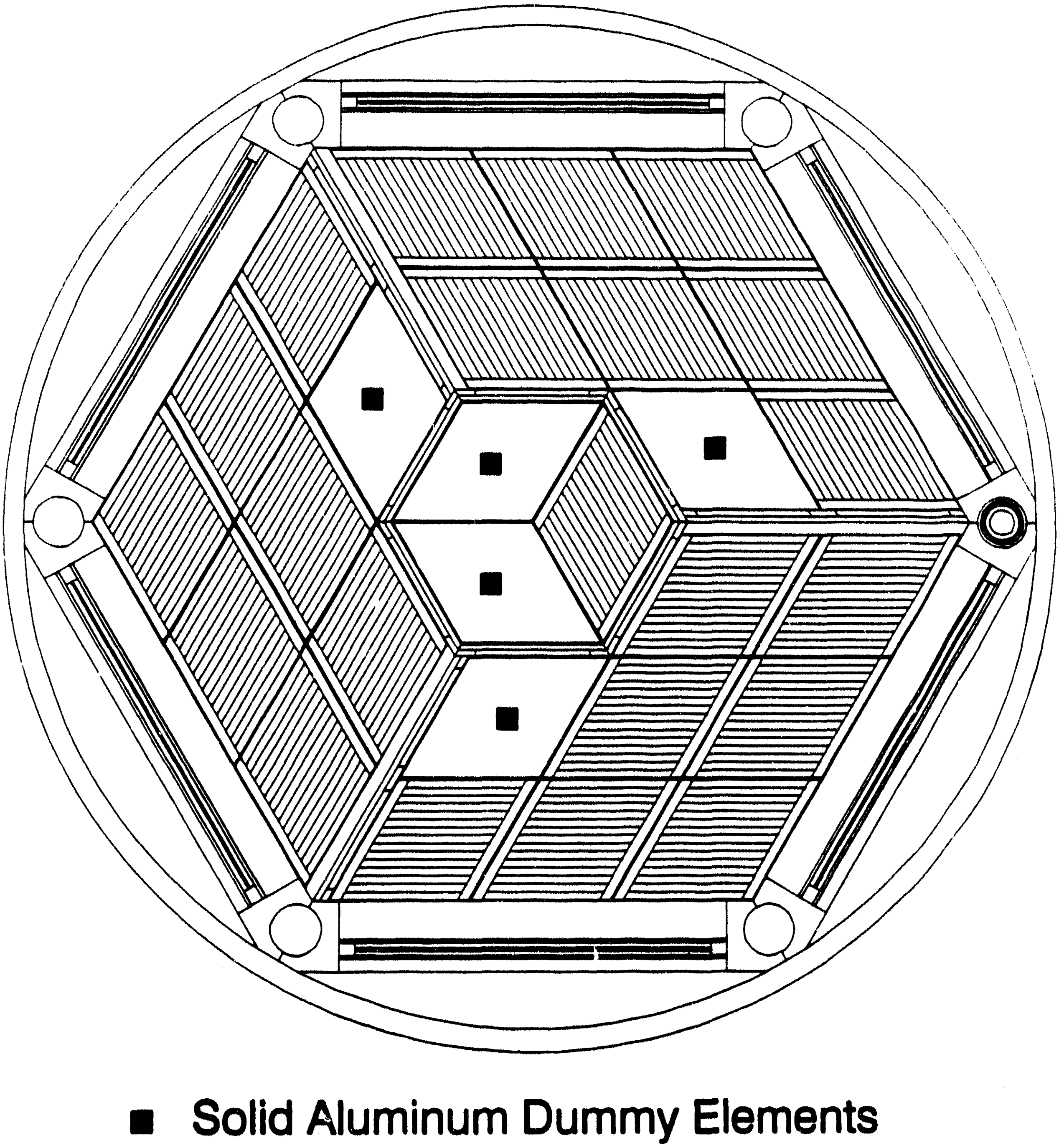


Figure 1.1. The geometry of the MITR-II Reactor (taken from Ref. [R-1]).

Chapter 2

The Time Dependent Nodal Balance Equation

2.1 Introduction

The equation that describes with adequate accuracy the space, time, energy, and directional distribution of neutrons in a reactor is the time dependent neutron transport equation [H-1]. The solution of this equation, however, is both difficult and costly in terms of computer resources. The diffusion equation is an alternative which, in a physical sense, is based on Fick's law, which states that the net neutron current is proportional to the gradient of the neutron flux. Mathematically, the diffusion equation can be derived from the P_1 form of the transport equation. However, the equation we will begin with can be derived from a balance equation performed on a control volume [H-1]. This equation is continuous in the space, energy, and time variables, and will be developed into diffusion theory with the application of Fick's law in the next chapter. The starting equation is, therefore, derivable directly from the neutron transport equation.

This chapter will develop the time dependent nodal balance equation.

2.2 The Nodal Balance Equation in Time Dependent Form

The time dependent, continuous energy balance equation is given by [H-1]:

$$\begin{aligned}\frac{\partial}{\partial t}\left[\frac{1}{\mathrm{v}(E)}\phi(\underline{r},E,t)\right] &= -\underline{\nabla}\cdot\underline{J}(\underline{r},E,t)-\Sigma_t(\underline{r},E,t)\phi(\underline{r},E,t)\\ &+\chi_p(E)(1-\beta_t)\int_0^\infty \nu\Sigma_f(\underline{r},E',t)\phi(\underline{r},E',t)dE' \qquad (2.1.a)\\ &+\int_0^\infty \Sigma_s(\underline{r},E'\rightarrow E,t)\phi(\underline{r},E',t)dE'+\sum_{iprec=1}^{nprec}\chi_{iprec}(E)\lambda_{iprec}C_{iprec}(\underline{r},t)\end{aligned}$$

$$\frac{\partial c_{iprec}(\underline{r},t)}{\partial t} = \beta_{iprec}\int_0^\infty \nu\Sigma_f(\underline{r},E',t)\phi(\underline{r},E',t)dE' - \lambda_{iprec}c_{iprec}(\underline{r},t) \qquad iprec = 1,\ nprec \tag{2.1.b}$$

where it has been assumed for simplicity that there is one fissioning isotope and no external source. The terms in (2.1) are defined as follows, where G is the total number of energy groups and nprec is the total number of delayed neutron precursor families:

$v(E)$ = neutron speed at energy E (cm/s),

$\phi(\underline{r},E,t)$ = scalar flux density at position $\underline{r}$, energy E, and time t (cm^{-2} s),

$\underline{J}(\underline{r},E,t)$ = net current density at position $\underline{r}$, energy E, and time t (cm^{-2} s),

$\Sigma_t(\underline{r},E,t)$ = total cross section at position $\underline{r}$, energy E, and time t (cm^{-1}),

$\Sigma_s(\underline{r},E'\rightarrow E,t)$ = scattering cross section at position $\underline{r}$, for scattering from energy E' to E, at time t (cm^{-1}),

$\chi_p(E)$ = prompt neutron fission spectrum at energy E,

$\nu\Sigma_f(\underline{r},E,t)$ = average number of neutrons per fission times the fission cross section (cm^{-1}),

$\chi_{iprec}(E)$ = delayed neutron fission spectrum for precursor family iprec,

β_{iprec} = fraction of fission neutrons that appear from precursor family iprec,

β_t = total number of fission neutrons that appear from all delayed neutron precursors nprec,

$$\beta_t = \sum_{iprec=1}^{nprec}\beta_{iprec}$$

$c_{iprec}(\underline{r},t)$ = density of delayed neutron precursor iprec at position $\underline{r}$ and time t (cm^{-3}),

The goal is to structure these equations in such a way that they can be solved on a computer. In general, this means constructing discreet forms of continuous variables, integrals, and derivatives. The range of energy is broken up into discreet energy 'bins' called groups. The spatial domain is broken up into a mesh of nodes, in which the cross sections in eqns. (2.1.a) and (2.1.b) are averaged to yield homogenized, spatially constant values. The spatial derivatives will be expressed in terms of node-homogenized quantities, and the subsequent solution to eqns. (2.1.a) and (2.1.b) will be the node and group average scalar flux densities.

If the volume of node l is V_l, and the range of energies is broken up into G total energy groups with each group being defined as the energy bin $\Delta E_g \equiv E_{g-1} - E_g$, then the energy and node averaged quantities may be defined as follows:

$$\bar{\bar{\phi}}_g^{(l)}(t) \equiv \frac{1}{V_l}\int_{V_l} dV \int_{\Delta E_g} dE\, \phi(\underline{r},E,t)$$

$$\frac{1}{\mathrm{v}_g} \equiv \frac{1}{\bar{\bar{\phi}}_g^{(l)}(t) V_l}\int_{V_l} dV \int_{\Delta E_g} dE \frac{1}{\mathrm{v}(E)}\phi(\underline{r},E,t)$$

$$\underline{J}_g(\underline{r},t) \equiv \int_{\Delta E_g} dE\, \underline{J}(\underline{r},E,t)$$

$$\Sigma_{t,g}^{(l)} \equiv \frac{1}{\bar{\bar{\phi}}_g^{(l)}(t) V_l}\int_{V_l} dV \int_{\Delta E_g} dE\, \Sigma_t(\underline{r},E,t)\phi(\underline{r},E,t)$$

$$\chi_{pg} \equiv \int_{\Delta E_g} dE\, \chi_p(E)$$

$$\nu\Sigma_{f,g}^{(l)} \equiv \frac{1}{\bar{\bar{\phi}}_g^{(l)}(t) V_l}\int_{V_l} dV \int_{\Delta E_g} dE\, \nu\Sigma_f(\underline{r},E,t)\phi(\underline{r},E,t)$$

$$\Sigma_{gg'}^{(l)} \equiv \frac{1}{\overline{\overline{\phi}}_g^{(l)}(t)V_l}\int_{V_l} dV \int_{\Delta E_g} dE \int_{\Delta E_{g'}} dE' \,\Sigma_s(\underline{r}, E' \to E, t)\phi(\underline{r}, E', t)$$

$$\chi_{iprec,g} \equiv \int_{\Delta E_g} dE\, \chi_{iprec}(E)$$

$$\overline{\overline{C}}_{iprec}^{(l)}(t) \equiv \frac{1}{V_l}\int_{V_l} dV\, c_{iprec}(\underline{r}, t)$$

In the above definitions, the superscript (l) on the cross sections signifies a spectrum weighted, homogenized quantity over the node l. The double bar and (l) on the scalar flux and precursor density signifies a volume averaged quantity over node l. We also combine the total cross section and the in-group scattering cross section (g to g):

$$\Sigma_g^{(l)} \equiv \Sigma_{t,g}^{(l)} - \Sigma_{gg}^{(l)}$$

If eqns. (2.1.a) and (2.1.b) are integrated over the volume of one node and the energy bin that defines group g, and the above definitions are substituted, we obtain equations which describe the neutron balance for a single node l and group g:

$$\begin{aligned}\frac{d}{dt}\frac{\overline{\overline{\phi}}_g^{(l)}(t)}{v_g} = &-\frac{1}{V_l}\int_{V_l} dV\, \underline{\nabla}\cdot \underline{J}_g(\underline{r}, t) - \Sigma_g^{(l)}(t)\overline{\overline{\phi}}_g^{(l)}(t) \\ &+\sum_{\substack{g'=1 \\ g'\neq g}}^{G} \Sigma_{gg'}^{(l)}(t)\overline{\overline{\phi}}_{g'}^{(l)}(t) + \chi_{pg}(1-\beta_t)\sum_{g'=1}^{G} \nu\Sigma_{fg'}^{(l)}(t)\overline{\overline{\phi}}_{g'}^{(l)}(t) \\ &+\sum_{iprec=1}^{nprec} \chi_{iprec,g}\lambda_{iprec}\overline{\overline{C}}_{iprec}^{(l)}(t)\end{aligned} \qquad (2.2.a)$$

$$\frac{d}{dt}\overline{\overline{C}}_{iprec}^{(l)}(t) = \beta_{iprec}\nu\Sigma_{fg}^{(l)}(t)\overline{\overline{\phi}}_g^{(l)}(t) - \lambda_{iprec}\overline{\overline{C}}_{iprec}^{(l)}(t) \quad iprec = 1, nprec \qquad (2.2.b)$$

The spatial and energy group discretization is complete for both (2.2.a) and (2.2.b) except for the leakage term involving the net current density. It will be of use later if the form of the leakage term is that of a surface-averaged quantity rather than a volume-averaged quantity. Gauss's divergence theorem allows the volume integral over the divergence of $\underline{J}_g(\underline{r},t)$ to be replaced by a surface integral:

$$\int_{V_l} \underline{\nabla} \cdot \underline{J}_g(\underline{r},t)dV = \sum_{p=1}^{P} \int_{A_p} \underline{J}_g(\underline{r},t) \cdot \underline{n}_p ds \tag{2.3}$$

where A_p is the surface area of face p of node l, $\underline{n}_p$ is the *outward directed* normal vector to face p, and P is the number of faces on node l. A face averaged net current density can be defined as:

$$\overline{\underline{J}}_g^{(l)}(p,t) = \frac{1}{A_p} \int_{A_p} \underline{J}_g(\underline{r},t)ds \Big|_{node\ l} \tag{2.4}$$

Combining eqn. (2.4) with the right hand side of eqn. (2.3) we can write:

$$\begin{aligned} \frac{d}{dt}\frac{\overline{\phi}_g^{(l)}(t)}{\mathrm{v}_g} = & -\frac{1}{V_l}\sum_{p=1}^{P} A_p \overline{\underline{J}}_g^{(l)}(p,t) \cdot \underline{n}_p - \Sigma_g^{(l)}(t)\overline{\phi}_g^{(l)}(t) \\ & + \sum_{\substack{g'=1 \\ g' \neq g}}^{G} \Sigma_{gg'}^{(l)}(t)\overline{\phi}_{g'}^{(l)}(t) + \chi_{pg}(1-\beta_t)\sum_{g'=1}^{G} \nu\Sigma_{fg'}^{(l)}(t)\overline{\phi}_{g'}^{(l)}(t) \\ & + \sum_{iprec=1}^{nprec} \chi_{iprec,g}\lambda_{iprec}\overline{C}_{iprec}^{(l)}(t) \end{aligned} \tag{2.5.a}$$

$$\frac{d}{dt}\overline{C}_{iprec}^{(l)}(t) = \beta_{iprec}\sum_{g'=1}^{G} \nu\Sigma_{fg'}^{(l)}(t)\overline{\phi}_{g'}^{(l)}(t) - \lambda_{iprec}\overline{C}_{iprec}^{(l)}(t) \quad iprec = 1, nprec \tag{2.5.b}$$

where it can be seen that the time dependent neutron balance is now expressed in terms of homogenized cross sections, volume averaged fluxes and precursor concentrations, and surface averaged currents. Equations (2.5.a) and (2.5.b) together form the time dependent nodal balance equation.

2.3 Summary

The goal is to be able to solve eqns. (2.1.a) and (2.1.b) efficiently and accurately on a computer. It was mentioned that the energy domain is broken up into discreet energy groups, and the spatial domain is broken up into nodes. The chief problem with solving eqn. (2.5) is that we require an additional relation between the surface averaged current and the volume averaged fluxes. The next chapter will discuss such a relation. It is important at this step to review our progress and the approximations used to make that progress. Equation (2.5) represents a discreet form of (2.1). It was made discreet by defining node-homogenized and group-averaged cross sections to replace the continuous cross sections in eqn.(2.1). Volume-averaged fluxes and precursor concentrations and surface averaged currents were also used. If the 'true flux' (i.e. the solution to eqn. (2.1)) were used to compute the node-homogenized and group-averaged cross sections, the solution to eqn. (2.5) would match exactly the solution of (2.1). This situation is not only circular but impossible because of the need for an additional relationship to solve (2.5). It points out, however, that eqn. (2.5) is a formally exact consequence of eqn. (2.1).

In practice, eqn. (2.5) contains homogenization errors. The magnitude of these homogenization errors depends upon the estimate of the flux used to compute the cross sections. These homogenization errors should be kept in mind as the equations are further developed and more approximations are made.

Chapter 3

Derivation of Static Nodal Diffusion Theory Equations for Equilateral Triangles

3.1 Introduction

In the last chapter, equations were derived which describe the overall neutron balance in a single node, for each energy group g, in terms of node and surface-averaged quantities. In this chapter, we shall simplify the basic problem by treating the static case. This simplification will not only make it easier to work with the equations and apply them to equilateral triangles, but is of direct use since, in many cases, it is only the static solution that we seek.

The first step will be to use Fick's Law to relate the net current density to the gradient of the scalar flux density. This is the diffusion approximation. The gradient operator in Fick's law contains spatial derivatives. In order to evaluate these spatial derivatives for a node of triangular-z geometry, an approximation will be made that the flux is flat within the node (the finite difference approximation). Discontinuity factors will be introduced to correct for sources of error: the homogenization of the cross sections, Fick's law, and the finite difference approximation.

The quadratic nodal method will then be introduced for triangular geometry. The derivation will first be shown for equilateral triangles in the 2-D plane, and then for the axial dimension. In the 2-D plane, three independent coordinate directions will be identified, each perpendicular to one face of the triangle and passing through the opposite point.

It will be assumed that the flux within each triangular node can be represented by three, independent, one-dimensional fluxes (one in each coordinate direction). Each one-dimensional flux will be modeled as a quadratic polynomial and forced to match the surface-averaged flux at one end and the flux at the point at the other end.

In previous implementations of the polynomial nodal method, the nodes have had an even number of faces [in 2-D: Cartesian = 4, Cylindrical (r only) = 2, Hexagonal = 6]. Hence, each coordinate direction passes through two surfaces, and not a surface and a point. Because we do not wish to solve for quantities at the points of the triangular nodes, the quadratic polynomials for each coordinate direction will be "connected" (equal in value and derivative) through the points to the quadratic polynomials in the nodes beyond the points. The consequence of using connected polynomials is that the traditional nearest-neighbor coupling is extended to include a "patchwork" of nodes surrounding each node of interest.

In the quadratic derivation, discontinuity factors will be introduced to correct for the same reasons as before, although the correction will be less for the spatial approximation.

The equations for both the finite difference and higher order approximations will be derived for cases when the node of interest is in the interior of the reactor, as well as when it is adjacent to the external boundary of the reactor.

3.2 The Static Nodal Balance Equation

The first step is to take the steady state version of (2.5.a) and (2.5.b). In steady state, the time dependence is dropped for all cross sections, currents, and fluxes. In addition, the time derivatives disappear. Thus, when (2.5.b) is used to eliminate the precursor terms in (2.5.a) we have:

$$\frac{1}{V_l}\sum_{p=1}^{P} A_p \underline{J}_g^{(l)}(p)\cdot\underline{n}_p + \Sigma_g^{(l)}\overline{\phi}_g^{(l)} = \sum_{\substack{g'=1\\ g'\neq g}}^{G}\Sigma_{gg'}^{(l)}\overline{\phi}_{g'}^{(l)} + \frac{\chi_g}{\lambda}\sum_{g'=1}^{G}\nu\Sigma_{fg'}^{(l)}\overline{\phi}_{g'}^{(l)} \qquad (3.1.a)$$

where

$$\chi_g = (1-\beta_l)\chi_{pg} + \sum_{iprec=1}^{nprec}\beta_{iprec}\chi_{iprec,g} \qquad (3.1.b)$$

and λ is the critical eigenvalue.

Eqn. (3.1.a) is written for a general node l with P faces, numbered p=1,2,3...P. Our goal is to solve (3.1.a) for nodes with triangular-z geometry. For reference, we introduce an equilateral triangle with side length S and height Z_d:

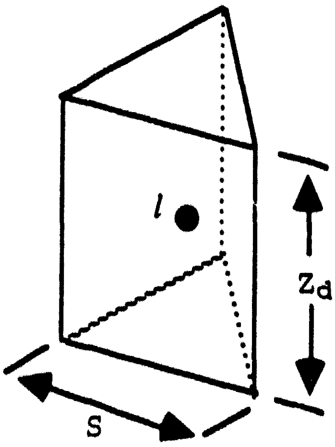


Figure 3.1. Three Dimensional View of Node 1.

The volume of this node is given by:

$$V_l = S^2\frac{\sqrt{3}}{4}Z_d \qquad (3.2)$$

The surface area for the top or bottom face of this node is:

$$A_{triangle} = S^2\frac{\sqrt{3}}{4} \qquad (3.3.a)$$

and for each of the three sides is:

$$A_z = S Z_d \tag{3.3.b}$$

A mesh of nodes in the 2-D plane is numbered as follows:

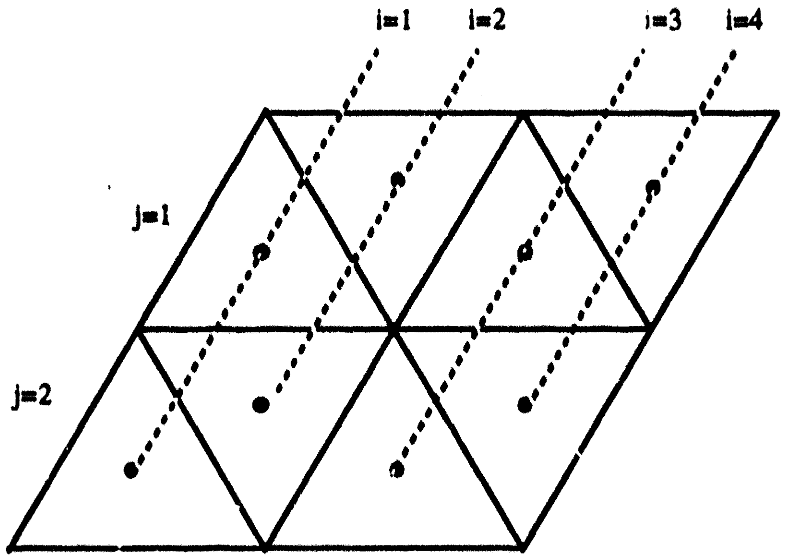


Figure 3.2. Mesh Layout in the 2-D Plane

The node numbering scheme for the axial direction is shown in Fig. 3.3.

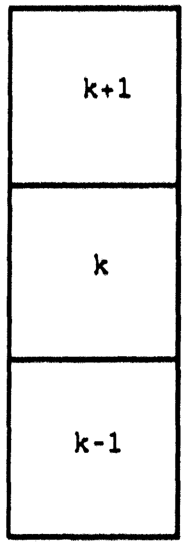


Figure 3.3. Axial Mesh Layout.

We also need to identify the surfaces between nodes in Fig. 3.2 and Fig. 3.3. It turns out to be convenient if we identify the surfaces by numbering them p=1,2,3...P for each node. Examining the 2-D mesh of triangles in Fig. 3.2 shows that nodes with even i are 'inverted' with respect to nodes with odd i. We label the surfaces for each node in such a way that surface 1 of node (i,j) is always adjacent to node (i-1,j), and surface 2 is adjacent to node (i+1,j). Surface 3 is then adjacent to either the node above or node below. Fig. 3.4 shows the surface numbering for the 2-D mesh when i is odd:

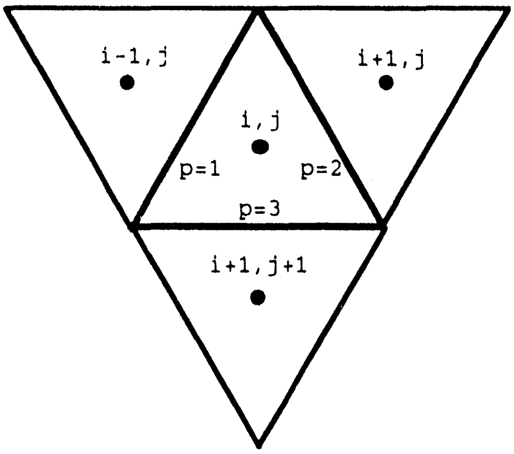


Figure 3.4. Side Numbering for Node i,j when i is odd.

Fig. 3.5 shows the surface numbering for the 2-D mesh when i is even:

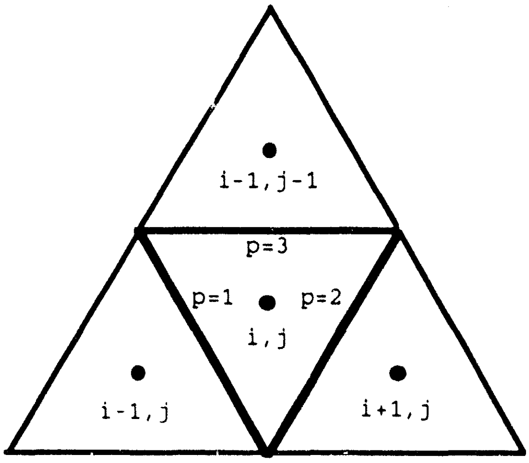


Figure 3.5. Side Numbering for Node i,j when i is even.

Each surface can be uniquely identified by node coordinate (i,j) and surface number (p).

The axial surfaces are numbered such that surface 4 is adjacent to the node above, and surface 5 is adjacent to the node below:

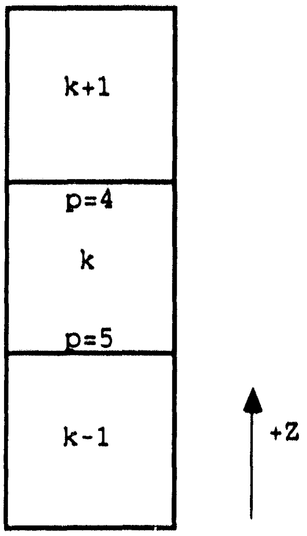


Figure 3.6. Surface Numbers for the Axial Dimension.

Consistent with the surface numbering scheme in the 2-D plane, we define three positive axes which define the positive direction relative to each face (p=1,2,3). Fig. 3.7 shows this numbering scheme.

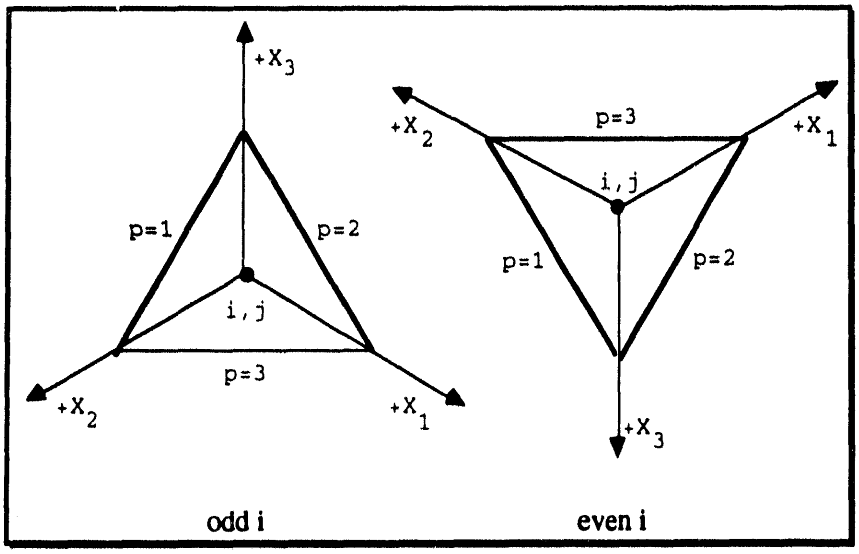


Figure 3.7. Positive Axes in the 2-D Plane.

For the axial surfaces, the positive axis points in the direction of increasing k, as can be seen in Fig. 3.6.

Rewriting eqn. (3.1) for node i,j,k yields:

$$\frac{1}{V_{i,j,k}}\sum_{p=1}^{P} A_p \overline{\underline{J}}_g^{(i,j,k)}(p)\cdot \underline{n}_p + \Sigma_g^{(i,j,k)}\overline{\overline{\phi}}_g^{(i,j,k)} = \sum_{\substack{g'=1\\ g'\neq g}}^{G}\Sigma_{gg'}^{(i,j,k)}\overline{\overline{\phi}}_{g'}^{(i,j,k)} + \frac{\chi_g}{\lambda}\sum_{g'=1}^{G}\nu\Sigma_{fg'}^{(i,j,k)}\overline{\overline{\phi}}_{g'}^{(i,j,k)} \tag{3.4}$$

Recall that the vector $\underline{n}_p$ is an outward directed normal to the surface p.

3.3 Fick's Law and The Finite Difference Approximation

The scalar flux density can be related to the net current densit; by Fick's Law, which states that the net current density is proportional to the gradient of the scalar flux:

$$J(\underline{r},E) \cong -D(\underline{r},E)\underline{\nabla}\phi(\underline{r},E) \tag{3.5}$$

We are interested in applying this at the interfaces between node i,j,k and its adjacent nodes. For simplicity, we treat the 2-D case first and derive an expression for face 1. The expressions for the other faces can be derived in a completely analogous way.

For face 1 of node (i,j), the adjacent node is node (i-1,j). This is the case if i is odd or i is even. Fig. 3.8 shows the two nodes with positive axis X_1 and transverse direction Y_1, where X_c is the center to center node distance, and is given in terms of S, the length of the triangle side:

$$X_c = S\frac{\sqrt{3}}{3} \tag{3.6}$$

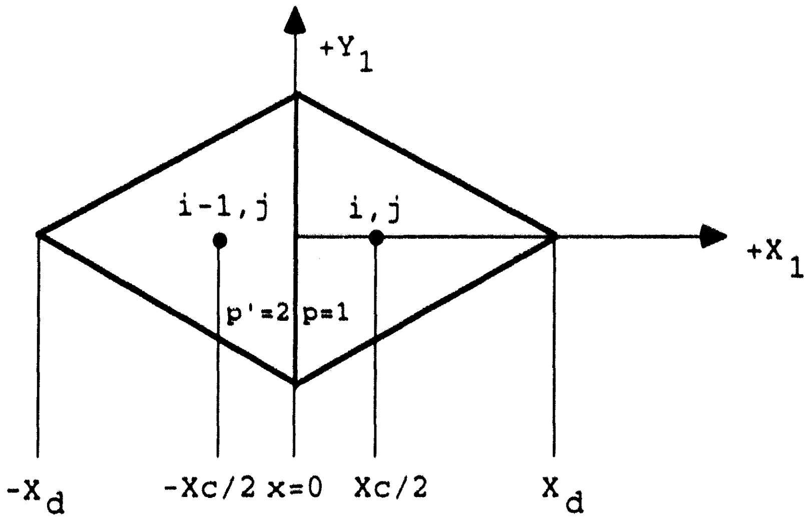


Figure 3.8. Node (i,j) and Adjacent Node (i-1,j).

X_d is the altitude of the triangle, and is given by:

$$X_d = S\frac{\sqrt{3}}{2} \tag{3.7}$$

Using the component of (3.5) in the $+X_1$ direction, we have:

$$J_{X_1}(x,y,E) \cong -D(x,y,E)\frac{\partial}{\partial x}\phi(x,y,E) \tag{3.8}$$

where the X_1 subscript means that the current points in the positive X_1 direction.

Integrating over the transverse direction y_1 and energy group g, and defining the following quantities:

$$\overline{J}^{(i,j)}_{g,X_1}(p=1) \equiv \frac{\int_{\Delta E_g} dE \int_{-S/2}^{S/2} dy J_{X_1}(x=0,y,E)}{\int_{-S/2}^{S/2} dy}$$

$$D_g^{(i,j)} \equiv \frac{\int_{\Delta E_g} dE \int_0^{X_d} dx \int_{-y(x)/2}^{y(x)/2} dy\, D(x,y,E)\phi(x,y,E)}{\int_{\Delta E_g} dE \int_0^{X_d} dx \int_{-y(x)/2}^{y(x)/2} dy \phi(x,y,E)}$$

$$\frac{\partial}{\partial x}\overline{\phi}_g^{(i,j)}(p=1) \equiv \frac{\int_{\Delta E_g} dE \int_{-S/2}^{S/2} dy \frac{\partial}{\partial x}\phi(x=0,y,E)}{\int_{-S/2}^{S/2} dy}$$

$$y(x) = S - \frac{2}{\sqrt{3}}x$$

we obtain:

$$\overline{J}^{(i,j)}_{g,X_1}(p=1) \cong -D_g^{(i,j)} \frac{\partial}{\partial x}\overline{\phi}_g^{(i,j)}(p=1) \qquad (3.9)$$

To approximate the derivative in eqn. (3.9), we write:

$$\overline{J}^{(i,j)}_{g,X_1}(p=1) \cong -D_g^{(i,j)}\left[\frac{\overline{\overline{\phi}}_g^{(i,j)} - \overline{\phi}_g^{(i,j)}(p=1)}{X_c/2}\right] \qquad (3.10.a)$$

where

$$\overline{\overline{\phi}}_g^{(i,j)} \equiv \frac{1}{V_{i,j}}\int_{\Delta E_g} dE \int_0^{X_d} dx \int_{-y(x)/2}^{y(x)/2} dy\, \phi(x,y,E) \qquad (3.10.b)$$

and

$$\overline{\phi}_g^{(i,j)}(p=1) \equiv \frac{1}{y(0)}\int_{\Delta E_g} dE \int_{-S/2}^{S/2} dy\, \phi(x=0,y,E) \qquad (3.10.c)$$

An analogous expression can be written in terms of node (i-1,j) quantities, where p' refers to the face of node (i-1,j) and the X_1 subscript on the face-averaged current means that it is positive in the $+X_1$ direction:

$$\overline{J}_{g,X_1}^{(i-1,j)}(p'=2) \cong -D_g^{(i-1,j)}\left[\frac{\overline{\phi}_g^{(i-1,j)}(p'=2)-\overline{\overline{\phi}}_g^{(i-1,j)}}{X_c/2}\right] \qquad (3.11)$$

Equations (3.10.a) and (3.11) can be used to eliminate the surface-averaged flux at x=0 by applying continuity of flux and current at x=0. Before this is done, however, we divide each surface-averaged flux by a correction factor for the appropriate side of the face at x=0. This correction factor will correct for the errors introduced in the homogenization of the cross sections, the approximation of diffusion theory, and the finite difference approximation. When the continuity of flux is applied, it is still applied to the uncorrected (physical) surface-averaged fluxes. Thus, if the correction factors for each side of the face are not equal, the *corrected* flux is *discontinuous* at the face. For this reason, the correction factors are known as *discontinuity factors*.

When discontinuity factors are introduced, eqns. (3.10.a) and (3.11) become:

$$\overline{J}_{g,X_1}^{(i,j)}(p=1) = -D_g^{(i,j)}\left[\frac{\overline{\overline{\phi}}_g^{(i,j)} - \dfrac{\overline{\phi}_g^{(i,j)}(p=1)}{f_{g,p=1}^{(i,j)}}}{X_c/2}\right] \qquad (3.12)$$

$$\overline{J}_{g,X_1}^{(i-1,j)}(p'=2) = -D_g^{(i-1,j)}\left[\frac{\dfrac{\overline{\phi}_g^{(i-1,j)}(p'=2)}{f_{g,p'=2}^{(i-1,j)}} - \overline{\overline{\phi}}_g^{(i-1,j)}}{X_c/2}\right] \qquad (3.13)$$

where $f_{g,p=1}^{(i,j)}$ is the discontinuity factor for the right side of the face at x=0, and $f_{g,p'=2}^{(i-1,j)}$ is the discontinuity factor for the left side of the face at x=0. Note that eqns. (3.12) and

(3.13) are formally exact due to the introduction of the discontinuity factors. Applying the continuity of the corrected fluxes at x=0:

$$\overline{\phi}_g^{(i,j)}(p=1)=\overline{\phi}_g^{(i-1,j)}(p'=2) \tag{3.14}$$

and the continuity of current at x=0, remembering that the current was defined as positive in the +X_1 direction:

$$\overline{J}_{g,X_1}^{(i,j)}(p=1)=\overline{J}_{g,X_1}^{(i-1,j)}(p'=2) \tag{3.15}$$

we obtain:

$$\overline{J}_{g,X_1}^{(i,j)}(p=1)=\frac{2}{X_c}\left[\frac{1}{D_g^{(i,j)}}+\frac{1}{D_g^{(i-1,j)}}\frac{f_{g,p'=2}^{(i-1,j)}}{f_{g,p=1}^{(i,j)}}\right]^{-1}\left[\frac{f_{g,p'=2}^{(i-1,j)}}{f_{g,p=1}^{(i,j)}}\overline{\phi}_g^{(i-1,j)}-\overline{\phi}_g^{(i,j)}\right] \tag{3.16}$$

An analogous expression can be derived for each face of the node. The relation for the other two faces of the equilateral triangle has the same form as (3.16). For the top (p=4) and bottom (p=5), the center to center distance X_c is replaced by the axial center to center distance, Z_d.

Substituting eqn. (3.16) for face 1 and the corresponding expressions for the other faces into eqn. (3.4), we get, for i odd, the discreet nodal balance equation for node (i,j,k):

$$\begin{aligned}
&\frac{A_{p=1}}{V_{i,j,k}}\frac{2}{X_c}\left[\frac{1}{D_g^{(i,j,k)}}+\frac{1}{D_g^{(i-1,j,k)}}\frac{f_{g,p'=2}^{(i-1,j,k)}}{f_{g,p=1}^{(i,j,k)}}\right]^{-1}\left[\overline{\overline{\phi}}_g^{(i,j,k)}-\frac{f_{g,p'=2}^{(i-1,j,k)}}{f_{g,p=1}^{(i,j,k)}}\overline{\overline{\phi}}_g^{(i-1,j,k)}\right]\\
&+\frac{A_{p=2}}{V_{i,j,k}}\frac{2}{X_c}\left[\frac{1}{D_g^{(i,j,k)}}+\frac{1}{D_g^{(i+1,j,k)}}\frac{f_{g,p'=1}^{(i+1,j,k)}}{f_{g,p=2}^{(i,j,k)}}\right]^{-1}\left[\overline{\overline{\phi}}_g^{(i,j,k)}-\frac{f_{g,p'=1}^{(i+1,j,k)}}{f_{g,p=2}^{(i,j,k)}}\overline{\overline{\phi}}_g^{(i+1,j,k)}\right]\\
&+\frac{A_{p=3}}{V_{i,j,k}}\frac{2}{X_c}\left[\frac{1}{D_g^{(i,j,k)}}+\frac{1}{D_g^{(i+1,j+1,k)}}\frac{f_{g,p'=3}^{(i+1,j+1,k)}}{f_{g,p=3}^{(i,j,k)}}\right]^{-1}\left[\overline{\overline{\phi}}_g^{(i,j,k)}-\frac{f_{g,p'=3}^{(i+1,j+1,k)}}{f_{g,p=3}^{(i,j,k)}}\overline{\overline{\phi}}_g^{(i+1,j+1,k)}\right]\\
&+\frac{A_{p=4}}{V_{i,j,k}}\frac{2}{X_d}\left[\frac{1}{D_g^{(i,j,k)}}+\frac{1}{D_g^{(i,j,k+1)}}\frac{f_{g,p'=5}^{(i,j,k+1)}}{f_{g,p=4}^{(i,j,k)}}\right]^{-1}\left[\overline{\overline{\phi}}_g^{(i,j,k)}-\frac{f_{g,p'=5}^{(i-1,j,k)}}{f_{g,p=4}^{(i,j,k)}}\overline{\overline{\phi}}_g^{(i,j,k+1)}\right]\\
&+\frac{A_{p=5}}{V_{i,j,k}}\frac{2}{X_d}\left[\frac{1}{D_g^{(i,j,k)}}+\frac{1}{D_g^{(i,j,k-1)}}\frac{f_{g,p'=4}^{(i,j,k-1)}}{f_{g,p=5}^{(i,j,k)}}\right]^{-1}\left[\overline{\overline{\phi}}_g^{(i,j,k)}-\frac{f_{g,p'=4}^{(i-1,j,k)}}{f_{g,p=5}^{(i,j,k)}}\overline{\overline{\phi}}_g^{(i,j,k-1)}\right] \qquad (3.17)\\
&+\Sigma_g^{(i,j,k)}\overline{\overline{\phi}}_g^{(i,j,k)}=\sum_{g'=1}^{G}\Sigma_{gg'}^{(i,j,k)}\overline{\overline{\phi}}_{g'}^{(i,j,k)}+\frac{\chi_g}{\lambda}\sum_{g'=1}^{G}\nu\Sigma_{fg'}^{(i,j,k)}\overline{\overline{\phi}}_{g'}^{(i,j,k)}
\end{aligned}$$

For i even, superscript (i+1,j+1,k) in the third line of eqn. (3.17) becomes (i-1,j-1,k). Eqn. (3.17) describes the neutron balance for an interior node (i,j,k) in terms of node homogenized cross sections, constants, discontinuity factors, and volume-averaged fluxes. The next section will derive an equation of the same form as (3.16) for nodes adjacent to the boundary of the reactor.

3.4 Boundary Conditions

In the previous section, a relationship between the surface-averaged currents and volume-averaged fluxes was derived by writing the finite difference approximation to eqn. (3.9). Eqns. (3.10.a) and (3.11) were used to eliminate the surface-averaged flux for face p=1 of node (i,j). If node (i,j) face p=1 is on a boundary, eqn. (3.11) cannot be written. In this case, we use a general equation for a boundary condition which relates the surface-averaged current to the surface-averaged flux:

$$\alpha_{g,p}^{(i,j)} \overline{\underline{J}}_g^{(i,j)}(p) \cdot \underline{n}_p = \beta_{g,p}^{(i,j)} \overline{\phi}_g^{(i,j)}(p) \qquad (3.18)$$

where, as before, $\underline{n}_p$ points out of the node and is normal to the surface p. The group and face dependent coefficients, $\alpha_{g,p}^{(i,j)}$ and $\beta_{g,p}^{(i,j)}$, are specified by the type of boundary condition desired. Table 3.1 gives appropriate values for typical boundary conditions.

Table 3.1
Values of $\alpha_{g,p}^{(i,j)}$ and $\beta_{g,p}^{(i,j)}$ for Typical Boundary Conditions.

Condition at Boundary	$\alpha_{g,p}^{(i,j)}$	$\beta_{g,p}^{(i,j)}$
Zero Current	1	0
Zero Flux	0	1
Zero Net Incoming Current	2	1
Albedo	1	C[1]

[1] C is the ratio of the outgoing current at the face to the face-averaged flux, as defined by eqn. (3.18) with alpha equal to unity.

Our goal is to obtain an expression analogous to (3.16). If we examine the case of face p=1 of node (i,j) on the boundary, then we re-write eqn. (3.18) as:

$$-\alpha_{g,p=1}^{(i,j)} \overline{J}_{g,x_1}^{(i,j)}(p=1) = \beta_{g,p=1}^{(i,j)} \overline{\phi}_g^{(i,j)}(p=1) \qquad (3.19)$$

where the dot product on the left hand side of eqn. (3.18) produced a -1 because the normal vector $\underline{n}_p$ points in the $-X_1$ direction.

Equation (3.12) remains valid and can be used with (3.19) to eliminate the surface-averaged flux at face 1 of node (i,j). Solving (3.19) for $\bar{\phi}_g^{(i,j)}(p=1)$ and inserting the result into (3.12) we obtain:

$$J_{g,x_1}^{(i,j)}(p=1) = -\frac{2}{X_c}\left[\frac{\beta_{g,p=1}^{(i,j)}}{D_g^{(i,j)}} + \frac{2}{X_c}\frac{\alpha_{g,p=1}^{(i,j)}}{f_{g,p=1}^{(i,j)}}\right]^{-1}\left[\beta_{g,p=1}^{(i,j)}\bar{\phi}_g^{(i,j)}\right] \qquad (3.20)$$

If face p=1, p=2, or p=3 of node (i,j,k) is on a boundary, the discreet nodal balance equation (eqn. (3.17)) is used with line 1,2, or 3 replaced by:

$$\frac{A_p}{V_{i,j,k}}\frac{2}{X_c}\left[\frac{\beta_{g,p}^{(i,j)}}{D_g^{(i,j)}} + \frac{2}{X_c}\frac{\alpha_{g,p}^{(i,j)}}{f_{g,p}^{(i,j)}}\right]^{-1}\left[\beta_{g,p}^{(i,j)}\bar{\phi}_g^{(i,j)}\right] \qquad (3.21)$$

If the face is p=4 or p=5, (3.21) is used with X_d in place of X_c, and inserted into eqn. (3.17) in line 4 or line 5, respectively.

In this section, the derivation of the discreet nodal balance equation was extended to include the case where face p was on the boundary of the reactor. Equation (3.17) was written for node (i,j,k) for odd i. A complimentary equation exists for even i. If node (i,j,k) is on the boundary of the reactor, eqn. (3.21) is used to replace the appropriate line in (3.17). These equations can be written for each node in the reactor, and solved on a computer to yield volume-averaged fluxes and a critical eigenvalue, provided that cross sections and discontinuity factors are available. The more traditional mesh-centered finite difference equations can be obtained from the above equations by setting all of the discontinuity factors to unity. Because the discontinuity factors correct for the use of the finite difference approximation, the above method will allow the use of larger nodes. We shall refer to the above method as the *Coarse Mesh Finite Difference (CMFD) Method*.

3.5 Quadratic Based Coupling Equations

Our goal in this section is to develop an improved relationship between the surface-averaged current and the volume-averaged fluxes.

3.5.1 Transverse-Averaged Flux in Quadratic Form for an Equilateral Triangle

We once again examine face 1 of node (i,j), shown in Fig. 3.8. Equation (3.10.c) was used to define the face-averaged flux for p=1 (x=0). We can generalize this and define a transverse-averaged flux as a function of x within the node (i,j):

$$\bar{\phi}_g^{(i,j)}(x) \equiv \frac{1}{y(x)}\int_{\Delta E_g} dE \int_{-y(x)/2}^{y(x)/2} dy\, \phi(x,y,E) \qquad (3.24)$$

The right hand side of eqn. (3.24) can be represented by a quadratic polynomial. This quadratic polynomial is forced, within node (i,j), to equal the face-averaged flux, $\bar{\phi}_g^{(i,j)}(p=1)$, when x=0, the value of the flux at the point, $\phi_g^{(i,j)}(X_d,0)$, when x=X_d, and when integrated over x according to eqn. (3.25),

$$\frac{1}{V_{i,j}}\int_0^{X_d} dx\, y(x)\bar{\phi}_g^{(i,j)}(x) = \bar{\bar{\phi}}_g^{(i,j)} \qquad (3.25)$$

it is forced to equal the volume-averaged flux for the node. Suppressing the group and node notation temporarily for simplicity, and defining a dimensionless parameter ξ where $\xi \equiv x/X_d$, we write the polynomial expression as:

$$\bar{\phi}(\xi) = \phi(X_d,0)P_1(\xi) + \bar{\phi}(0)P_2(\xi) + \bar{\bar{\phi}}P_3(\xi) \qquad (3.26)$$

where

$$
\begin{aligned}
P_1(\xi) &\equiv C_{1a} + C_{1b}\xi + C_{1c}\xi^2 \\
P_2(\xi) &\equiv C_{2a} + C_{2b}\xi + C_{2c}\xi^2 \\
P_3(\xi) &\equiv C_{3a} + C_{3b}\xi + C_{3c}\xi^2
\end{aligned}
\tag{3.27}
$$

The coefficients in (3.27) are found by forcing $P_1(\xi)$, $P_2(\xi)$, and $P_3(\xi)$ to meet the following requirements:

$$
\begin{aligned}
&P_1(\xi=0)=0 \quad P_2(\xi=0)=1 \quad P_3(\xi=0)=0 \\
&P_1(\xi=1)=1 \quad P_2(\xi=1)=0 \quad P_3(\xi=1)=0
\end{aligned}
\tag{3.28}
$$

and

$$
\begin{aligned}
&\frac{\int_0^1 y(\xi)P_1(\xi)d\xi}{\int_0^1 y(\xi)d\xi} = 0 \\
&\frac{\int_0^1 y(\xi)P_2(\xi)d\xi}{\int_0^1 y(\xi)d\xi} = 0 \\
&\frac{\int_0^1 y(\xi)P_3(\xi)d\xi}{\int_0^1 y(\xi)d\xi} = 1 \qquad y(\xi) \equiv \frac{2}{\sqrt{3}}(1-\xi)
\end{aligned}
\tag{3.29}
$$

The result is:

$$
\begin{aligned}
P_1(\xi) &= -\xi + 2\xi^2 \\
P_2(\xi) &= 1 - 5\xi + 4\xi^2 \\
P_3(\xi) &= 6\xi - 6\xi^2
\end{aligned}
\tag{3.30}
$$

Putting these results into eqn. (3.26), we obtain the quadratic expression for the transverse-averaged flux:

$$
\begin{aligned}
\bar{\phi}(x) = &\left[-\frac{x}{X_d} + \frac{2x^2}{X_d^2}\right]\phi(X_d,0) + \left[1 - \frac{5x}{X_d} + \frac{4x^2}{X_d^2}\right]\bar{\phi}(0) \\
&+\left[\frac{6x}{X_d} - \frac{6x^2}{X_d^2}\right]\bar{\phi}
\end{aligned}
\tag{3.31}
$$

The analytic derivative of (3.31) with respect to x is:

$$\frac{d\bar{\phi}(x)}{dx}=\frac{1}{X_d}\left[-1+\frac{4x}{X_d}\right]\phi(X_d,0)+\frac{1}{X_d}\left[-5+\frac{8x}{X_d}\right]\bar{\phi}(0)+\frac{1}{X_d}\left[6-\frac{12x}{X_d}\right]\bar{\bar{\phi}} \qquad (3.32)$$

Writing eqn. (3.32) in terms of node (i,j) quantities and returning to the group and node notation, we have:

$$\frac{d\bar{\phi}_g^{(i,j)}(x)}{dx}=\frac{1}{X_d}\left[-1+\frac{4x}{X_d}\right]\phi_g^{(i,j)}(X_d,0)+\frac{1}{X_d}\left[-5+\frac{8x}{X_d}\right]\bar{\phi}_g^{(i,j)}(p=1)+\frac{1}{X_d}\left[6-\frac{12x}{X_d}\right]\bar{\bar{\phi}}_g^{(i,j)} \qquad (3.33)$$

We can now use this expression in eqn. (3.9) to produce a relation between the surface-averaged current and the volume-averaged fluxes which is more accurate than eqn. (3.10.a). Evaluating eqn. (3.33) at x=0 and substituting the result into eqn. (3.9) gives:

$$J_{g,x_1}^{(i,j)}(p=1)\cong-\frac{D_g^{(i,j)}}{X_d}\left[\phi_g^{(i,j)}(X_d,0)+5\bar{\phi}_g^{(i,j)}(p=1)-6\bar{\bar{\phi}}_g^{(i,j)}\right] \qquad (3.34)$$

In order to write an analogous equation for face p=2 of node (i-1,j), we must first re-write (3.32) for a node of opposite orientation where x varies from x=0 to x=-X$_d$:

$$\frac{d\bar{\phi}(x)}{dx}=\left[\frac{1}{X_d}+\frac{4x}{X_d^2}\right]\phi(-X_d,0)+\left[\frac{5}{X_d}+\frac{8x}{X_d^2}\right]\bar{\phi}(0)-\left[\frac{6}{X_d}+\frac{12x}{X_d^2}\right]\bar{\bar{\phi}} \qquad (3.35)$$

In terms of node (i-1,j) quantities, eqn. (3.35) becomes:

$$\frac{d\bar{\phi}_g^{(i-1,j)}(x)}{dx} = \left[\frac{1}{X_d} + \frac{4x}{X_d^2}\right]\phi_g^{(i-1,j)}(-X_d,0) + \left[\frac{5}{X_d} + \frac{8x}{X_d^2}\right]\bar{\phi}_g^{(i-1,j)}(p=2)$$
$$-\left[\frac{6}{X_d} + \frac{12x}{X_d^2}\right]\bar{\bar{\phi}}_g^{(i-1,j)} \qquad (3.36)$$

The equivalent of (3.34) for face 2 of node (i-1,j) is then:

$$\bar{J}_{g,x_1}^{(i-1,j)}(p=2) \cong$$
$$\frac{D_g^{(i-1,j)}}{X_d}\left[-\phi_g^{(i-1,j)}(-X_d,0) - 5\bar{\phi}_g^{(i-1,j)}(p=2) + 6\bar{\bar{\phi}}_g^{(i-1,j)}\right] \qquad (3.37)$$

At this point, we could apply a continuity of surface-averaged flux and current at the face at x=0, and eliminate the surface-averaged flux with eqns. (3.34) and (3.37), but the resulting equation would contain the point fluxes $\phi_g^{(i,j)}(X_d,0)$ and $\phi_g^{(i-1,j)}(-X_d,0)$. We do not want to solve for the point fluxes during the solution, so we require additional equations to eliminate the point fluxes from eqns. (3.34) and (3.37).

In order to obtain these additional equations, we expand our view of the nodes surrounding node (i,j) from the nearest neighbor coupling of Figs. 3.4 and 3.5 to the extended coupling of Figs. 3.9 and 3.10. Fig. 3.9 shows the extended coupling for nodes of odd i, and Fig. 3.10 shows the extended coupling for nodes of even i.

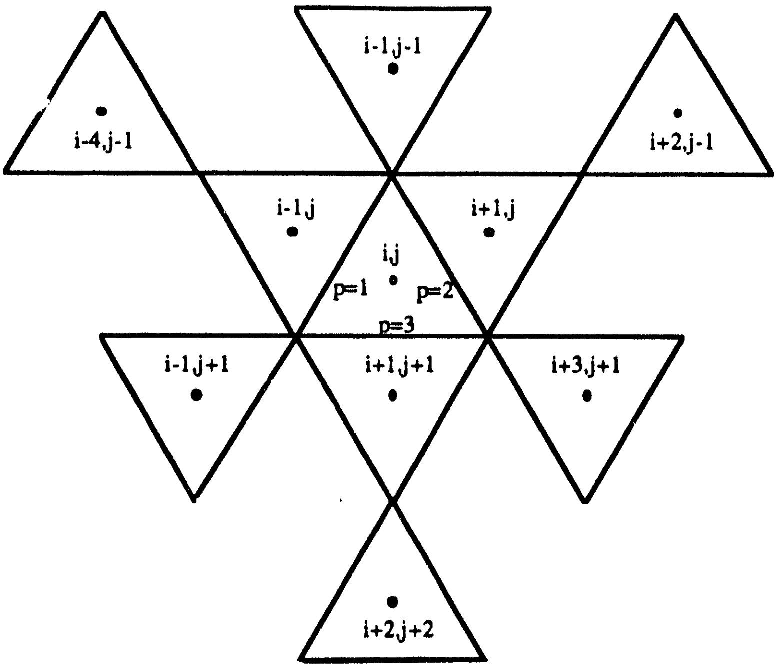


Figure 3.9. Extended Coupling for Node (i,j) when i is odd.

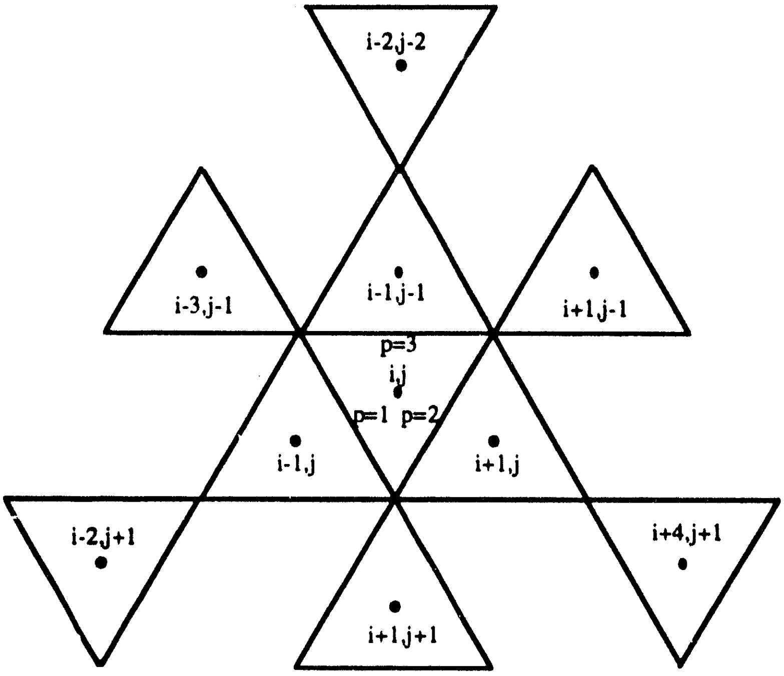


Figure 3.10. Extended Coupling for Node (i,j) when i is Even.

For face 1 of node (i,j), the nodes of interest are now:

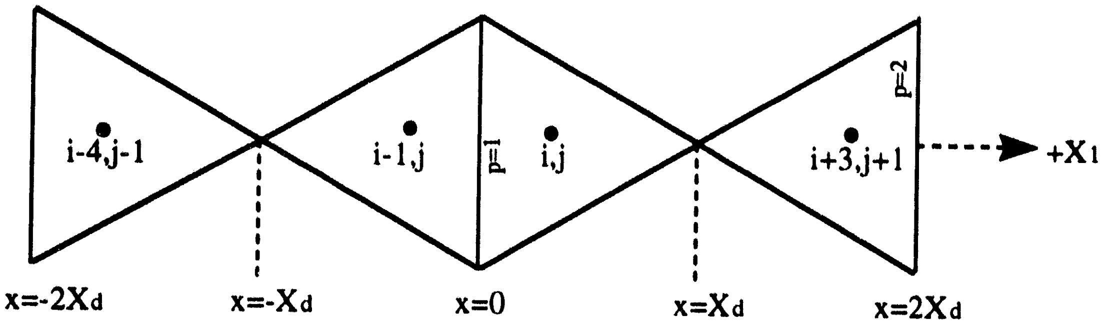


Figure 3.11. String of Nodes for Face 1 of Node (i,j) where i is odd.

The next sub-section will develop expressions for the surface-averaged current at x=X$_d$ in terms of node (i,j) quantities and node (i+3,j+1) quantities. In addition, an expression for the surface-averaged current at x=2X$_d$ will be written in terms of node (i+3,j+1) quantities. These equations will be used with (3.34) to obtain an expression for the surface-averaged current for the right side of x=0 that does not contain the point flux $\phi_g^{(i,j)}(X_d,0)$.

3.5.2 Eliminating the Point Flux at x=X$_d$

Strictly speaking, the surface-averaged current cannot be written for the point at x=X$_d$. We therefore write an expression for the surface-averaged current across a surface, perpendicular to the X$_1$ axis, a very small distance, ε, to the left of the point x=X$_d$. In the limit as this distance ε shrinks to zero, we approximate the surface-averaged current at x=X$_d$. Evaluating eqn. (3.33) at x=X$_d$ - ε and substituting the result into eqn. (3.9) we get:

$$\overline{J}_{g,X_1}^{(i,j)}(x=X_d-\varepsilon)\cong$$

$$-\frac{D_g^{(i,j)}}{X_d}\left[3\phi_g^{(i,j)}(X_d,0)+3\overline{\phi}_g^{(i,j)}(p=1)-6\overline{\phi}_g^{(i,j)}\right] \tag{3.38}$$

Using the same argument as above, the surface-averaged current can also be written for x=X$_d$ in terms of node (i+3,j+1) quantities. This is done by evaluating eqn. (3.36) at x=X$_d$ + ε and substituting the result into eqn. (3.9). This gives:

$$\overline{J}_{g,X_1}^{(i+3,j+1)}(x = X_d + \varepsilon) \cong$$
$$-\frac{D_g^{(i+3,j+1)}}{X_d}\left[-3\phi_g^{(i+3,j+1)}(X_d,0) - 3\overline{\phi}_g^{(i+3,j+1)}(p=2) + 6\overline{\overline{\phi}}_g^{(i+3,j+1)}\right] \qquad (3.39)$$

The surface-averaged current for face p=2 of node (i+3,j+1) can be written by evaluating (3.36) at x=0 and inserting the result into eqn. (3.9). This gives:

$$\overline{J}_{g,X_1}^{(i+3,j+1)}(p=2) \cong$$
$$-\frac{D_g^{(i+3,j+1)}}{X_d}\left[\phi_g^{(i+3,j+1)}(X_d,0) + 5\overline{\phi}_g^{(i+3,j+1)}(p=2) - 6\overline{\overline{\phi}}_g^{(i+3,j+1)}\right] \qquad (3.40)$$

The goal is to obtain an expression for the surface-averaged current for the right side of the face at x=0 that does not contain the flux at the point x=X$_d$, nor the surface-averaged flux at x=2X$_d$. The procedure is outlined as follows:

1) Apply a continuity of current and continuity of flux at the point x=X$_d$. As the distance ε shrinks to zero, the small surface for which eqn. (3.38) was written approaches x=X$_d$ from the left and the small surface for eqn. (3.39) approaches from the right. We assume that these surfaces become one surface, and the current is continuous across that one face. Use equations (3.38) and (3.39) to arrive at (3.41):

$$\phi_g(X_d,0) \cong \frac{2\left[D_g^{(i+3,j+1)}\overline{\overline{\phi}}_g^{(i+3,j+1)} + D_g^{(i,j)}\overline{\overline{\phi}}_g^{(i,j)}\right]}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)}\right]}$$
$$-\frac{\left[D_g^{(i+3,j+1)}\overline{\phi}_g^{(i+3,j+1)}(p=2) + D_g^{(i,j)}\overline{\phi}_g^{(i,j)}(p=1)\right]}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)}\right]} \qquad (3.41)$$

where, because of the continuity of flux at x=X$_d$,

$$\phi_g(X_d,0) \equiv \phi_g^{(i+3,j+1)}(X_d,0) = \phi_g^{(i,j)}(X_d,0) \qquad (3.42)$$

There is no theoretical justification for applying the continuity of current at x=X$_d$, but results in Chapter 6 should testify to the adequacy of this approximation.

2) Substitute Eqn. (3.41) into (3.34) to obtain (3.43):

$$\begin{aligned} \frac{X_d}{D_g^{(i,j)}} \bar{J}_{g,X_1}^{(i,j)}(p=1) &\cong 5\bar{\phi}_g^{(i,j)}(p=1) - 6\bar{\bar{\phi}}_g^{(i,j)} \\ &+ \frac{2\left[D_g^{(i+3,j+1)}\bar{\bar{\phi}}_g^{(i+3,j+1)} + D_g^{(i,j)}\bar{\bar{\phi}}_g^{(i,j)}\right]}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)}\right]} \\ &- \frac{\left[D_g^{(i+3,j+1)}\bar{\phi}_g^{(i+3,j+1)}(p=2) + D_g^{(i,j)}\bar{\phi}_g^{(i,j)}(p=1)\right]}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)}\right]} \end{aligned} \qquad (3.43)$$

3) Apply a continuity of flux at x=X$_d$. Add eqns. (3.34) and (3.40) and solve for the surface-averaged flux at x=2X$_d$. This gives:

$$\begin{aligned} \bar{\phi}_g^{(i+3,j+1)}(p=2) &\cong \bar{\phi}_g^{(i,j)}(p=1) - \frac{6}{5}\left[\bar{\bar{\phi}}_g^{(i,j)} - \bar{\bar{\phi}}_g^{(i+3,j+1)}\right] \\ &- \frac{X_d}{5}\left[\frac{\bar{J}_{g,X_1}^{(i,j)}(p=1)}{D_g^{(i,j)}} + \frac{\bar{J}_{g,X_1}^{(i+3,j+1)}(p=2)}{D_g^{(i+3,j+1)}}\right] \end{aligned} \qquad (3.44)$$

4) Substitute eqn. (3.44) into (3.43). This gives:

$$
\begin{aligned}
&\frac{X_d}{20D_g^{(i,j)}}\left[\frac{4D_g^{(i+3,j+1)}+5D_g^{(i,j)}}{D_g^{(i+3,j+1)}+D_g^{(i,j)}}\right]\bar{J}_{g,x_1}^{(i,j)}(p=1)\cong\bar{\phi}_g^{(i,j)}(p=1)\\
&+\frac{X_d}{20}\frac{\bar{J}_{g,x_1}^{(i+3,j+1)}(p=2)}{\left[D_g^{(i+3,j+1)}+D_g^{(i,j)}\right]}-\frac{\left[20D_g^{(i,j)}+24D_g^{(i+3,j+1)}\right]}{20\left[D_g^{(i+3,j+1)}+D_g^{(i,j)}\right]}\bar{\phi}_g^{(i,j)}\\
&+\frac{D_g^{(i+3,j+1)}}{5\left[D_g^{(i+3,j+1)}+D_g^{(i,j)}\right]}\bar{\phi}_g^{(i+3,j+1)}
\end{aligned}
\tag{3.45}
$$

3.5.3 Eliminating the Point Flux at $x=-X_d$

The procedure that was outlined in the last sub-section for eliminating the point flux at $x=X_d$ from (3.34) can be followed to eliminate the point flux at $x=-X_d$ from eqn. (3.37). When this is done, we get:

$$
\begin{aligned}
&\frac{X_d}{20D_g^{(i-1,j)}}\left[\frac{4D_g^{(i-4,j-1)}+5D_g^{(i-1,j)}}{D_g^{(i-4,j-1)}+D_g^{(i-1,j)}}\right]\bar{J}_{g,x_1}^{(i-1,j)}(p=2)\cong-\bar{\phi}_g^{(i-1,j)}(p=2)\\
&+\frac{X_d}{20}\frac{\bar{J}_{g,x_1}^{(i-4,j-1)}(p=1)}{\left[D_g^{(i-4,j-1)}+D_g^{(i-1,j)}\right]}+\frac{\left[20D_g^{(i-1,j)}+24D_g^{(i-4,j-1)}\right]}{20\left[D_g^{(i-4,j-1)}+D_g^{(i-1,j)}\right]}\bar{\phi}_g^{(i-1,j)}\\
&-\frac{D_g^{(i-4,j-1)}}{5\left[D_g^{(i-4,j-1)}+D_g^{(i-1,j)}\right]}\bar{\phi}_g^{(i-4,j-1)}
\end{aligned}
\tag{3.46}
$$

3.5.4 Eliminate the Surface-Averaged Flux at $x=0$

Eqns. (3.45) and (3.46) express the surface-averaged current at x=0 in terms of node and surface-averaged quantities to the right of the face at x=0, and the left of the face at x=0, respectively. These two equations can be used to eliminate the surface-averaged flux at x=0 by applying a continuity of flux across the face. Before this is done, however, we once again introduce discontinuity factors for the same reasons as before. Eqn. (3.45) becomes:

$$\frac{X_d f_{g,p=1}^{(i,j)}}{20 D_g^{(i,j)}} \left[\frac{4 D_g^{(i+3,j+1)} + 5 D_g^{(i,j)}}{D_g^{(i+3,j+1)} + D_g^{(i,j)}} \right] \overline{J}_{g,X_1}^{(i,j)}(p=1) = \overline{\phi}_g^{(i,j)}(p=1)$$
$$+ \frac{X_d f_{g,p=1}^{(i,j)}}{20} \frac{\overline{J}_{g,X_1}^{(i+3,j+1)}(p=2)}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} - \frac{f_{g,p=1}^{(i,j)} \left[20 D_g^{(i,j)} + 24 D_g^{(i+3,j+1)} \right]}{20 \left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} \overline{\overline{\phi}}_g^{(i,j)}$$
$$+ \frac{f_{g,p=1}^{(i,j)} D_g^{(i+3,j+1)}}{5 \left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} \overline{\overline{\phi}}_g^{(i+3,j+1)} \qquad (3.47)$$

where $f_{g,p=1}^{(i,j)}$ is the discontinuity factor for the right side of the face at x=0. When the discontinuity factor for the left side of the face at x=0, $f_{g,p=2}^{(i-1,j)}$, is introduced into eqn. (3.46), we get:

$$\frac{X_d f_{g,p=2}^{(i-1,j)}}{20 D_g^{(i-1,j)}} \left[\frac{4 D_g^{(i-4,j-1)} + 5 D_g^{(i-1,j)}}{D_g^{(i-4,j-1)} + D_g^{(i-1,j)}} \right] \overline{J}_{g,X_1}^{(i-1,j)}(p=2) = -\overline{\phi}_g^{(i-1,j)}(p=2)$$
$$+ \frac{X_d f_{g,p=2}^{(i-1,j)}}{20} \frac{\overline{J}_{g,X_1}^{(i-4,j-1)}(p=1)}{\left[D_g^{(i-4,j-1)} + D_g^{(i-1,j)} \right]} + \frac{f_{g,p=2}^{(i-1,j)} \left[20 D_g^{(i-1,j)} + 24 D_g^{(i-4,j-1)} \right]}{20 \left[D_g^{(i-4,j-1)} + D_g^{(i-1,j)} \right]} \overline{\overline{\phi}}_g^{(i-1,j)}$$
$$- \frac{f_{g,p=2}^{(i-1,j)} D_g^{(i-4,j-1)}}{5 \left[D_g^{(i-4,j-1)} + D_g^{(i-1,j)} \right]} \overline{\overline{\phi}}_g^{(i-4,j-1)} \qquad (3.48)$$

Adding eqns. (3.47) and (3.48), with:

$$\overline{\phi}_g^{(i,j)}(p=1) = \overline{\phi}_g^{(i-1,j)}(p=2) \qquad (3.49)$$

gives a higher order expression for the surface-averaged current at x=0 in terms of volume-averaged fluxes, diffusion coefficients, discontinuity factors, X_d, and surface-averaged currents at $-2X_d$ and $2X_d$:

$$
\begin{aligned}
&\overline{J}_{g,X_1}^{(i,j)}(p=1) = \\
&\left\{ \frac{X_d}{20\,D_g^{(i-1,j)}} \left[\frac{5\,D_g^{(i-1,j)} + 4\,D_g^{(i-4,j-1)}}{D_g^{(i-1,j)} + D_g^{(i-4,j-1)}} \right] + \frac{X_d}{20\,D_g^{(i,j)}} \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \left[\frac{5\,D_g^{(i,j)} + 4\,D_g^{(i+3,j+1)}}{D_g^{(i,j)} + D_g^{(i+3,j+1)}} \right] \right\}^{-1} \\
&\cdot \left\{ \begin{aligned}
&- \frac{D_g^{(i-4,j-1)}}{5\left[D_g^{(i-4,j-1)} + D_g^{(i-1,j)} \right]} \overline{\overline{\phi}}_g^{(i-4,j-1)} + \left[\frac{D_g^{(i-1,j)} + 6/5\,D_g^{(i-4,j-1)}}{\left[D_g^{(i-4,j-1)} + D_g^{(i-1,j)} \right]} \right] \overline{\overline{\phi}}_g^{(i-1,j)} \\
&- \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \left[\frac{D_g^{(i,j)} + 6/5\,D_g^{(i+3,j+1)}}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} \right] \overline{\overline{\phi}}_g^{(i,j)} + \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \left[\frac{D_g^{(i+3,j+1)}}{5\left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} \right] \overline{\overline{\phi}}_g^{(i+3,j+1)} \\
&+ \frac{X_d}{20} \frac{\overline{J}_{g,X_1}^{(i-4,j-1)}(p=1)}{\left[D_g^{(i-4,j-1)} + D_g^{(i-1,j)} \right]} + \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \frac{X_d}{20} \frac{\overline{J}_{g,X_1}^{(i+3,j+1)}(p=2)}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]}
\end{aligned} \right\}
\end{aligned}
\tag{3.50}
$$

Completely analogous expressions to eqn. (3.50) can be written for faces p=2 and p=3 of node (i,j). The next subsection will develop analogous expressions for faces 4 and 5.

3.5.5 Quadratic Based Coupling Equations for the Axial Dimension

Polynomial representations for the transverse-averaged flux have been developed and researched in Cartesian geometry for as high as 4th order [Z-1,G-1]. In this work, a quadratic polynomial will be used for the axial dimension.

For face 5 of a node (i,j,k), Fig. 3.12 shows the adjacent node (i,j,k-1), where it has been assumed that the height of both (i,j,k) and (i,j,k-1) is the same and equal to Z_d.

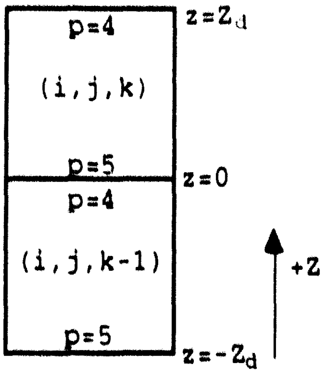


Figure 3.12. Conventions Used for the Quadratic Expansion in the Axial Direction.

Using Fig. 3.12 as a reference, the quadratic form of the transverse-averaged flux for $0 \le z \le Z_d$ turns out to be:

$$\bar{\phi}(z) = \left[1 - \frac{4z}{Z_d} + \frac{3z^2}{Z_d^2}\right]\bar{\phi}(z=0) + \left[-\frac{2z}{Z_d} + \frac{3z^2}{Z_d^2}\right]\bar{\phi}(z=z_d) + \left[\frac{6z}{Z_d} - \frac{6z^2}{Z_d^2}\right]\bar{\bar{\phi}} \tag{3.51}$$

Eqn. (3.51) can be used for $-Z_d \le z \le 0$ by replacing $\bar{\phi}(z=z_d)$ with $\bar{\phi}(z=-z_d)$. Taking the derivative of (3.51), and applying the node and group notation shown in Fig. 3.12 for node (i,j,k), we get:

$$\frac{d\bar{\phi}_g^{(i,j,k)}(z)}{dz} = \frac{1}{Z_d}\left[-4 + \frac{6z}{Z_d}\right]\bar{\phi}_g^{(i,j,k)}(p=5) + \frac{1}{Z_d}\left[-2 + \frac{6z}{Z_d}\right]\bar{\phi}_g^{(i,j,k)}(p=4) + \frac{1}{Z_d}\left[6 - \frac{12z}{Z_d}\right]\bar{\bar{\phi}}_g^{(i,j,k)} \tag{3.52}$$

Evaluating eqn. (3.52) at z=0, and inserting the result into eqn. (3.9), we obtain and equation for the surface-averaged current at face 5 of node (i,j,k):

$$\overline{J}_{g,z}^{(i,j,k)}(p=5) \cong -\frac{D_g^{(i,j,k)}}{Z_d}\left[-4\overline{\phi}_g^{(i,j,k)}(p=5)-2\overline{\phi}_g^{(i,j,k)}(p=4)+6\overline{\overline{\phi}}_g^{(i,j,k)}\right] \qquad (3.53)$$

We can eliminate the surface-averaged flux at face 4 of node (i,j,k) by evaluating (3.52) at z=Z$_d$, inserting the result into eqn. (3.9), and solving the resulting expression for $\overline{\phi}_g^{(i,j,k)}(p=4)$. When this is inserted into (3.53), we get:

$$-\frac{Z_d}{3D_g^{(i,j,k)}}\overline{J}_{g,z}^{(i,j,k)}(p=5) \cong -\overline{\phi}_g^{(i,j,k)}(p=5)+\overline{\overline{\phi}}_g^{(i,j,k)} + \frac{Z_d}{6D_g^{(i,j,k)}}\overline{J}_{g,z}^{(i,j,k)}(p=4) \qquad (3.54)$$

An entirely analogous expression can be obtained for face 4 of node (i,j,k-1). This is simply given as:

$$-\frac{Z_d}{3D_g^{(i,j,k-1)}}\overline{J}_{g,z}^{(i,j,k-1)}(p=4) \cong \overline{\phi}_g^{(i,j,k-1)}(p=4)-\overline{\overline{\phi}}_g^{(i,j,k-1)} + \frac{Z_d}{6D_g^{(i,j,k-1)}}\overline{J}_{g,z}^{(i,j,k-1)}(p=5) \qquad (3.55)$$

Equations (3.54) and (3.55) can be used to eliminate the surface-averaged flux at z=0 by applying a continuity of flux across the face. Before this is done, however, we introduce discontinuity factors to both equations. Equation (3.54) becomes:

$$-\frac{Z_d f_{g,p=5}^{(i,j,k)}}{3D_g^{(i,j,k)}}\overline{J}_{g,z}^{(i,j,k)}(p=5) = -\overline{\phi}_g^{(i,j,k)}(p=5)+f_{g,p=5}^{(i,j,k)}\overline{\overline{\phi}}_g^{(i,j,k)} + \frac{Z_d f_{g,p=5}^{(i,j,k)}}{6D_g^{(i,j,k)}}\overline{J}_{g,z}^{(i,j,k)}(p=4) \qquad (3.56)$$

and eqn. (3.55) becomes:

$$-\frac{Z_d f^{(i,j,k-1)}_{g,p=4}}{3D_g^{(i,j,k-1)}}\bar{J}^{(i,j,k-1)}_{g,z}(p=4)=\bar{\phi}^{(i,j,k-1)}_g(p=4)-f^{(i,j,k-1)}_{g,p=4}\bar{\bar{\phi}}^{(i,j,k-1)}_g$$
$$+\frac{Z_d f^{(i,j,k-1)}_{g,p=4}}{6D_g^{(i,j,k-1)}}\bar{J}^{(i,j,k-1)}_{g,z}(p=5) \qquad (3.57)$$

Adding eqns. (3.56) and (3.57), with:

$$\bar{\phi}^{(i,j,k)}_g(p=5)=\bar{\phi}^{(i,j,k-1)}_g(p=4) \qquad (3.58)$$

we get:

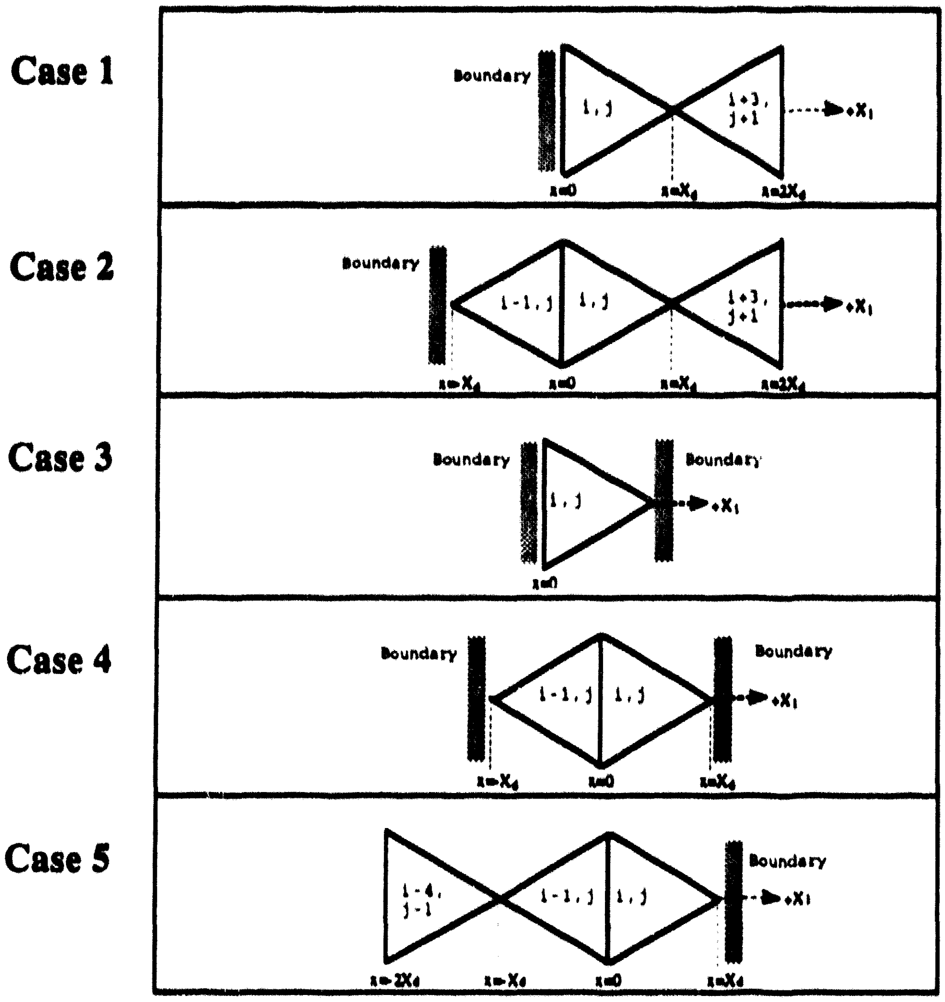
$$\bar{J}^{(i,j,k)}_{g,z}(p=5)=\left\{\frac{Z_d}{3D_g^{(i,j,k-1)}}+\frac{f^{(i,j,k)}_{g,p=5}}{f^{(i,j,k-1)}_{g,p'=4}}\frac{Z_d}{3D_g^{(i,j,k)}}\right\}^{-1}\cdot$$
$$\left\{\begin{array}{l}-\dfrac{f^{(i,j,k)}_{g,p=5}}{f^{(i,j,k-1)}_{g,p'=4}}\bar{\bar{\phi}}^{(i,j,k)}_g+\bar{\bar{\phi}}^{(i,j,k-1)}_g\\ -\dfrac{Z_d}{6D_g^{(i,j,k)}}\dfrac{f^{(i,j,k)}_{g,p=5}}{f^{(i,j,k-1)}_{g,p'=4}}\bar{J}^{(i,j,k)}_{g,z}(p=4)\\ -\dfrac{Z_d}{6D_g^{(i,j,k-1)}}\bar{J}^{(i,j,k-1)}_{g,z}(p=5)\end{array}\right\} \qquad (3.59)$$

Thus, for an internal node, we may replace eqn. (3.16) with eqn. (3.50) for faces 1,2, and 3. For faces 4 and 5, we may use eqn. (3.52). Using these expressions for the surface-averaged current, we obtain an equation describing the neutron balance for a node (i,j,k) which replaces eqn. (3.17). 'he only task left to fully describe the equations which govern the static problem is to treat the case in which node (i,j,k) is on the boundary of the reactor. This will be done in the next section.

3.6 Boundary Conditions for the Quadratic Based Coupling Equations

It turns out that the extended coupling of the quadratic coupling equations in the 2-D plane introduces a number of different boundary conditions which must be addressed before the method can be applied. There are a total of five possible boundary condition situations which may be encountered. For face 1 of a node (i,j,k), these are given in Table 3.2:

Table 3.2
Possible Boundary Condition Cases for Extended Coupling in the 2-D Plane



To derive the quadratic based coupling equations for these five cases, the procedure outlined in the last section is followed. When a boundary is encountered, eqn. (3.18) is used to relate the surface-averaged current to the surface-averaged flux. The derivation for each case will not be given here, but it should be noted that these derivations are somewhat simpler than the fully internal node derivation of the last section. This is due to the fact that in these cases, the surface-averaged fluxes at x=2X$_d$ and/or at x=-2X$_d$ do not appear, and therefore do not need to be eliminated. Also, in the cases where the point of the triangle is on a boundary, the surface-averaged flux is given by the boundary condition and does not need to be eliminated.

3.6.1 Case 1 Boundary Condition

In case 1, the left side of the surface at x=0 is described by eqn. (3.18). The right side of the surface at x=0 is described by eqn. (3.47). Using these, we obtain:

$$J^{(i,j)}_{g,X_1}(p=1)= \\ \beta^{(i,j)}_{g,p=1}\left[\Gamma_1\right]^{-1}\cdot\left\{-\Gamma_2\bar{\bar{\phi}}^{(i,j)}_g+\Gamma_3\bar{\bar{\phi}}^{(i+3,j+1)}_g+\Gamma_3 J^{(i+3,j+1)}_{g,X_1}(p=2)\right\} \tag{3.60}$$

where, to simplify the appearance of (3.60), the following constants have been used:

$$\begin{aligned}
\Gamma_1 &\equiv \frac{X_d}{20}\frac{f^{(i,j)}_{g,p=1}}{D^{(i,j)}_g}\beta^{(i,j)}_{g,p=1}\left[\frac{5D^{(i,j)}_g+4D^{(i+3,j+1)}_g}{\left[D^{(i,j)}_g+D^{(i+3,j+1)}_g\right]}\right]+\alpha^{(i,j)}_{g,p=1}\\
\Gamma_2 &\equiv f^{(i,j)}_{g,p=1}\left[\frac{D^{(i,j)}_g+6/5\,D^{(i+3,j+1)}_g}{\left[D^{(i,j)}_g+D^{(i+3,j+1)}_g\right]}\right]\\
\Gamma_3 &\equiv f^{(i,j)}_{g,p=1}\left[\frac{D^{(i+3,j+1)}_g}{5\left[D^{(i,j)}_g+D^{(i+3,j+1)}_g\right]}\right]\\
\Gamma_4 &\equiv f^{(i,j)}_{g,p=1}\frac{X_d/20}{\left[D^{(i,j)}_g+D^{(i+3,j+1)}_g\right]}
\end{aligned} \tag{3.61}$$

3.6.2 Case 2 Boundary Condition

For case 2, the boundary condition occurs at x=-X$_d$, which is at the point of the triangular node (i-1,j). The boundary condition is therefore applied to a small surface, perpendicular to the X_1 axis, which is located a very small distance, ε , to the right of $x=-X_d$. In this case, eqn. (3.18) is replaced by:

$$\alpha_{g,point}^{(i-1,j)} \underline{J}_g^{(i-1,j)}(\varepsilon\ surface) \cdot \underline{n}_{\varepsilon\ surface} = \beta_{g,point}^{(i-1,j)} \overline{\phi}_g^{(i-1,j)}(\varepsilon\ surface) \qquad (3.62)$$

where $\underline{n}_{\varepsilon\ surface}$ is the outward directed normal to the small ε surface. Using (3.62), the equation for the surface-averaged current turns out to be:

$$\begin{gathered} J_{g,X_1}^{(i,j)}(p=1) = \\ \{\Gamma_1\}^{-1} \cdot \left\{ \Gamma_2 \overline{\phi}_g^{(i-1,j)} - \Gamma_3 \overline{\phi}_g^{(i,j)} + \Gamma_4 \overline{\phi}_g^{(i+3,j+1)} + \Gamma_5 J_{g,X_1}^{(i+3,j+1)}(p=2) \right\} \end{gathered} \qquad (3.63)$$

where, as before, the following constants are used:

$$\begin{aligned} \Gamma_1 &\equiv \frac{X_d}{20 D_g^{(i,j)}} \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \left[\frac{5 D_g^{(i,j)} + 4 D_g^{(i+3,j+1)}}{D_g^{(i,j)} + D_g^{(i+3,j+1)}} \right] \\ &\quad + \frac{X_d}{D_g^{(i-1,j)}} \left[\frac{X_d \beta_{g,point}^{(i-1,j)} + 3 D_g^{(i-1,j)} \alpha_{g,point}^{(i-1,j)}}{5 X_d \beta_{g,point}^{(i-1,j)} + 12 D_g^{(i-1,j)} \alpha_{g,point}^{(i-1,j)}} \right] \\ \Gamma_2 &\equiv \left[\frac{12 D_g^{(i-1,j)} \alpha_{g,point}^{(i-1,j)} + 6 X_d \beta_{g,point}^{(i-1,j)}}{5 X_d \beta_{g,point}^{(i-1,j)} + 12 D_g^{(i-1,j)} \alpha_{g,point}^{(i-1,j)}} \right] \\ \Gamma_3 &\equiv \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \left[\frac{D_g^{(i,j)} + 6/5\, D_g^{(i+3,j+1)}}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} \right] \\ \Gamma_4 &\equiv \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \left[\frac{D_g^{(i+3,j+1)}}{5 \left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} \right] \\ \Gamma_5 &\equiv \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}} \frac{X_d/20}{\left[D_g^{(i+3,j+1)} + D_g^{(i,j)} \right]} \end{aligned} \qquad (3.64)$$

In general, the boundary condition parameters for the outer surfaces of the reactor (which appear in eqn. (3.18)) will be specified by the user, but the boundary condition parameters for each $\mathcal{E}$ surface (i.e. each triangle point which lies on the surface) will not. This is not a problem if the outward directed normal to the $\mathcal{E}$ surface is coincident with the outward directed normal to the reactor surface. In that case, the user specified parameters may be used in eqn. (3.62). Unfortunately, the normal to the $\mathcal{E}$ surface is often *not* coincident with the outward directed normal to the reactor surface. Fig. 3.13 illustrates one such occurrence, where the +X axis points in the direction of the outward normal to the $\mathcal{E}$ surface at the point.

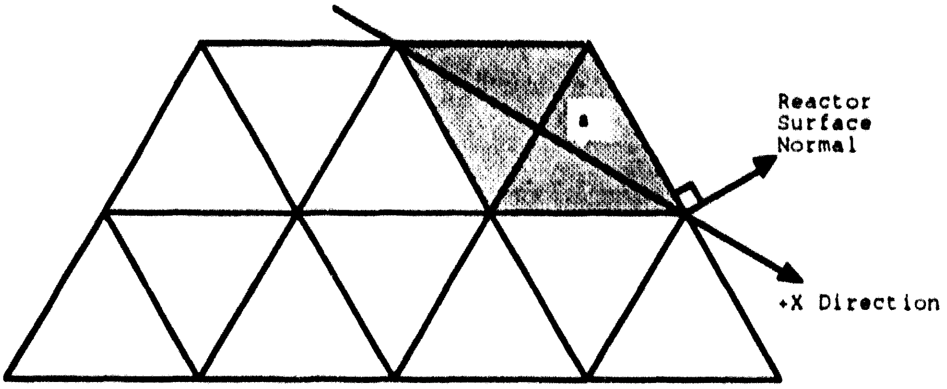


Figure 3.13. Discrepancy Between the Normal to the Reactor Surface and the +X Direction for the Point of Node a.

In addition to the situation illustrated in Fig. 3.13, there are other occurrences in which the relationship between the reactor surface normal and the $\mathcal{E}$ surface normal is different. It is not practical to require the user to input the boundary conditions for every 'triangle point' on the reactor surface, so the parameters at the point must either be obtained from the user input parameters, or an approximation must be made that the parameters at the point are *equal to* the user input parameters. This matter will be discussed further when results are presented. Appendix A briefly outlines a technique for using the known boundary condition parameters at the surface of the reactor for the parameters at a point.

3.6.3 Case 3 Boundary Condition

In case 3, there are two boundary conditions. The left boundary occurs at face 1 of node (i,j), and the right boundary occurs at the point of node (i,j). The surface-averaged current for this case is given by:

$$\overline{J}_{g,X_1}^{(i,j)}(p=1) = -\beta_{g,p=1}^{(i,j)}\left[\Gamma_1\right]^{-1}\cdot\left\{\Gamma_2\overline{\phi}_g^{(i,j)}\right\} \tag{3.65}$$

where the constants are defined as:

$$\begin{aligned}\Gamma_1 &\equiv \frac{X_d}{D_g^{(i,j)}} f_{g,p=1}^{(i,j)}\beta_{g,p=1}^{(i,j)}\left[\frac{X_d\beta_{g,point}^{(i,j)} + 3D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}{5X_d\beta_{g,point}^{(i,j)} + 12D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}\right] + \alpha_{g,p=1}^{(i,j)}\\ \Gamma_2 &\equiv f_{g,p=1}^{(i,j)}\left[\frac{6X_d\beta_{g,point}^{(i,j)} + 12D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}{5X_d\beta_{g,point}^{(i,j)} + 12D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}\right]\end{aligned} \tag{3.66}$$

Note the mixture of different boundary condition parameters.

3.6.4 Case 4 Boundary Condition

In case 4 there are two boundaries, each at a point. The surface-averaged current is:

$$\overline{J}_{g,X_1}^{(i,j)}(p=1) = \left[\Gamma_1\right]^{-1}\cdot\left\{-\Gamma_2\overline{\phi}_g^{(i,j)} + \Gamma_3\overline{\phi}_g^{(i-1,j)}\right\} \tag{3.67}$$

The constants are:

$$\begin{aligned}\Gamma_1 &\equiv \frac{X_d}{D_g^{(i,j)}}\frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}}\left[\frac{X_d\beta_{g,point}^{(i,j)} + 3D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}{5X_d\beta_{g,point}^{(i,j)} + 12D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}\right]\\ &\quad + \frac{X_d}{D_g^{(i-1,j)}}\left[\frac{X_d\beta_{g,point}^{(i-1,j)} + 3D_g^{(i-1,j)}\alpha_{g,point}^{(i-1,j)}}{5X_d\beta_{g,point}^{(i-1,j)} + 12D_g^{(i-1,j)}\alpha_{g,point}^{(i-1,j)}}\right]\\ \Gamma_2 &\equiv \frac{f_{g,p=1}^{(i,j)}}{f_{g,p=2}^{(i-1,j)}}\left[\frac{6X_d\beta_{g,point}^{(i,j)} + 12D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}{5X_d\beta_{g,point}^{(i,j)} + 12D_g^{(i,j)}\alpha_{g,point}^{(i,j)}}\right]\\ \Gamma_3 &\equiv \left[\frac{6X_d\beta_{g,point}^{(i-1,j)} + 12D_g^{(i-1,j)}\alpha_{g,point}^{(i-1,j)}}{5X_d\beta_{g,point}^{(i-1,j)} + 12D_g^{(i-1,j)}\alpha_{g,point}^{(i-1,j)}}\right]\end{aligned} \tag{3.68}$$

3.6.5 Case 5 Boundary Condition

The surface-averaged current for boundary condition case 5 was not derived. It turns out that a case 5 from the point of view of node (i,j) is identical to a case 2 from the point of view of node (i-1,j). Therefore, if a case 5 is encountered for node (i,j), the node of interest is switched to the adjacent node, (i-1,j), and a case 2 is solved.

3.6.6 Boundary Condition for the Axial Dimension

In the axial dimension, a boundary could occur at face 4 or face 5 of node (i,j,k). The case of a boundary at face 5 is depicted in Fig. 3.14.

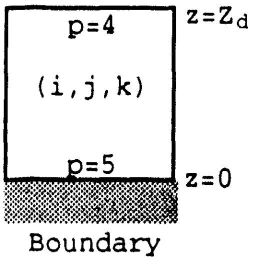


Figure 3.14. Boundary Condition for Face 5.

For this case, the surface-averaged current becomes:

$$\bar{J}_{g,z}^{(i,j,k)}(p=5) = \beta_{g,p=5}^{(i,j,k)} \left\{ \alpha_{g,p=5}^{(i,j,k)} + f_{g,p=5}^{(i,j,k)} \beta_{g,p=5}^{(i,j,k)} \frac{Z_d}{3D_g^{(i,j,k)}} \right\}^{-1} \cdot \left\{ \begin{array}{l} -f_{g,p=5}^{(i,j,k)} \bar{\phi}_g^{(i,j,k)} \\ -\dfrac{Z_d}{6D_g^{(i,j,k)}} f_{g,p=5}^{(i,j,k)} \bar{J}_{g,z}^{(i,j,k)}(p=4) \end{array} \right\} \qquad (3.69)$$

The surface-averaged current when face 4 is on the boundary is:

$$\overline{J}_{g,z}^{(i,j,k)}(p=4) = \beta_{g,p=4}^{(i,j,k)} \left\{ \alpha_{g,p=4}^{(i,j,k)} + f_{g,p=4}^{(i,j,k)} \beta_{g,p=4}^{(i,j,k)} \frac{Z_d}{3 D_g^{(i,j,k)}} \right\}^{-1} \cdot \left\{ \begin{aligned} & -f_{g,p=4}^{(i,j,k)} \overline{\overline{\phi}}_g^{(i,j,k)} \\ & -\frac{Z_d}{6 D_g^{(i,j,k)}} f_{g,p=4}^{(i,j,k)} \overline{J}_{g,z}^{(i,j,k)}(p=5) \end{aligned} \right\} \tag{3.70}$$

3.7 Summary

In this chapter, the static form of the nodal diffusion theory equations were derived for equilateral triangles. The Coarse Mesh Finite Difference (CMFD) equations were derived first, utilizing Fick's Law and the finite difference approximation to discretize the equations in space. Discontinuity Factors were introduced to correct for errors in homogenization, the use of Fick's Law, and the spatial approximation of assuming that the flux is linear within a node. It was argued that the spatial approximation could be improved by modeling the flux as a quadratic polynomial within the node. Quadratic based coupling equations were then derived with this model, and discontinuity factors were once again introduced. It was mentioned that a difficulty was encountered with the specification of boundary conditions at the points of nodes on the edge of the reactor. This topic will be discussed further in a later chapter.

In the next chapter, the transient equations will be derived. Chapter 5 will outline solution methods for both static and transient problems.

Chapter 4

Derivation of the Transient Nodal Diffusion Theory Equations for Equilateral Triangles

4.1 Introduction

In this chapter, the discreet form of the transient nodal diffusion equations will be derived. We shall start with the semi-discreet equations (2.5.a) and (2.5.b), and apply the same procedure to the leakage terms as was presented in Chapter 3. This will lead to a set of equations which describe the overall neutron balance for each node in terms of node- and group- averaged fluxes, precursor densities, discontinuity factor ratios, and constants. All will depend upon the continuous time variable, t. In order to simplify the notation, these equations will then be put in matrix form. The theta difference approximation will be used to discretize the prompt equation in time, and a direct integration method will be used to discretize the precursor equations.

4.2 Time Dependent CMFD and Quadratic Equations

Eqns. (2.5.a) and (2.5.b) describe the time dependent neutron balance for a homogenized node, l, and energy group, g. In Chapter 3, these equations were written in steady state form for a node (i,j,k), group g, and discretized using the finite difference approximation. In this section, we apply the results of Chapter 3 to (2.5.a) and (2.5.b) and obtain the time dependent version of the coarse mesh finite difference equations.

In the time dependent case, the cross sections and discontinuity factors become time-varying. The time-dependent expression for the surface-averaged current for face p of node (i,j,k) is:

$$\bar{J}_{g,X_p}^{(i,j,k)}(p,t)=\frac{2}{X_c}\left[\frac{1}{D_g^{(i,j,k)}(t)}+\frac{1}{D_g^{(i-1,j,k)}(t)}\frac{f_{g,p'}^{(i-1,j,k)}(t)}{f_{g,p}^{(i,j,k)}(t)}\right]^{-1} \cdot\left[\frac{f_{g,p'}^{(i-1,j,k)}(t)}{f_{g,p}^{(i,j,k)}(t)}\bar{\phi}_g^{(i-1,j,k)}(t)-\bar{\phi}_g^{(i,j,k)}(t)\right] \tag{4.1}$$

Equation (4.1) can be written for each face, p, of a fully internal node. If a face, p, is on the external boundary of the reactor, the expression becomes:

$$\bar{J}_{g,X_p}^{(i,j,k)}(p,t)=-\frac{2}{X_c}\left[\frac{\beta_{g,p}^{(i,j)}}{D_g^{(i,j,k)}(t)}+\frac{2}{X_c}\frac{\alpha_{g,p=1}^{(i,j,k)}}{f_{g,p}^{(i,j,k)}(t)}\right]^{-1} \cdot\left[\beta_{g,p=1}^{(i,j)}\bar{\phi}_g^{(i,j,k)}(t)\right] \tag{4.2}$$

where the boundary coefficient terms are assumed to be time independent.

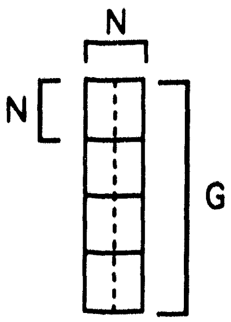
Inserting eqns. (4.1) and (4.2) for faces of node (i,j,k) in the 2-D plane (p=1,2,3), and appropriate expressions for the axial dimension into (2.5.a), the time-dependent version of the coarse mesh finite difference equations are obtained.

The development of Section 3.5 for the quadratic based coupling equations is also valid in the time-dependent case. The cross sections and discontinuity factors are again time-varying. For the case of a fully internal node, the diffusion coefficients, discontinuity factors, volume averaged fluxes, and surface-averaged currents of eqn. (3.50) become dependent on the time variable, t. This is also true for the boundary condition cases of Section 3.6, except that the boundary condition parameters are assumed to be time independent.

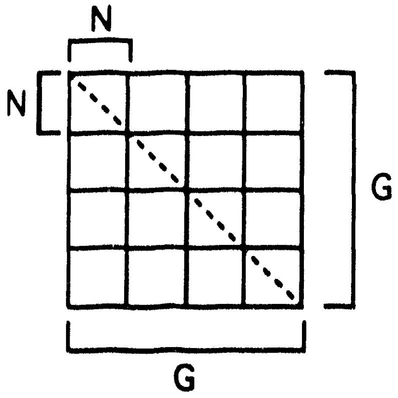
4.3 Time Dependent Equations in Matrix Form

In order to simplify the notation of the transient equations for subsequent discussions, matrix notation will be used. A bold, uppercase letter will be used to represent matrices. In this section, the definition of each matrix will also include an illustration which shows the structure of each matrix.

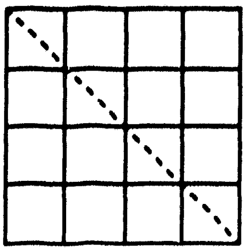
G total number of energy groups.
N total number of nodes.



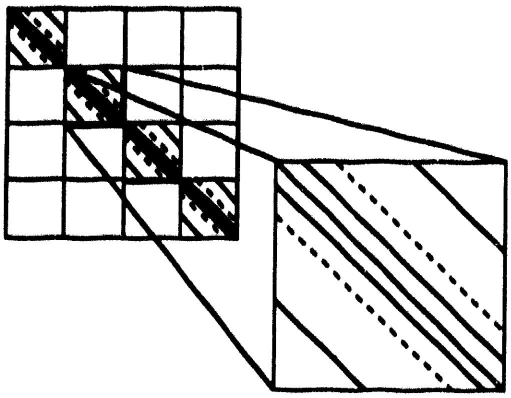
$\mathbf{\Phi}$ a G-element supervector consisting of N-element subvectors of the nodal fluxes $\overline{\phi}_g^{(i,j,k)}(t)$.



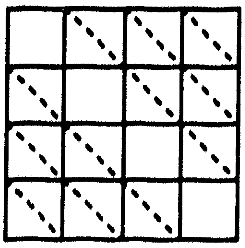
$\mathbf{V}^{-1}$ a GxG supermatrix consisting of NxN element submatrices. The diagonal elements of the diagonal submatrices are the inverse group speeds, $\frac{1}{v_g}$.



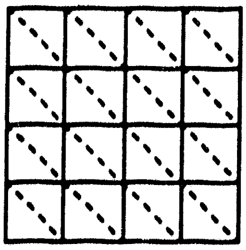
$\mathbf{\Sigma}$ a GxG supermatrix consisting of NxN element submatrices. The diagonal elements of the diagonal submatrices are the total minus the in-group scattering cross sections, $\Sigma_g^{(i,j,k)}(t)$.



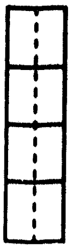
D a GxG supermatrix consisting of NxN element submatrices. Each diagonal submatrix is an NxN, 7 stripe matrix (5 stripe for 2-D problems) of the coupling terms. The two solid lines on either side of the diagonal represent coupling through face 1 and 2. The broken lines represent coupling through face 3, which may be above or below. The solid lines furthest from the diagonal represent coupling to the plane above (face 4) and below (face 5).



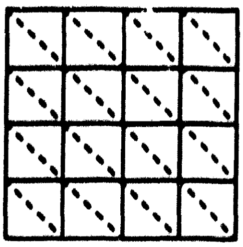
$\boldsymbol{\Sigma}_{\mathbf{scat}}$ a GxG supermatrix consisting of NxN element submatrices. The diagonal elements of the off-diagonal submatrices are the scattering cross sections for group g' to g, $\Sigma_{gg'}^{(i,j,k)}(t) \quad g' \neq g$.



$\mathbf{M}_{\mathbf{p}}$ a GxG supermatrix consisting of NxN element submatrices. The diagonal elements of each submatrix are the prompt fission operator, $\chi_{p,g}\left(1-\beta_{tot}\right)\nu\Sigma_{fg'}^{(i,j,k)}(t)$.



$\mathbf{C}_{\mathbf{iprec}}$ a G-element supervector of N-element subvectors. Each subvector consists of the node averaged precursor concentrations for family *iprec* multiplied by the delayed neutron spectrum, $\chi_{d,g}\overline{\overline{C}}_{iprec}^{(i,j,k)}(t)$.



$\mathbf{M}_{\mathbf{iprec}}$ a GxG supermatrix consisting of NxN element submatrices. The diagonal elements of each submatrix are the delayed fission operator for family *iprec*, $\chi_{d,g}\beta_{iprec}\nu\Sigma_{fg'}^{(i,j,k)}(t)$.

In addition, we define:

$$\mathbf{L} \equiv \mathbf{D} + \mathbf{\Sigma} - \mathbf{\Sigma}_{\mathbf{scat}} \tag{4.5}$$

Combining these matrix definitions with the time dependent CMFD or quadratic equations, we get:

$$\mathbf{V}^{-1}\left[\frac{d\mathbf{\Phi}(t)}{dt}\right] = \left[\mathbf{M}_{\mathbf{p}}(t) - \mathbf{L}(t)\right]\mathbf{\Phi}(t) + \sum_{iprec=1}^{nprec}\lambda_{iprec}\mathbf{C}_{\mathbf{iprec}}(t) \tag{4.6.a}$$

$$\frac{d\mathbf{C}_{\mathbf{iprec}}(t)}{dt} = \mathbf{M}_{\mathbf{iprec}}(t)\mathbf{\Phi}(t) - \lambda_{iprec}\mathbf{C}_{\mathbf{iprec}}(t) \quad iprec = 1, nprec \tag{4.6.b}$$

4.4 Time Integration Scheme

4.4.1 Theta Difference of the Prompt Equation

The theta difference method expresses the flux at the new time step (n+1) in terms of a combination of the last time step values and new time step values. The most common values of θ are $\theta = 0$, which gives a fully explicit method, $\theta = 1/2$, which gives the Crank-Nicholson method, and $\theta = 1$, which gives a fully implicit method.

$$\begin{aligned}\mathbf{V}^{-1}\left[\frac{\mathbf{\Phi}^{(n+1)} - \mathbf{\Phi}^{(n)}}{\Delta t_n}\right] &= \theta\left\{\left[\mathbf{M}_{\mathbf{p}}^{(n+1)} - \mathbf{L}^{(n+1)}\right]\mathbf{\Phi}^{(n+1)} + \sum_{iprec=1}^{nprec}\lambda_{iprec}\mathbf{C}_{\mathbf{iprec}}^{(n+1)}\right\} \\ &+ (1-\theta)\left\{\left[\mathbf{M}_{\mathbf{p}}^{(n)} - \mathbf{L}^{(n)}\right]\mathbf{\Phi}^{(n)} + \sum_{iprec=1}^{nprec}\lambda_{iprec}\mathbf{C}_{\mathbf{iprec}}^{(n)}\right\}\end{aligned} \tag{4.7}$$

4.4.2 Direct Integration of the Delayed Neutron Precursor Equations

The theta difference method could be used on the delayed neutron precursor equations. However, due to the form of these equations, it is possible to use an integrating factor and integrate the equations directly.

The first step is to define the following:

$$\begin{aligned} u &\equiv \lambda_{iprec}\left(t - t_n\right) \\ u_{max} &\equiv \lambda_{iprec}\left(t_{n+1} - t_n\right) \end{aligned} \qquad (4.8)$$

Also define the time dependent vector, $\mathbf{B}(t)$, for each precursor family:

$$\mathbf{B}_{\mathbf{iprec}}(t) \equiv \mathbf{M}_{\mathbf{iprec}}(t)\mathbf{\Phi}(t) \qquad iprec = 1, nprec \qquad (4.9)$$

Substituting these definitions, we obtain:

$$\frac{d\mathbf{C}_{\mathbf{iprec}}(u)}{du} + \mathbf{C}_{\mathbf{iprec}}(u) = \frac{1}{\lambda_{iprec}}\mathbf{B}_{\mathbf{iprec}}(u) \qquad iprec = 1, nprec \qquad (4.10)$$

Using the integrating factor e^u, we can write (4.10) as:

$$e^{-u}\frac{d}{du}\left[e^u\mathbf{C}_{\mathbf{iprec}}(u)\right] = \frac{1}{\lambda_{iprec}}\mathbf{B}_{\mathbf{iprec}}(u) \qquad iprec = 1, nprec \qquad (4.11)$$

Integrating (4.11) from $u = 0$ to $u = u_{max}$, we get:

$$e^{u_{max}}\mathbf{C}_{\mathbf{iprec}}\left(u_{max}\right) = C_{\mathbf{iprec}}(0) + \frac{1}{\lambda_{iprec}}\int_0^{u_{max}} e^u\mathbf{B}_{\mathbf{iprec}}(u)du \qquad iprec = 1, nprec \qquad (4.12)$$

In order to evaluate the integral on the right hand side of (4.12), we need to know the function $\mathbf{B}_{iprec}(u)$ across the time step from $u = 0$ to $u = u_{max}$. We will assume that the function is linear across the time step:

$$\mathbf{B}_{\mathbf{iprec}}(u)=\left[1-\frac{u}{u_{max}}\right]\mathbf{B}_{\mathbf{iprec}}(0)+\left[\frac{u}{u_{max}}\right]\mathbf{B}_{\mathbf{iprec}}(u_{max}) \tag{4.13}$$

Substituting this expression for $\mathbf{B}_{iprec}(u)$ into (4.12) and integrating yields:

$$\begin{aligned}\mathbf{C}_{\mathbf{iprec}}(u_{max}) = e^{-u_{max}}\mathbf{C}_{\mathbf{iprec}}(0) &+ \left[\frac{\left(1-e^{-u_{max}}\right)}{u_{max}}-e^{-u_{max}}\right]\frac{\mathbf{B}_{\mathbf{iprec}}(0)}{\lambda_{iprec}} \\ &+\left[1-\frac{\left(1-e^{-u_{max}}\right)}{u_{max}}\right]\frac{\mathbf{B}_{\mathbf{iprec}}(u_{max})}{\lambda_{iprec}}\end{aligned} \tag{4.14}$$

Returning to the original notation, and defining the following functions which depend only on λ_{iprec} and the time step size $t_{n+1}-t_n$:

$$\begin{aligned} g_1(iprec,\Delta t_n) &\equiv e^{-\lambda_{iprec}(t_{n+1}-t_n)} \\ g_2(iprec,\Delta t_n) &\equiv \frac{\left(1-e^{-\lambda_{iprec}(t_{n+1}-t_n)}\right)}{\lambda_{iprec}\left(t_{n+1}-t_n\right)}-e^{-\lambda_{iprec}(t_{n+1}-t_n)} \\ g_3(iprec,\Delta t_n) &\equiv 1-\frac{\left(1-e^{-\lambda_{iprec}(t_{n+1}-t_n)}\right)}{\lambda_{iprec}\left(t_{n+1}-t_n\right)}\end{aligned} \tag{4.15}$$

we obtain:

$$\begin{aligned}\mathbf{C}_{\mathbf{iprec}}^{(n+1)} &= g_1(iprec,\Delta t_n)\mathbf{C}_{\mathbf{iprec}}^{(n)} \\ &+ g_2(iprec,\Delta t_n)\frac{\mathbf{M}_{\mathbf{iprec}}^{(n)}\mathbf{\Phi}^{(n)}}{\lambda_{iprec}} \\ &+ g_3(iprec,\Delta t_n)\frac{\mathbf{M}_{\mathbf{iprec}}^{(n+1)}\mathbf{\Phi}^{(n+1)}}{\lambda_{iprec}}\end{aligned} \tag{4.16}$$

4.5 Summary

In this chapter, the time dependent nodal balance equation was fully discretized. The spatial discretization was achieved by applying the results of Chapter 3, but with time dependent cross sections and discontinuity factors. The temporal discretization of the prompt equation was achieved by using the theta difference method, and the delayed neutron precursor equation was treated using a direct integration technique.

The equations describing the static and transient neutron balance for nodes of triangular-z geometry have now been completely developed. The next chapter will discuss solution methods.

Chapter 5

Solution Methods for Static and Transient Problems

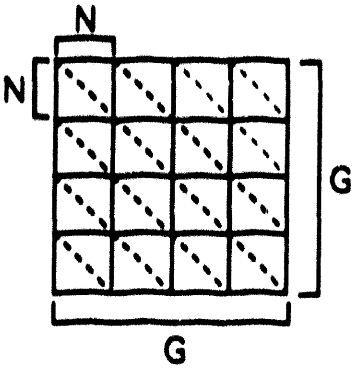
5.1 Introduction

In previous chapters, the equations describing the static and dynamic neutron balance in each node were derived. In this chapter, the solution of these equations will be discussed. The material in this chapter closely follows Chapter 4 of Gehin [G-1].

5.2 Solution of the Static Equations

5.2.1 <u>Numerical Solution of the CMFD Equations</u>

The Coarse Mesh Finite Difference equations, derived in Chapter 3, can be put into matrix form using the applicable definitions from Chapter 4. If, in addition to the loss matrix **L**, we define a total fission operator **M** as:



M a GxG supermatrix consisting of NxN element submatrices. The diagonal elements of each submatrix are the total fission operator, $\chi_g \nu\Sigma_{fg'}^{(i,j,k)}(t)$, where χ_g is defined by eqn. (3.1.b).

we can write:

$$\mathbf{L\Phi} = \frac{1}{\lambda}\mathbf{M\Phi} \qquad (5.1)$$

This is an eigenvalue problem with the largest eigenvalue being equal to the critical eigenvalue for the reactor, and the associated, all positive eigenvector being equal to the solution of the neutron flux in the reactor.

A common and effective solution method for eigenvalue problems is the power iteration method [N-1]. The power iteration method estimates the fission source, on the right side of eqn. (5.1), using the flux and eigenvalue estimate from the last iteration:

$$\mathbf{L}\mathbf{\Phi}^{(m+1)} = \frac{1}{\lambda^{(m)}}\mathbf{M}\mathbf{\Phi}^{(m)} \tag{5.2}$$

where m refers to the iteration number. During the solution process, a new eigenvalue can be estimated using the Rayleigh quotient [N-1]:

$$\lambda^{(m+1)} = \lambda^{(m)}\frac{\left\langle \mathbf{w},\mathbf{\Phi}^{(m+1)}\right\rangle}{\left\langle \mathbf{w},\mathbf{\Phi}^{(m)}\right\rangle} \tag{5.3}$$

where the brackets in eqn. (5.3) indicate integration over energy and space, and the weight vector, **w** is arbitrary but does affect the convergence of the problem. A good choice for the weight vector has been found to be the flux (or fission source) from the last iteration.

At each iteration, the matrix **L** must be inverted to obtain the new solution vector, $\mathbf{\Phi}^{(m+1)}$. For problems with greater than two groups, this inversion is best achieved using iterative methods. In this implementation, an iterative method is always used, and thus a 2 level iteration scheme is used to solve the CMFD equations. The power iteration scheme is commonly referred to as the *outer* iteration, and the iteration scheme to invert **L** is referred to as *inner* or *flux* iterations.

5.2.2 Convergence of the Iterative Methods

The iteration matrix associated with eqn. (5.1) is found by multiplying both sides of (5.1) by the inverse of the loss matrix:

$$\mathbf{\Phi}^{(m+1)} = \frac{1}{\lambda}\mathbf{R}\mathbf{\Phi}^{(m)} \tag{5.4}$$

where:

$$\mathbf{R} \equiv \mathbf{L}^{-1}\mathbf{M} \tag{5.5}$$

Recall that the mesh-centered finite difference equations can be obtained from the Coarse Mesh Finite Difference equations by setting all of the discontinuity factors to unity. The **L** matrix for the mesh-centered equations can be shown to be real, irreducible, and diagonally dominant [N-1]. In addition, the diagonal elements of **L** are positive and the off-diagonal elements are non-positive (a matrix with these properties is an "S-matrix"). The inverse therefore exists and has all positive elements [N-1]. The fission matrix, **M**, is a non-symmetric, non-negative matrix. The resulting **R**matrix, therefore, is a non-symmetric, non-negative matrix. The convergence of the outer iterations (to the largest eigenvalue and its associated, all positive eigenvector) using the power method is guaranteed when the **R** matrix is irreducible and non-negative [N-1].

A method for accelerating the convergence rate of the outer iterations is Wielandt's Method [N-1]. In this method, the power iteration method is re-formulated by subtracting the quantity:

$$\left(\frac{1}{\lambda'}\mathbf{M}\right)\mathbf{\Phi} \tag{5.6}$$

from both sides of eqn. (5.2) and defining:

$$\frac{1}{\Lambda} \equiv \frac{1}{\lambda} - \frac{1}{\lambda'} \tag{5.7}$$

where λ' is known as the eigenvalue shift. This gives:

$$\left(\mathbf{L}-\frac{1}{\lambda'}\mathbf{M}\right)\mathbf{\Phi}^{(m+1)}=\frac{1}{\Lambda^{(m)}}\mathbf{M}\mathbf{\Phi}^{(m)} \tag{5.8}$$

The new eigenvalue problem is solved with:

$$\Lambda^{(m+1)}=\Lambda^{(m)}\frac{\left\langle \mathbf{w},\mathbf{\Phi}^{(m+1)}\right\rangle}{\left\langle \mathbf{w},\mathbf{\Phi}^{(m)}\right\rangle} \tag{5.9}$$

and the eigenvalue we seek can be found by:

$$\lambda^{(m+1)}=\frac{\Lambda^{(m+1)}\lambda'}{\Lambda^{(m+1)}+\lambda'} \tag{5.10}$$

The benefit of Wielandt's method can be seen by comparing the dominance ratio of the shifted problem (i.e. eqn. (5.8)) to the unshifted problem. The dominance ratio is a measure of the asymptotic convergence rate of the outer iterations. The dominance ratio for the power method is given by the ratio of the second largest eigenvalue to the largest eigenvalue:

$$d\equiv\frac{|\lambda_1|}{|\lambda|} \tag{5.11}$$

The closer the dominance ratio is to unity, the slower the convergence rate. The dominance ratio for the shifted problem is:

$$d\equiv\left[\frac{1/\lambda-1/\lambda'}{1/\lambda_1-1/\lambda'}\right] \tag{5.12}$$

which will be less than the dominance ratio of the unshifted problem if λ' is larger than λ.

The nature of the acceleration method, as proposed by eqn. (5.8), is such that all the energy groups must be solved simultaneously. Sutton [S-1] has demonstrated that the

Wielandt acceleration procedure can be adapted to group-wise solution methods by introducing a diagonal NxN matrix of "spectrum ratios." These spectrum ratios are the group g' to group g fluxes. Formally, we define an NxN spectrum ratio matrix, **A**, as:

$$\mathbf{A}_{gg'} \equiv \left[\boldsymbol{\phi}_g \mathbf{I}\right]^{-1} \boldsymbol{\phi}_{g'} \qquad (5.13)$$

where $\boldsymbol{\phi}_g$ is an N-length column vector of the group g fluxes. The matrix equation used for the solution of the group g equations becomes:

$$\left(\mathbf{L}_g - \frac{1}{\lambda'}\sum_{g'=1}^{G}\mathbf{M}_{gg'}\mathbf{A}_{gg'}^{(m)}\right)\boldsymbol{\phi}_g^{(m+1)} = \frac{1}{\Lambda^{(m)}}\sum_{g'=1}^{G}\mathbf{M}_{gg'}\boldsymbol{\phi}_{g'}^{(m)} \qquad (5.14)$$

where it can be seen that the spectrum ratios from the last iteration are used to compute the fluxes at the current iteration. Eqns. (5.9) and (5.10) are used to compute the eigenvalue, as before.

The use of Wielandt's method accelerates the convergence of the outer iterations, but decreases the convergence rate of the inner iterations. This is because the matrix to be inverted at each outer iteration (left side of eqn, (5.14)) is *nearly* singular if λ' is close to the current value of λ (the choice of λ' will be discussed shortly). The best overall convergence is achieved if an appropriate value of λ' is chosen such that a balance is obtained between inner and outer convergence rates [S-2].

The inner iterations, used to invert the matrix on the left side of eqn. (5.14), can be solved using common methods for linear systems. The Jacobi method, Gauss-Seidel method, or Successive Over Relaxation (SOR) method (and others) [H-2] could be used and each are guaranteed to converge for irreducibly diagonally dominant matrices. The method which will be used here is the Chebyshev Cyclic Semi-Iterative

(CCSI) method. CCSI is similar to SOR except that instead of a single, constant relaxation parameter, the relaxation parameter is modified from iteration to iteration in order to achieve a better overall convergence rate. Formally, the CCSI method is obtained by partitioning the Jacobi iteration matrix in such a way that the spatial mesh is ordered in a checkerboard of "red" nodes and "black" nodes [H-2,S-2,G-2]. Chebyshev acceleration is then applied to this partitioned matrix. The checkerboard pattern for a triangular mesh is shown in Figure 5.1.

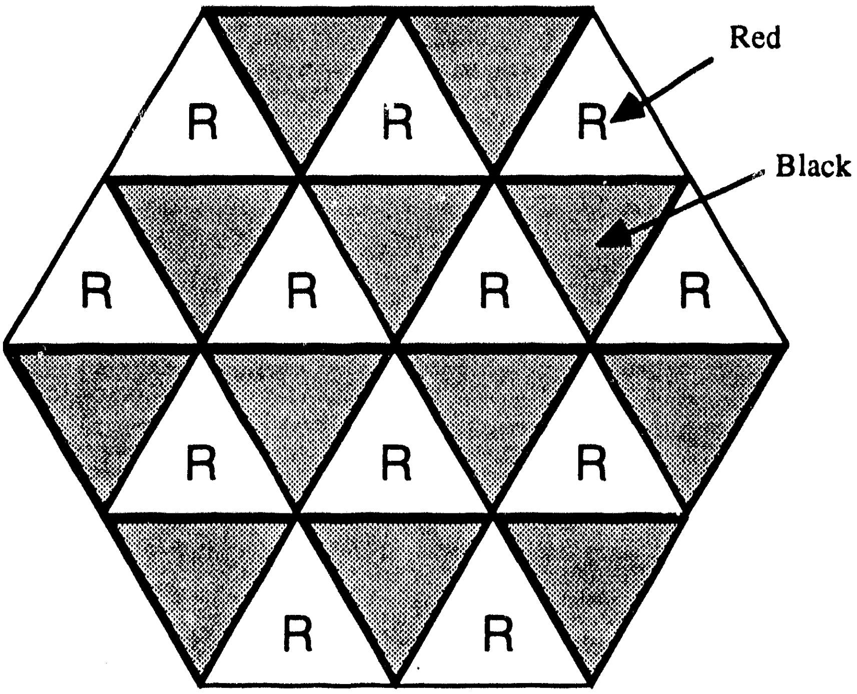


Figure 5.1. Checkerboard Pattern for Inner Iteration Solutions Using CCSI.

It can be seen in Fig. 5.1 that each red node is entirely surrounded by black nodes, and vice versa. The iterative procedure calls for the solution of the red nodes on the first iteration. The next iteration solves for the black nodes using the solution of the reds, and so on. The relaxation parameters for the first iterations are given by:

$$\omega_{red}^{(1)} = 1 \qquad \omega_{black}^{(1)} = \frac{1}{1 - \frac{1}{2}\rho_{Jacobi}^2} \tag{5.15}$$

and for subsequent iterations:

$$\omega_{red}^{(m+1)} = \frac{1}{1 - \frac{1}{4}\rho_{Jacobi}^{2}\omega_{black}^{(m)}} \qquad \omega_{black}^{(m+1)} = \frac{1}{1 - \frac{1}{4}\rho_{Jacobi}^{2}\omega_{red}^{(m+1)}} \tag{5.16}$$

The ρ_{Jacobi} which appears in eqns. (5.15) and (5.16) is the spectral radius of the Jacobi iteration matrix. This spectral radius can be estimated at the beginning of the calculation by performing Jacobi inner iterations and computing:

$$\left(\rho_{Jacobi}\right)^{(m+1)} = \frac{\left\|\Phi^{(m+1)} - \Phi^{(m)}\right\|_1}{\left\|\Phi^{(m)} - \Phi^{(m-1)}\right\|_1} \tag{5.17}$$

where it can be seen that the 1-norm has been used to compute the Jacobi spectral radius. In practice, it is more convenient to perform Gauss-Seidel iterations in order to compute the spectral radius. The Gauss-Seidel spectral radius is equal to the square of the Jacobi spectral radius.

During the solution, the outer iterations are performed until the specified convergence criteria have been met. The criteria used here are the eigenvalue convergence and the convergence of power in the fueled nodes. That is:

$$\left|\lambda^{(m+1)} - \lambda^{(m)}\right| \le \varepsilon_{eig}^{crit}$$

$$\max_i \left|\frac{P_i^{(m+1)} - P_i^{(m)}}{P_i^{(m)}}\right| \le \varepsilon_{pwr}^{crit}$$

where ε_{eig}^{crit} is the user-specified eigenvalue convergence and ε_{pwr}^{crit} is the user-specified convergence of the power in fueled regions. $P_i^{(m+1)}$ is the current iteration's estimate of the power in node i, and $P_i^{(m)}$ is the power in node i from the last outer iteration.

At each outer iteration, the inner iterations must be performed until a specified criterion is met for the solution of the flux. Smith [S-2] has shown that the overall convergence of the problem is optimized if a pre-determined number of inners are performed per outer, rather than converging the inner iterations at every outer. The number of inners per outer is computed at the same time as the spectral radius, and for a user-specified reduction in the error, $\varepsilon_{inners}^{crit}$, this quantity is estimated by [G-1]:

$$n_g \approx \frac{\log\left(\varepsilon_{inners}^{crit}/2\right)}{\log\left(\left(\omega_{best}\right)_g - 1\right)} \tag{5.18}$$

where ω_{best} is the asymptotic relaxation factor for CCSI, which is equal to the "best" relaxation factor for SOR. Appropriate values for the error reduction, $\varepsilon_{inners}^{crit}$, range from 0.1 to 0.4 [S-2].

5.2.3 Numerical Solution of the Quadratic Equations

The quadratic expressions for the surface-averaged current [eqns. (3.50), (3.59), etc.] were derived in Chapter 3 to replace the finite difference expressions for the surface-averaged currents [eqn. (3.16)]. As was mentioned in Section 3.5, the quadratic expressions for the currents could be inserted into the nodal balance equation [eqn. (3.4)], and the result would replace the discretized Coarse Mesh Finite Difference equations [eqn. (3.17)]. The consequences of doing so, however, are substantial. First, the computer memory storage requirements rise dramatically, due to the extended coupling of the quadratic equations, and the need to store the coupling coefficients. Second, the beneficial convergence properties of the CMFD equations, which were outlined in the last sub-section could be lost. A recent implementation of this approach for a quadratic method in hexagonal geometry showed that this solution method can be quite inefficient [S-3].

An improved way proposed by Smith [S-4], and successfully implemented in several computer codes [Z-1,B-1,G-1] is the non-linear iteration method. It proposes that the structure of the CMFD equations be preserved, and the discontinuity factor ratios, which appear in (3.17), be used to correct the CMFD equations to match the quadratic solution. In this way, the convergence properties and the moderate storage requirements of the CMFD equations can be maintained. This approach requires that the discontinuity factor ratios be computed during the solution, and updated as the solution progresses. It is important to point out that the non-linear iteration method contains *two* sets of discontinuity factor ratios. The first set is contained in the CMFD equations. This set is computed during the solution by the quadratic currents to force the CMFD equations to match the quadratic solution. The second set is contained in the quadratic current equations themselves (see (3.50)). Recall that these were introduced in Section 3.5.4 to correct for the use of Fick's law, homogenization, and somewhat for the spatial approximation. These discontinuity factor ratios are supplied by the user and obtained from an external, detailed calculation. In this work, we assume that these discontinuity factor ratios are available, and only concern ourselves with the computation of the discontinuity factor ratios which correct the CMFD equations to match the quadratic solution. From this point on, the discontinuity factor ratios which appear in the quadratic current equations and correct for heterogeneity etc. shall be called *heterogeneity* factors. The discontinuity factor ratios which force the CMFD equations to match the quadratic solution will retain the name *discontinuity factor ratios*.

The solution begins with the discontinuity factor ratios all set to unity. The CMFD equations are solved to obtain an estimate of the node-averaged fluxes. Equation (3.50), the quadratic expression for the surface-averaged currents, is then solved using the most recent node-averaged fluxes to obtain quadratic-based currents. Using these surface-averaged currents, and the most recent node-averaged fluxes, the discontinuity factor ratios may be obtained. For the 2-D, internal node shown in Fig. 3.8, the expression is:

$$\frac{f^{(i-1,j)}_{g,p'=2}}{f^{(i,j)}_{g,p=1}}=\frac{\left[\overline{\phi}^{(i,j)}_{g}-\frac{X_c}{2}\frac{\overline{J}^{(i,j)}_{g,X_1}(p=1)}{D^{(i,j)}_{g}}\right]}{\left[\overline{\phi}^{(i-1,j)}_{g}-\frac{X_c}{2}\frac{\overline{J}^{(i-1,j)}_{g,X_1}(p'=2)}{D^{(i-1,j)}_{g}}\right]} \tag{5.19}$$

Equation (5.19) was obtained by solving both eqn. (3.12) and (3.13) for the discontinuity factors, and taking the ratio of the results. If the node surface is on the boundary of the reactor, eqn. (5.19) is replaced by:

$$\frac{\alpha^{(i,j)}_{g,p=1}}{f^{(i,j)}_{g,p=1}}=\frac{\left[\overline{\phi}^{(i,j)}_{g}-\frac{X_c}{2}\frac{\overline{J}^{(i,j)}_{g,X_1}(p=1)}{D^{(i,j)}_{g}}\right]}{\left[\overline{J}^{(i,j)}_{g,X_1}(p=1)\Big/\beta^{(i,j)}_{g,p=1}\right]} \tag{5.20}$$

Equations (3.12) and (3.19) were used to obtain (5.20). Note that eqn. (3.50), and the corresponding equations for boundary condition cases contain surface-averaged currents on the right hand side. Previous studies [G-1,B-1] have revealed that attempting to solve the quadratic current equations simultaneously does not converge. An acceptable method for solving the quadratic equations is to utilize surface-averaged currents which are consistent with the nodal fluxes and discontinuity factors from the last outer iteration for the right hand side of (3.50). In other words, after an outer iteration is completed, the nodal fluxes and the most recent set of discontinuity factor ratios are used in eqn. (3.16) to obtain a complete set of surface-averaged currents. These surface-averaged currents are then used on the right hand side of the quadratic current equations.

These quadratic currents are then used in (5.19) and (5.20) to obtain a new set of discontinuity factor ratios. Experience has shown that for most problems, the discontinuity factor ratios do not need to be updated at every outer iteration. Typically, between 2 to 5 outer iterations per update works well.

Figure 5.2 shows the non-linear iteration procedure.

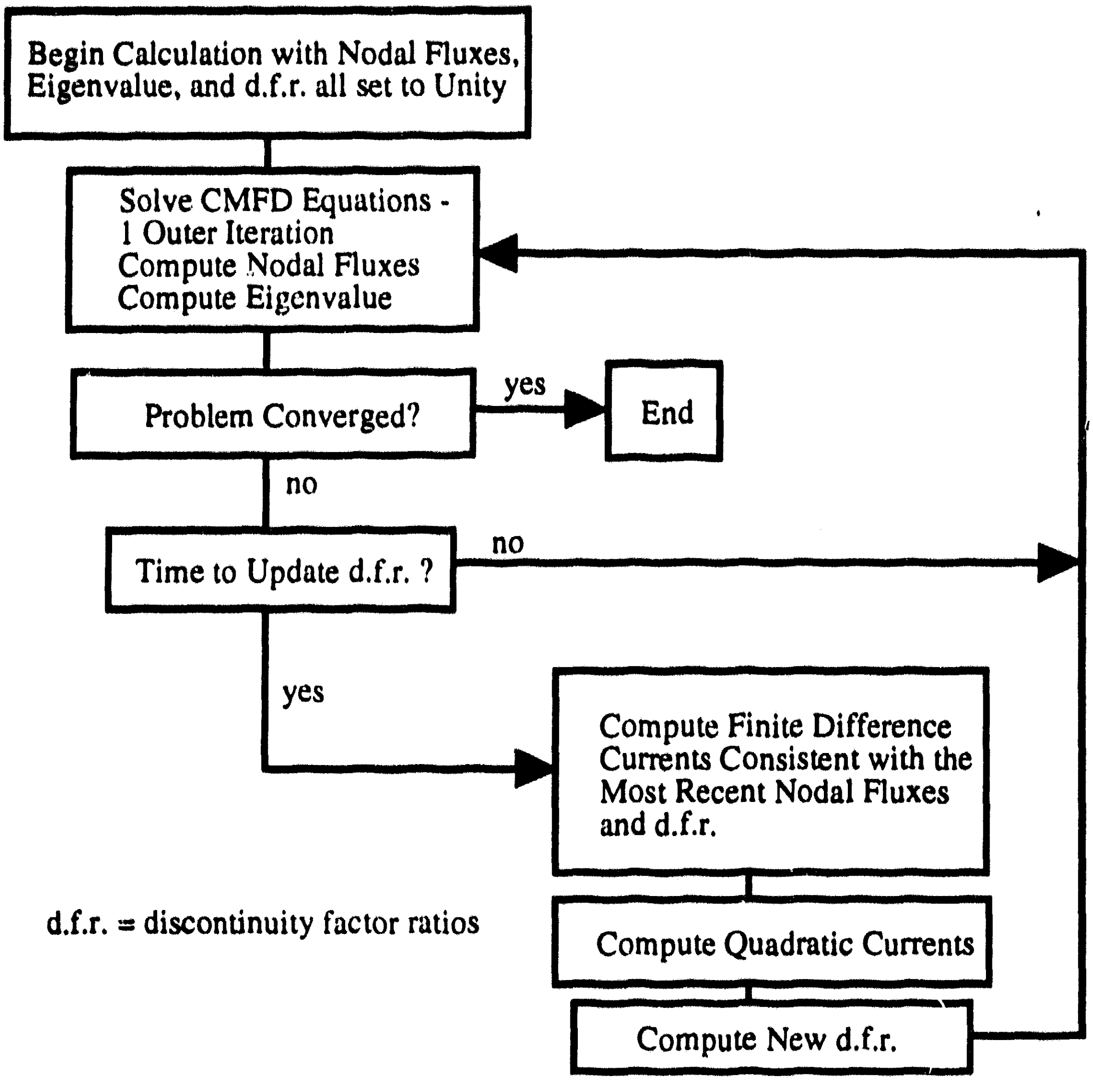

Begin Calculation with Nodal Fluxes, Eigenvalue, and d.f.r. all set to Unity
Solve CMFD Equations - 1 Outer Iteration Compute Nodal Fluxes Compute Eigenvalue
Problem Converged?
yes
End
no
Time to Update d.f.r. ?
no
yes
Compute Finite Difference Currents Consistent with the Most Recent Nodal Fluxes and d.f.r.
d.f.r. = discontinuity factor ratios
Compute Quadratic Currents
Compute New d.f.r.

Figure 5.2. The Non-Linear Iteration Scheme.

Sutton [S-1] has demonstrated that the Wielandt outer iteration method works with the non-linear iteration procedure, but because of the constantly-changing discontinuity factor ratios, the shift factor, λ' , must be updated during the solution. The overall static solution procedure proposed by Sutton [S-1] and successfully implemented in several computer codes [Z-1,G-1], will be used here. The procedure starts with the nodal fluxes and eigenvalue set to unity. An initial, non-accelerated (power) outer iteration is performed to obtain a non-flat nodal flux vector. The Gauss-Seidel method is used on inner iterations until the spectral radii are computed (see below). The second outer iteration is performed using an estimate of the shift which is significantly larger than the expected eigenvalue of the problem. The shift must be larger than the expected eigenvalue to ensure that the problem converges to the fundamental mode. In this application, a shift of $\lambda'=1.5$ is used during this stage. A few outer iterations are performed using this large shift until a convergence of 10^{-2} is obtained in the eigenvalue. The next outer iteration is devoted to computing the spectral radii. Inner iterations are performed until the spectral radius for each group converges to the user-specified criterion ε_{sr}^{crit} :

$$\left|\rho_{G-S}^{(m'+1)}-\rho_{G-S}^{(m')}\right|\leq\varepsilon_{sr}^{crit}$$

Note the m' refers to the inner iteration number, and the subscript "G-S" refers to Gauss-Seidel. After the spectral radii are computed, the (fixed) number of inners per outer (which will be used for the duration of the calculation) are computed using eqn. (5.18). The method allows a different number of inners per outer for each group. Inner iterations from this point on are computed using CCSI. On the next and subsequent outer iterations, the shift factor is computed using the current estimate of the eigenvalue plus a small constant:

$$\lambda'=\lambda^{(m)}+\delta\lambda$$
$$\delta\lambda=constant$$

The value of the small constant does affect the convergence rate of the outer iterations, and several studies have been performed to determine the optimum choice [S-1,S-2,Z-1,G-1]. Smith [S-2] reports an optimum choice of 0.04 or 0.05 for light water systems.

5.2.4 Adjoint Problems

The mathematical adjoint of a specified problem may be computed using the above methods, with the following adjustments:

- The scattering matrix is transposed in energy.
- The coupling matrix is transposed in space and energy.
- The $\nu\Sigma_{fg}$ cross sections are interchanged with the χ_g cross sections. Both are then transposed in energy.
- The initial guess for the eigenvalue and nodal fluxes is the solution from the forward calculation.
- The discontinuity factor ratios are held constant and equal to the converged values from the forward calculation.
- The CCSI inner iteration acceleration can be used and Wielandt outer iteration acceleration can be used. The eigenvalue shift, however, must be larger than typical shift values used in forward problems. Shift values of 0.5 to 1.0 appear to work well for the problems tested.
- The group structure is flipped, the first group becomes g=G and the last group becomes g=1. This is necessary because the dominant scattering

direction has changed from downscatter to upscatter.

- The adjoint flux is normalized.

The solution of the adjoint problem will be important for transient analyses.

5.2.5 Thermal Hydraulic Feedback

The final major component of the solution for static problems is the adjustment of cross sections to account for temperature and density. In this work , we assume that cross section sets at different temperatures are available, and that the reference temperatures bracket the temperatures computed during the solution. During the solution process, the most recent nodal fluxes are used in a thermal hydraulic model to compute fuel-averaged temperatures, coolant-averaged temperatures and coolant-averaged densities for each node in the reactor. These temperatures and densities are then used to interpolate in the reference tables of cross sections to obtain cross sections which reflect the temperature and density profiles in the reactor. The thermal hydraulic feedback, therefore, consists of two parts: the estimation of the temperatures and densities, and the interpolation of the cross sections. Due to the general nature of this work, a very simple thermal hydraulic model was incorporated. The cross section interpolation procedure which was used assumed that the cross sections were quadratic functions of fuel and coolant temperature. If this method is to be applied to a specific reactor, more sophisticated models may be desired. The computer code developed as a part of this work has been set up in such a way that different models may be used in place of those currently implemented.

The WIGL thermal hydraulic model [V-1] was used for the temperature calculations because of its use in many other studies [S-2,G-1]. It is a lumped heat capacity model, and is described in Appendix B. The WIGL model provides node-averaged fuel and coolant temperatures, and can be used for steady state calculations as well as transient thermal feedback calculations.

The cross section interpolation procedure is similar to the one implemented by Byers [B-1]. It assumes that the cross sections are a quadratic function of fuel and coolant temperature, but not local coolant density. Figure 5.3 shows the relationship between the cross sections at reference temperatures and the cross sections at four other points. The interpolation is two dimensional. As can be seen in the figure, left of center is a set of cross section at reference fuel temperature and low coolant temperature. Right of center is a set at reference fuel temperature and high coolant temperature. Above and below center are the high and low fuel temperatures at reference coolant temperature, respectively.

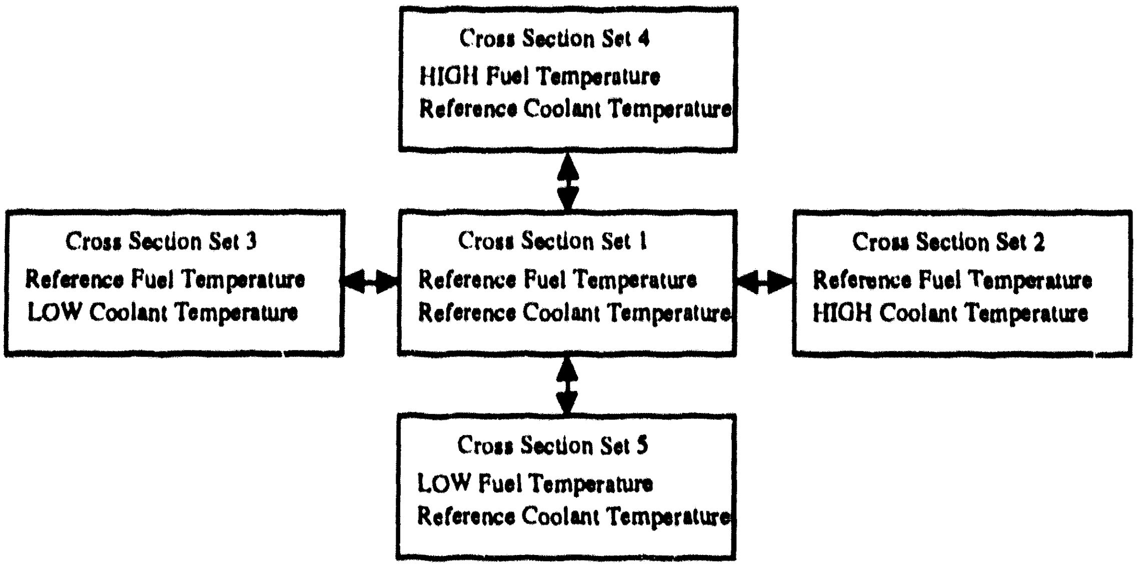


Figure 5.3. The Cross Section Interpolation Model.

The quadratic interpolation is performed first in the coolant temperature direction and then in the fuel temperature direction (the order is arbitrary). The results from the two independent interpolations are combined to obtain a new cross section. Using the symbol Σ to represent a cross section, this combination is:

$$\Sigma\left(T_{fuel},T_{cool}\right)=\Sigma\left(T_{fuel}^{Ref},T_{cool}\right)+\Sigma\left(T_{fuel},T_{cool}^{Ref}\right)-\Sigma\left(T_{fuel}^{Ref},T_{cool}^{Ref}\right) \quad (5.21)$$

The next section will discuss solution methods for transient problems.

5.3 Solution of the Transient Equations

The transient nodal diffusion equations, described by eqn. (4.7) and eqn. (4.16), form a set of first order differential equations in time. Common solution methods for first order differential equations cannot be used for this system, however, due to the characteristics of the constants which appear in the equations. It turns out that the time constants associated with these equations have widely differing values because of the fast response of the prompt neutrons and relatively slow response of delayed neutron precursors to changes in the reactor. Equations of this type are "stiff", and for unconditional stability, an implicit method is required. It turns out that the fully implicit form of eqn. (4.7) is the best choice, which means that we shall use $\theta=1$ for all applications of (4.7).

5.3.1 <u>The Improved Quasi-Static Method</u>

A very common approach to solving the transient equations is to factor the time dependent neutron flux into a shape function and an amplitude function. That is:

$$\mathbf{\Phi}(t)=\mathbf{S}(t)\mathrm{T}(t) \quad (5.22)$$

where $\mathbf{S}(t)$ is a vector containing the shape function and the scalar T(t) is the amplitude. A definition of the amplitude is given by [H-1]:

$$\mathrm{T}(t)\equiv\mathbf{w}^{\mathrm{T}}\mathrm{V}^{-1}\mathbf{\Phi}(t) \quad (5.23)$$

where the vector **w** has been introduced as a weight function. Using eqns. (5.22) and (5.23), a normalization requirement for the shape function can be obtained. Substituting eqn. (5.22) into (5.23) for the flux vector on the right hand side gives:

$$\begin{gathered} \mathrm{T}(t) = \mathbf{w}^{\mathrm{T}}\mathbf{V}^{-1}\{\mathbf{S}(t)\mathrm{T}(t)\} \\ \therefore\ \mathbf{w}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{S}(t) = 1 \end{gathered} \tag{5.24}$$

Many types of solution methods have been developed using this flux facLorization technique. Its success derives from the fact that f r many transient problems, the flux shape, after a quick i: .tial adjustment to composition changes, changes much more slowly than the amplitude. The shape function and the amplitude function can be computed separately, and the nodal flux solution can be constructed at any time by using (5.22). An equation for the amplitude function can be derived in a formally exact way by multiplying eqn. (4.7) and (4.16) by an arbitrary weight function, substituting (5.22), using (5.23) and (5.24), and defining the reactivity, the effective delayed fraction, the prompt neutron lifetime, and the effective delayed neutron precursor concentration, respectively [H-1]:

$$\begin{gathered} \rho(t) \equiv \frac{\mathbf{w}^{\mathrm{T}}[\mathbf{M}(t) - \mathbf{L}(t)]\mathbf{S}(t)}{\mathbf{w}^{\mathrm{T}}[\mathbf{M}(t)]\mathbf{S}(t)} \\ \beta^{eff}_{iprec}(t) \equiv \frac{\mathbf{w}^{\mathrm{T}}\left[\mathbf{M}_{\mathrm{Iprec}}(t)\right]\mathbf{S}(t)}{\mathbf{w}^{\mathrm{T}}[\mathbf{M}(t)]\mathbf{S}(t)} \qquad \beta^{eff} \equiv \sum_{iprec=1}^{nprec} \beta^{eff}_{iprec} \\ \Lambda(t) \equiv \frac{\mathbf{w}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{S}(t)}{\mathbf{w}^{\mathrm{T}}[\mathbf{M}(t)]\mathbf{S}(t)} \\ C^{eff}_{iprec}(t) \equiv \frac{\mathbf{w}^{\mathrm{T}}\mathbf{C}_{\mathrm{Iprec}}(t)}{\mathbf{w}^{\mathrm{T}}\mathbf{V}^{-1}\mathbf{S}(t)} \end{gathered} \tag{5.25}$$

The result is the point kinetics equations:

$$\frac{d}{dt}\mathrm{T}(t) = \frac{\rho(t) - \beta^{eff}(t)}{\Lambda(t)}\mathrm{T}(t) + \sum_{iprec=1}^{nprec}\lambda_{iprec}C_{iprec}^{eff}(t) \qquad (5.26)$$

$$\frac{d}{dt}C_{iprec}^{eff}(t) = \frac{\beta_{iprec}^{eff}(t)}{\Lambda(t)}\mathrm{T}(t) - \lambda_{iprec}C_{iprec}^{eff}(t) \qquad (5.27)$$

It is important to point out that the point kinetics parameters, defined in (5.25), were derived in a formally exact way [H-1], which means that if the correct shape function is known, the correct point kinetics parameters are known and the correct amplitude can be computed using (5.26) and (5.27).

The methodology for obtaining the shape function is the distinguishing feature of the many flux factorization approaches. In traditional point kinetics, the shape function is obtained from a static calculation and held constant throughout the transient. The attraction of point kinetics is that once the point kinetics parameters have been computed, the solution of (5.26) and (5.27) can be very fast. The result is a solution for the amplitude as a function of time. However, the use of the static shape neglects distortions caused by the direct perturbations to the system, as well as any distortions caused by feedback during the transient. In general, point kinetics is only valid for transients in which the shape does not change significantly during the transient. The adiabatic method improves upon the point kinetics approach by estimating the shape during the transient using several static calculations [O-1]. This has been done using pre-computed shape functions, which account for distortions due to composition changes but do not reflect feedback effects. The adiabatic method has also been implemented using static calculations during the transient, which do account for feedback effects. The adiabatic methods suffer from the fact that static calculations assume that the delayed neutron precursors are in equilibrium, which they of course are not. In addition, the time derivative is neglected and the static calculations are expensive [O-1]. The quasi-static method estimates the shape function by substituting (5.22) into the transient equation and neglecting the time derivative. The resulting shape equation includes the amplitude function, which comes from the solution of the point kinetics equations [(5.26) and (5.27)] over the last time interval. The improved quasi-static method does not neglect the time derivative, but rather approximates it using a first order backward difference [O-1] (recall that eqn. (4.7) used this approximation for the derivative).

Kao [K-1] proposed that the improved quasi-static method be implemented using the transient nodal equations, (4.7) and (4.16), for the shape function. The nodal flux solution from these equations could be used to determine the shape function by applying eqn. (5.22) and imposing the normalization requirement [eqn. (5.24)]. In this way, the amplitude does not appear in the shape function equation.

The general approach which will be used here is to solve eqns. (4.7) and (4.16) using a large time step. The solution of these equations yields the nodal fluxes and precursor densities at each time step. The amplitude associated with the nodal fluxes can be obtained using eqn. (5.23) (the choice of weight function will be discussed shortly). The shape associated with the nodal fluxes can be obtained using eqn. (5.22). At the end of each large "shape step", the shape function is expected to be accurate while the amplitude is expected to be poor. The amplitude is therefore re-computed over the time step using point kinetics. The point kinetics parameters (i.e. reactivity, etc.) are obtained using the definitions in (5.25) by interpolating the shape function between the beginning of step and end of step. Because the shape is expected to be accurate, and the point kinetics parameters were derived using a formally exact procedure, the amplitude estimate from point kinetics is expected to be excellent. Figure 5.4 shows the time stepping algorithm.

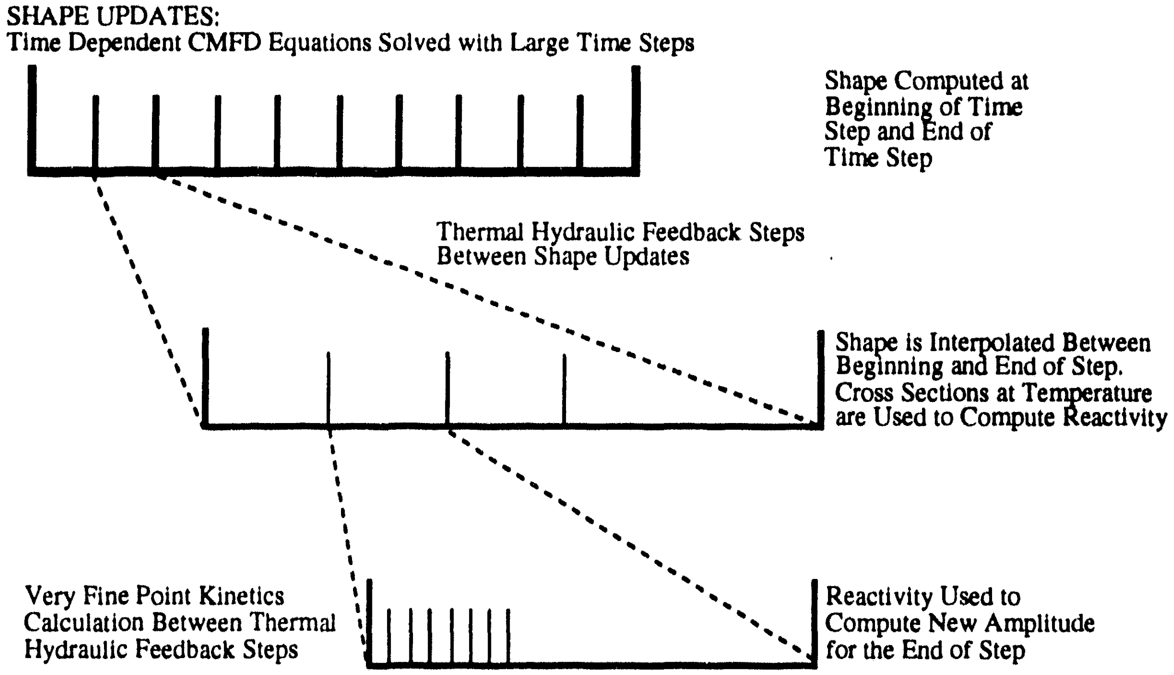

SHAPE UPDATES:
Time Dependent CMFD Equations Solved with Large Time Steps
Shape Computed at Beginning of Time Step and End of Time Step
Thermal Hydraulic Feedback Steps Between Shape Updates
Shape is Interpolated Between Beginning and End of Step. Cross Sections at Temperature are Used to Compute Reactivity
Very Fine Point Kinetics Calculation Between Thermal Hydraulic Feedback Steps
Reactivity Used to Compute New Amplitude for the End of Step

Figure 5.4. The Time Stepping Scheme for Transient Problems.

In Fig. 5.4, it can be seen that there are three levels of time steps. The top level is the shape update level. The transient nodal equations are solved using large time steps. Each time step is then re-computed using point kinetics. The second level shows that thermal feedback is modeled during the point kinetics calculation. At this level, the point kinetics parameters are computed at each feedback step. Gehin [G-1] demonstrated that if thermal feedback is not allowed during the point kinetics calculation, the amplitude can be severely over or under estimated. This is because the point kinetics parameters are computed using the cross sections, and if the cross sections reflect "old" temperatures, the combination of "new" shape and "old" cross sections can give poor estimations of the reactivity.

The third level of computation is the point kinetics calculation. The point kinetics parameters used for the solution of eqns. (5.26) and (5.27) between thermal feedback steps are found by interpolating between the parameters at the beginning and the end of the feedback step.

At the end of each (shape) time step, the nodal fluxes are scaled by the ratio of the point kinetics-computed amplitude to the shape-computed amplitude.

5.3.2 Solution of The Point Kinetics Equations

The discretization of the point kinetics equations is completely analogous to the discretization of the transient nodal equations. The theta difference approximation is used for the prompt equation, and a direct integration procedure is used for the delayed equations. The prompt equation becomes:

$$\begin{aligned}\frac{\left(\mathrm{T}^{(n+1)}-\mathrm{T}^{(n)}\right)}{\Delta t_n^{pk}} &= \theta\left\{\frac{\rho(t_{n+1})-\beta^{eff}(t_{n+1})}{\Lambda(t_{n+1})}\mathrm{T}^{(n+1)}+\sum_{iprec=1}^{nprec}\lambda_{iprec}C_{iprec}^{eff}(t_{n+1})\right\}\\ &\quad+(1-\theta)\left\{\frac{\rho(t_n)-\beta^{eff}(t_n)}{\Lambda(t_n)}\mathrm{T}^{(n)}+\sum_{iprec=1}^{nprec}\lambda_{iprec}C_{iprec}^{eff}(t_n)\right\}\end{aligned} \tag{5.28}$$

The Δt_n^{pk} in (5.28) refers to the time step size used for the point kinetics solution, which will be different from the time step size in eqn. (4.7).

The direct integration of the precursor equations follows directly from Section 4.4.2. The result is:

$$
\begin{aligned}
C^{eff}_{iprec}(t_{n+1}) = {} & g_1(iprec,\Delta t_n)C^{eff}_{iprec}(t_n) \\
& + g_2(iprec,\Delta t_n)\frac{\beta^{eff}_{iprec}(t_n)}{\lambda_{iprec}\Lambda(t_n)}\mathrm{T}^{(n)} \\
& + g_3(iprec,\Delta t_n)\frac{\beta^{eff}_{iprec}(t_{n+1})}{\lambda_{iprec}\Lambda(t_{n+1})}\mathrm{T}^{(n+1)}
\end{aligned}
\tag{5.29}
$$

where the functions g_1, g_2, and g_3 are defined by eqn. (4.15).

The best choice for the weight function, which appears in eqn. (5.23) and in the definitions of (5.25), is the adjoint flux from the static problem. The adjoint flux can be shown to be the best choice for minimizing errors in the point kinetics parameters when the shape function is not known exactly [H-1]. The use of the adjoint as a weight function, however, is less important for the improved quasi-static method than in the traditional point kinetics method because the shape function is updated frequently during the transient. Nevertheless, the adjoint will be used as the weight function for all transient computations.

5.3.3 Numerical Solution of the Transient Problem

An equation for the nodal flux at each time step can be obtained by substituting the equation for the nodal precursors, eqn. (4.16), into eqn. (4.7). The result is:

$$
\begin{aligned}
& \left\{\frac{\mathbf{V}^{-1}}{\Delta t_n} - \mathbf{M}_{\mathbf{p}}^{(n+1)} + \mathbf{L}^{(n+1)} - \sum_{iprec=1}^{nprec} g_3(iprec,\Delta t_n)\mathbf{M}_{\mathbf{iprec}}^{(n+1)}\right\}\mathbf{\Phi}^{(n+1)} \\
& = \left\{\frac{\mathbf{V}^{-1}}{\Delta t_n} + \sum_{iprec=1}^{nprec} g_2(iprec,\Delta t_n)\mathbf{M}_{\mathbf{iprec}}^{(n)}\right\}\mathbf{\Phi}^{(n)} + \left\{\sum_{iprec=1}^{nprec} g_1(iprec,\Delta t_n)\lambda_{iprec}\mathbf{C}_{\mathbf{iprec}}^{(n)}\right\}
\end{aligned}
\tag{5.30}
$$

The matrix on the left hand side of eqn. (5.30) must be inverted at each time step in order to solve for the nodal fluxes. This inversion is similar to the inversion of the loss matrix for the static problem. Motivated by the same arguments, the inversion will be accomplished using the Chebyshev Cyclic Semi-Iterative (CCSI) method.

The transient quadratic equations will be solved using the non-linear iteration scheme. Thus, the procedure at each time step is to perform inner and outer iterations. The outer iterations in the transient case are not used to compute an eigenvalue, as was the case in the static problem, but they are used to compute the discontinuity factor ratios. At each time step, the discontinuity factor ratios from the last time step are used in the transient CMFD equations [eqn.(5.30)]. The transient CMFD equations are solved by inner iterations, using the CCSI acceleration procedure. Unlike the static problem, however, a fixed number of inners is not used. Rather, the inners are converged to a user-specified criterion at each time step. The quadratic currents and the discontinuity factor ratios are re-computed using these nodal fluxes, and the outer iterations are continued until the nodal fluxes between iterations have converged to a user-specified criterion.

The terms on the right hand side of eqn. (5.30) are all obtained using last time step values. The right hand side is therefore treated as a source. An adaptive time stepping scheme was not used for the solution of eqn. (5.30). Instead, the shape at each time step is computed over fixed, user-defined 'time domains.' A time domain is a time span during which a constant time step size is used to compute the shape function. Time domains can be defined in such a way that certain portions of a transient are modeled with relatively small time steps, and others are modeled with relatively coarse time steps. The spectral radii of the inner iteration matrix, which is required by CCSI, are computed at th3 beginning of each time domain on the first time step of the domain. The thermal hydraulic feedback steps, the second level in the improved quasi-static scheme, is also user-defined and given as a fixed number of feedback steps per shape update. The number of feedback steps per shape update can be different for each time domain.

The point kinetics calculation for the amplitude, which is performed between thermal feedback steps, is solved using adaptive time stepping. The equation for the amplitude can be obtained by substituting eqn. (5.29) into eqn.(5.28). Before this is done, however, eqn. (5.28) is put into fully implicit form by setting $\theta=1$. The fully implicit form is used for the same reasons as before. The result is:

$$\left\{1-\frac{\Delta t_n^{pk}}{\Lambda(t_{n+1})}\left[\rho(t_{n+1})-\beta^{eff}(t_{n+1})+\sum_{iprec=1}^{nprec} g_3(iprec,\Delta t_n^{pk})\beta_{iprec}^{eff}(t_{n+1})\right]\right\}\mathrm{T}^{(n+1)}$$
$$=\left\{1+\frac{\Delta t_n^{pk}}{\Lambda(t_n)}\sum_{iprec=1}^{nprec} g_2(iprec,\Delta t_n^{pk})\beta_{iprec}^{eff}(t_n)\right\}\mathrm{T}^{(n)} \qquad (5.31)$$
$$+\Delta t_n^{pk}\sum_{iprec=1}^{nprec} g_1(iprec,\Delta t_n^{pk})\lambda_{iprec}C_{iprec}^{eff}(t_n)$$

The *pk* superscript on the time step size in (5.31) is used to distinguish between the time step size in the point kinetics calculation and the time step size used in the shape calculation. The right hand side of (5.31) is computed using the last (point kinetics) time step values and is treated as a source. The adaptive time stepping routine used for the point kinetics calculations is "step doubling" [P-2,G-1]. In step doubling, the amplitude for the end of the current time step is computed twice. The first is computed using one step, and the second is computed using two steps of half the size. The truncation error is then estimated using these two estimations of the amplitude. For $\theta=1$, this error is given by:

$$\varepsilon_{pk}\approx\frac{\mathrm{T}_{double}^{(n+1)}-\mathrm{T}_{single}^{(n+1)}}{\mathrm{T}_{double}^{(n+1)}} \qquad (5.32)$$

The current time step is then either rejected or accepted based upon a user-defined acceptable error, ε_{pk}^{crit} . If the step is rejected, the time step is redone with a smaller step size. If the step is accepted, the next step size is estimated using [G-1]:

$$\Delta t_{n+1}^{pk} = 0.9 \Delta t_{n}^{pk} \left| \frac{\varepsilon_{pk}}{\varepsilon_{pk}^{crit}} \right|^{-0.5} \tag{5.33}$$

If the error is much smaller than the acceptable error, a step of twice the size is used for the next step.

5.4 Summary

In this chapter, the solution methods for static and transient problems were described. For static problems, the non-linear iteration procedure was introduced, describing an efficient method for solving the quadratic equations. A two-level iteration scheme is used, with acceleration methods used on both levels. The solution of the adjoint problem was discussed, as well as thermal hydraulic feedback. For transient problems, the improved quasi-static method was introduced. This three-level time stepping method allows relatively large shape steps to be taken using the transient nodal equations, with very fine point kinetics calculations between each shape step to correct the amplitude.

These static and transient solution methods have been implemented for testing into a computer code with the name QUARTZ (**QUA**dratic **R**eactor code in **T**riangular-**Z** geometry). QUARTZ has been written in FORTRAN 77 for general (1-D, 2-D or 3-D) problems in any number of groups, with upscatter. The computer code has been tested on different systems successfully (IBM-PC Compatibles, DEC VS3100, IBM RS/6000, CRAY XMP 416), and should be portable to other systems with minor changes. Details of the structure and use of QUARTZ can be found in the User's Manual, which is distributed with the source code.

The next chapter will present the results of selected static and transient problems.

Chapter 6

Testing the Static and Transient Methods

6.1 Introduction

In this chapter, the static and transient methods are tested. The static methods are tested using simple benchmark problems, and the nature of the quadratic equations is explored. The transient methods are tested using two problems. An analytic solution was available for the first, which consisted of a simple zero current cell. The second problem consisted of a large heterogeneous heavy water reactor. QUARTZ results were compared to published results for this problem.

6.2 Testing the Static Method

6.2.1 A Simple Analytic Test

An initial test of the static methods in QUARTZ was made using a simple test problem for which an analytic solution was known. The first test consisted of an infinite medium calculation with the cross sections given in Appendix C, Section 1. The second test was similar to the first except that an infinite slab of finite thickness was modeled.

The analytic solution for the eigenvalue of an infinite, two group reactor is given by [H-1]:

$$k_\infty = \frac{\left(\nu\Sigma_{f1}\Sigma_2 + \nu\Sigma_{f2}\Sigma_{21}\right)}{\left(\Sigma_1\Sigma_2\right)}$$

Using this equation with the cross sections given in Appendix C, the critical eigenvalue is 1.35439. The computed value using QUARTZ was also 1.35439.

The analytic solution for the eigenvalue for a two group infinite slab, where the leakage is modeled using geometric buckling is given by:

$$k = \frac{\left(\nu\Sigma_{f1}\Sigma_2' + \nu\Sigma_{f2}\Sigma_{21}\right)}{\left(\Sigma_1'\Sigma_2'\right)}$$

$$\Sigma_g' \equiv \Sigma_g + D_g B^2$$

where B^2 is the geometric buckling. Neglecting any extrapolation length, the buckling for this problem is given in terms of the thickness of the slab, H:

$$\mathrm{B}^2 = \left(\frac{\pi}{\mathrm{H}}\right)^2 \ (\mathrm{cm}^{-2})$$

Using the cross sections given in Appendix C, and assuming a slab thickness of 250 cm, the eigenvalue is 1.20444. The computed value using QUARTZ was 1.20444.

The next sub-section compares QUARTZ solutions to a 2-D, finite difference nodal code in triangular geometry.

6.2.2 Comparisons to MITHEX

In this sub-section, two problems are used to compare QUARTZ to a 2-D, finite difference nodal code with the name MITHEX [G-2], which has the capability to solve problems in triangular geometry. In these two problems, QUARTZ was run in finite difference mode, with the quadratic updating of the discontinuity factor ratios turned off. All 'heterogeneity' factors were unity. The first problem was a simple two-group, heterogeneous, 24 node cell model with a zero current boundary condition. The details of the cell, including cross sections are given in Appendix C, Section 2. The problem was solved by MITHEX and QUARTZ using an eigenvalue and flux convergence criterion of 10^{-6}. The results are presented in Table 6.1.

Table 6.1

Results from the MITHEX Infinite Cell Comparison

		MITHEX	QUARTZ
Eigenvalue		1.61436	1.61436
Node 1*	Group 1 Flux	6.268E+00	6.2678E+00
	Group 2 Flux	1.619E+01	1.6189E+01
Node 6	Group 1 Flux	6.268E+00	6.2678E+00
	Group 2 Flux	1.619E+01	1.6189E+01
Node 7	Group 1 Flux	7.295E+00	7.2955E+00
	Group 2 Flux	1.473E+01	1.4732E+01
Node 8	Group 1 Flux	1.044E+01	1.0435E+01
	Group 2 Flux	1.015E+01	1.0151E+01

* The node numbers are shown on the figure in Appendix C, Section 2. The model has 60 degree rotational symmetry, so nodes 1,6,7, and 8 completely describe the results.

In Table 6.1, it can be seen that QUARTZ matches the MITHEX-calculated eigenvalue exactly. The table also shows that the flux distribution matches. This is to be expected, because both QUARTZ and MITHEX solve the same finite difference equations.

The second comparison problem was a two-group, 96 node, core-sized heterogeneous problem. Four different material types were used. The overall core geometry was a hexagon, with the outer nodes being reflector material. Appendix C, Section 3 gives the details of the model. The boundary condition used for this model was zero net returning current (referred to as "albedo" here). The MITHEX eigenvalue was computed to be 1.07740. The QUARTZ eigenvalue was also computed to be 1.07740. The fluxes and relative power densities for each node were also compared. The MITHEX and QUARTZ solutions for the group and node-averaged fluxes matched, as was the case for the cell problem. It turns out that MITHEX, when computing relative power densities, averages over all nodes in the reactor, including non-fueled regions. QUARTZ, on the other hand, averages over fueled zones only. This difference was accounted for by multiplying the MITHEX-computed relative power densities by the ratio of fueled zones to non-fueled zones. When this adjustment was made, the relative power densities matched. This problem was not encountered in the cell problem because all regions were fueled in that model.

These two MITHEX comparison problems demonstrate that QUARTZ is capable of properly solving the mesh-centered finite difference equations for triangular geometry. The quadratic methods incorporated into QUARTZ may now be tested and compared with the mesh-centered finite difference method. The next sub-section shows that the quadratic equations reduce to the Coarse Mesh Finite Difference (CMFD) equations in the limit of small mesh size.

6.2.3 Reducing the Quadratic Method to the CMFD Method in the Limit of Small Mesh Size

It can be shown algebraically [H-3] that the equations for the quadratic currents (eqns. 3.50,3.74,etc.) can be cast into a form of first order terms in X_c (similar to the finite difference expression for the current) and second order terms in X_c. It is expected that, in the limit of very small mesh spacing, the quadratic terms will disappear more quirkly than the first order terms. Thus, the quadratic method should reduce to the Coarse Mesh Finite Difference (CMFD) method when very small mesh spacing is used. To test that this is the case, a simple test problem was used. Appendix C, Section 4 gives the geometry and cross sections used for the test problem. The problem consisted of a 6 node hexagon, with homogeneous material and an albedo (zero net returning current) boundary condition. This problem was run initially with a node size of 5 cm ("size" here means the side of the triangular node). The node size was then reduced, and the eigenvalue was recorded for the CMFD and Quadratic methods. Fig. 6.1 shows the relative difference between the quadratic and finite difference eigenvalues as a function of S, the node side. As the node size shrinks (to the left of the figure), the quadratic terms in the quadratic method disappear, and the quadratic eigenvalue reduces exactly to the finite difference eigenvalue.

It should be noted that some of the eigenvalues produced by this study were very small and non-physical (~ 3.E-6 for S =1.0E-4 cm). The problems were run in double precision and the eigenvalues were converged to 10^{-12}.

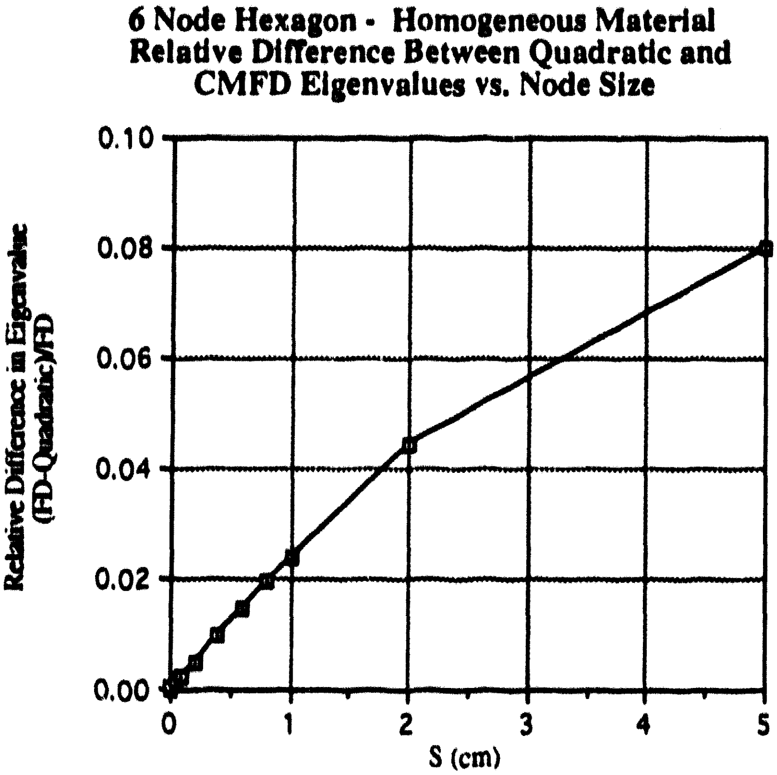

6 Node Hexagon - Homogeneous Material Relative Difference Between Quadratic and CMFD Eigenvalues vs. Node Size

Figure 6.1. Relative Difference Between the Quadratic and CMFD Eigenvalues vs. Node Size.

It was mentioned in Chapter 5 that the Coarse Mesh Finite Difference equations, in the limit of very small mesh size, are guaranteed to converge to the exact solution. Because the Quadratic method becomes the CMFD method in the limit of very small mesh size, it too is guaranteed to converge to the true solution in the limit.

Another important result of Fig. 6.1 is to add credibility to the application of continuity of current through the point of a triangle in Chapter 3. That approximation was vital to the derivation of the quadratic equations for both internal nodes and nodes with boundary conditions. This result shows that if the approximations in Chapter 3 are poor, problems associated with the approximations disappear in the limit.

6.2.4 Comparing the Quadratic Method to the CMFD Method

The quadratic equations were derived and implemented in order to improve the spatial approximation. In general, we expect that for a given node size, the quadratic method will give a more accurate representation of the node-averaged fluxes than the finite difference method. If this is the case, then larger nodes may be used, which will result in less computational work and faster run times. The penalty associated with using the quadratic method is the computational work required to compute discontinuity factor ratios during the solution. In general, the quadratic method will be attractive if the computational cost of solving the quadratic equations (using the non-linear iteration procedure) is less than the cost of solving the CMFD equations with a finer mesh (a mesh size which gives comparable accuracy).

Three sample problems were used to compare the quadratic method to the CMFD method. The eigenvalues and relative power densities were used for comparison. The relative power densities are defined such that the average power density in the fueled nodes is unity. The problem descriptions are given in detail in Appendix C, Section 5. In each of the cases, a 24 node, 2-D problem was used. The tests consisted of refining the mesh, which in the case of equilateral triangles, can only be achieved by subdividing each equilateral triangle into four sub-triangles. Figure 6.2 shows one subdivision of a triangular node.

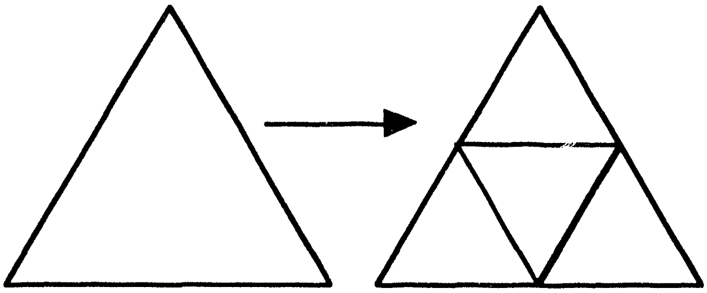


Figure. 6.2. A Single Subdivision of a Triangular Node.

Each subdivision increases the number of nodes by a factor of four. Thus, for a model of 24 nodes, one subdivision gives 96 nodes, two subdivisions gives 384 nodes, and so on. In each of the comparisons, the eigenvalue and the relative power densities were compared to a reference, 6144 node (4 subdivisions of 24 nodes), fine mesh solution. The percent eigenvalue error is computed by:

$$\left(\frac{\lambda - \lambda_{reference}}{\lambda_{reference}}\right)\times 100$$

The absolute, maximum error in the relative power density over all fueled regions is used for comparison and is computed by:

$$\max_{fueled\ nodes,i}\left(\frac{\left|\mathrm{P}_i^{ref} - \mathrm{P}_i\right|}{\mathrm{P}_i^{ref}}\right)\times 100$$

The absolute, average error in the relative power density is also presented. This is given by:

$$\frac{1}{V_{core}}\sum_i\left(\frac{\left|\mathrm{P}_i^{ref} - \mathrm{P}_i\right|}{\mathrm{P}_i^{ref}}\right)V_i$$

$$V_{core} \equiv \sum_{fueled\ nodes,\ i} V_i$$

The first problem, which will be referred to as case I, was a homogeneous, two-group model with a zero net returning current boundary condition. The quadratic and CMFD methods were used to solve the case I problem with 0, 1, 2, and 3 subdivisions. The eigenvalue error as a function of mesh size is given in Figure 6.3 (figures and tables are located at the end of this sub-section). In this case, both the quadratic and CMFD eigenvalue errors decrease linearly with respect to the mesh spacing squared. Both the quadratic and CMFD eigenvalue errors approach zero in the limit of small mesh size, but for larger mesh sizes the quadratic eigenvalue error is much larger than the CMFD eigenvalue error. The results for the eigenvalue error comparison are also given in Table 6.2. In the table, it can be seen (where it is a bit difficult to see on the graph) that the error for the quadratic eigenvalue actually changes sign at the fine mesh of 64 nodes per assembly (3 subdivisions). To minimize the possibility that the problem was falsely converged, the eigenvalue was converged to 10^{-12}. In order to minimize roundoff error, the problems were run in double precision on a CRAY XMP 416. Several other problems have been run with different cross sections and numbers of nodes. The cases with albedo boundary conditions consistently show the eigenvalue error approaching zero from below and crossing over (changing sign) at small mesh size. It is difficult to further reduce the mesh size for these problems because the number of nodes increase by a factor of four each time. In addition, the spectral radii become very close to unity for the very small mesh, and even the accelerated convergence rates are very slow.

The results of Section 6.2.3 demonstrated that the quadratic method does reduce to the CMFD method in the limit of small mesh size, so it appears that there is a region between ~1 cm and ~10 cm, within which the quadratic method displays inconsistent error reduction with decreasing mesh size.

The maximum error in the relative power density is given in Figure 6.4. It can be seen in the figure that the spatial discretization error for the CMFD method is on the order of the mesh size squared, and reduces to zero as the mesh spacing approaches the fine mesh reference. For the quadratic method, the maximum error is much lower than the CMFD error at coarse mesh, but as the mesh size decreases to two subdivisions (mesh size of ~10 cm), the error increases slightly and then decreases.

For this case, it may be concluded that, with large mesh size, the quadratic method gives a much better estimate of the nodal relative power densities, but a poorer eigenvalue. At fine mesh, the solution is comparable to the CMFD solution, but displays inconsistent error reduction with decreasing mesh size.

The second problem, case II, was identical to the first except that a zero flux boundary condition was used. Tables 6.4 and 6.5 show the results for the eigenvalue error and relative power density error, respectively. Figures were not constructed for these data because the results qualitatively matched Figs. 6.3 and 6.4. However, in Table 6.4, it can be seen that at the small mesh sizes, the quadratic eigenvalue error does not change sign, and decreases consistently. In Table 6.5, it can be seen that the maximum and average relative power density error decreases consistently in small mesh sizes also. The difference between case I and case II is the boundary condition applied to the outer surface. Recall that the boundary conditions are specified by two parameters in eqn. (3.18), which is reproduced here for convenience:

$$\alpha_{g,p}^{(i,j)} \overline{\underline{J}}_g^{(i,j)}(p) \cdot \underline{n}_p = \beta_{g,p}^{(i,j)} \overline{\phi}_g^{(i,j)}(p) \qquad (3.18)$$

The albedo boundary condition, used in case I, is specified by setting $\alpha = 2.0$ and $\beta = 1.0$. The zero flux boundary condition is obtained by setting $\alpha = 0$ and $\beta = 1.0$. Recall also the discussion in Section 3.6.2 about the difficulties associated with identifying the proper normal to the reactor surface. It is proposed that the quadratic method's inconsistent error reduction in the eigenvalue and relative power for small mesh size is due to the boundary condition difficulties. The difficulties do not appear in the zero flux cases because the alpha parameter is zero, which causes the entire left hand side of eqn. (3.18) to become zero.

The third problem, case III, was constructed to test the two methods without the interference of the boundary condition difficulties. Case III is a 24 node, 2-D, heterogeneous problem with a zero current boundary condition. The results from the quadratic and CMFD calculations can be seen in Figs. 6.5 and 6.6. In these figures, it is apparent that the quadratic method gives significantly better results than the CMFD method for both the eigenvalue and the relative power densities. Tables 6.6 and 6.7 give the numerical results for this case. In Table 6.7, for example, it can be seen that in order to achieve a maximum error in the relative power of less than 1 percent, we can run either a quadratic calculation with 1 node per assembly (0 subdivisions, or 24 nodes total) or a CMFD calculation with 16 nodes per assembly (2 subdivisions, or 384 nodes total). The run times for these two cases, for a VAX 3100 machine (and no numerical acceleration used) were 1.1 seconds for the quadratic case and 27 seconds for the CMFD case.

Based upon the eigenvalue results for the zero current boundary condition case, it appears that the nodes on the boundary of the reactor may be causing problems with the eigenvalue calculations at coarse mesh. The reaction rates and leakages were examined for a node at the corner of the 24 node hexagon. This node is shown as node 1 in Fig. 6.7. Table 6.8 shows the percent errors in the group 2 absorption rate and the group 2 in-scatter rate for the CMFD and the quadratic methods with 1 node per assembly (0 subdivisions). It can be seen in the table that both reaction rates are much better for the quadratic method. This indicates that the nodal fluxes are better for the quadratic method, and this is consistent with the low power density errors in Table 6.5.

Table 6.9 shows the percent errors in the leakages for the three faces of node 1. The leakage for side 2 is zero because of symmetry (i.e. the flux in node 1 is equal to the flux in node 2). The error in the leakage for side 1 is lower for the quadratic method, but the error is higher for side 3. The quadratic calculation of the surface-averaged current for side 3 is performed using the "case 5" boundary condition equations (see Chapter 3, Table 3.2, where node 1 is "node (i,j)"). The boundary condition at the "top" point (the point opposite side 3) is used in that calculation. It is proposed that the poor estimation of the leakage at side 3 is influenced by the fact that a boundary condition is imposed at the opposite point.

A study performed by Gehin [G-1], comparing quadratic, cubic, and quartic polynomial nodal methods in Cartesian geometry, has shown that the quadratic nodal method is capable of producing accurate node-averaged fluxes, but can give poor estimations of the current at surfaces. This is because the quadratic polynomial has few degrees of freedom. The quadratic method is even more rigid in triangular geometry because of the requirement that the polynomial must match the value (i.e. flux) and derivative (i.e. current) at the point to the polynomial in the node beyond the point. This is done without the benefit of discontinuity factors at the point.

The error in the leakage for side 3 may be caused by the inability of the quadratic polynomial to satisfy the boundary condition at the point and accurately predict the current at the face. The difficulty in accurately predicting the leakages at faces along the boundary is likely to be the reason for the relatively poor eigenvalue estimates using the quadratic method at coarse mesh. The accuracy of the quadratic eigenvalue calculation improves with finer mesh because the boundary nodes become less and less important with more nodes, and the errors become averaged with the better interior nodes.

These results indicate that the quadratic method may perform poorly in cases where the flux changes drastically, such as reactor boundaries and material interfaces. Preliminary tests with reflectors have given mixed results. A reactor model with a light water reflector was briefly tested. The quadratic method was found to give poor results for the eigenvalue (similar to previously presented results for bare cores), and even led to some numerical instability. It was *suspected* that the leakage in the nodes on the fuel/reflector interface and the reactor boundary adversely affected the eigenvalue calculation. Instabilities have been experienced in other geometries with light water reflectors [G-1,B-1], requiring that the discontinuity factor ratios be set to unity in the reflector. In this brief test, the discontinuity factor ratios were not unity.

Another preliminary test was run with a heavy water system with a heavy water reflector. The eigenvalue errors for the quadratic method were on the same order as the CMFD method. The reflector may have improved the quadratic method by decreasing the severity of the flux change at the reactor boundary. It should be emphasized that these two tests were very brief, and in order to fully understand the nature of the quadratic method with material interfaces, further tests are strongly recommended.

In conclusion, the quadratic method has been shown in certain cases to give significant benefit over the CMFD method. In fact, the quadratic method has produced better node-averaged fluxes at coarse mesh for every case tested. However, the quadratic method produces relatively poor estimates of the leakages in nodes along the boundary of the reactor and at severe material interfaces. These leakage errors can produce poor estimates of the eigenvalue at coarse mesh. It was mentioned that the quadratic polynomial may not have enough flexibility to accurately model the surface-averaged current at the points and surfaces of the triangular nodes. Experience in Cartesian geometry (see Ref. [G-1]) supports this argument. Certainly, more research is needed in this area.

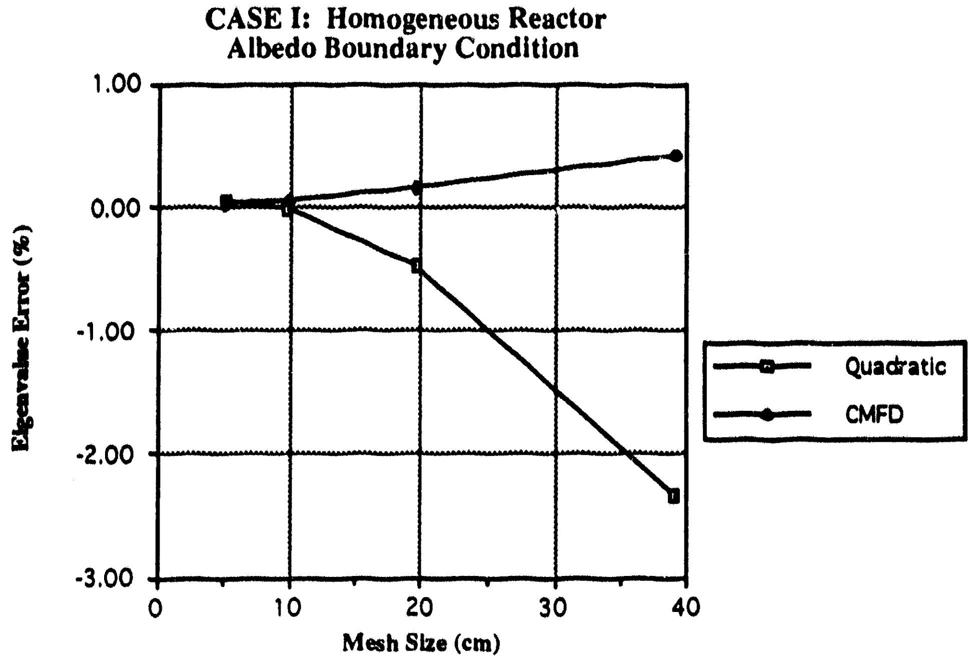
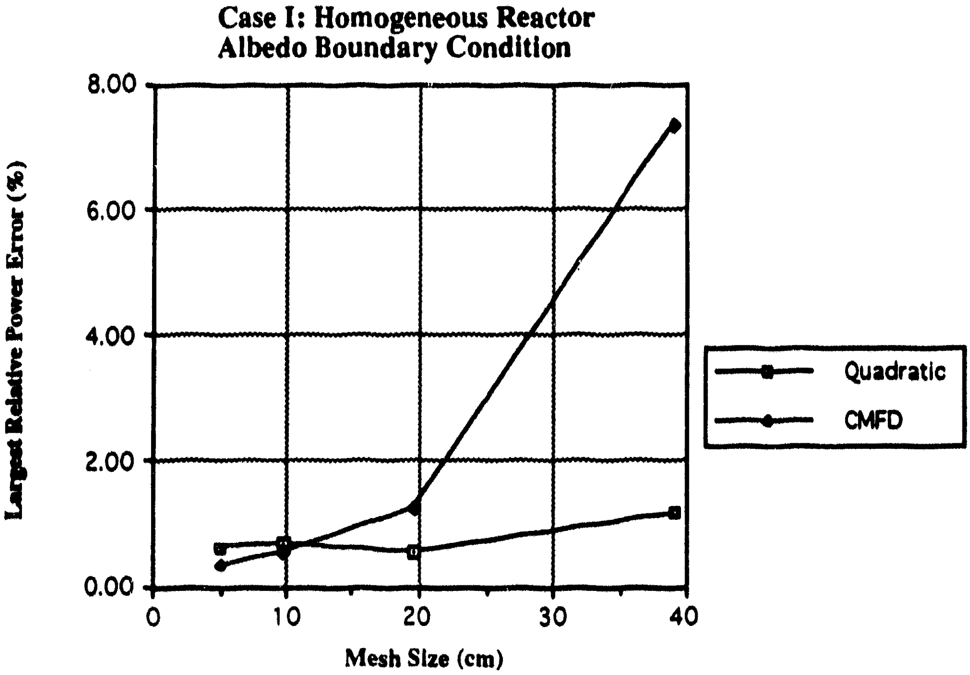


Figure 6.3. Comparison of the Quadratic and CMFD Eigenvalue Errors in Case I.



Figiire 6.4. Comparison of the Quadratic and CMFD Absolute Relative Power Errors in Case I.

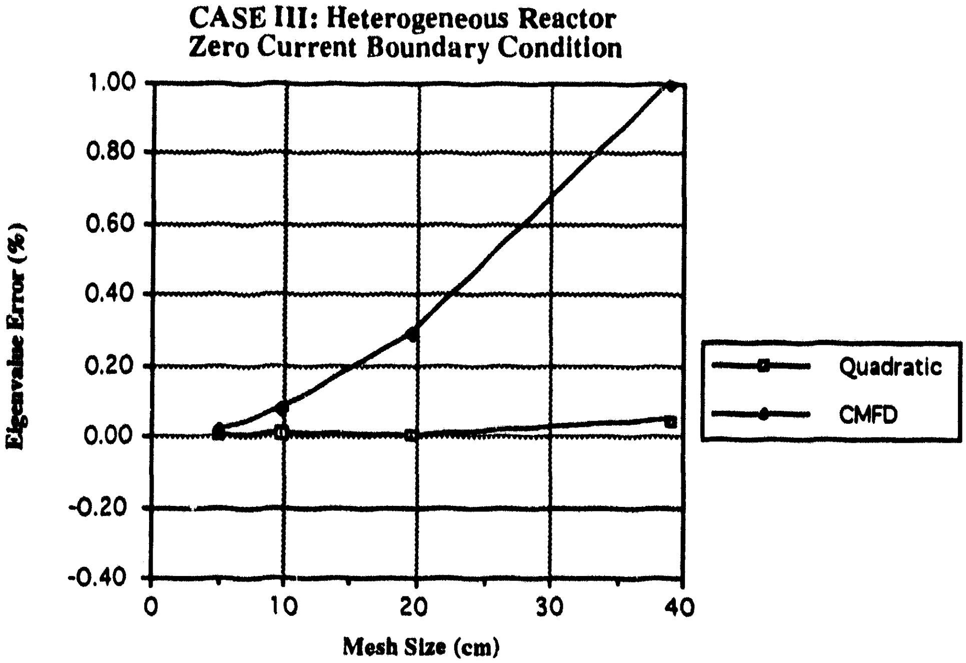


Figure 6.5. Comparison of the Quadratic and CMFD Eigenvalue Errors in Case III.

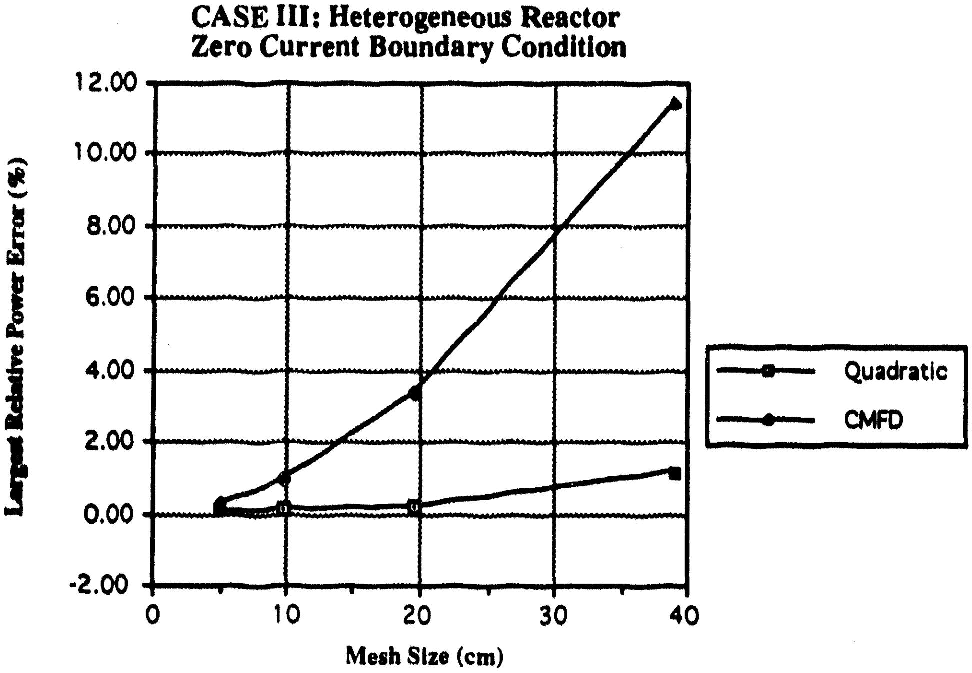


Figure 6.6. Comparison of the Quadratic and CMFD Absolute Relative Power Errors in Case III.

Table 6.2

Eigenvalue Errors for the Homogeneous Model with an Albedo Boundary Condition

CASE I: EIGENVALUE COMPARISONS

Nodes Per "Assembly"	Quadratic Eigenvalue	Quadratic Eigenvalue Error (%)	CMFD Eigenvalue	CMFD Eigenvalue Error (%)
1	1.01452	-2.34 %	1.04314	+0.42 %
4	1.03381	-0.48 %	1.04032	+0.15 %
16	1.03842	-0.04 %	1.03920	+0.04 %
64	1.03928	+0.05 %	1.03889	+0.01 %
REFERENCE EIGENVALUE: 256 Nodes Per Assembly = 1.03879				

Table 6.3

Absolute Maximum and Average Relative Power Errors for the Homogeneous Model with a Albedo Boundary Condition

CASE I: POWER COMPARISONS

Nodes Per "Assembly"	Quadratic MAX Relative Power Error (ABS)	Quadratic AVE Relative Power Error (ABS)	CMFD MAX Relative Power Error (ABS)	CMFD AVE Relative Power Error (ABS)
1	1.15 %	0.55 %	7.33 %	4.64 %
4	0.53 %	0.26 %	1.24 %	0.56 %
16	0.69 %	0.29 %	0.56 %	0.26 %
64	0.61 %	0.25 %	0.33 %	0.17 %
REFERENCE CASE: 256 Nodes Per Assembly				

Table 6.4

Eigenvalue Errors for the Homogeneous Model with a Zero Flux Boundary Condition

CASE II: EIGENVALUE COMPARISONS

Nodes Per "Assembly"	Quadratic Eigenvalue	Quadratic Eigenvalue Error (%)	CMFD Eigenvalue	CMFD Eigenvalue Error (%)
1	0.99953	-2.67 %	1.03209	+0.50 %
4	1.02026	-0.65 %	1.02879	+0.18 %
16	1.02536	-0.15 %	1.02745	+0.05 %
64	1.02655	-0.04 %	1.02707	+0.01 %
REFERENCE EIGENVALUE: 256 Nodes Per Assembly = 1.02694				

Table 6.5

Absolute Maximum and Average Relative Power Errors for the Homogeneous Model with a Zero Flux Boundary Condition

CASE II: POWER COMPARISONS

Nodes Per "Assembly"	Quadratic MAX Relative Power Error (ABS)	Quadratic AVE Relative Power Error (ABS)	CMFD MAX Relative Power Error (ABS)	CMFD AVE Relative Power Error (ABS)
1	1.29 %	0.50 %	8.96 %	5.63 %
4	0.57 %	0.27 %	1.37 %	0.63 %
16	0.53 %	0.26 %	0.54 %	0.25 %
64	0.31 %	0.16 %	0.32 %	0.16 %
REFERENCE CASE: 256 Nodes Per Assembly				

Table 6.6

Eigenvalue Errors for the Heterogeneous Model with a Zero Current Boundary Condition

CASE III: EIGENVALUE COMPARISONS

Nodes Per "Assembly"	Quadratic Eigenvalue	Quadratic Eigenvalue Error (%)	CMFD Eigenvalue	CMFD Eigenvalue Error (%)
1	1.23939	+0.04 %	1.25123	+1.00 %
4	1.23889	+0.00 % *	1.24247	+0.29 %
16	1.23882	+0.01 %	1.23981	+0.07 %
64	1.23884	+0.00 % *	1.23910	+0.02 %
REFERENCE EIGENVALUE: 256 Nodes Per Assembly = 1.23889				
* Not Exactly Zero: Less Than 0.004 %				

Table 6.7

Absolute Maximum and Average Relative Power Errors for the Heterogeneous Model with a Zero Current Boundary Condition

CASE III: POWER COMPARISONS

Nodes Per "Assembly"	Quadratic MAX Relative Power Error (ABS)	Quadratic AVE Relative Power Error (ABS)	CMFD MAX Relative Power Error (ABS)	CMFD AVE Relative Power Error (ABS)
1	1.15 %	0.59 %	11.38 %	4.38 %
4	0.18 %	0.09 %	3.36 %	1.29 %
16	0.09 %	0.04 %	0.92 %	0.34 %
64	0.06 %	0.03 %	0.21 %	0.08 %
REFERENCE CASE: 256 Nodes Per Assembly				

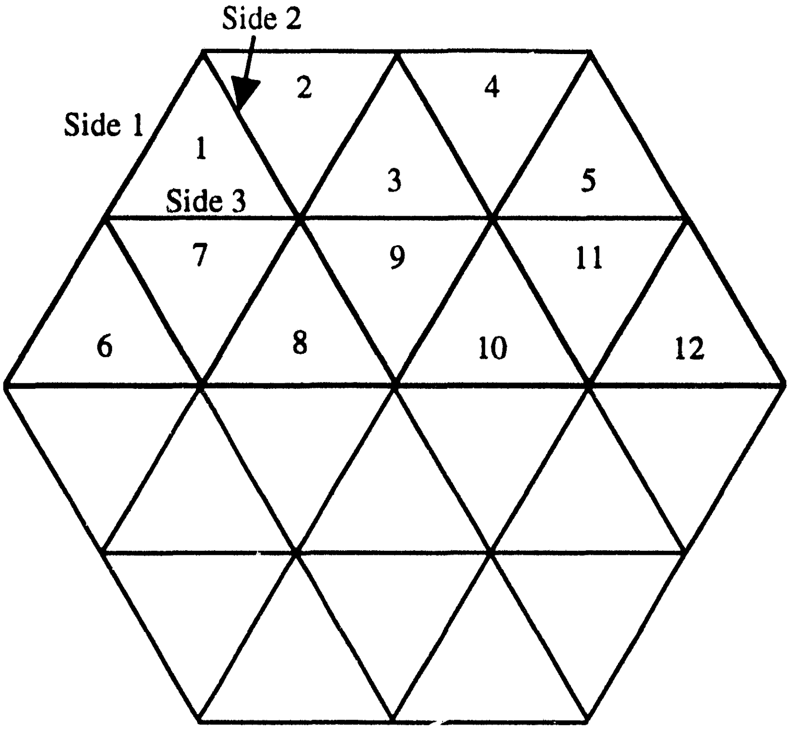


Figure 6.7. Relative Position of Node 1 for the Homogeneous Quadratic Tests.

Table 6.8

Node 1, Group 2 Reaction Rate Errors for the Homogeneous Model with a Zero Flux Boundary Condition

	CMFD (% Error)	Quadratic (% Error)
Absorption Rate	-8.5 %	+ 0.24 %
In-Scatter Rate	-8.7 %	+ 1.5 %

Table 6.9

Node 1, Group 2 Leakage Rate Errors for the Homogeneous Model with a Zero Flux Boundary Condition

	CMFD (% Error)	Quadratic (% Error)
Side 1	-8.5 %	+ 3.1 %
Side 2	0.0 %	0.0 %
Side 3	+ 10.3 %	+ 22.2 %

6.2.5 Numerical Instability of the Quadratic Method

In the results presented in the previous sub-section, all the cases converged. However, it was discovered that certain computations, performed at small mesh sizes, can lead to a numerical instability when the quadratic equations are employed. A rough mapping of the unstable cases showed a relationship between the number of nodes in the model and the size of the nodes used. Figure 6.8 shows this mapping for a homogeneous problem with a zero flux boundary condition. The stability line for same problem with an albedo boundary condition is a straight line at 0.01 cm.

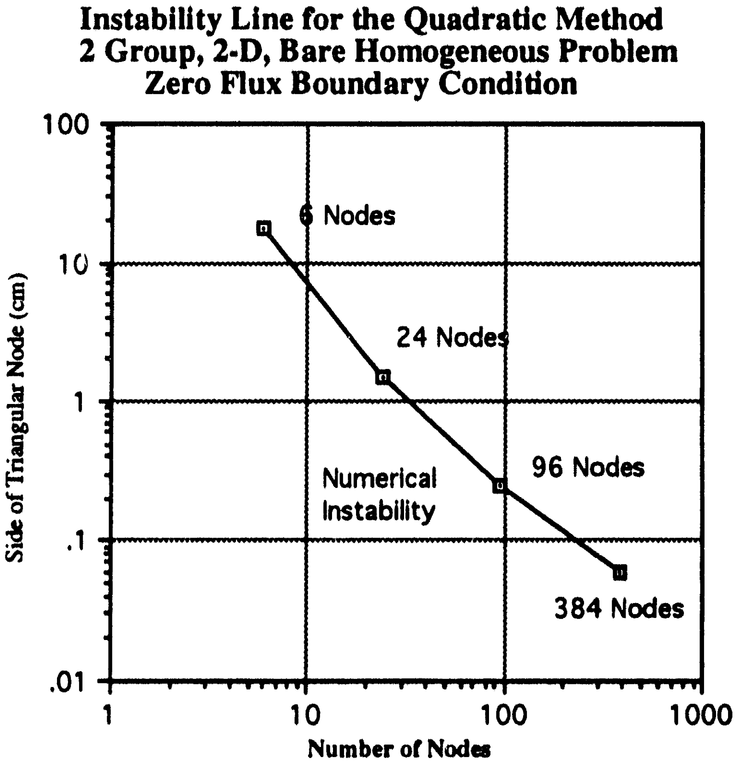


Figure 6.8. A Rough Mapping of the Numerical Stability of the Quadratic Method for 5 Outers per Update and a Zero Flux Boundary Condition.

In each of the cases run for Fig. 6.8, and in the cases where an albedo boundary condition was used, the discontinuity factor ratios were updated every 5 outer iterations. This number was used because previous experience with problems of small mesh size proved that 5 outers per update worked efficiently. Further study into the characteristics of the unstable region showed that the number of outers per update had a large effect on whether or not the problem converged. Surprisingly, the 6 node case was found to converge for all mesh sizes tested (as low as 1E-3 cm) if either 2 or 1 outers per update were used.

In order to further understand the nature of the instability, the 24 node case was tested using an albedo boundary condition. In these tests, the computer code was modified to test for diagonal dominance of the iteration matrix. Recall that a necessary condition for convergence is diagonal dominance, and if this is lost, the method is not guaranteed to converge. Because the discontinuity factor ratios are not constrained in value, and these discontinuity factor ratios are computed during the solution by the quadratic equations, it was thought that the value of these ratios may be disrupting diagonal dominance.

A summary of the results is presented in Table 6.10. It was previously mentioned that for an albedo boundary condition and 5 outer iterations per update, the instability was encountered at mesh sizes below ~0.01 cm. This can be seen in the second column of the table. It is difficult to explain the apparent trend (in the lower section of the table) of *convergence* with *even* numbers of outers per update and *non-convergence* with *odd* numbers of outers per update. This trend appears when diagonal dominance is lost for all outers after the first update.

Details from a few select cases of this brief study will be presented in figures showing eigenvalue convergence as a function of iteration. The eigenvalue convergence is defined as the relative difference between the current eigenvalue estimate and the estimate from the last iteration. For reference, the first figure presents a case which converges well. Figure 6.9 shows the eigenvalue convergence as a function of iteration for a 5 cm case, where the discontinuity factor ratios were updated every 3 outer iterations. The first update is performed after a specified convergence criterion is met. This can be seen in the figure between iteration 5 and 9, where the convergence is close to zero (it is not as close to zero as it looks). The spike at iteration 9 represents the first update of the discontinuity factor ratios. The update "changes the problem" by correcting the Coarse Mesh Finite Difference (CMFD) equations to match the Quadratic solution. The convergence drops in the subsequent 3 iterations because the problem is converging to the solution which is consistent with the most recent values of the discontinuity factor ratios. The second spike at approximately iteration 12 represents the second update, and this continues until the spikes "dampen out" and discontinuity factor ratios approach a constant value.

The next case tested was for a node side of 1 cm. Two outer iterations per update was used in this solution. Figure 6.10 shows that the spike is much larger than the one previously seen in Fig. 6.9. The diagonal dominance was lost on iterations 15 and 16, but restored for subsequent iterations. One possible explanation for this behavior is that the first estimate of the discontinuity factor ratios for some or all of the nodes were such that they disrupted diagonal dominance, but the *converged* (or asymptotic) values of the ratios were not. The problem converged despite the problems with diagonal dominance.

The next case demonstrates the effect that the number of outers per update can have on the convergence. The node size was chosen to be 0.1 cm. Figure 6.11 shows the results for the case of 5 outers per update. Note that the problem converges more slowly in the first few iterations (before the first update) than in previous cases because the spectral radii of the iteration matrix approach unity as smaller mesh sizes are used. The usual spikes are apparent in the figure (although it is interesting that the first spike is not the largest in this case), and the problem is seen to converge. Diagonal dominance was lost from iteration 65 until approximately iteration 95. Figure 6.12 shows the results when 1 outer iteration per update is used. In this case, the problem did not converge. The eigenvalue and eigenvalue convergence oscillated until the case was terminated at approximately 1080 iterations. It is interesting to note that the problem appeared to be converging well between iteration 100 and 200, but then began to oscillate. The diagonal dominance was lost on every outer iteration after the first update.

The last cases to be presented are for a node size of 0.001 cm. These cases also show dependence on the number of outer iterations per update. For each of these cases, the diagonal dominance was lost for all outers after the first discontinuity factor ratio update. Figure 6.13 shows the results for 5 outer iterations per update. The results show milder spikes than in previous problems, which continue for approximately 50 iterations, then dampen out. The problem appears to be converging when "ripples" begin. The error then begins to oscillate with increasing magnitude and the solution fails. Figure 6.14 presents the results for 4 outer iterations per update. The initial behavior is almost identical to Figure 6.13. In this case, however, the "ripples" never occur, and the problem converges. Figure 6.15 shows the results when 3 outer iterations per update are used. The initial behavior is again very similar to Fig. 6.13, but when "ripples" begin here, they lead to oscillations. The oscillations were followed until approximately 1000 iterations, and then the problem was terminated. The case of 2 outer iterations per update gave results almost identical to Fig. 6.14. The case of 1 outer iteration per update is presented in Fig. 6.16. The initial behavior is similar to the previous cases, but the solution immediately afterward leads to failure.

In conclusion, the numerical instabilities were only found to occur for the quadratic method at small mesh size. These difficulties should not be a problem for typical reactor applications. Nevertheless, an in-depth study is badly needed to better understand the nature of the non-linear iteration scheme and its use with the quadratic equations.

Table 6.10

A Summary of the Instability Tests for a 24 Node, Homogeneous Model with an Albedo Boundary.

	Number of Outers Per Update					
Node Size[1]	5	4	3	2	1	Converged Eigenvalue
5 cm	Converged in 170 Outers	Converged in 145 Outers	Converged in 149 Outers	Converged in 121 Outers	Converged in 51 Outers	4.31761E-1
1 cm	Converged in 160 Outers Diag. Dom. Lost on Outer # 20-25	Converged in 81 Outers Diag. Dom. Lost on Outer # 18-22	Converged in 118 Outers Diag. Dom. Lost on Outer # 16-18	Converged in 98 Outers Diag. Dom. Lost on Outer # 15-16	Converged in 58 Outers Diag. Dom. Lost on Outer #12	7.70863E-2
0.5 cm	Converged in 176 Outers Diag. Dom. Lost on Outer # 27-31	Converged in 168 Outers Diag. Dom. Lost on Outer # 25-28	Converged in 127 Outers Diag. Dom. Lost on Outer # 23-25	Converged in 106 Outers Diag. Dom. Lost on Outer # 21-22	Converged in 66 Outers Diag. Dom. Lost on Outer # 19-23	3.66196E-2
0.1 cm	Converged in 280 Outers Diag. Dom. Lost on Outer # 65-95	Converged in 249 Outers Diag. Dom. Lost on Outer # 73-124	Converged in 215 Outers Diag. Dom. Lost on Outer # 72-119	Converged in 165 Outers Diag. Dom. Lost on Outer # 70-97	Oscillation: No Convergence Diag. Dom. Lost on all Outers after # 69	7.00668E-3
0.01 cm	Oscillation: No Convergence Diag. Dom. Lost on all Outers after first update	Converged in 821 Outers Diag. Dom. Lost on all Outers after first update	Oscillation: No Convergence Diag. Dom. Lost on all Outers after first update	Converged in 401 Outers Diag. Dom. Lost on all Outers after first update	Oscillation: No Convergence Diag. Dom. Lost on all Outers after first update	6.93602E-4
0.001 cm	Instability Diag. Dom. Lost on all Outers after first update	Converged in 3433 Outers Diag. Dom. Lost on all Outers after first update	Oscillation: No Convergence Diag. Dom. Lost on all Outers after first update	Converged in 3432 Outers Diag. Dom. Lost on all Outers after first update	Instability Diag. Dom. Lost on all Outers after first update	6.92898E-5

[1] The side of an equilateral triangle

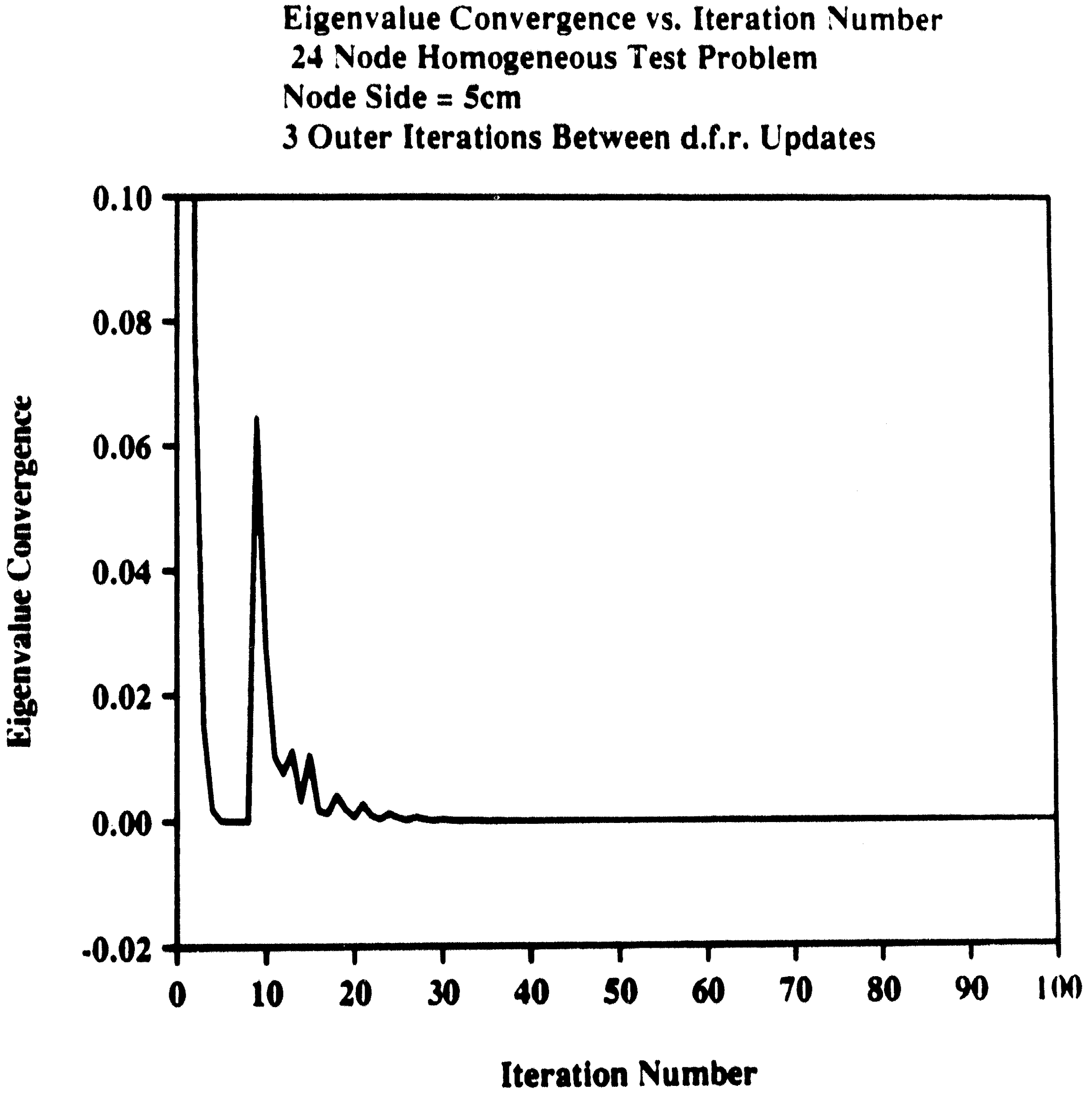


Figure 6.9. Eigenvalue Convergence for Node Size of 5 cm and 3 Outers per Update.

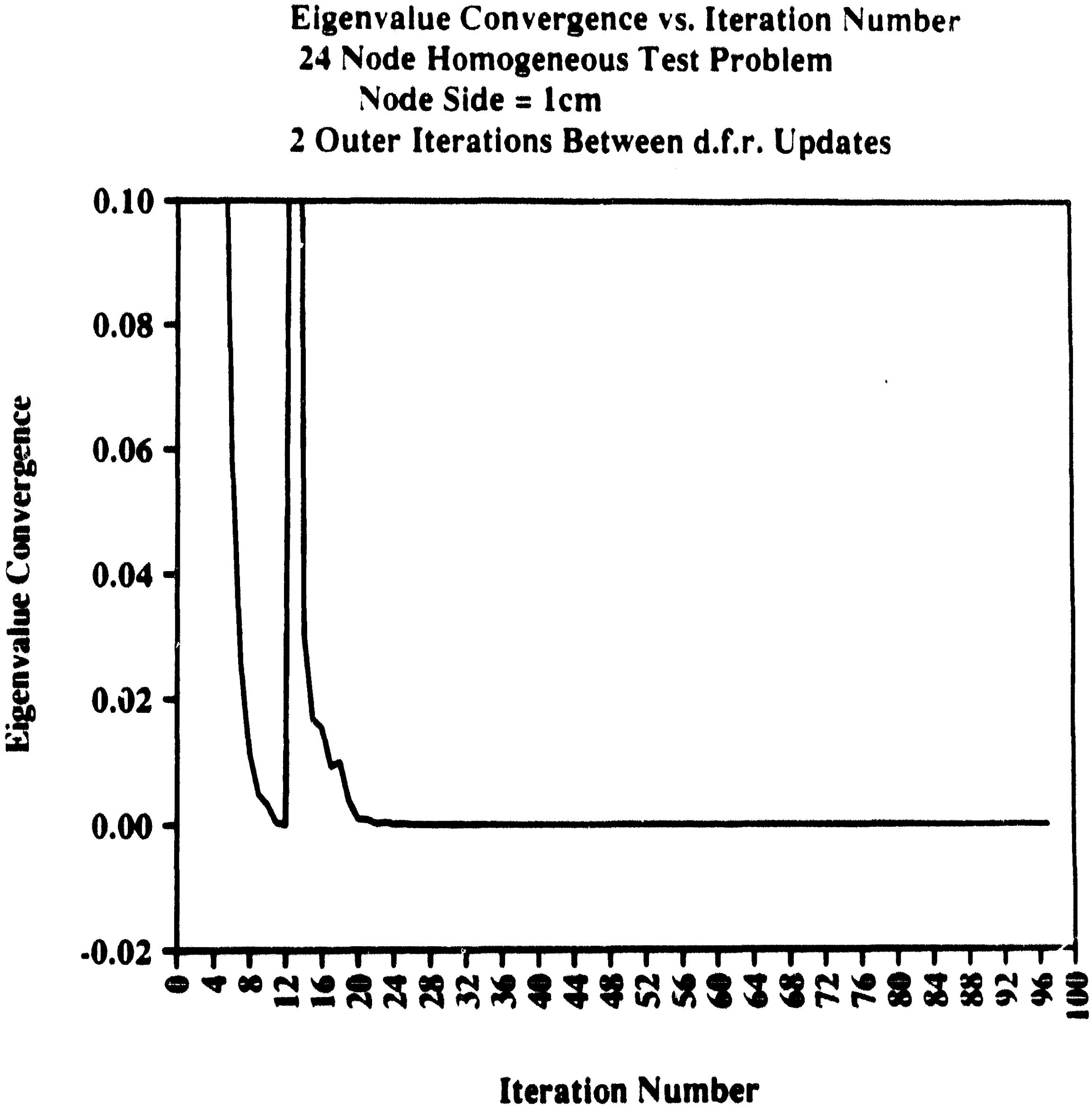


Figure 6.10. Eigenvalue Convergence for Node Size of 1 cm and 2 Outers per Update.

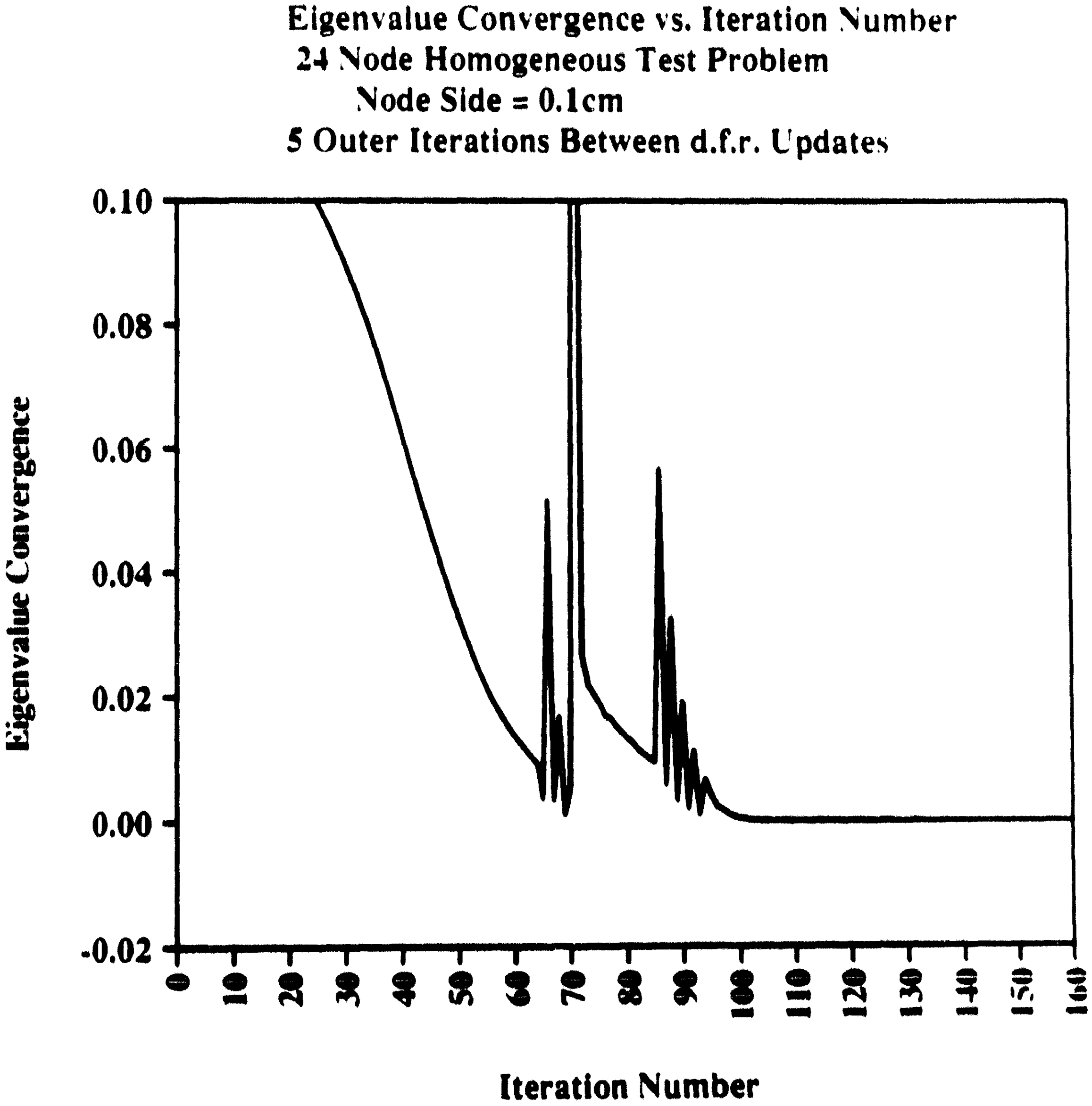


Figure 6.11. Eigenvalue Convergence for Node Size of 0.1 cm and 5 Outers per Update.

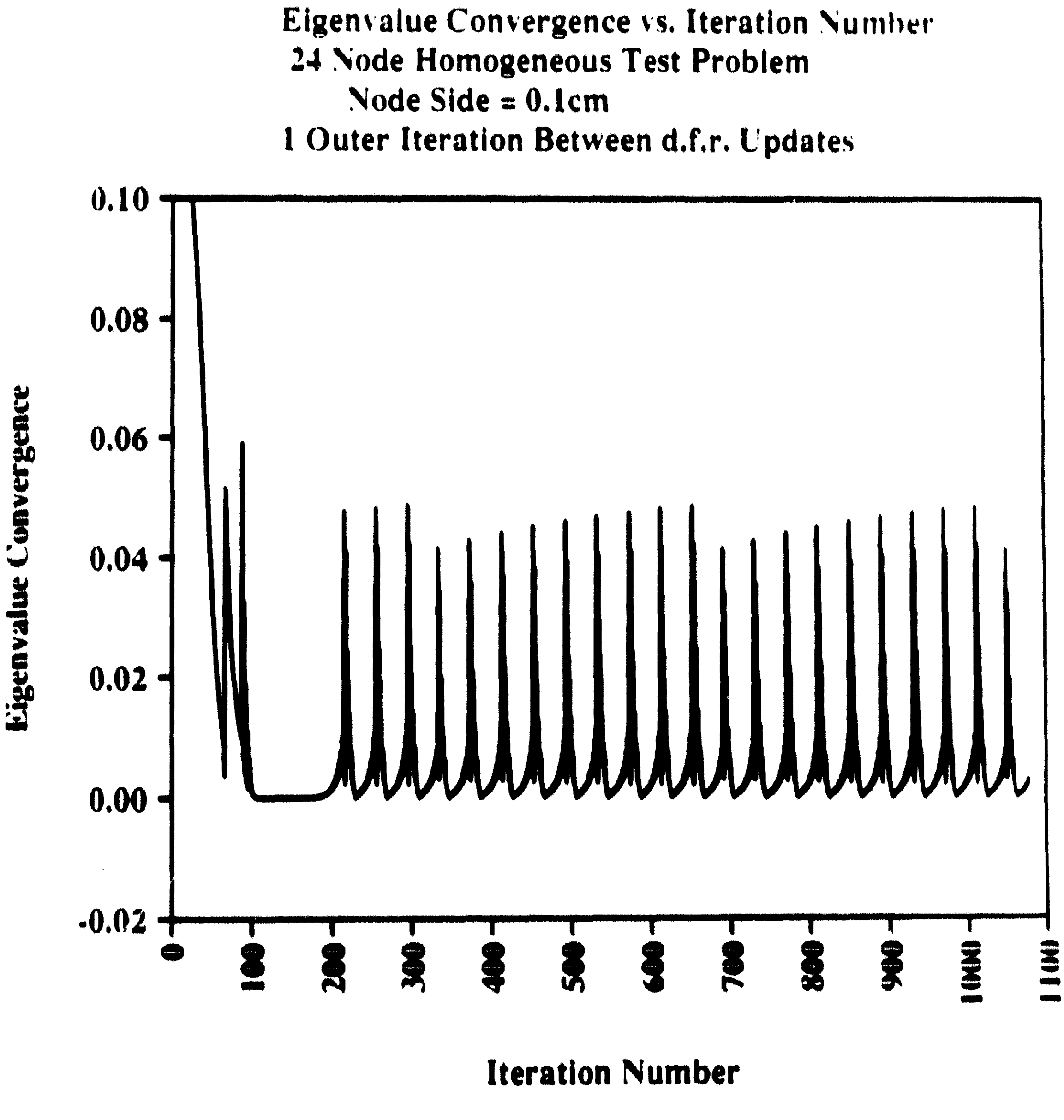


Figure 6.12. Eigenvalue Convergence for Node Size of 0.1 cm and 1 Outers per Update.

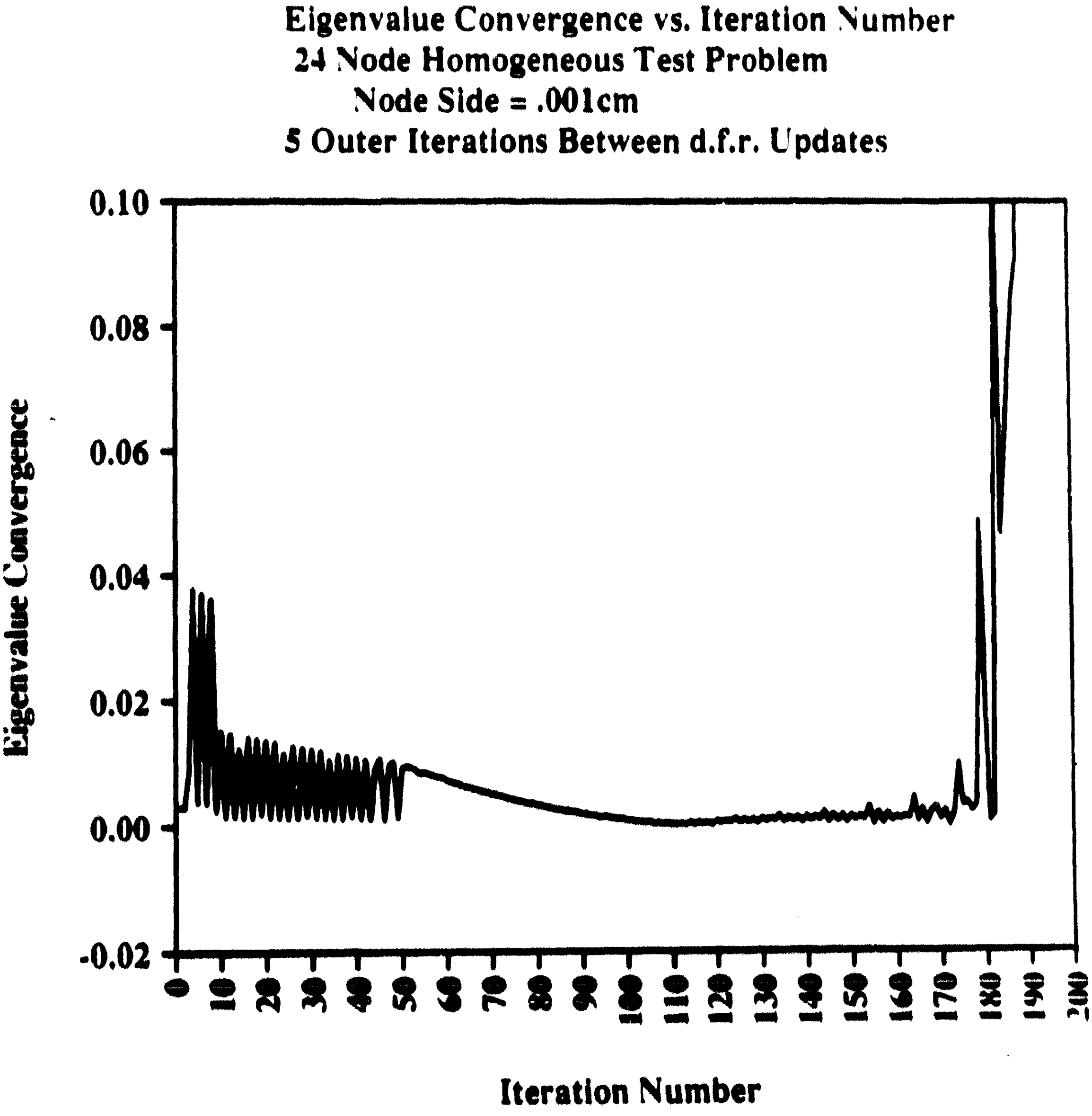


Figure 6.13. Eigenvalue Convergence for Node Size of 0.001 cm and 5 Outers per Update.

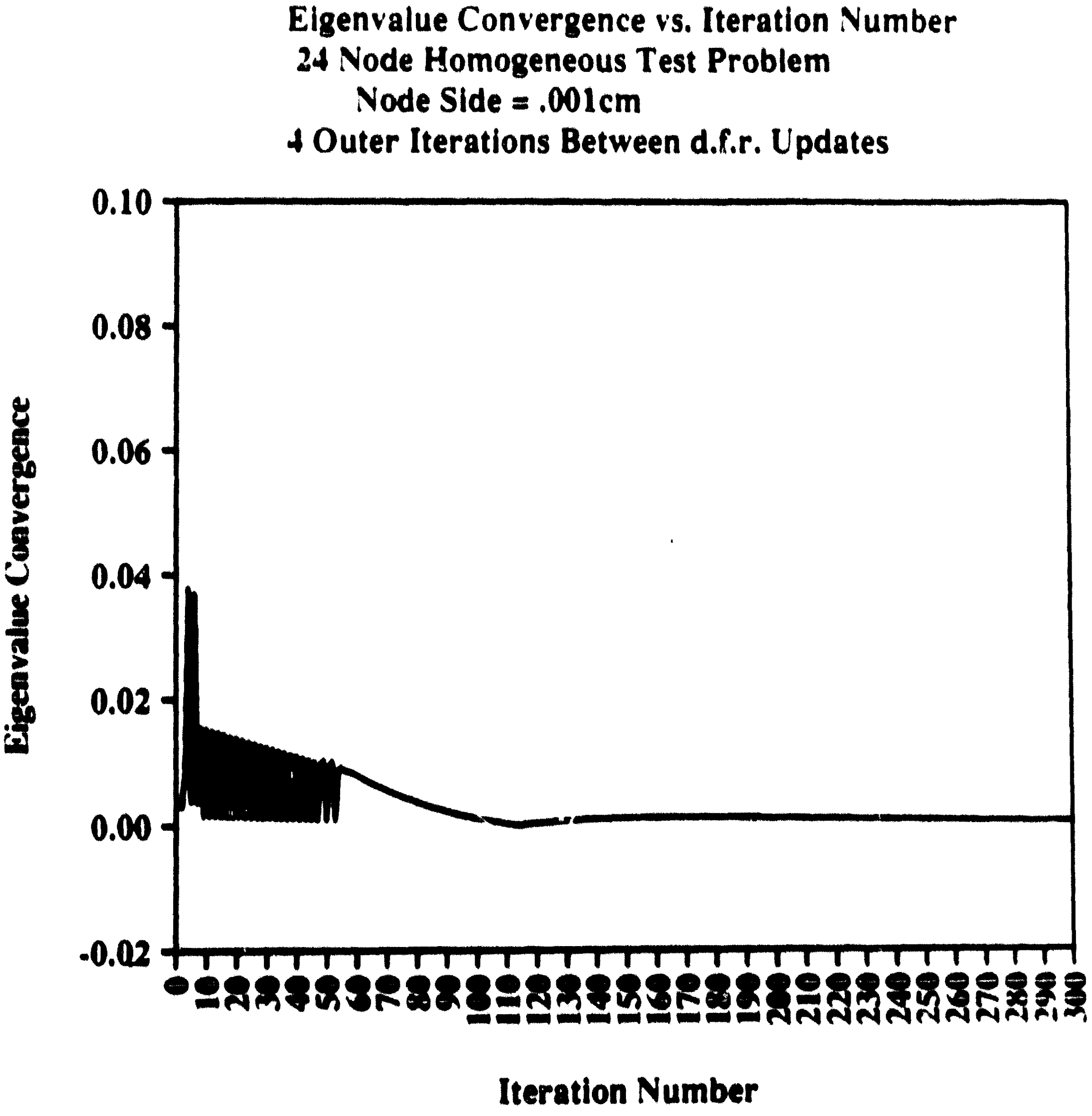


Figure 6.14. Eigenvalue Convergence for Node Size of 0.001 cm and 4 Outers per Update.

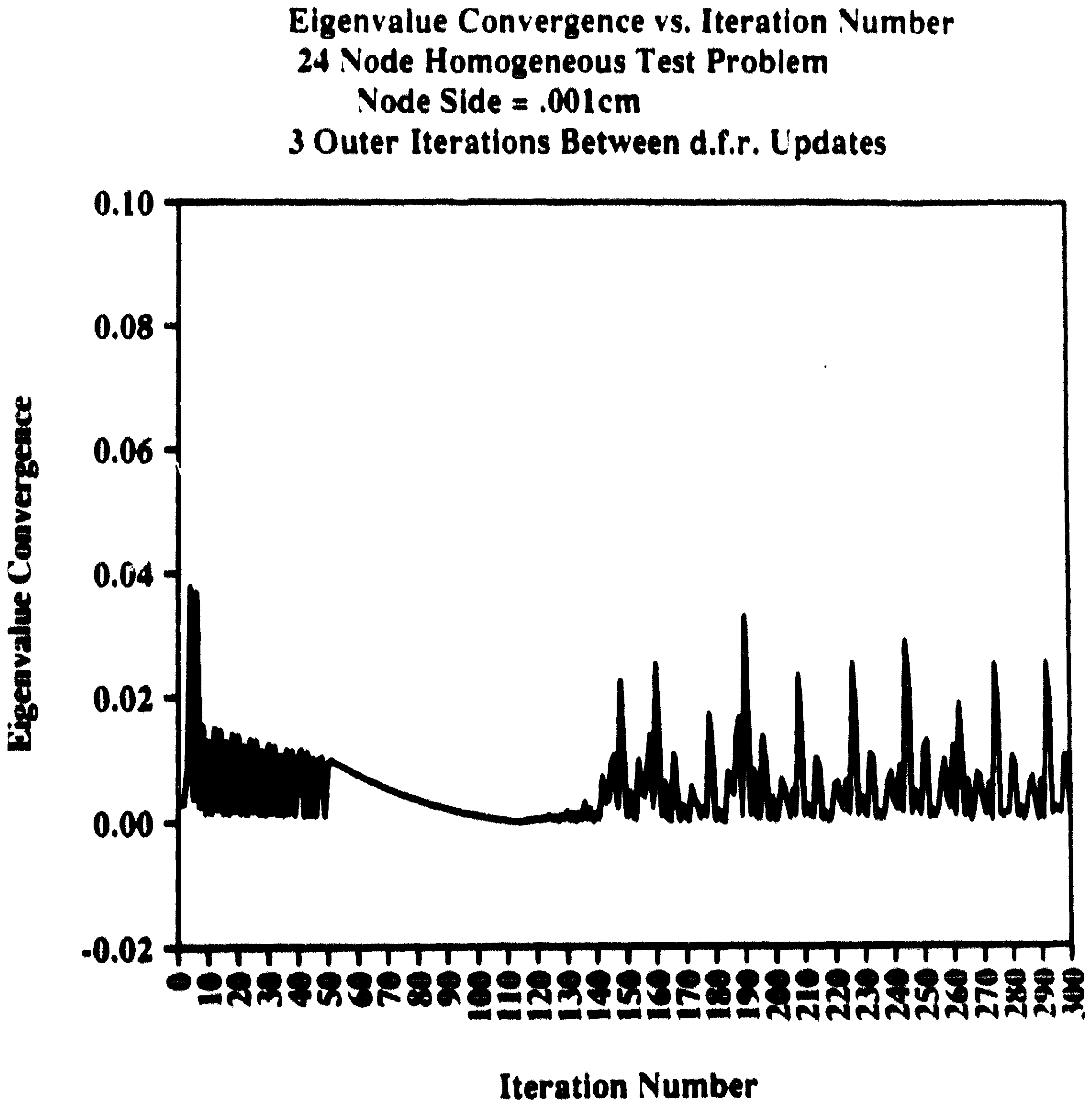


Figure 6.15. Eigenvalue Convergence for Node Size of 0.001 cm and 3 Outers per Update.

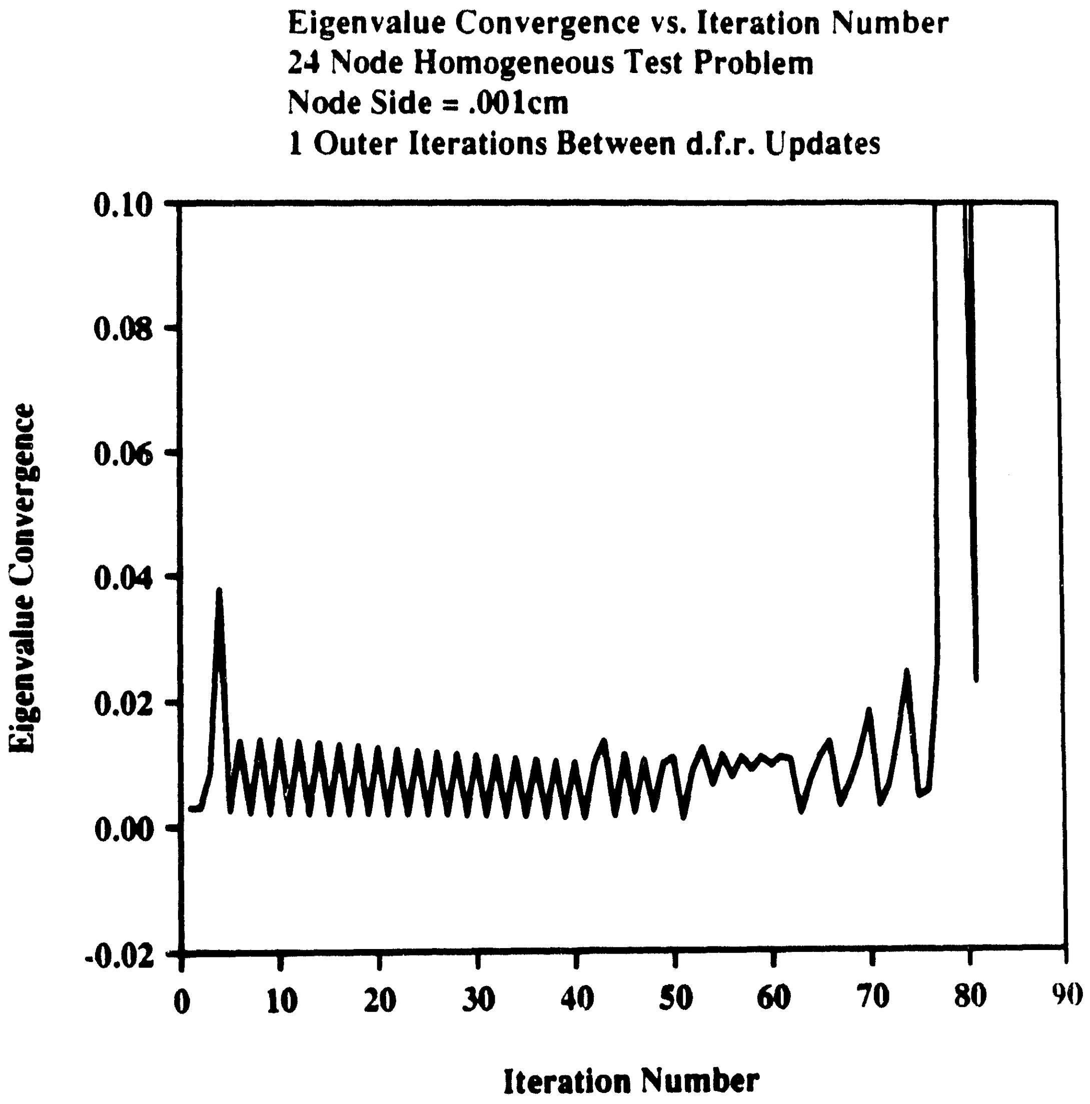


Figure 6.16. Eigenvalue Convergence for Node Size of 0.001 cm and 1 Outers per Update.

6.3 Testing the Transient Method

In order to test the transient solution methods, two benchmark problems were run. The first was a simple model with a zero current boundary condition involving a uniform step change at time zero. The analytic solution to this problem was available from Ref. [B-1]. The second benchmark problem consisted of a non-uniform step change transient in a large, heterogeneous heavy water reactor. This problem was originally used in the verification of the computer code TRIMHX [P-2], a three-dimensional diffusion theory code for hexagonal-z geometry which was developed at Savannah River Laboratory. Results obtained from QUARTZ are compared to the TRIMHX results given by Ref. [B-2].

Because of the difficulty in obtaining other benchmark problems for triangular-z geometry, the remaining studies were used as simple 'consistency checks', to verify that the transient methods in QUARTZ give physically acceptable results to selected transients.

6.3.1 Transient Analytic Test Problem

The analytic test problem, constructed by Byers [B-1] (from the work of Dias [D-1]), was used to test the transient solution methods in the computer code ZAQ. It is a simple model, independent of spatial mesh because of the zero current boundary condition on the outer surfaces. Appendix C, Section 6 presents the cross sections and constants used for the model. The analytic solution of the reactor power as a function of time is given as [B-1]:

$$p(t) = a_1 e^{\omega_1 t} + a_2 e^{\omega_2 t} + a_3 e^{\omega_3 t} \qquad (6.1)$$

where the constants are given in Table 6.11. The eigenvalue for this problem is exactly 1.00.

Table 6.11

Constants for the Analytic Solution

a_1	1.5069321349E+00
a_2	-5.0756822664E-01
a_3	6.3909171724E-04

ω_1	4.0582083930E-02
ω_2	-1.7804156798E+02
ω_3	-1.1072207901E+05

A static calculation of the perturbed system gave a reactivity estimate of 2.5236E-3 (the eigenvalue was 1.00253). The transient problem was solved by the quasi-static method and the point kinetics method using the computer code QUARTZ. The transient was initiated by a step decrease in the sigma cross section in group 2 at time zero. The quasi-static solution is shown in Fig. 6.17. As can be seen in the figure, the QUARTZ solution matches the analytic solution very well. It should be noted that the neutron speeds for the reference solution were not given, so typical speeds were used (see Appendix C). The fact that the prompt jump occurs sooner for the quasi-static method in Fig. 6.17 indicates that the assumed speeds were higher than the speeds used in the reference problem.

The asymptotic reactivity computed during the solution was 2.5253E-3.

The point kinetics solution is shown in Fig. 6.18. In the figure, it can be seen that the point kinetics solution is acceptable, but does not quite match the reference solution.

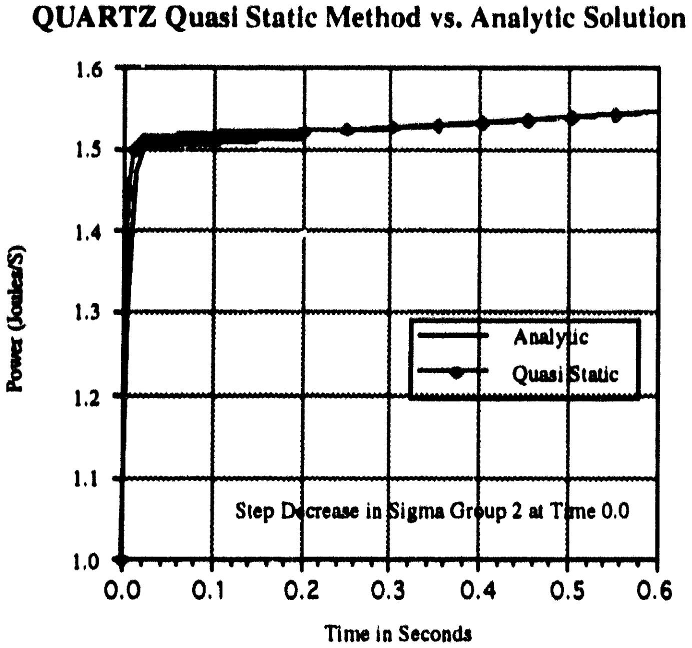


Figure 6.17. QUARTZ Quasi Static Solution to the Analytic Benchmark Problem.

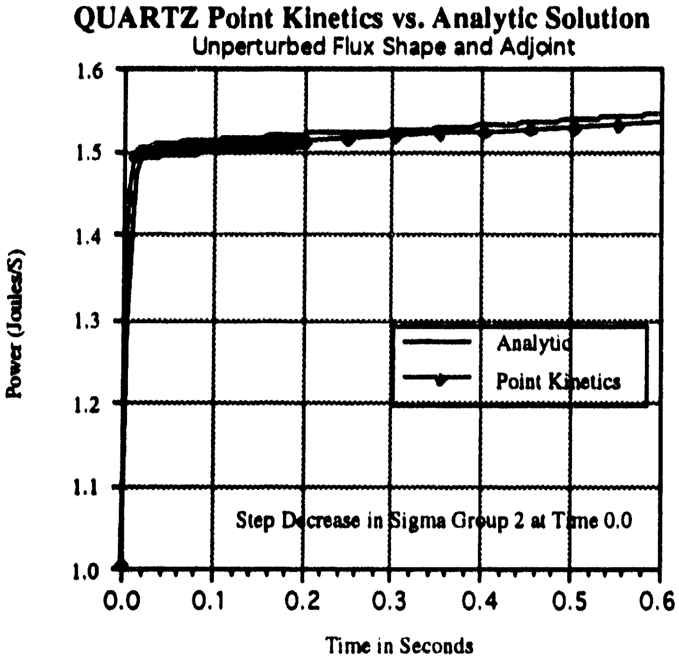


Figure 6.18. Point Kinetics Solution to the Analytic Problem Using Static Flux and Adjoint.

The point kinetics solution uses the unperturbed flux and unperturbed adjoint to compute the point kinetics parameters. Point kinetics usually gives poor results when the flux shape during the transient differs significantly from the static flux shape (i.e. the shape that was used to compute the point kinetics parameters). In the analytic benchmark problem, the perturbation is spatially uniform, meaning that the spatial shape does not change during the transient. However, because the perturbation is made to the group 2 cross section only, the fast-to-slow flux ratio (the ratio of the group 1 flux to the group 2 flux) changes, and thus the point kinetics parameters used during the transient reflect the incorrect, static energy shape. To test this, a static calculation was run using the perturbed cross sections. The forward and adjoint fluxes were edited and then used to compute point kinetics parameters. The transient was then run exactly as before, except that the improved point kinetics parameters were used. The result is shown in Fig. 6.19. It can be seen that the solution now matches the reference very well.

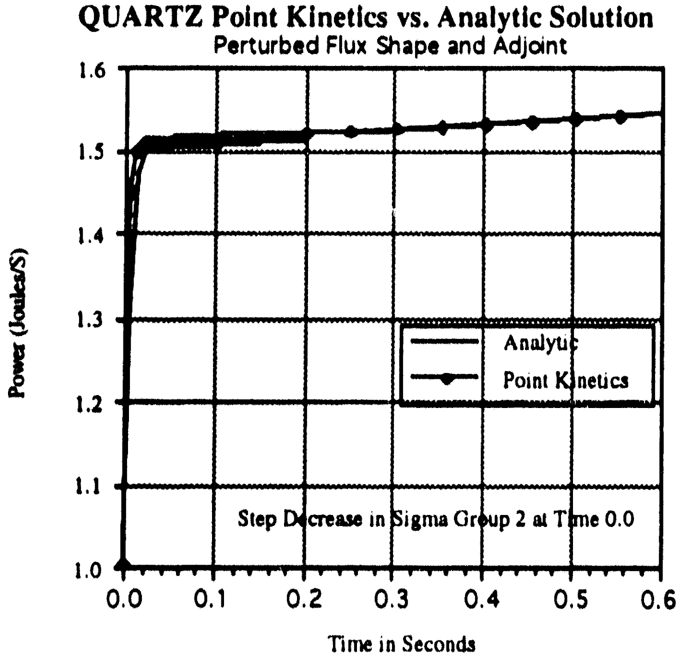


Figure 6.19. Point Kinetics Solution to the Analytic Problem Using Perturbed Flux and Adjoint.

6.3.2 Buckner-Stewart Heterogeneous Test Problem

The Buckner-Stewart benchmark problem was obtained from Ref. [B-2], and consists of a non-uniform step change in the central region of a large, heterogeneous heavy water reactor. The details of the model are given in Appendix C, Section 7. It was originally used to test the finite difference, hexagonal diffusion theory code TRIMHX [B-2]. The core consists of hexagonal assemblies arranged in a pattern with 60 degree rotational symmetry. The model has five unique axial planes, and is symmetric about the center plane. The perturbation consisted of an axially uniform 15% decrease in the thermal capture cross section in the central patch of assemblies at time zero. Thermal hydraulic feedback is not modeled.

The 1/6, full height core was modeled using QUARTZ. The total number of nodes, using six triangles per hexagon was 4468. The quasi-static method was used with two time domains. The first time domain, from 0 to 1 second, used a time step size of 5 msec. The second time domain, from 1 second to 5 seconds, used a time step size of 50 msec. The spatial model was the CMFD method. The total calculation time for this problem was 77 minutes on a VAX 3100 computer.

The reactivity as a function of time is given in Fig. 6.20. The asymptotic reactivity was computed to be 5.8738E-3, or 90.37 cents (beta is .0065 for the single delayed neutron precursor group). The reactivity computed by TRIMHX, and reported in Ref. [B-2] was 87.15 cents. In this transient, the shape of the neutron flux changes dramatically over the first second. Figure 6.21 shows the radial distribution of the thermal neutron flux at midplane as a function of time, where radial position 0 is at the center of the reactor, and position 12 is at the outer surface. All four of the thermal flux profiles are normalized in this figure such that the flux in the central node (radial position 0) is unity. The reactivity reaches its asymptotic value when the flux reaches its asymptotic shape. The flux shape at 1 second in Fig. 6.21 qualitatively matches the asymptotic flux shape reported in Ref. [B-2].

The normalized central node thermal flux as a function of time is shown in Fig. 6.22. Because the reactivity is close to prompt critical, the three to four percent difference between the QUARTZ and TRIMHX reactivity has a relatively large effect on the power as a function of time. The QUARTZ solution for the power is consistent with the larger estimate of the reactivity.

One major difference between the TRIMHX model and the QUARTZ model is the spatial mesh. The QUARTZ model uses six nodes per hexagon, and the TRIMHX model uses one node per hexagon. This difference may account for the difference in the reactivity estimate.

This benchmark problem was also solved by the quadratic hexagonal code DIF3D [T-1]. Ref. [T-1] did not specifically state the spatial model used in the calculation. However, it is implied that the quadratic nodal method (in hexagonal geometry) was used. The published asymptotic reactivity was given to be 84 cents.

It is difficult to draw strong conclusions about the transient methods based upon the results of this benchmark problem. The reactivity calculation, using the quasi-static method, predicted the correct value within 4% of reference, but because the reactivity was close to prompt critical, the power was significantly overpredicted. It appears that the solution is sensitive to the spatial model used. The reference calculation used 1 node per hexagon in CMFD. The QUARTZ model used 6 nodes per hexagon, in CMFD. The DIF3D model used 1 node per hexagon with a quadratic method. All three gave different estimates of the reactivity.

The next logical step would have been to use the quadratic method in the QUARTZ model. Unfortunately, the quadratic method in QUARTZ can only be used for problems with uniform boundary conditions on the outer periphery of the reactor. The benchmark problem is a 1/6 core, with zero current boundary condition on two sides. A full core model could have been used, but the size of the problem would have been prohibitively large.

The requirement of a uniform outer boundary condition is not imposed by limitations in the method, but rather by the limitations of the computer code. The difficulties are caused by the inability to properly locate the points of the triangles on the reactor surface. The fact that a point lies on the surface can be ascertained by the fact that the node beyond the point does not exist. However, sufficient information is not available to specifically locate the point on one side of the reactor or another. This problem chiefly arises at the corners of the reactor, where two surfaces meet with different outer boundary conditions.

In short, it is disappointing that the quadratic method could not be used for this problem. In the future, these difficulties may be circumvented by customizing the triangular quadratic method to the specific reactor type of interest. In that way, a systematic way of identifying points on the reactor surface would be tractable. In an effort to keep the computer code QUARTZ very general, a systematic way of identifying points on the surface became too difficult to overcome.

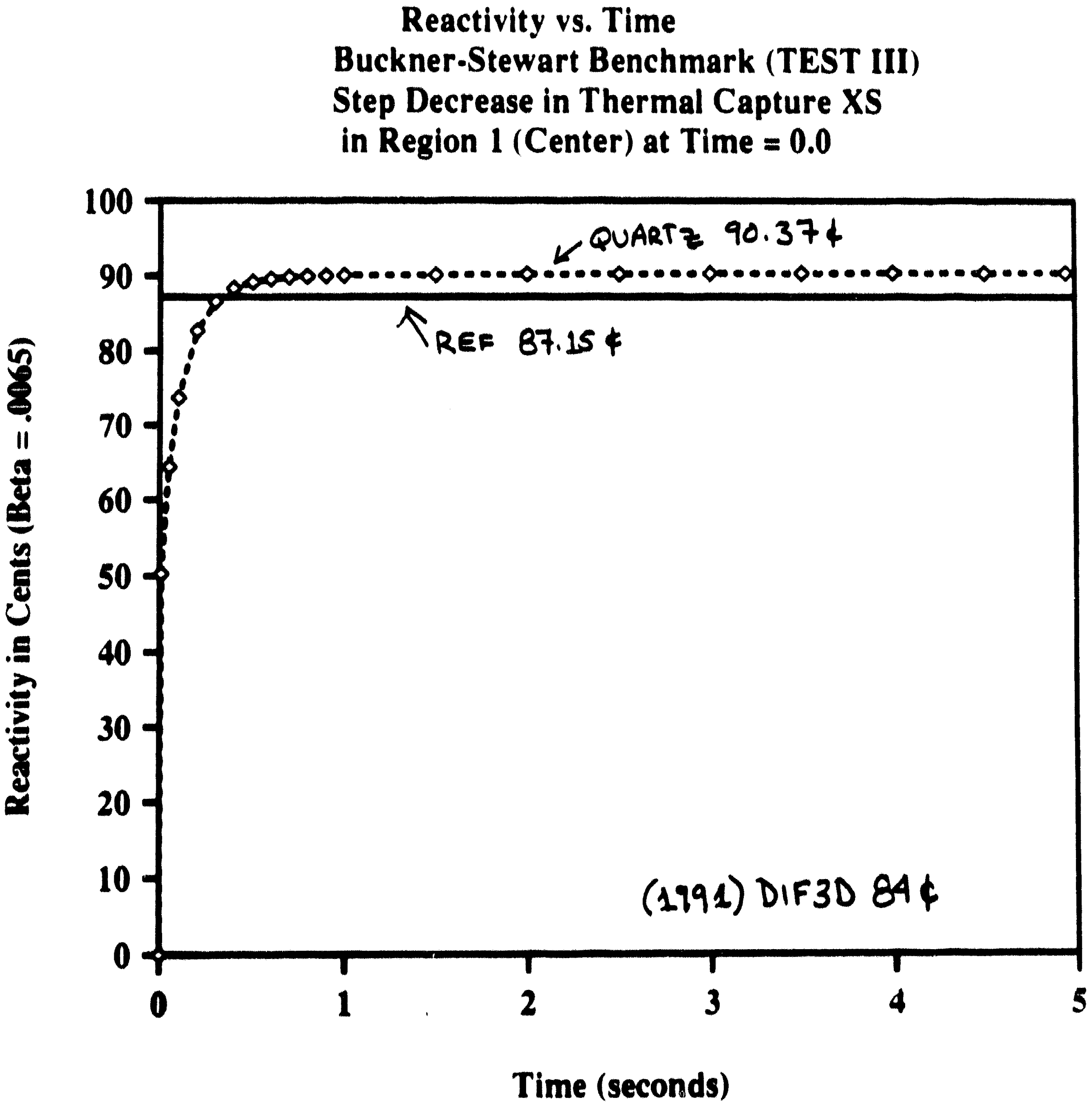


Figure 6.20. Reactivity as a Function of Time for the Buckner-Stewart Benchmark Problem.

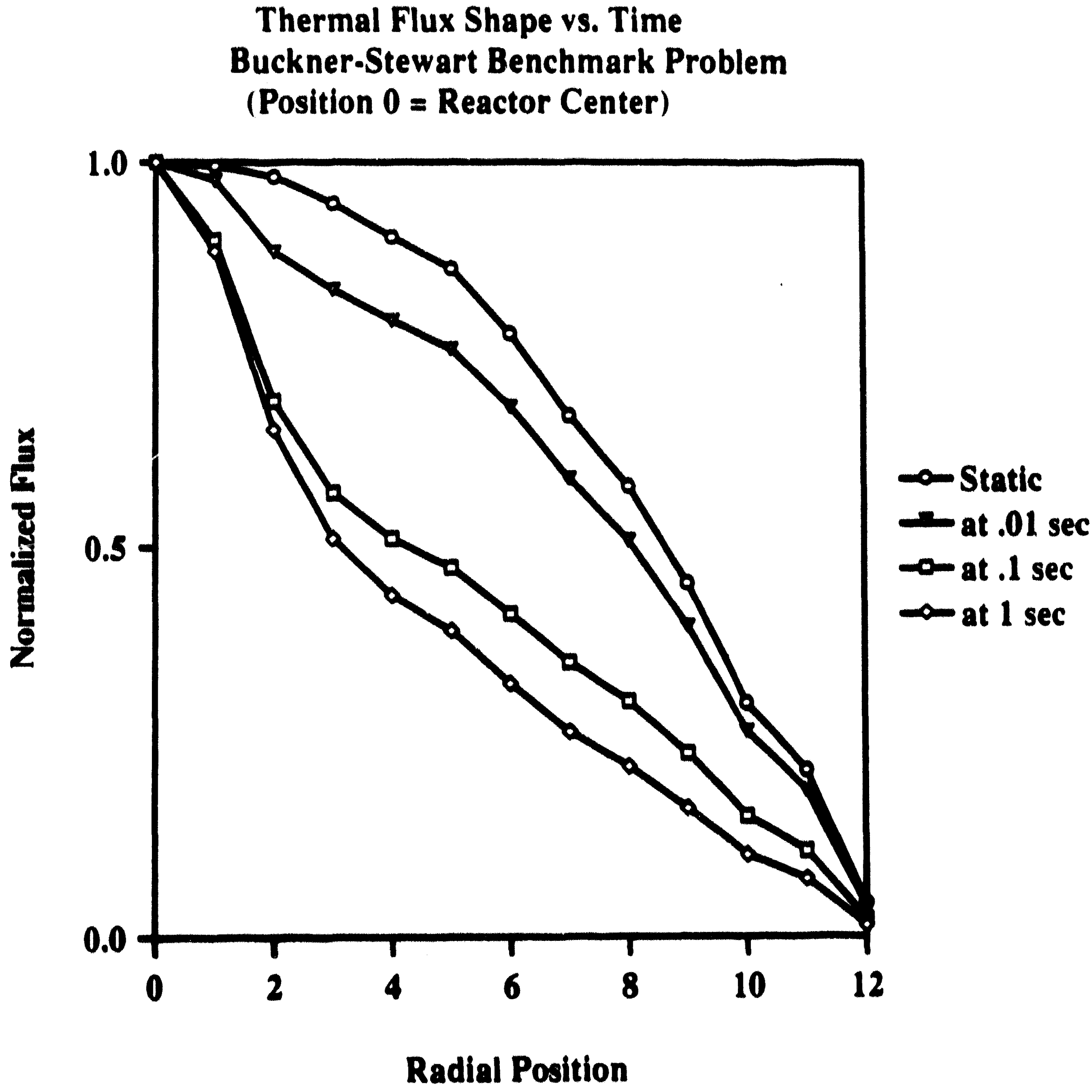


Figure 6.21. Thermal Flux Shape as a Function of Time for the Buckner-Stewart Benchmark Problem.

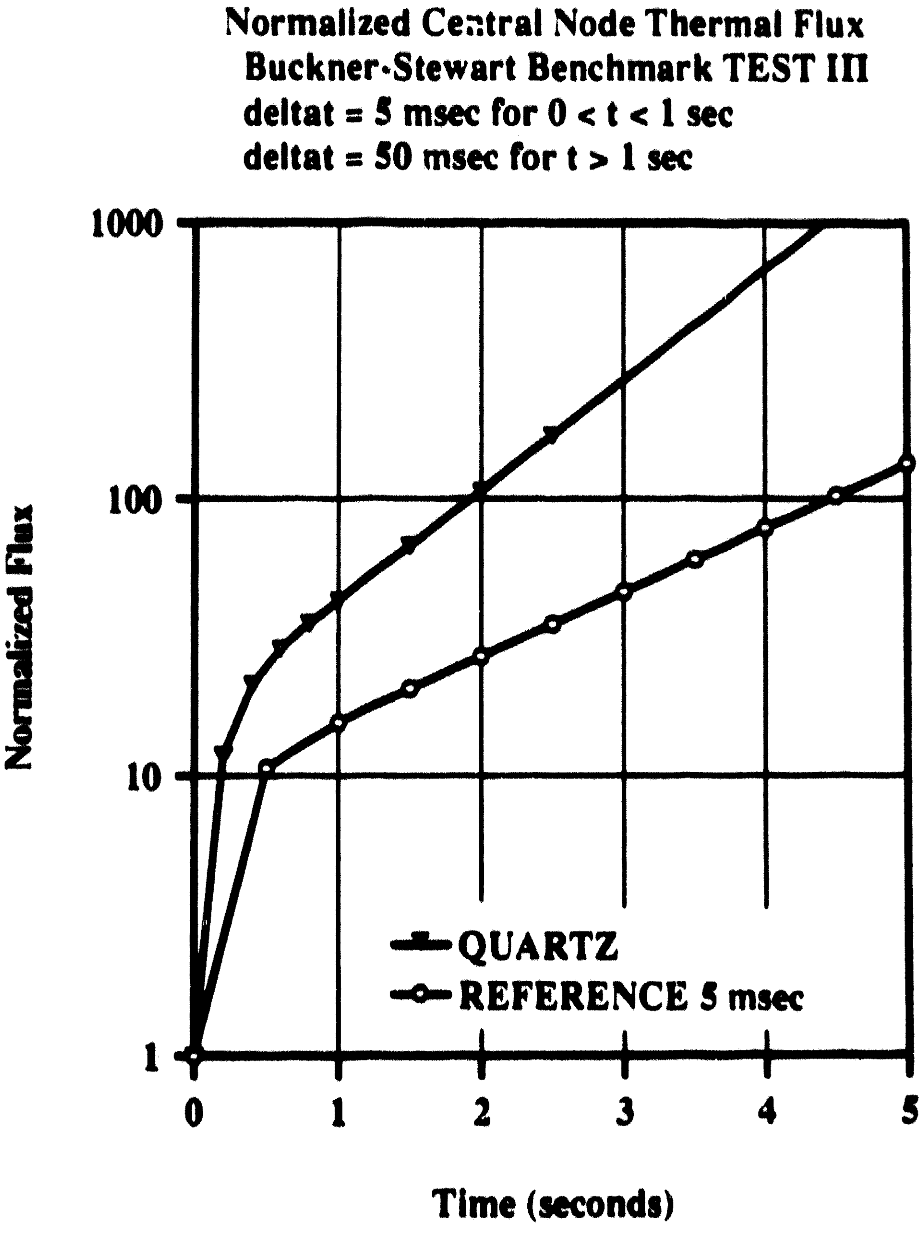


Figure 6.22. Central Node Thermal Flux as a Function of Time for the Buckner-Stewart Benchmark Problem.

6.3.3 Other Consistency Studies

In order to further test the transient methods, three generic problems were created to check the power response of the method to common transients. In each of the three tests, a 3-D, two-group, homogeneous model was used.

The first test consisted of a uniform step decrease in the thermal absorption cross section at t=5 seconds. The problem was run with and without thermal feedback. Figure 6.23 shows the power response as a function of time for the feedback case (dotted line) and the no feedback case (solid line). The case without feedback immediately jumps to a positive and constant value of reactivity. The power increases without bound, as expected. The case with feedback is shown as the dotted line in the figure. The power initially increases because of the decrease in the absorption cross section, but as the temperatures rise, the cross section feedback eliminates the positive reactivity and the power goes to a constant value at a higher level.

The second test consisted of a step decrease in the inlet flow rate to 80% of the initial value from t=5 seconds until t=30 seconds, when the flow rate was restored to 100%. Inlet coolant temperature was assumed constant throughout the transient. Figure 6.24 shows the total reactor power response to the transient. In this figure, the solid line represents the power response, and the dotted line represents the steady state power level. The decrease in the flow rate causes the temperatures in the reactor to rise. The cross section feedback gives a negative reactivity, and the power decreases. At 30 seconds, the flow rate is restored. The temperatures drop, the reactivity becomes positive, and the power level goes to its initial, steady state level.

The third test consisted of a step increase in the inlet coolant temperature at t=10 seconds, and restoration of the initial temperature at t=30 seconds. The coolant flow rate was assumed to be constant throughout the transient. Figure 6.25 shows the results for the test. The solid line is the power response and the dotted line is the initial power level. The power decreases at t=10 seconds due to the increase in temperature, and returns to its steady state level a short time after the inlet temperature is restored.

In conclusion, the transient thermal feedback models have been shown to give expected results to three types of transients. This further indicates that the transient methods have been implemented correctly.

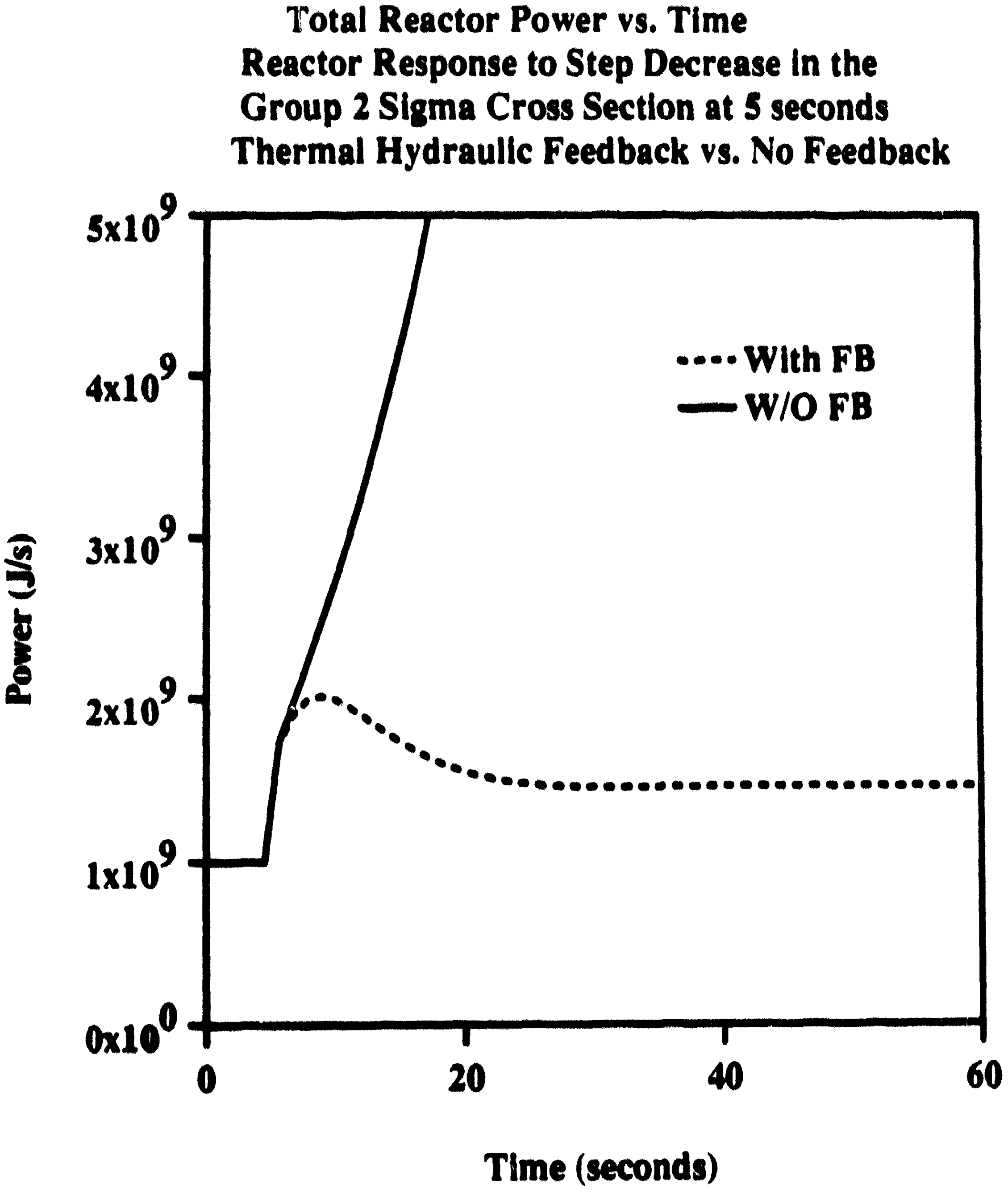


Figure. 6.23. The Reactor Power as a Function of Time for the First Consistency Check.

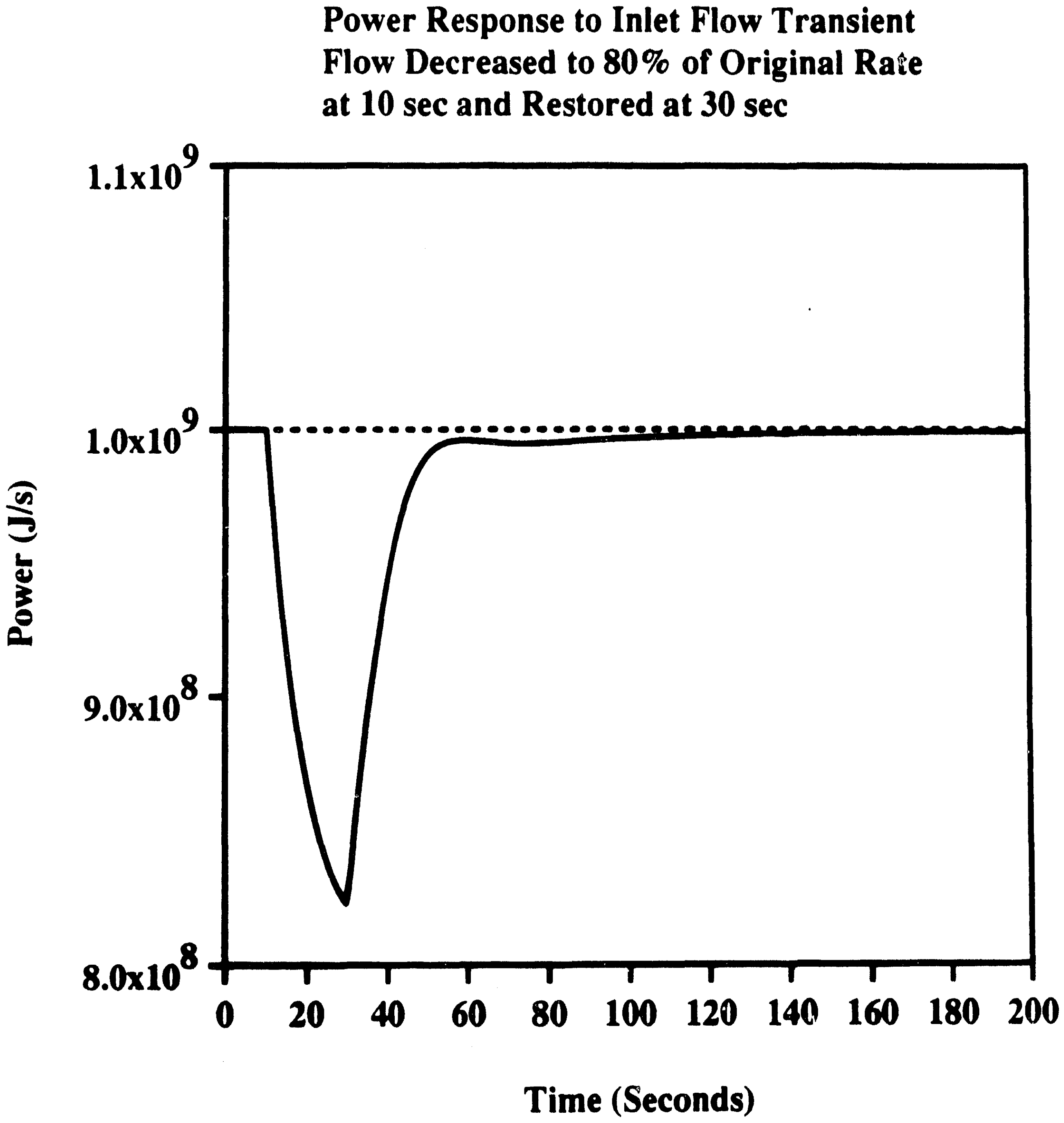


Figure. 6.24. The Reactor Power as a Function of Time for the Second Consistency Check.

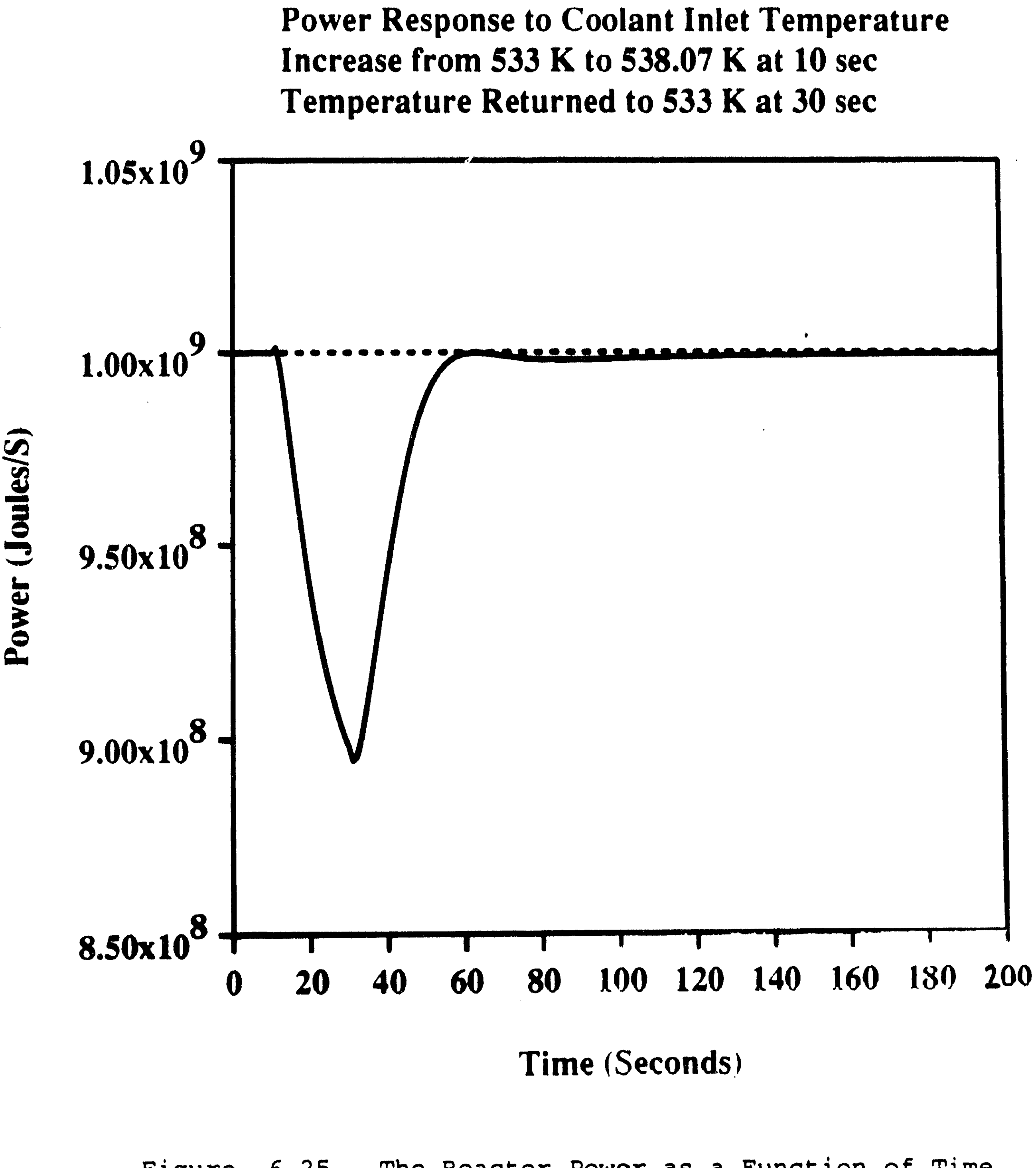


Figure. 6.25. The Reactor Power as a Function of Time for the Third Consistency Check.

6.4 Summary

In this chapter, the static and transient solution methods have been tested. The static method was initially shown to match very simple infinite cell and infinite slab analytic solutions. The static method was also compared to results from a 2-D, finite difference triangular code MITHEX. These comparisons demonstrated that the Coarse Mesh Finite Difference (CMFD) equations were correctly implemented and solved. The quadratic equations were then shown to reduce to the CMFD equations in the limit of small mesh size. This was a very important result because the CMFD equations, in the limit of small mesh size, are guaranteed to converge to the true solution of the nodal diffusion equations.

The efficiency and accuracy of the quadratic method was then compared to the CMFD method. The results from a homogeneous test problem were presented first. The quadratic method was shown to give a more accurate estimate of the nodal fluxes at coarse mesh for both zero flux and zero net returning current boundary conditions. However, the eigenvalue estimates were less accurate. Also, for the zero net returning current boundary condition case, the quadratic method exhibited inconsistent error reduction with mesh size when small mesh sizes were used. This problem was not seen in the zero flux boundary condition case, and is attributed to the boundary condition approximations mentioned in Chapter 3 and discussed in Appendix A. A heterogeneous, zero current case was run to test the two methods without the influence of the boundary conditions. The quadratic method produced very accurate estimates of the eigenvalue and fluxes at coarse mesh. It was proposed that the poor estimations of the eigenvalue by the quadratic method was caused by the inability of the quadratic polynomial to accurately predict the current at the points and surfaces of the triangular nodes. Previous work by Gehin [G-1] in Cartesian geometry has shown that quadratic polynomials can accurately predict node-averaged fluxes, but can have difficulty predicting the current at the surfaces of the node.

The quadratic method was found to be numerically unstable when very small mesh sizes were used. This instability was investigated for a 24 node, 2-D problem, and the results were presented. It was mentioned that these problems should not be encountered in typical reactor applications.

The transient solution methods were tested using a simple analytic problem, and the Buckner-Stewart benchmark problem. A few consistency test cases were also run. The simple analytic problem demonstrated that the fundamental transient equations worked properly, although the problem did not include leakage. The results from the Buckner-Stewart problem were difficult to interpret. The shape distortion due to the initiating perturbation was presented, and it was mentioned that the results "qualitatively" matched the published results (quantitative numbers were not published). The reactivity was computed to be 3-4% higher than the published, reference value. It was also mentioned that results from this problem were published by a third, independent researcher, using DIF3D. The DIF3D calculation under predicted (with respect to the reference) the reactivity by 3-4%. Thus it appears that this benchmark problem is difficult to solve, and three different computer codes have given three different estimates of the reactivity. It was proposed that the different estimations of the reactivity were caused by the differences in the spatial models. The QUARTZ model used the CMFD method with 6 nodes per hexagon. The TRIMHX model used the CMFD method with 1 node per hexagon, and the DIF3D model used a quadratic method with 1 node per hexagon. The triangular quadratic method could not be run for this problem because of limitations in the computer code, QUARTZ. QUARTZ, when running in quadratic mode, models one boundary condition for the periphery of the reactor per axial plane.

The remaining problems, which were run as general consistency checks, demonstrated that the thermal hydraulic feedback model was capable of giving physically expected results.

Chapter 7

Conclusions and Recommendations for Future Work

7.1 Summary of Results

A quadratic nodal method has been developed in triangular-z geometry for static and transient problems. Solution methods have been presented, and several test problems have been used to demonstrate the nature of the quadratic method. A summary of the test results is given as follows:

- The quadratic equations were shown to reduce to the Coarse Mesh Finite Difference (CMFD) equations in the limit of small mesh size. Thus, if the assumptions made in the derivation are poor, the problems disappear in the limit of small mesh size.

- Comparing the node-averaged fluxes for the quadratic and the CMFD methods to reference solutions showed that the quadratic method produced much better results at coarse mesh for all models tested. At smaller mesh size, however, the quadratic method exhibited inconsistent error reduction with shrinking mesh size when an albedo boundary condition was used. This behavior was not seen when either a zero flux or zero current boundary condition was used. It is suspected that the problem is caused by incorrectly treated boundary condition parameters, which only appear in albedo boundary conditions.

- Comparing the eigenvalue for both methods to reference solutions showed that the quadratic method gave poor results at large mesh size when either an albedo or zero flux boundary condition was used. When a zero current, heterogeneous model was used, the quadratic method gave excellent results. It is suspected that the problem is caused by incorrect leakages in nodes on the boundary of the reactor. It was mentioned that the quadratic representation of the transverse-averaged flux may be too rigid to model accurately the current at the points and surfaces of the triangular nodes. Previous work in Cartesian geometry [G-1] showed that quadratic polynomials were capable of accurately predicting node-averaged fluxes, but sometimes incapable of accurately predicting the currents at the surfaces. The problem is more severe for the triangular quadratic method because the polynomials are forced to match value and derivative to other polynomials at the points of the triangular nodes. It was mentioned that this rigidity may lead to problems at any location where the flux changes drastically.

- A numerical instability was encountered for extreme (very small mesh) cases. The cause of the instability was found to be the loss of diagonal dominance due to the values of the discontinuity factor ratios.

- The transient solution to a simple, zero current test problem using the quasi-static method was shown to match the analytic solution.

- A large, heterogeneous, heavy water benchmark problem was used to test the space-time transient methods. The reactivity computed using the quasi-static method was shown to be within 3-4% of published results. Because the amount of reactivity was very high (i.e. close to prompt critical), the 3-4% difference led to large differences in power as a function of time. It was mentioned that the shape distortion caused by the non-homogeneous perturbation qualitatively matched published results. It was proposed that the difference between the QUARTZ, TRIMHX, and DIF3D estimates of the reactivity could be attributed to the different spatial models used.

- Three thermal hydraulic test problems were used to demonstrate the adequacy of the thermal feedback model. These tests showed that transients could be initiated by changing the inlet thermal hydraulic conditions, and that the thermal feedback to other types of transients behaved as expected.

7.2 Conclusions

In conclusion, the quadratic nodal method in triangular geometry has been shown to give much better estimations of the nodal fluxes than the CMFD method at the same mesh size. The eigenvalue estimations, however, were shown to be poorer than the CMFD values in certain cases. It is suspected that the cause of these eigenvalue difficulties is the quadratic representation of the transverse-averaged flux. The results consistently show that the quadratic method can accurately predict the node-averaged fluxes, but can have difficulty predicting the surface-averaged currents. The quadratic polynomial simply does not have enough flexibility to match the surface-averaged current at the face and at the point. The triangular quadratic method is more 'rigid' than the Cartesian quadratic method because the polynomial is forced to match the value and derivative at the point of the triangle to the polynomial in the node beyond without the benefit of discontinuity factors. The triangular quadratic method may therefore give poor leakage estimations for nodes at material interfaces or reactor boundaries. These leakage errors may then adversely affect the eigenvalue.

One topic which has not been discussed in detail is the effect of the "heterogeneity factors." Recall that these are the discontinuity factors which appear in the quadratic current equations, and correct for heterogeneity, Fick's law, and the spatial approximation. If the "reference" calculation to get the heterogeneity factors was performed using a patch of nodes (which is likely because reference calculations typically model assemblies, and there will always be two or more triangles per assembly), then the heterogeneity factors should somewhat correct for the leakage difficulties. If this is the case, then the quadratic method would give accurate node-average fluxes and the heterogeneity factors would improve the eigenvalue solution.

In select cases, the quadratic method has been shown to give significant benefit over the CMFD method. The essential question of whether or not the quadratic method is useful must be answered for each specific reactor type. It appears that the method has its greatest difficulty near reactor boundaries, and it is conjectured that it also suffers at interfaces across which there are severe material discontinuities. To the extent that these severe conditions exist, the quadratic method may or may not be very useful.

In the current implementation, numerical instabilities were encountered when very small mesh sizes were used. In typical reactor applications, the mesh sizes will be much larger, and therefore the instabilities should not be of major concern.

The transient methods were tested using a simple analytic problem, three problems with thermal feedback, and a detailed heterogeneous problem. The results from these tests showed, in a preliminary way, that the transient methods were developed and implemented correctly. However, there was a difficulty in obtaining good benchmark problems for triangular geometry. Further testing is recommended with a wider variety of problems.

7.3 Recommendations for Future Research

During the course of this work, the nature of the quadratic nodal equations for triangular geometry was investigated. Many questions remain, however. It is recommended, for future research, that the following areas be investigated:

- It has been mentioned several times that the quadratic polynomial may not have enough flexibility to accurately predict the currents at both the points of the triangular nodes and the faces. It is recommended that higher order polynomials be investigated. Methods as high as fourth order have been developed successfully in Cartesian geometry [G-1]. The development of higher order methods, however, might pose a considerable challenge because of the unusual geometry.

- It is strongly recommended that the quadratic method be tested thoroughly for other reactor systems. Results have indicated that the quadratic method can give significant benefit over the CMFD method for certain reactor configurations. It is necessary to develop a knowledge and understanding of which reactor systems the method is suited for, and why.

- An in-depth investigation of the numerical stability of the quadratic method is recommended. The instabilities were only found for extreme cases, but a full understanding of the causes and remedies is needed. An important question is whether or not the triangular nodal method is particularly susceptible to numerical instability, or if this type of instability is generic to all nodal methods.

- The non-linear iteration scheme currently uses surface-averaged currents from the last iteration on the right hand side of eqn. (3.50). Previous implementations of the non-linear iteration scheme (in other geometries) have found that attempting to solve these equations simultaneously for the currents is unstable. In the triangular nodal method, however, the extended coupling may allow the stable, simultaneous solution of the currents. Additionally, extended coupling might be considered for the axial dimension. Extended coupling is not required for Cartesian geometry, but it might allow the currents to be solved for simultaneously. The overall goal is to reduce the run time by converging the quadratic currents as quickly as possible, which might be accomplished by simultaneous solution.

- A very simple thermal hydraulic feedback model was incorporated to test the general feedback response of the static and transient methods. A more detailed, elaborate model could be employed for application to a specific reactor system.

References

[A-1] AIAA Document, "A Historical Collection of Papers on Nuclear Thermal Propulsion," AIAA Nuclear Thermal Propulsion Technical Committee, CP925, Washington, DC (1992).

[B-1] M.E. Bvers, "A Transient Nodal Method for Reactor Models in R-Z Geometry," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (August 1992).

[B-2] M.R. Buckner and J.W. Stewart, "Multidimensional Space-Time Nuclear-Reactor Kinetics Studies-Part I: Theoretical," *Nucl. Sci. Eng.*, **59**, 289-297 (1976).

[B-3] J.F. Briesmeister, et al., "MCNP-A General Monte Carlo Code for Neutron and Photon Transport," LA-7396-M, Los Alamos National Laboratory (1986).

[D-1] A.F.V. Dias, "Systematic Derivation, From 3-D Nodal Equations, of Simpler Models for Describing Reactor Transients," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (May 1987).

[F-1] T.B. Fowler, D.R. Vondy, and G.W. Cunningham, "Nuclear Reactor Core Analysis Code: CITATION," ORNL-TM-2496, Oak Ridge National Laboratory (1971).

[G-1] J.C. Gehin, "A Quasi-Static Polynomial Nodal Method for Nuclear Reactor Analysis," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (September 1992).

[G-2] J.C. Gehin, "A Nodal Method for the Solution of the Static, Few-Group Diffusion Equations in Hexagonal Geometry," S.M. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (February 1990).

[H-1] A.F. Henry, *Nuclear Reactor Analysis*, MIT Press, Cambridge, MA (1975).

[H-2] L.A. Hageman and D.M. Young, *Applied Iterative Methods*, Academic Press, New York (1981).

[H-3] A.F. Henry, Personal Communication,(1993).

[K-1] P.W. Kao, "Application of Supernodal Methods to Transient Analysis," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (August 1988).

[M-1] L.S. Mason, et al., "Nuclear Electric Propulsion for Mars Cargo Missions," *Trans. 7th Symposium on Space Nuclear Power Systems*, Albuquerque, NM, 8-11 January 1990.

[N-1] S. Nakamura, *Computational Methods in Engineering and Science with Applications to Fluid Dynamics and Nuclear Systems*, John Wiley & Sons, New York (1977).

[O-1] K.O. Ott, D.A. Meneley, "Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics," *Nucl. Sci. Eng.*, **36**, 402 (1969).

[P-1] P.B. Parks, et. al.,"Multidimensional Space-Time Nuclear-Reactor Kinetics Studies-Part II: Experimental," *Nucl. Sci. Eng.*, **59**, 298 (1976).

[P-2] W.H. Press, et al., *Numerical Recipes in FORTRAN*, Cambridge University Press, NY (1992).

[P-3] J. Powell, et al., "Particle Bed Reactor Multimegawatt Concepts," Brookhaven National Laboratory, BNL-39495, (March 1987).

[R-1] E. Redmond II, "Monte Carlo Modeling of the MITR-II," Report from Special Term Project 22.39, MIT, (December 1990).

[S-1] T.M. Sutton, "Wielandt Iteration as Applied to the Nodal Expansion Method," *Nucl. Sci. Eng.*, **98**, 169 (1988).

[S-2] K.S. Smith, "An Analytic Nodal Method for Solving the Two-Group, Multidimensional, Static and Transient Neutron Diffusion Equations," S.M. and N.E. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (March 1979).

[S-3] Y.A. Shatilla, "A Simple Quadratic Nodal Model for Hexagonal Geometry," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (September 1992).

[S-4] K.S. Smith, "Nodal Method Storage Reduction by Nonlinear Iteration," *Trans. Am. Nucl. Soc.*, **44**, 265 (1983).

[T-1] T.A. Taiwo and H.S. Khalil, "The DIF3D Nodal Kinetics Capability in Hex-Z Geometry -- Formulation and Preliminary Tests," *Proc. Advances in Mathematics, Computations, and Reactor Physics*, Pittsburgh, PA (April 1991).

[V-1] A.V. Vota, N.J. Curlee, Jr., and A.F. Henry, "WIGL3-A Program for the Steady-State and Transient Solution of the One-Dimensional, Two-Group, Space-Time Diffusion Equations Accounting for Temperature, Xenon, and Control Feedback," WAPD-TM-788 (February 1969).

[Z-1] M.L. Zerkle, "Development of a Polynomial Nodal Method with Flux and Current Discontinuity Factors," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (June 1992).

Appendix A

Special Boundary Condition Considerations

In Chapter 3, a difficulty was encountered when a boundary condition was applied to the point of a triangular node. It was mentioned that the boundary condition parameters which appear in eqn. (3.55) are not necessarily equal to the 'traditional' boundary condition parameters in eqn. (3.18). The computer code QUARTZ currently assumes that the boundary condition parameters which appear in eqn. (3.55) are equal to the parameters input by the user. This appendix will discuss the relationship between the two sets of boundary condition parameters, and evaluate the assumptions made in the implementation.

A general, albedo boundary condition will be assumed for the work in this section, so that both alpha and beta are non-zero. Traditionally, the parameters 'alpha' and 'beta' are input by the user to describe the condition at the outer surface of the reactor. This relationship is given by:

$$\frac{\beta}{\alpha} = \frac{\overline{J} \cdot \underline{n}_{reactor}}{\overline{\phi}} \qquad \text{(A.1)}$$

where $\underline{n}_{reactor}$ is an outward-directed normal to the reactor surface. In Chapter 3, an equation for the boundary condition at a point of a triangular node (which lies on the boundary of the reactor) was introduced as eqn. (3.55). This equation is similar in form to eqn. (A.1), but although both are applied at the outer surface of the reactor, eqn. (3.55) uses the component of the net current density which is normal to a small "ε surface" (see Chapter 3). Dropping the group and node notation, eqn. (3.55) is rewritten as:

$$\left(\frac{\beta}{\alpha}\right)_{point} = \frac{\overline{J} \cdot \underline{n}_{\varepsilon\ surface}}{\overline{\phi}} \qquad \text{(A.2)}$$

For a particular node, if $\underline{n}_{\varepsilon\ surface}$ does not point in the same direction as $\underline{n}_{reactor}$, then:

$$\left(\frac{\beta}{\alpha}\right)_{point} \neq \frac{\beta}{\alpha} \tag{A.3}$$

The goal is to allow the user to specify one alpha and one beta (for the whole reactor, or for a whole side of the reactor) which describes the boundary condition according to eqn. (A.1). The appropriate boundary condition parameters at any given triangular point can be determined by:

$$\left(\frac{\beta}{\alpha}\right)_{point} = \gamma\frac{\beta}{\alpha} \tag{A.4}$$

where γ is a constant which is determined for each triangular point which lies on the reactor surface. For a regular pattern of triangular nodes, there are few unique values of γ. In general, the γ constant depends on the orientation of the node to the reactor surface, and the nature of the adjacent nodes. A systematic way of obtaining this constant is currently not available. Under many conditions, however, this constant is unity. In order to evaluate the approximation that this constant is unity for all points on the reactor surface, it is useful to examine specific cases. Figure A.1 shows the point "A" on the reactor surface.

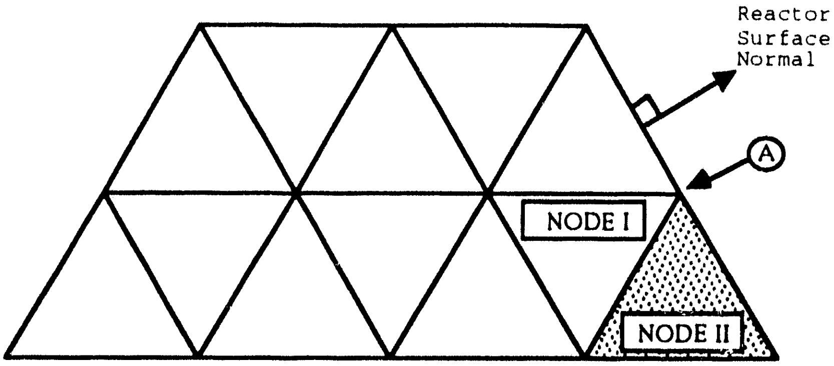


Figure A.1 Two Triangular Nodes Which Share the Same Point on the Reactor Surface.

We shall examine the point "A" for node II. Figure A.2 shows the tip of node II. The small area of interest is bounded by the "ε surface" below, and the two sides of the triangle near the tip. In the figure, the surface currents are shown on each of the three surfaces (the subscripts do not carry any special meaning). As the ε surface approaches point A, the reaction rates within this small area become negligible, and the balance of neutrons is then largely determined by the leakage in or out of each face. The net current into this area should equal the net current out. That is:

$$J_1 = J_2 + J_3 \tag{A.5}$$

As the ε surfaces for both node I and node II approach point A, the leakage between these two regions becomes small. In the limit, $J_3 \approx 0$. The current at the ε surface is then equal to the current on the reactor surface. Assuming that the **boundary** condition parameters at the point of the triangle (i.e. ε surface) are equal to the boundary condition parameters at the surface is equivalent to assuming that the current at the ε surface is equal to the current at the surface. Thus, the boundary condition approximation is valid insofar as $J_3 = 0$.

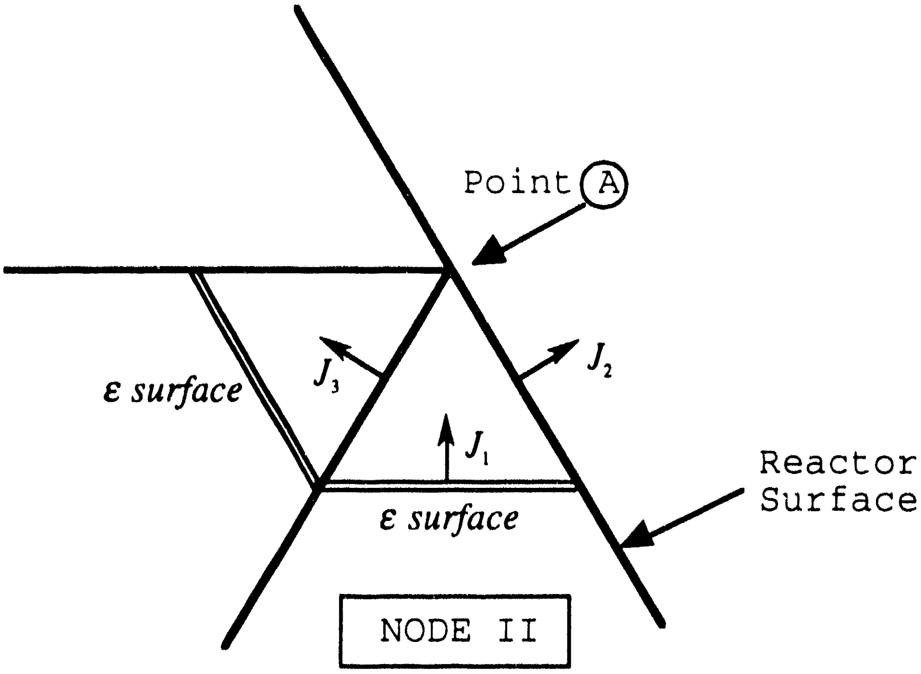


Figure A.2. Surface-Averaged Currents at the Tip of Node II.

Another situation which must be considered is when the point of a triangular node is at the corner of the reactor. Figure A.3 shows this situation for node III. The point of interest is labeled 'B.'

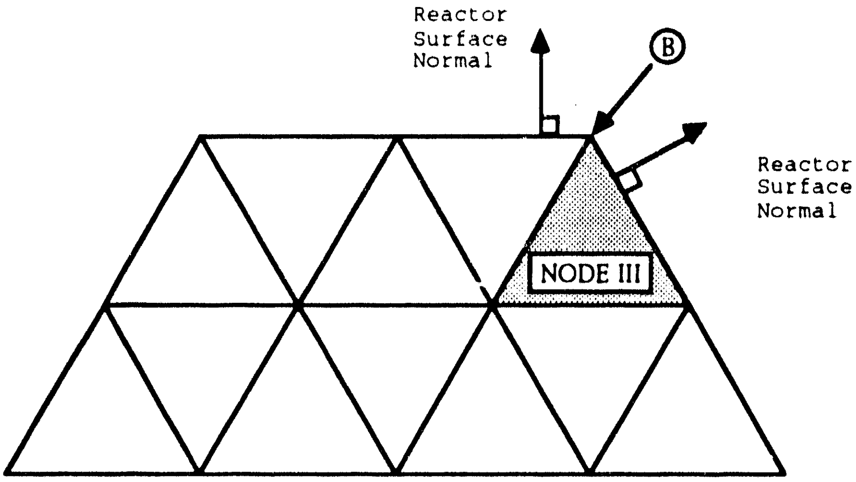


Figure A.3 A Point on the Reactor Surface which Occurs at the Corner of the Reactor.

For node III of Fig. A.3, the tip of the triangle is shown in Fig. A.4.

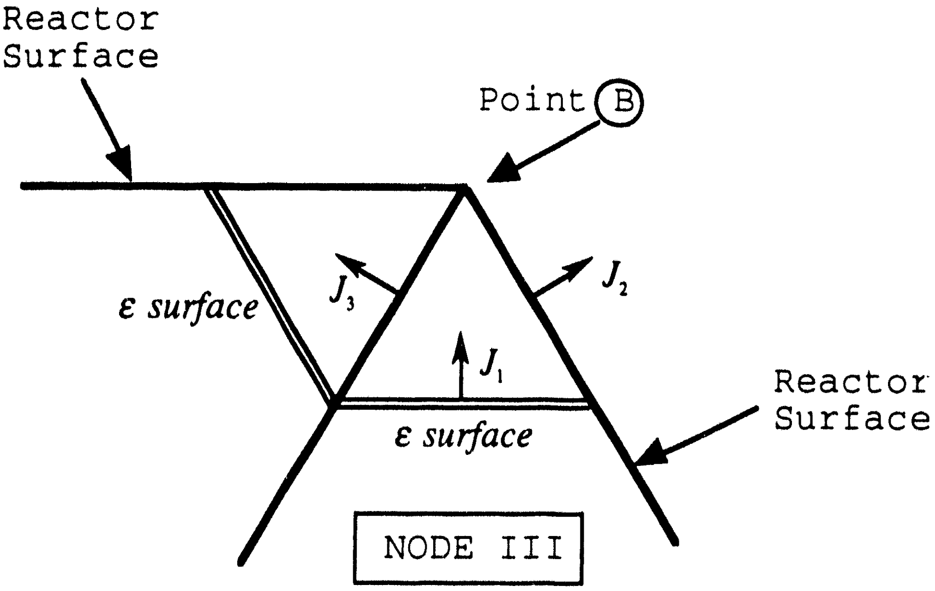


Figure A.4. Surface-Averaged Currents at the Tip of Node III.

The same arguments made in the previous case hold here. The current at the ε surface will equal the current at the reactor surface if $J_3 = 0$.

A very unusual case is the point of a single triangular node. This situation is shown in Fig. A.5. In this case, the current at the ε surface is equal to the sum of the other two currents. If the external boundary condition is the same for both sides of the tip, the γ constant is equal to 2.0.

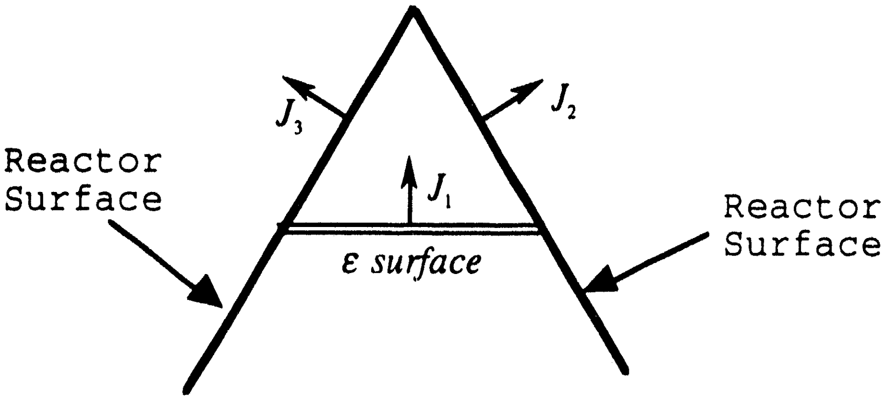


Figure A.5. A View of the Tip of a Single Triangular Node.

Appendix B

The WIGL Model

The WIGL thermal hydraulics model [V-1] is a lumped heat capacity model which gives node-averaged fuel and coolant temperatures for static and transient applications. The WIGL model assumes that no boiling occurs within the nodes. The transient form of the WIGL equation for the node-averaged fuel temperature in node (i,j,k), $\overline{T}_f^{(i,j,k)}$, is:

$$\rho_f V_f^{(i,j,k)} C_f \frac{d\overline{T}_f^{(i,j,k)}}{dt} = (1-r)(q''')^{(i,j,k)} V_f^{(i,j,k)}$$

$$-V_c^{(i,j,k)} \left[\frac{1}{A_H U} + \frac{1}{A_H h_o} \left(\frac{W}{W_o} \right)^{0.8} \right]^{-1} \left(\overline{T}_f^{(i,j,k)} - \overline{T}_c^{(i,j,k)} \right) \qquad \text{(B.1)}$$

and for the node-averaged coolant temperature, $\overline{T}_c^{(i,j,k)}$:

$$V_c^{(i,j,k)} \left(\frac{\partial \rho_c H}{\partial \overline{T}_c} \right) \frac{d\overline{T}_c^{(i,j,k)}}{dt} =$$

$$V_c^{(i,j,k)} \left[\frac{1}{A_H U} + \frac{1}{A_H h_o} \left(\frac{W}{W_o} \right)^{0.8} \right]^{-1} \left(\overline{T}_f^{(i,j,k)} - \overline{T}_c^{(i,j,k)} \right) \qquad \text{(B.2)}$$

$$+2 W_r^{(i,j,k)} C_c \left(T_b^{(i,j,k)} - \overline{T}_c^{(i,j,k)} \right) + r(q''')^{(i,j,k)} V_f^{(i,j,k)}$$

where the coolant inlet temperature for node (i,j,k) is $T_b^{(i,j,k)}$ (b is for 'bottom'), and for any k greater than 1, is given in terms of the node-averaged coolant temperature and the inlet coolant temperature of node (i,j,k-1):

$$T_b^{(i,j,k)} = 2\overline{T}_c^{(i,j,k-1)} - T_b^{(i,j,k-1)} \qquad k > 1 \qquad \text{(B.3)}$$

The notation in (B.1), (B.2), and (B.3) is used in order to match previous implementations [S-2,B-1,G-1]. The quantities used in the WIGL model are defined as:

COMPUTED QUANTITIES:

$\overline{T}_f^{(i,j,k)}$	= node-averaged fuel temperature for node (i,j,k) (K)
$\overline{T}_c^{(i,j,k)}$	= node-averaged coolant temperature for node (i,j,k) (K)
$T_b^{(i,j,k)}$	= node coolant inlet temperature for node (i,j,k) (if k > 1) (K)

USER INPUT QUANTITIES:

T_{bo}	= core inlet coolant temperature at time t (K)
ρ_f	= fuel density (g/cm^3)
C_f	= specific heat of the fuel (erg/g K)
C_c	= specific heat of the coolant (erg/g K)
r	= fraction of fission power deposited directly into the coolant
$V_f^{(i,j,k)}$	= volume of fuel in node (i,j,k) (cm^3)
$V_c^{(i,j,k)}$	= volume of coolant in node (i,j,k) (cm^3)
A_H	= total heat transfer area / coolant volume in node (i,j,k) (cm^{-1})
h_o	= convective heat transfer coefficient at the initial core flow rate (erg/s cm^2 K)
U	= conductivity/conduction lengths of the fuel, gap, and cladding
W_o	= initial core (i.e. total) mass flow rate (g/s)
W	= core mass flow rate at time t (g/s)
$W_r^{(i,j,k)}$	= mass flow rate in node (i,j,k) at time t (g/s)
$\left(\dfrac{\partial \rho_c H}{\partial \overline{T}_c}\right)$	= the energy required to raise the temperature of a unit volume of coolant one degree K (erg/cm^3 K)

SOLUTION-GENERATED QUANTITIES:

$(q''')^{(i,j,k)}$	= volumetric energy generation rate in node (i,j,k) at time t (erg/cm^3)

The volumetric energy generation rate in each node is computed using the nodal power densities at each time step. If a composition change in the reactor initiates a transient, the nodal power densities change as a function of time, and new temperatures are computed using the WIGL equations. The temperatures affect the solution by adjusting the cross sections.

Two other mechanisms for initiating a transient have been implemented. The core inlet temperature can be adjusted as a function of time, and the core flow rate.

The thermal hydraulic equations are solved using a fully implicit integration over time, identical to the implementation by Gehin [G-1].

The static WIGL equations are used to compute the steady state temperature distributions during static problems. The static WIGL equations can be obtained from (B.1) and (B.2) by setting the time derivatives to zero, and using the steady state values of the inlet coolant temperature and core mass flow rate.

Appendix C

Model Descriptions

In this appendix, the descriptions of several models which were used to verify the code QUARTZ are given.

C.1 Static Analytic Test Problem

The first of the analytic tests was a zero current problem with the following 2 group cross sections (taken from Byers [B-1]):

D	Chi	Fission	Nu Fission	Sigma
1.49750E+0	1.00000E+0	1.38519E-4	3.46297E-4	3.32021E-3
1.15689E+0	0.0	2.35888E-3	5.89720E-3	3.37634E-3

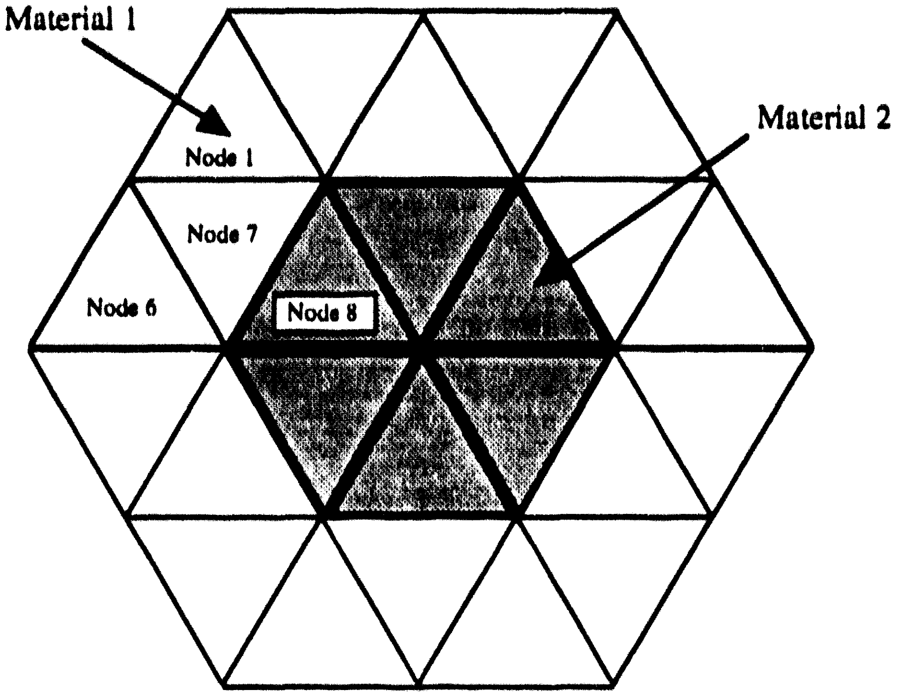
Scattering
$\Sigma_{1\rightarrow 2} = 2.37634E\text{-}3$
$\Sigma_{2\rightarrow 1} = 0.0$

The second of the static analytic tests involved a bare, homogeneous slab (the computer code QUARTZ has the option of solving problems in 1-D Cartesian geometry). The thickness of the slab was taken to be 250 cm. This gives a buckling of:

$$\mathrm{B}^2 = \left(\frac{\pi}{\mathrm{H}}\right)^2 = \frac{\pi^2}{(250\,cm)^2} = 1.57914\mathrm{E}-4\ (\mathrm{cm}^{-2})$$

C.2 MITHEX Infinite Cell Comparison Problem

The first problem used to compare QUARTZ to MITHEX was a heterogeneous, 24 node cell with a zero current boundary condition. The geometrical layout of the model is given as:



where the cross sections for materials 1 and 2 were taken from Ref. [G-2] (note that the cross sections differ slightly from published values in [G-2]):

MATERIAL 1

D	Chi	Fission	Nu Fission	Sigma
1.29000E+0	1.00000E+0	0.0	0.0	8.02913E-3
8.83000E-1	0.0	0.0	0.0	7.96268E-5

Scattering
$\Sigma_{1\to 2}$ = 8.00000E-3
$\Sigma_{2\to 1}$ = 0.0

MATERIAL 2

D	Chi	Fission	Nu Fission	Sigma
1.38504E+0	1.00000E+0	1.16741E-3	2.85416E-3	8.86505E-3
8.98526E-1	0.0	1.52190E-2	3.70308E-2	2.18824E-2

Scattering

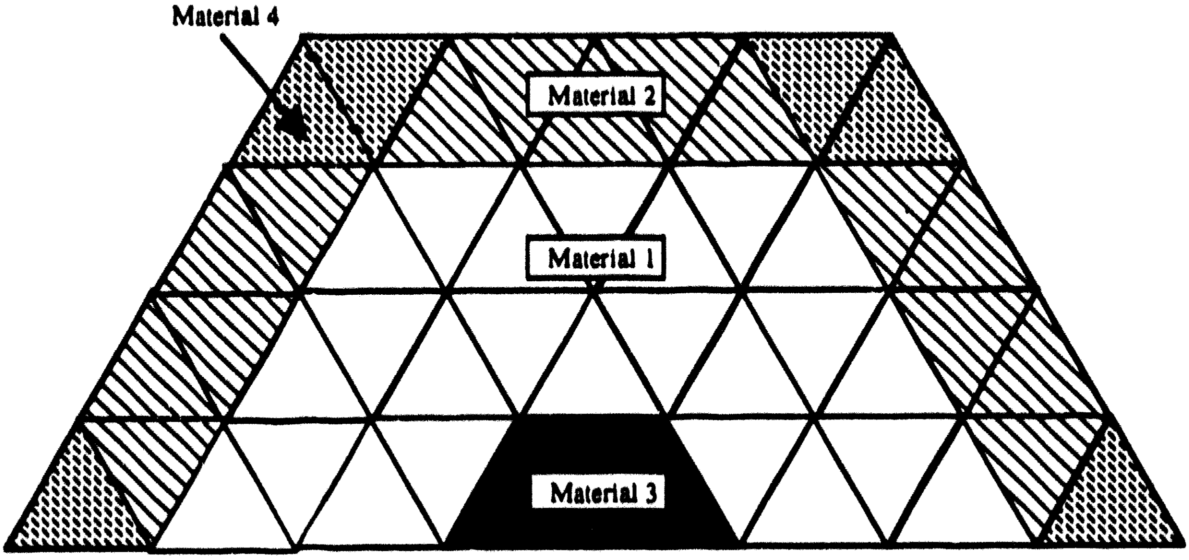
$\Sigma_{1\rightarrow 2}$ = 6.44203E-3
$\Sigma_{2\rightarrow 1}$ = 4.39126E-5

Other Data:

LENGTH OF NODE SIDE : 10.2650 cm
BOUNDARY CONDITION: Zero Current

C.3 MITHEX Core-Sized Comparison Problem

The second problem used to compare QUARTZ to MITHEX was a 96 node, heterogeneous, 2 group problem. The geometry is given as follows, where only the top half of the symmetric layout is shown:



The cross sections are defined as follows:

MATERIAL 1

D	Chi	Fission	Nu Fission	Sigma
1.38504E+0	1.00000E+0	1.16741E-3	2.85416E-3	8.86505E-3
8.98526E-1	0.0	1.52190E-2	3.70308E-2	2.18824E-2

Scattering
$\Sigma_{1\rightarrow 2}$ = 6.44203E-3
$\Sigma_{2\rightarrow 1}$ = 4.39126E-5

MATERIAL 2

D	Chi	Fission	Nu Fission	Sigma
1.31919E+0	0.0	0.0	0.0	1.20594E-2
9.01342E-1	0.0	0.0	0.0	5.63096E-3

Scattering
$\Sigma_{1\to2}$ = 1.16912E-2
$\Sigma_{2\to1}$ = 3.44567E-5

MATERIAL 3

D	Chi	Fission	Nu Fission	Sigma
1.17097E+0	1.00000E+0	5.69906E-4	1.50390E-3	1.29105E-2
8.80244E-1	0.0	3.62256E-3	8.81442E-3	1.36547E-2

Scattering
$\Sigma_{1\to2}$ = 7.67664E-3
$\Sigma_{2\to1}$ = 3.57473E-5

MATERIAL 4

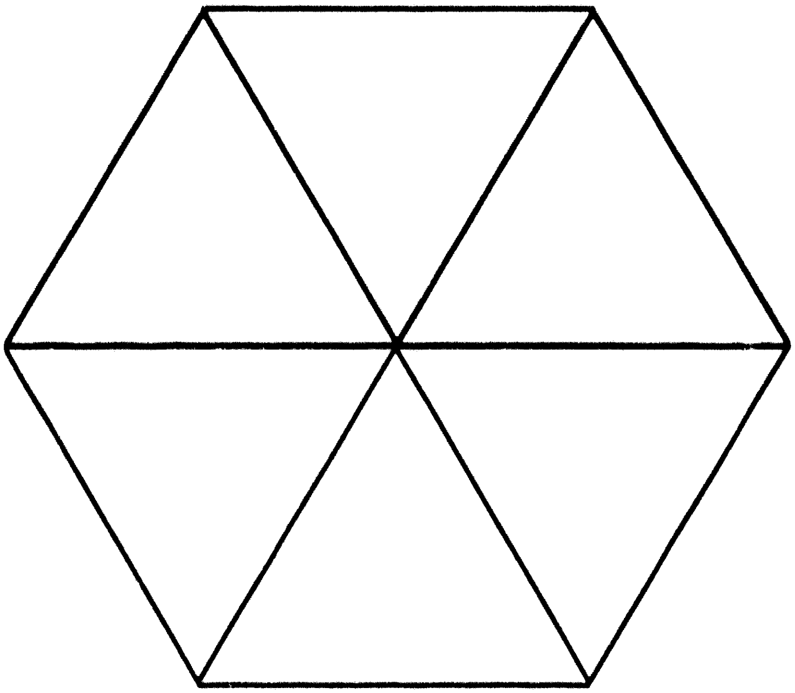
D	Chi	Fission	Nu Fission	Sigma
1.29000E+0	0.0	0.0	0.0	8.02913E-3
8.83000E-1	0.0	0.0	0.0	7.96268E-5

Scattering
$\Sigma_{1\to2}$ = 8.00000E-3
$\Sigma_{2\to1}$ = 0.0

Other Data:
LENGTH OF NODE SIDE : 17.7805 cm
BOUNDARY CONDITION: Albedo (No Net Returning Current)

C.4 Reduction of Quadratic to CMFD Limit Problem

This model was used to test the quadratic equations at very small mesh sizes, to see if the quadratic equations reduced to the CMFD equations in the limit. The model is a 2-D, 6 node homogeneous problem arranged in an overall hexagonal pattern:



The cross sections are given by:

D	Chi	Fission	Nu Fission	Sigma
1.42391E+0	1.00000E+0	1.60000E-2	4.00000E-2	5.00000E-2
3.56310E-1	0.0	3.20000E-2	8.00000E-2	8.00000E-2

Scattering
$\Sigma_{1\rightarrow 2}$ = 1.00000E-2
$\Sigma_{2\rightarrow 1}$ = 0.0

Other Data:

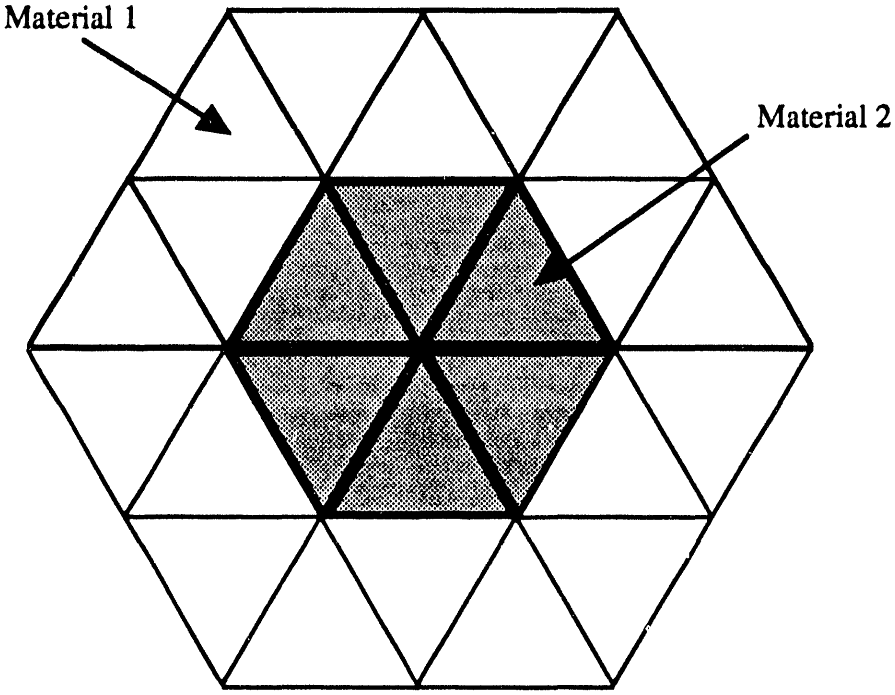
LENGTH OF NODE SIDE :

Tests:
5.00 cm
2.00 cm
1.00 cm
0.80 cm
0.60 cm
0.40 cm
0.20 cm
0.10 cm
0.05 cm
0.01 cm
0.005 cm
0.001 cm
5.0E-4 cm
1.0E-4 cm
5.0E-5 cm
1.0E-5 cm

BOUNDARY CONDITION: Albedo (No Net Returning Current)
CONVERGENCE: 10^{-12} on eigenvalue, double precision
1 Outer Iteration Between d.f.r. Updates
THERMAL FEEDBACK: None

C.5 Quadratic Test Problems

Two models were used for the quadratic tests. The first was a 2-D, homogeneous model with 24 nodes. The second was a heterogeneous version of the same model. Case I and Case II used the homogeneous model with albedo and zero flux boundary conditions, respectively. Case III used the heterogeneous model. The heterogeneous model is represented by:



The homogeneous model used type 1 cross sections everywhere. The heterogeneous model used both type 1 and type 2, in the patterns shown in the figure. The two-group cross sections are:

MATERIAL 1

D	Chi	Fission	Nu Fission	Sigma
1.49750E+0	1.00000E+0	1.38519E-4	3.46297E-4	3.32021E-3
1.15689E+0	0.0	2.35888E-3	5.89720E-3	3.37634E-3

Scattering
$\Sigma_{1\to 2} = 2.37634E\text{-}3$
$\Sigma_{2\to 1} = 0.0$

MATERIAL 2

D	Chi	Fission	Nu Fission	Sigma
1.27288E+0	1.00000F+0	1.24667E-4	3.11667E-4	4.15026E-3
9.83360E-1	0.0	2.12299E-3	5.30748E-3	4.22043E-3

Scattering
$\Sigma_{1\to 2} = 2.47634E\text{-}3$
$\Sigma_{2\to 1} = 0.0$

Other Data:

LENGTH OF NODE SIDE :

1 Node /Assembly	67.7805 cm
4 Nodes/Assembly	33.8903 cm
16 Nodes/Assembly	16.9451 cm
64 Nodes/Assembly	8.4726 cm

BOUNDARY CONDITION:

CASE I	Albedo (No Net Returning Current)
CASE II	Zero Flux
CASE III	Zero Current

CONVERGENCE:

Eigenvalue:	10^{-6}
Static Power:	10^{-4}

C.6 Transient Analytic Test Problem

The transient analytic test problem consists of a single node with a zero current boundary condition on all sides. The details were obtained from Ref. [B-1], although the neutron speeds were not given, and typical values were assumed. The problem is two group, with one delayed neutron group.

The cross sections are given as:

D	Chi	Fission	Nu Fission	Sigma
1.42391E+0	1.00000E+0	1.60000E-2	4.00000E-2	5.00000E-2
3.56310E-1	0.0	3.20000E-2	8.00000E-2	8.00000E-2

Scattering
$\Sigma_{1\rightarrow 2}$ = 1.00000E-2
$\Sigma_{2\rightarrow 1}$ = 0.0

Other Data:

LENGTH OF NODE SIDE : 1.35561E+2 cm
BOUNDARY CONDITION: Zero Current
CONVERGENCE:
Eigenvalue: 10^{-6}
Static Power: 10^{-4}
Flux at Every Time Step: 10^{-7}

METHOD: CMFD
THERMAL FEEDBACK: None

Transient Data:

Precursor Group	Precursor Half Life	Delayed Neutron Fraction
1	0.08	0.0075

Energy Group	Delayed Spectrum	Neutron Speed
1	1.0000E+0	1.2500E+7
2	0.0	2.5000E+5

Time Step Size: 0.0 < t < 0.2 sec deltat = 5E-5 sec
0.2 < t < 0.9 sec deltat = 5E-4 sec

Cross Section Adjustments:

@ TIME = 0.0 seconds

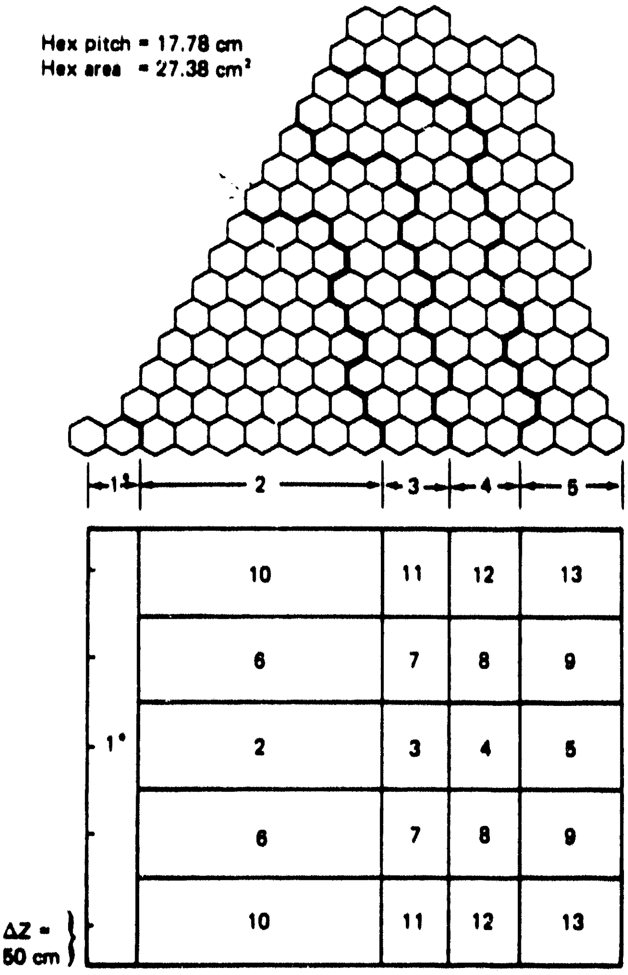
TYPE = Absolute Change in Cross Section

D	Chi	Fission	Nu Fission	Sigma
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-1.00000E-3

Scattering
$\Sigma_{1\rightarrow2} = 0.0$
$\Sigma_{2\rightarrow1} = 0.0$

C.7 Buckner-Stewart Benchmark Problem

The data for this benchmark problem was obtained from Ref. [B-2]. The problem is a 3-D, heterogeneous, heavy water reactor. The 60 degree sector is given as follows:



The cross sections are given as:

D	Chi	Fission	Nu Fission	Sigma
1.37000E+0	1.00000E+0	8.16650E-4	1.98446E-3	1.30000E-2
8.40000E-1	0.0	6.12490E-3	1.48835E-2	**

Scattering

$\Sigma_{1\rightarrow2}$ = 1.00000E-2
$\Sigma_{2\rightarrow1}$ = 0.0

** The group 2 sigma differed for each region:

	SIGMA		SIGMA
Region 1	1.355E-2	**Region 8**	1.325E-2
Region 2	1.355E-2	**Region 9**	1.320E-2
Region 3	1.350E-2	**Region 10**	1.315E-2
Region 4	1.345E-2	**Region 11**	1.310E-2
Region 5	1.340E-2	**Region 12**	1.305E-2
Region 6	1.335E-2	**Region 13**	1.300E-2
Region 7	1.330E-2		

Other Data:

LENGTH OF NODE SIDE : 17.78 cm
BOUNDARY CONDITION: Zero Flux
CONVERGENCE:

Eigenvalue:	10^{-6}
Static Power:	10^{-4}
Flux at Every Time Step:	10^{-6}
Point Kinetics Amplitude:	10^{-6}

METHOD: CMFD
THERMAL FEEDBACK: None

Transient Data:

Precursor Group	Precursor Half Life	Delayed Neutron Fraction
1	0.1	0.0065

Energy Group	Delayed Spectrum	Neutron Speed
1	1.0000E+0	1.000E+30
2	0.0	5.0000E+6

Time Step Size:	0 < t < 1 sec	deltat = 5 msec
	1 < t < 6 sec	deltat = 50 msec

Cross Section Adjustments:

```
     @ TIME = 0.0 seconds
       TYPE = Absolute Change in Cross Section
       LOCATION = Region 1
```

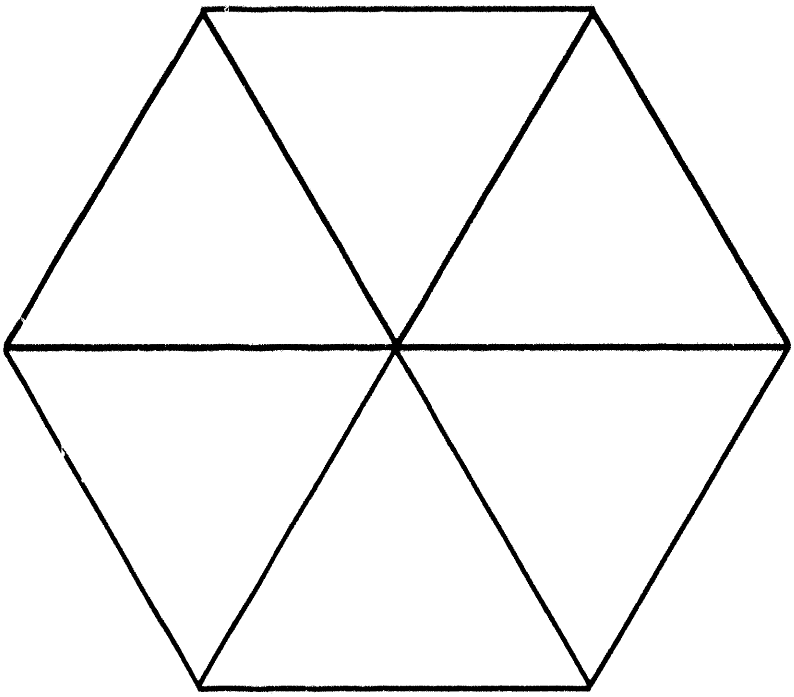
D	Chi	Fission	Nu Fission	Sigma
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-1.11377E-3

Scattering
$\Sigma_{1\rightarrow 2} = 0.0$
$\Sigma_{2\rightarrow 1} = 0.0$

C.8 Transient Thermal Hydraulic Test Problems

Three transient thermal hydraulic tests were performed on a 3-D, homogeneous, 2-group model. The first test consisted of a composition change, which was run with and without thermal feedback. The second test consisted of a step decrease in the inlet flow rate, and the third was a step decrease in the inlet temperature.

The model consists of four homogeneous axial planes, each with the 2-D pattern:



The cross sections are given at three fuel temperatures: reference, high, and low. The cross sections were obtained from Appendix B.2 of Reference [G-1]. The data is for the "LMW LWR Transient Problem" (see [G-1] for details). The reference gives cross sections at the reference fuel temperature and cross section derivatives at 533 K. In order to use these data in QUARTZ, which does not accept cross section derivatives (recall that a quadratic interpolation model is used for feedback), the derivatives were used to obtain cross sections at 583 K and 483 K. These are:

AT REFERENCE FUEL TEMPERATURE: 533 K

D	Chi	Fission	Nu Fission	Sigma
1.42391E+0	1.00000E+0	2.59108E-3	6.47769E-3	2.79576E-2
3.56310E-1	0.0	4.50931E-2	1.12733E-1	8.76622E-2

Scattering

$\Sigma_{1\to2}$ = 1.75555E-2
$\Sigma_{2\to1}$ = 0.0

AT HIGH FUEL TEMPERATURE: 583 K

D	Chi	Fission	Nu Fission	Sigma
1.43207E+0	1.00000E+0	2.59108E-3	6.47769E-3	2.80326E-2
3.64754E-1	0.0	4.46781E-2	1.11724E-1	8.68372E-2

Scattering

$\Sigma_{1\to2}$ = 1.74805E-2
$\Sigma_{2\to1}$ = 0.0

AT LOW FUEL TEMPERATURE: 483 K

D	Chi	Fission	Nu Fission	Sigma
1.41585E+0	1.00000E+0	2.59108E-3	6.47769E-3	2.78826E-2
3.48241E-1	0.0	4.55081E-2	1.13741E-1	8.84872E-2

Scattering

$\Sigma_{1\to2}$ = 1.76305E-2
$\Sigma_{2\to1}$ = 0.0

Other Data:

LENGTH OF NODE SIDE : 1.35561E+2 cm
BOUNDARY CONDITION: Albedo (No Net Returning Current)
CONVERGENCE:

Eigenvalue:	10^{-6}
Static Power:	10^{-4}
Flux at Every Time Step:	10^{-6}
Point Kinetics Amplitude:	10^{-6}

METHOD: CMFD
THERMAL FEEDBACK: WIGL Model

Transient Data:

Precursor Group	Precursor Half Life	Delayed Neutron Fraction
1	0.0127	0.000247
2	0.0317	0.0013845
3	0.1150	0.001222
4	0.3110	0.0026455
5	1.40	0.000832
6	3.87	0.000169

Energy Group	Delayed Spectrum	Neutron Speed
1	1.0000E+0	1.2500E+7
2	0.0	2.5000E+5

Time Step Size: deltat = 0.1 sec

WIGL DATA (see Appendix B for Definitions):

T_{bo}	533 K
ρ_f	10.3 g/cm3
C_f	2.46E+6 (erg/g K)
C_c	5.43E+7 (erg/g K)
r	0.0
$V_c/(V_c+V_f)$	0.559
A_H	2.59 (cm^{-1})
U	2.2E+6 ergs/ cm^2 s K
h_o	2.71E+7 (erg/s cm^2 K)
W_o	=2.2E+6 (g/s)
$\left(\dfrac{\partial \rho_c H}{\partial \overline{T}_c}\right)$	1.60E+7 (erg/cm^3 K)

TEST I

Cross Section Adjustments:

TIME = perturbed at 10 seconds, reset at 30 seconds

TYPE = Absolute Change in Cross Section

LOCATION = All Regions

D	Chi	Fission	Nu Fission	Sigma
0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-2.50000E-4

Scattering
$\Sigma_{1\to 2} = 0.0$
$\Sigma_{2\to 1} = 0.0$

TEST II

Inlet flow rate reduced to 80% of original value at time = 10 seconds, returned to original value at 30 seconds.

TEST III

Inlet temperature increased from original value of 533 K to 538.07 K at time = 10 seconds, returned to original value at time = 30 seconds.

END

DATE
FILMED

1/12/94