AN ABSTRACT OF THE THESIS OF

Byung-Oh Cho for the degree of Doctor of Philosophy in Nuclear Engineering presented on December 3, 1990.

Title: Nodal Methods for Calculating Nuclear Reactor Transients, Control Rod Patterns, and Fuel Pin Powers

Abstract approved: Signature redacted for privacy.

Nodal methods which are used to calculate reactor transients, control rod patterns, and fuel pin powers are investigated. The 3-D nodal code, STORM, has been modified to perform these calculations.

Several numerical examples lead to the following conclusions: (1) By employing a "thermal leakage-to-absorption ratio" (TLAR) approximation for the spatial shape of the thermal fluxes for the 3-D Langenbuch-Maurer-Werner (LMW) and the superprompt critical transient problems, the convergence of the conventional two-group scheme is accelerated. The resulting computing time is reduced eight to fourteen times while maintaining computational accuracy. The TLAR acceleration scheme assumes that even if the thermal leakage rate and absorption rate varies significantly in a given node, the ratio of those two changes little during a small time step; (2) By employing the steepest-ascent hill climbing search with heuristic strategies, Optimum Control Rod Pattern Searcher (OCRPS) is developed for solving control rod positioning problem in BWRs. Using the method of approximation programming the objective function and the nuclear and thermal-hydraulic constraints are modified as heuristic functions that guide the search. The test calculations have demonstrated that, for the first cycle of the Edwin Hatch Unit #2 reactor,

OCRPS shows excellent performance for finding a series of optimum control rod patterns for six burnup steps during the operating cycle. Computing costs are modest even if the initial guess patterns have extremely deteriorated core characteristics; and (3) For the modified two-dimensional EPRI-9R problem, the least square second-order polynomial flux expansion method was demonstrated to be computationally about 30 times faster than a fine-mesh finite difference calculation in order to achieve comparable accuracy for pin powers. The basic assumption of this method is that the reconstructed flux can be expressed as a product of an assembly form function and a second-order polynomial function. The assembly form function is calculated by solving an assembly criticality calculation. The polynomial function is determined by minimizing the least-squares difference between the intra-nodal quantities obtained from the global two-dimensional coarse-mesh nodal solutions and those obtained from the evaluation of the polynomial function.

Nodal Methods for Calculating Nuclear Reactor Transients, Control Rod Patterns, and Fuel Pin Powers

by

Byung-Oh Cho

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed December 3, 1990

Commencement June 1991

APPROVED:

Signature redacted for privacy.

Professor of Nuclear Engineering in Charge of Major

Signature redacted for privacy.

Head of Department of Nuclear Engineering

Signature redacted for privacy.

Dean of Graduate School

Date thesis is presented December 3, 1990

Typed by Byung-Oh Cho for Byung-Oh Cho

ACKNOWLEDGEMENTS

It is a great pleasure to acknowledge the untiring support of my advisor Dr. Alan H. Robinson, who always gave me the monetary and moral support to continue work on this thesis. His help and encouragement to the organization and completion of this dissertation is gratefully appreciated. I would like to thank the members of my thesis committee, Drs. Malcolm Daniels, Kenneth Funk, Andrew Klein, and Jose Reyes, for their helpful comments and suggestions. I wish to recognize my friends Drs. Kevin Greek and Chungchan Lee for their kind assistance on several occasions.

I would like to acknowledge my grandparents, mother, brother, and sisters. No acknowledgement could be complete without mention of my wife, Yongsoon and child, Joongsuk, who patiently offered the encouragement to complete this study. Finally, I wish to dedicate this work to my deceased father who have left me a lot of love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NODAL METHODS FOR CALCULATING NUCLEAR REACTOR TRANSIENTS, CONTROL ROD PATTERNS, AND FUEL PIN POWERS

Chapter 1

INTRODUCTION

The efficient and safe operation of a nuclear power plant represents the major challenge to nuclear reactor design/operation engineers. The prediction of the performance of a nuclear reactor is essential to this task.

The reactor design/operation engineer is called upon to provide the essential physics information required in reactor design, operation, and control. Several characteristics of the reactor must be established to ensure that the reactor can be started up, operated at a steady state over its entire operating lifetime, and shut down as needed in a safe and economic manner. Some of the parameters and phenomena, which must be predicted with sufficient accuracy, are the spatial distribution of the power generation throughout the operational period, the positioning of control rods within the core as required for control of the nuclear reactor, and the time-dependent behavior of the spatial power distribution during off-normal and accident situations.

For the demonstration of prediction accuracy of such problems, it is frequently sufficient to know the average flux and power produced in certain subregions of a reactor. In these instances it is an enormous waste of effort and economy to go through the detailed multi-dimensional models furnished by finite-difference techniques with small mesh spacing. Therefore, so-called "nodal" methods have been designed to compute only the average flux and power values in predetermined subregions.

The remainder of this dissertation discusses the computationally efficient nodal methods used to calculate reactor transients, control rod patterns, and fuel pin powers and the results that have been obtained. The dissertation is presented in three major parts of which the first, second, and third part are Chapter 2, 3, and 4, respectively. Chapter 2 describes the thermal leakage-to-absorption ratio acceleration scheme for solving reactor kinetics. A search technique using a heuristic method for generation of optimum control rod pattern is depicted in Chapter 3. Chapter 4 presents a polynomial flux expansion method for reconstruction of fuel pin powers.

Chapter 2

AN EFFICIENT COMPUTATIONAL TECHNIQUE FOR THERMAL REACTOR TRANSIENTS

2.1 Introduction

An important aspect in the design and safe operation of a nuclear reactor is the behavior of a reactor during a transient, or non steady-state condition. Efforts to answer safety questions which arise in conjunction with actual and hypothetical accident scenarios often require the knowledge of multi-dimensional transient power distributions.

The accurate prediction of reactor behavior is costly because it involves the determination of a multi-dimensional power distribution throughout a large and often geometrically complicated core. Even if it is possible to determine the group diffusion theory parameters that accurately predict the average group fluxes within small, explicitly represented regions such as homogenized fuel cells, control rods, burnable poison lumps, etc., the solution of the corresponding group diffusion equations is expensive. The detailed description of a thermal reactor can easily involve a million mesh points and several energy groups. For problems of these magnitudes, time-dependent group-diffusion calculations based on a finite difference approximation are unmanageable.

Because of this situation, the problem is usually decomposed into two classes of methods. The first class of methods determines the equivalent homogenized group-diffusion theory parameters representing fairly large subregions such as an entire fuel assembly. The author will not address the many problems associated with homogenization technique

[1,2,3], but rather approach the task of determining the spatial power distribution within a reactor which has been partitioned into "homogenized" regions. The equivalent homogenized diffusion theory parameters for each region are assumed to be known.

The second class of methods circumscribes the technique of how to solve actually the reactor kinetics equations with the equivalent homogenized group-diffusion theory parameters. Many solution methods are presently available to the nuclear reactor community. A complete description of each of these methods is too lengthy to be included in this review. Instead, only the most widely used methods are summarized in this section.

In finite difference schemes [4,5,6,7], low order difference approximations are used to represent the leakage term. These finite difference methods possess several advantages over most other schemes. For example, these methods are conceptually simple and the resulting algebraic equations are such that only adjacent nodes are coupled by the spatial leakage terms. The most important property of the finite difference technique is that it can be shown to converge to the exact solution of the multi-group diffusion equations as the mesh spacing becomes increasingly small. The only real disadvantage of the finite difference schemes is that very fine spatial meshes which result in computer memory storage problems and huge execution time are required to achieve acceptable accuracy.

In past years, many researchers have applied finite element techniques [8,9,10] to solve the multi-group diffusion equations. In the finite element methods, the spatial shapes of the multi-group fluxes are represented as polynomials over large homogeneous regions. The finite element schemes also converge to the exact solution of the multi-group diffusion equations as the mesh spacing becomes increasingly small. The major disadvantage is that the coupling of the finite element equations is much more extensive than with the finite difference equations.

In nodal methods [11,12,13,14], the quantities of interest are usually the group-dependent neutron fluxes averaged over large spatial subregions(nodes) and the neutron currents averaged over the faces of the nodes. Determining the coupling constants that relate the current flowing from one node to its neighbor to the average fluxes of these two neighbors is an important matter. The difficulty with the nodal methods is that the relationship between the node-averaged fluxes and the face-averaged currents must be obtained. Once the relationship between them is specified, nodal equations with a structure similar to finite difference equations can be constructed.

Many space- and time-differencing methods for solving the time-dependent(kinetic) multi-group diffusion equations are available. These methods include, quasistatic method [15,16], adiabatic method [17], synthesis method [18], prompt-jump approximation [19,20], and matrix splitting method [21,22,23,24,25,26,27].

In the quasistatic method, the spatial- and time-dependent flux vector is separated into a product of the shape function that is slowly varying with time, $\Psi(\mathbf{r},t)$, and a more rapidly varying amplitude, T(t). The main idea in the quasistatic method is that a low-order approximation with large integration time steps can be used for the calculation of the shape function whereas a high-order approximation with a small integration time step is used for the calculation of the amplitude function.

Most transients and accidents of importance in LWR safety analyses are relatively slow compared to the neutron life time. It is therefore reasonable to assume that during a transient the time retardation in the shape of the precursor distribution may be ignored so that the neutron flux goes immediately to its equilibrium value. This is the basis of the so-called "adiabatic approximation" [28]. With this assumption, the neutron flux distribution in a given time may be determined from a modified steady-state calculation. The adiabatic method is different from the quasistatic method in which the time derivative of the amplitude function appears in the shape function.

In the synthesis method, the flux is expanded into a series of time dependent modes with time dependent coefficients. The precomputed expansion functions are obtained over large regions of the reactor. These methods offer the largest reduction in the number of spatial unknowns. The unfortunate drawback of the synthesis methods is that there is no systematic procedure for choosing the expansion functions. Furthermore, rigorous error bounds on the synthesized solutions do not exist. This lack of error bounds has limited wide use of the synthesis method.

Quasistatic method, adiabatic method, and synthesis method are characterized by relatively small computational time requirements. However, these methods are also known by a lack of definitive error bounds on the final solution.

For reactor power transients dominated by delayed neutrons, the prompt-jump approximation is widely used in the numerical study of the point kinetics equations since it permits the use of large iterative time steps. But the success of multi-dimensional and multi-group reactor kinetics model is not yet known.

As computational capabilities increase, direct solution techniques which use finite differencing in time and space become practical for multi-dimensional calculations.

The WIGLE [6], TWIGL [7], and GAKIN [29] methods represent successful application of direct solution methods to the one- and two-dimensional problems, respectively. The TWIGL program uses a Gauss-Seidel iteration for the inner iteration and a cyclic Chebyshev polynomial method for the outer iteration.

The GAKIN method has been extended to two- and three-dimensional space-kinetics equations in conjunction with the alternating direction implicit (ADI) method [21,22] as well as alternating direction explicit (ADE) method [23]. The major advantage of the multi-dimensional version of the GAKIN method is that it does not require the two loops of iterative schemes as in the standard methods. In other words, one time step needs only one sweep of grid points, so the computing time for one time step is shorter than the standard implicit method. The numerical study shows, however, that the multi-dimensional version of the GAKIN method for fast transient calculations needs much smaller time steps than the standard implicit method.

Alternating direction implicit and alternating direction explicit are parts of a general class of matrix splitting methods. These methods consist of two kind of sweeps: one inverting horizontally from the bottom to top rows for half time steps, another inverting vertically from the leftmost to the rightmost column for next time step. A modification of

ADI, ADI-B [24], which closely resembles the TWIGL scheme results in block-tridiagonal difference equations that are amendable to noniterative solution techniques. It has been proved that the accuracy of ADI and ADE methods can be greatly improved by the application of a transform of variables known as the exponential transformation [23,25]. The numerical study performed by the author of the method shows that the scheme is stable for all the problems tested and the computing time is three to ten times faster than the TWIGL scheme on the same accuracy basis for a specific class of problem tested. For a more difficult class of problems the running time was equivalent to the TWIGL scheme because the ADI-B scheme requires a much smaller time step if the transient is fast and the flux distribution is complicated. Alternating direction checkerboard (ADC) [26] and locally one-dimensional (LOD) [27] methods are also parts of a general class of matrix splitting methods. The sweeps of these methods are different from both ADI methods and ADE methods.

The objective of this study is to develop computationally efficient numerical methods for solving the two group, multi-dimensional transient neutron diffusion equations with moderator and fuel temperature feedback. By employing an approximation for the spatial shape of the thermal fluxes(which is called "thermal leakage-to-absorption ratio approximation"), the convergence of the iteration scheme can be accelerated and the resulting computing time can be greatly reduced while maintaining computational accuracy.

This study discusses the thermal leakage-to-absorption ratio acceleration scheme, hereafter referred to as TLAR acceleration scheme, for solving the two-group, multi-dimensional reactor kinetics equations and the results that have been obtained. In Section 2.2, the derivations of the finite difference nodal diffusion equations are presented. In Section 2.3, the TLAR acceleration scheme is discussed. The thermal-hydraulic feedback model is included in Section 2.4. Results of a benchmark problem are discussed in Section 2.5. Finally, a summary of this investigation and recommendations for future research are given in Section 2.6.

2.2 Neutronics Model

The reactor power distribution is calculated by using a model representing the reactor as a group of geometrically identical nodes. A node is defined as a homogeneous nuclear property volume whose boundaries are specified by mesh lines. Mesh lines are chosen to lie between fuel assemblies and along the core periphery.

The power distribution calculation is based on a finite difference approximation to the kinetics equations. Fluxes are calculated at the center of each node. An averaging scheme is then used to calculate node averaged fluxes. The power distribution is calculated from the node average flux distributions.

2.2.1 Formulation of Difference Equations for Two-Group Diffusion Equations

In most situations encountered in the analysis of LWR's, it is sufficient to model the neutronics behavior of the reactor by a low order approximation to the formally exact neutron transport equation. The most frequently used of these approximations is the two-group neutron diffusion theory. The general two-group diffusion theory equations are derived from a fundamental conservation law which states that the rate of change of neutron density is equal to the rate at which neutrons are produced minus the rate at which they are absorbed or escape within a volume of interest. For this model, the set of time- and space-dependent coupled partial differential equations for which the two-group diffusion and delayed precursor equations are sought can be written as:

$$\frac{1}{v_1}\frac{\partial}{\partial t}\phi_1(\vec{r},t) = \nabla\bullet D_1(\vec{r},t)\,\nabla\phi_1(\vec{r},t)$$
$$-\Sigma_{r1}(\vec{r},t)\,\phi_1(\vec{r},t) + (1-\beta)\,[\nu\Sigma_{f1}\phi_1(\vec{r},t) \qquad (2.2.1.a)$$
$$+\nu\Sigma_{f2}(\vec{r},t)\,\phi_2(\vec{r},t)] + \sum_{d=1}^{DG}\lambda_d C_d(\vec{r},t)$$

$$\frac{1}{v_2}\frac{\partial}{\partial t}\phi_2(\vec{r},t) = \nabla\bullet D_2(\vec{r},t)\,\nabla\phi_2(\vec{r},t) \qquad (2.2.1.b)$$
$$-\Sigma_{a2}(\vec{r},t)\,\phi_2(\vec{r},t) + \Sigma_{s12}(\vec{r},t)\,\phi_1(\vec{r},t)$$

$$\frac{\partial}{\partial t}C_d(\vec{r},t) = \beta_d\,[\nu\Sigma_{f1}\phi_1(\vec{r},t)$$
$$+\nu\Sigma_{f2}(\vec{r},t)\,\phi_2(\vec{r},t)] - \lambda_d C_d(\vec{r},t), \qquad (2.2.1.c)$$
$$d = 1,2,\ldots,DG.$$

where

ϕ_1, ϕ_2 = fast and thermal fluxes

C_d = density of delayed neutron precursor in family d

v_1, v_2 = fast and thermal neutron speeds

D_1, D_2 = fast and thermal diffusion coefficients

Σ_{r1} = fast removal cross section (= Σ_{a1} + Σ_{s12})

Σ_{a1} = fast absorption cross section

Σ_{s12} = fast slowing down cross section

Σ_{a2} = thermal absorption cross section

$\nu\Sigma_{f1}$, $\nu\Sigma_{f2}$ = fast and thermal ν-fission cross section

DG = total number of delayed neutron precursor families

λ_d = decay constant for delayed neutron precursor family d

β_d = fractional yield of delayed neutron precursor family d

β = total fractional yield of delayed neutron precursors per fission.

The static solution to Eqs.(2.2.1) is obtained by assuming that the reactor is in a "critical" configuration. That is, all of the properties of the reactor are independent of time and hence all of the time derivatives of Eqs.(2.2.1) are identically zero. The static solution to Eqs.(2.2.1) is obtained by varying ν(mean number of neutrons emitted per fission) such that a nontrivial solution to the static two-group equations exists.

If the distribution of material properties in space and time, the initial fast and thermal flux distributions in space, and the boundary conditions are specified, a unique solution to Eqs.(2.2.1) exists. The two most widely used boundary conditions applied to the outer surface of the reactor core are that the neutron flux or the incoming neutron current for each energy group is identically zero. At any internal surfaces, continuity of neutron flux and current are imposed for each energy group.

The global reactor problem is treated in three-dimensional Cartesian geometry, where x, y, and z represent the three coordinate directions. The spatial domain is subdivided into a regular array of nodes. The mesh point is placed at the center of the node of height d_z, width d_x, and depth d_y, as shown in Figure 2.2.1.

The time-difference form of Eqs.(2.2.1) is obtained by discretizing the continuous time domain into a sequence of absolute time values t_0, $t_1, \ldots,$ $t_n, \ldots,$ t_T where t_0 and t_T are the initial and final time, respectively. The time dependent quantities and space dependent quantities can be symbolized by using superscript, n, and subscript, i, respectively. Thus, for example, ϕ_{1i}^n refers to the fast flux at node i and time t_n. We then approximate the time derivative for each time interval

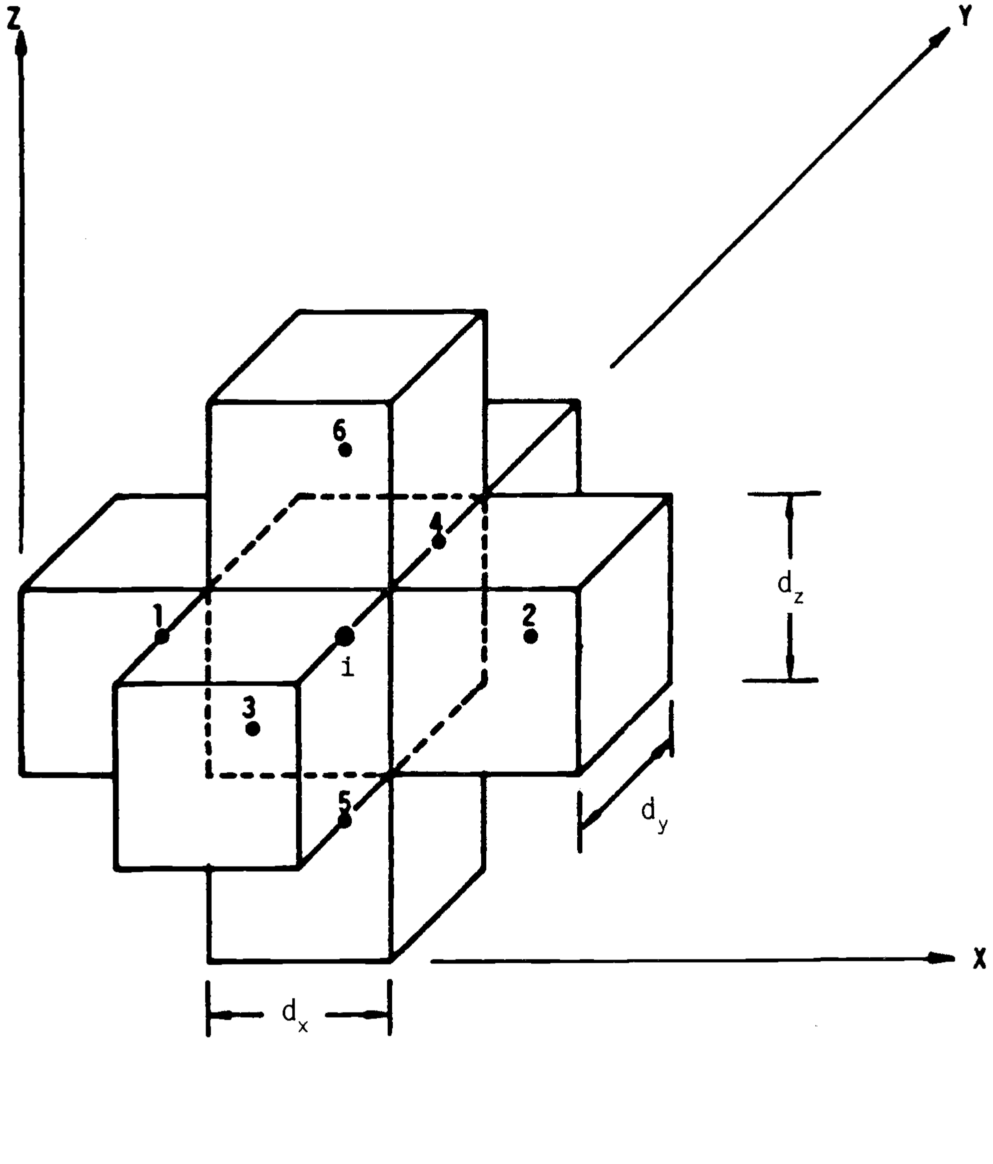


Figure 2.2.1 Three-Dimensional Mesh Description in X-Y-Z Geometry

$\Delta t_n = t_{n+1} - t_n$ as

$$\frac{\partial}{\partial t} F_i = \frac{F_i^{n+1} - F_i^n}{\Delta t_n}, \tag{2.2.2}$$

where F represents ϕ_1, ϕ_2, or C_d. The delayed neutron precursor terms are central differenced over this time interval as

$$C_{di} = \frac{1}{2} (C_{di}^{n+1} + C_{di}^n) . \tag{2.2.3}$$

By using Eqs.(2.2.2) and (2.2.3), the implicit form of the Eq.(2.2.1.c) can be written as

$$\frac{C_{di}^{n+1} - C_{di}^n}{\Delta t_n} = \beta_d (\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1} + \nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1}) - \lambda_d (\frac{C_{di}^{n+1} + C_{di}^n}{2}) . \tag{2.2.4}$$

Equation (2.2.4) can be rearranged to obtain

$$C_{di}^{n+1} = (\frac{1 - \frac{\lambda_d}{2}\Delta t_n}{1 + \frac{\lambda_d}{2}\Delta t_n}) C_{di}^n + B_d^n \Delta t_n (\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1} + \nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1}) , \tag{2.2.5}$$

where

$$B_d^n = \frac{\beta_d}{1 + \frac{\lambda_d}{2}\Delta t_n} . \tag{2.2.6}$$

Again by using Eqs.(2.2.2) and (2.2.3), the implicit form of the Eq.(2.2.1.a) can be written as

$$\begin{aligned} \frac{1}{v_1}\frac{\phi_{1i}^{n+1} - \phi_{1i}^n}{\Delta t_n} &= \nabla \bullet D_{1i}^{n+1} \nabla \phi_{1i}^{n+1} - \Sigma_{r1i}^{n+1}\phi_{1i}^{n+1} \\ &+ (1-\beta)\ [\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1} + \nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1}] \\ &+ \sum_{d=1}^{DG} \frac{\lambda_d}{2}\ (C_{di}^{n+1} + C_{di}^n) . \end{aligned} \tag{2.2.7}$$

Equation (2.2.5) can be substituted into Eq.(2.2.7) to obtain

$$\begin{aligned} \frac{1}{v_1}\frac{\phi_{1i}^{n+1} - \phi_{1i}^n}{\Delta t_n} &= \nabla \bullet D_{1i}^{n+1} \nabla \phi_{1i}^{n+1} - \Sigma_{r1i}^{n+1}\phi_{1i}^{n+1} \\ &+ (1-\beta)\ [\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1} + \nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1}] \\ &+ \sum_{d=1}^{DG} \frac{\lambda_d}{2}\ [\frac{2}{1 + \frac{\lambda_d}{2}\Delta t_n} C_{di}^n \\ &+ B_d^n \Delta t_n (\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1} + \nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1})\ ] . \end{aligned} \tag{2.2.8}$$

Performing some subsequent rearrangement yields

$$\begin{aligned} \frac{1}{v_1}\frac{\phi_{1i}^{n+1} - \phi_{1i}^n}{\Delta t_n} &= \nabla \bullet D_{1i}^{n+1} \nabla \phi_{1i}^{n+1} - \Sigma_{r1i}^{n+1}\phi_{1i}^{n+1} \\ &+ (1-B^n)\ [\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1} + \nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1}] \\ &+ \sum_{d=1}^{DG} \frac{\lambda_d}{1 + \frac{\lambda_d}{2}\Delta t_n} C_{di}^n , \end{aligned} \tag{2.2.9}$$

where

$$B^n = \sum_{d=1}^{DG} B_d^n = \sum_{d=1}^{DG} \frac{\beta_d}{1 + \frac{\lambda_d}{2}\Delta t_n} . \tag{2.2.10}$$

Similarly, the implicit form of the Eq.(2.2.1.b) can be written as

$$\frac{1}{v_2}\frac{\phi_{2i}^{n+1} - \phi_{2i}^{n}}{\Delta t_n} = \nabla\bullet D_{2i}^{n+1}\nabla\phi_{2i}^{n+1} - \Sigma_{a2i}^{n+1}\phi_{2i}^{n+1} + \Sigma_{s12i}^{n+1}\phi_{1i}^{n+1} . \tag{2.2.11}$$

In order to obtain the time dependent spatial finite-differenced two-group diffusion equations, we integrate Eqs.(2.2.5), (2.2.9), and (2.2.11) over a node volume V.

In general, integration of the leakage term is accomplished by first transforming the volume integral to a surface integral by means of Green's theorem

$$\int_V \nabla\cdot D\nabla\phi \; dV = \int_S D\nabla\phi \; dS. \tag{2.2.12}$$

The surface integral is carried out over the six faces of the node.

Evaluation of the flux gradient at the interface is easily understood by considering a one-dimensional case where two-mesh points, i and i+1, are separated by a distance, d, and equidistant from a mesh line separating two regions. The region to the left of the mesh line has diffusion coefficient D_i, and that region to the right has D_{i+1}.

Using difference approximations, the continuity of the neutron current of the node interface gives

$$-D_i \frac{\phi_{i+\frac{1}{2}} - \phi_i}{\frac{d}{2}} = -D_{i+1} \frac{\phi_{i+1} - \phi_{i+\frac{1}{2}}}{\frac{d}{2}} \quad . \qquad (2.2.13)$$

ϕ_i, $\phi_{i+1/2,}$ and ϕ_{i+1} refer to the fluxes at point i, the mesh line, and point i+1, respectively. The flux at the interface is given by

$$\phi_{i+\frac{1}{2}} = \frac{D_i\phi_i + D_{i+1}\phi_{i+1}}{D_i + D_{i+1}} \quad . \qquad (2.2.14)$$

Using this expression, the flux gradient at the interface can be written in terms of ϕ_i and ϕ_{i+1}

$$\nabla\phi_{i+\frac{1}{2}} = \left(\frac{D_{i+1}}{D_i + D_{i+1}}\right) \left(\frac{\phi_{i+1} - \phi_i}{\frac{d}{2}}\right) \quad . \qquad (2.2.15)$$

Finally, the time dependent spatial finite-differenced two-group difference equations for mesh point i become

$$\begin{aligned} \frac{1}{V_1} \frac{\phi_{1i}^{n+1} - \phi_{1i}^{n}}{\Delta t_n} = &-\sum_{m=1}^{6} \frac{D_{1i}^{n+1} D_{1m}^{n+1}}{D_{1i}^{n+1} + D_{1m}^{n+1}} \frac{(\phi_{1i}^{n+1} - \phi_{1m}^{n+1})}{\frac{d_m}{2}} \frac{A_m}{V_i} \\ &- \Sigma_{r1i}^{n+1}\phi_{1i}^{n+1} + (1 - B^n)\,[\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1} + \nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1}] \qquad (2.2.16) \\ &+ \sum_{d=1}^{DG} \frac{\lambda_d}{1 + \frac{\lambda_d}{2}\Delta t_n} C_{di}^{n} \ , \end{aligned}$$

$$\frac{1}{v_2}\,\frac{\phi_{2i}^{n+1}-\phi_{2i}^{n}}{\Delta t_n} = -\sum_{m=1}^{6}\frac{D_{2i}^{n+1}D_{2m}^{n+1}}{D_{2i}^{n+1}+D_{2m}^{n+1}}\;\frac{(\phi_{2i}^{n+1}-\phi_{2m}^{n+1})}{\frac{d_m}{2}}\;\frac{A_m}{V_i} \qquad (2.2.17)$$

$$-\Sigma_{a2i}^{n+1}\phi_{2i}^{n+1}+\Sigma_{s12i}^{n+1}\phi_{1i}^{n+1}\;,$$

and

$$C_{di}^{n+1} = \left(\frac{1-\frac{\lambda_d}{2}\Delta t_n}{1+\frac{\lambda_d}{2}\Delta t_n}\right)C_{di}^{n} \qquad (2.2.5)$$

$$+B_d^n\Delta t_n\,(\nu\Sigma_{f1i}^{n+1}\phi_{1i}^{n+1}+\nu\Sigma_{f2i}^{n+1}\phi_{2i}^{n+1})\;,$$

where

A_m = area of boundary between mesh point i and adjacent mesh point m,

V_m = volume associated with mesh point i,

d_m = distance between mesh point i and adjacent mesh point m.

The summation is over all six nodes adjacent to node i, as shown in Figure 2.2.1.

Since all transient problems are assumed to start from a steady state, the initial precursor concentration terms at node i are calculated by the steady state relation

$$C_{di}^{0} = \frac{\beta_d}{\lambda_d}\,(\nu\Sigma_{f1i}^{0}\phi_{1i}^{0}+\nu\Sigma_{f2i}^{0}\phi_{2i}^{0})\;. \qquad (2.2.18)$$

At each time step n, Eqs.(2.2.16) and (2.2.17) each constitute a set of inhomogeneous algebraic equations. These equations may be written as two coupled matrix equations

$$\begin{aligned} [\boldsymbol{u_1}]\,[\Phi_1] &= [\boldsymbol{w_2}]\,[\Phi_2] + [z_1] \\ [\boldsymbol{u_2}]\,[\Phi_2] &= [\boldsymbol{w_1}]\,[\Phi_1] + [z_2] \;, \end{aligned} \qquad (2.2.19)$$

where both $[\phi_1]$ and $[\phi_2]$ are column vectors composed of a number of components equal to the number of spatial solution points; $[u_1]$ and $[u_2]$ are matrices representing the finite difference formulas; $[w_1]$ and $[w_2]$ are diagonal matrices and $[z_1]$ and $[z_2]$ are known vectors.

The standard numerical solution method to solve Eq.(2.2.19) involves two kinds of iterative schemes: inner iterations and outer iterations. The direct inversions of Eq.(2.2.19) is not practical for the multi-dimensional case because $[u_1]$ and $[u_2]$ in Eq.(2.2.19) are not tridiagonal.

A fission source overrelaxation (FSOR) or a Chebyshev polynomial method is necessary for Eq.(2.2.19) and is called the "outer iteration". Each outer iteration needs the solution of the left sides of Eq.(2.2.19) whenever the right sides are given. The iterative scheme to solve the inhomogeneous problem is called the "inner iteration". The Gauss-Seidel scheme for the inner iteration and FSOR for the outer iteration will be used for both the steady state and transient solutions.

Finally, Eq.(2.2.16) and Eq.(2.2.17) can be recast into a form more convenient for performing flux iterations. That is,

$$\phi_{1i}^{n+1} = \frac{S_{1i}^{n+1} + \sum_{m=1}^{6} f_{1m}^{n+1} \phi_{1m}^{n+1}}{f_{1i7}^{n+1}} , \tag{2.2.20}$$

$$\phi_{2i}^{n+1} = \frac{S_{2i}^{n+1} + \sum_{m=1}^{6} f_{2m}^{n+1} \phi_{2m}^{n+1}}{f_{2i7}^{n+1}} , \tag{2.2.21}$$

where,

$$\begin{aligned} S_{1i}^{n+1} = \frac{\phi_{1i}^{n+1}}{v_1 \Delta t_n} + (1 - B^n) \; [\nu \Sigma_{f1i}^{n+1} \phi_{1i}^{n+1} \\ + \nu \Sigma_{f2i}^{n+1} \phi_{2i}^{n+1}] + \sum_{d=1}^{DG} \frac{\lambda_d}{1 + \frac{\lambda_d}{2} \Delta t_n} C_{di}^{n} , \end{aligned} \tag{2.2.22}$$

$$f_{1m}^{n+1} = \frac{2 D_{1i}^{n+1} D_{1m}^{n+1}}{d_m^2 (D_{1i}^{n+1} + D_{1m}^{n+1})} , \qquad m=1, \ldots, 6 \tag{2.2.23}$$

$$f_{1i7}^{n+1} = \frac{1}{v_1 \Delta t_n} + \Sigma_{r1i}^{n+1} + \sum_{m=1}^{6} f_{1m}^{n+1} , \tag{2.2.24}$$

$$S_{2i}^{n+1} = \frac{\phi_{2i}^{n+1}}{v_2 \Delta t_n} + \Sigma_{s12i}^{n+1} \phi_{1i}^{n+1} , \tag{2.2.25}$$

$$f_{2m}^{n+1} = \frac{2 D_{2i}^{n+1} D_{2m}^{n+1}}{d_m^2 (D_{2i}^{n+1} + D_{2m}^{n+1})} , \qquad m=1, \ldots, 6 \tag{2.2.26}$$

$$f_{2i7}^{n+1} = \frac{1}{v_2 \Delta t_n} + \Sigma_{a2i}^{n+1} + \sum_{m=1}^{6} f_{2m}^{n+1} \quad . \qquad (2.2.27)$$

2.2.2 Node Average Power Calculation

A coarse mesh diffusion theory model tends to predict current densities poorly at node interfaces. This is particularly true if there is a large difference between the diffusion coefficients of the two adjacent nodes or if the node is on the core periphery. This can lead to a fairly large error in the prediction of power distribution. In an effort to reduce this error Borresen's averaging scheme [30] is employed. In Borresen's scheme the node flux calculations are improved by the introduction of special weight factors determined by off-line calculations. The node average fast flux, $\bar{\phi}_{1i}$, is calculated as a weighted average of the midpoint flux, ϕ_{1i}, in that node and the flux values on the interfaces to the six neighboring nodes,

$$\bar{\Phi}_{1i} = B_1 \Phi_{1i} + 2C_1 (\sum_{j=1}^{4} \Phi_{1i}^{j} + r\sum_{k=1}^{2} \Phi_{1i}^{k}) \quad , \qquad (2.2.28)$$

where

$$B_1 = \frac{3a_1}{3a_1 + (1-a_1)(r+2)} \quad , \qquad (2.2.29)$$

$$C_1 = \frac{1-a_1}{12a_1+4(1-a_1)(r+2)} , \tag{2.2.30}$$

$$r = \frac{d_x^2}{d_z^2} , \tag{2.2.31}$$

ϕ_{1i}^{j} = interface fast flux between node i and j in horizontal plane,

ϕ_{1i}^{k} = interface fast flux between node i and k in vertical plane,

where

a_1 = input weight factor for fast flux.

Similarly, the node average thermal flux, $\bar{\phi}_{2i}$, is calculated as

$$\bar{\Phi}_{2i} = B_2 \Phi_{2i} + 2C_2 \left(\sum_{j=1}^{4} \Phi_{2i}^{j} + r\sum_{k=1}^{2} \Phi_{2i}^{k}\right) , \tag{2.2.32}$$

where

$$B_2 = \frac{3a_2}{3a_2+(1-a_2)(r+2)} , \tag{2.2.33}$$

$$C_2 = \frac{1-a_2}{12a_2+4(1-a_2)(r+2)} , \tag{2.2.34}$$

ϕ_{2i}^{j} = interface thermal flux between node i and j in horizontal plane,

ϕ_{2i}^{k} = interface thermal flux between node i and k in vertical plane,

where

a_2 = input weight factor for thermal flux.

Two different values for the weight factors, a_1 and a_2, are used to calculate the node averaged fast and thermal fluxes. Borresen recommends 0.3 and 0.7 for the fast and thermal groups, respectively.

Finally, the power distribution is obtained from the node averaged flux distributions,

$$P_i = E_R(\nu\Sigma_{f1i}\overline{\Phi}_{1i} + \nu\Sigma_{f2i}\overline{\Phi}_{2i}) \quad , \tag{2.2.35}$$

where

E_R = conversion factor determined by reactor power.

2.2.3 Transient Solution Algorithm for the Two-Group Diffusion Equations

The full set of temporally- and spatially-discretized diffusion equations are given by Eq.(2.2.20) and (2.2.21). To advance the solution from one time step to the next requires that many of the elements of the coefficient matrix be updated because of feedback effects. Hence, these matrix elements must be recalculated at each time step. The detailed thermal-hydraulic feedback mechanism will be discussed later in Section 2.4.

A description of the transient solution algorithm for the finite-differenced two-group diffusion equations is outlined below:

1. Choose the times(t_0, t_1,..., t_t,..., t_T) which partition the transient problem.
2. Calculate the steady state fluxes $[\phi_1]^0$, $[\phi_2]^0$, and precursor densities $[C_d]^0$.
3. Obtain $[\bar{\phi}_1]^0$ and $[\bar{\phi}_2]^0$ with Eq.(2.2.28) and Eq.(2.2.32).
4. Obtain $[P]^0$ with Eq.(2.2.35).
5. Alter cross sections to correspond to transient conditions such as control rod position change and/or feedback effects at time t_{n+1}.
6. Solve Eq.(2.2.20) and Eq.(2.2.21) iteratively for $[\phi_1]^{n+1}$, $[\phi_2]^{n+1}$, respectively.
7. Obtain $[C_d]^{n+1}$ at time t_{n+1} with Eq.(2.2.5).
8. Obtain $[\bar{\phi}_1]^{n+1}$ and $[\bar{\phi}_2]^{n+1}$ with Eq.(2.2.28) and Eq.(2.2.32).
9. Obtain $[P]^{n+1}$ with Eq.(2.2.35).
10. Repeat step 5-9 for each time step until the end of the last time domain.

The iterative scheme described above step 6 is similar to that of solving the steady state two-group diffusion equation. Equations (2.2.20) and (2.2.21) are solved using the inner-outer iteration technique.

2.3 Thermal Leakage-to-Absorption Ratio Acceleration Method

In Section 2.2, the temporally- and spatially-discretized nodal diffusion equations were obtained for solving the two-group diffusion equations. In this section, the two-group finite difference kinetics equations with TLAR acceleration scheme will be derived and an algorithm for solving these kinetics equations will be presented.

2.3.1 Formulation

Since the modified one-group diffusion theory scheme [30,31,32] runs remarkably faster than the two-group diffusion theory scheme, the modified one-group method has been widely used for the analysis of static and transient problems. However, it has been shown that the computational accuracy of the modified one-group method appears a bit deteriorated in transient applications [31,32]. This is due to the simplicity of the neglecting the thermal leakage effect. Therefore, the modified-one group method may not be valid for problems in which the thermal flux gradients between nodes are severe, and the thermal neutron diffusion has a significant space-time effect.

The inclusion of the thermal neutron diffusion effect is a key to improve the accuracy of modified one-group diffusion theory compared to that of two-group diffusion theory. Since the thermal neutron leakage rate is directly related to the thermal neutron shape function which is invariant or slowly varying during transient with small time step, a new method can be proposed assuming that even if the thermal leakage rate and absorption rate varies significantly in a given node, the ratio of those two changes little during that period. Even in a region where rapid flux changes occur such as after a control rod motion, the change in thermal leakage to absorption ratio is limited to the fairly small region around the moved rod. Since the thermal leakage terms in nodal balance equations are generally small in magnitude compared to the absorption terms, the errors introduced by this assumption are generally quite acceptable. The basic assumption of this method is that:

$$R(\vec{r},t) = R(\vec{r},t^{-}) \ , \quad t=t^{-}+\Delta t \qquad (2.3.1.a)$$

where

$$R(\vec{r},t^{-}) = \frac{-\nabla\cdot D_2(\vec{r},t^{-})\,\nabla\phi_2(\vec{r},t^{-})}{\Sigma_{a2}(\vec{r},t^{-})\,\phi_2(\vec{r},t^{-})} \qquad (2.3.1.b)$$

$$R(\vec{r},t) = \frac{-\nabla\cdot D_2(\vec{r},t)\,\nabla\phi_2(\vec{r},t)}{\Sigma_{a2}(\vec{r},t)\,\phi_2(\vec{r},t)} \qquad (2.3.1.c)$$

and R denotes the ratio of the thermal leakage to the absorption rate. The use of Eq.(2.3.1) leads to

$$\begin{aligned}-\nabla\cdot D_2(\vec{r},t)\,\nabla\phi_2(\vec{r},t) &= \\ &R(\vec{r},t^{-})\,\Sigma_{a2}(\vec{r},t)\,\phi_2(\vec{r},t) \ .\end{aligned} \qquad (2.3.2)$$

Substitution of Eq.(2.3.2) into the right side of Eq.(2.2.1.b) yields the simplified thermal group flux,

$$\frac{1}{v_2}\frac{\partial}{\partial t}\phi_2(\vec{r},t) = [1+R(\vec{r},t^-)]\ \Sigma_{a2}(\vec{r},t)\phi_2(\vec{r},t)$$
$$+\Sigma_{s12}(\vec{r},t)\phi_1(\vec{r},t)\quad . \qquad (2.3.3)$$

The finite-difference approximation for the thermal group can be written as:

$$\frac{1}{v_2}\ \frac{\phi_{2i}^{n+1}-\phi_{2i}^{n}}{\Delta t_n} = -(1+R_i^n)\ \Sigma_{a2i}^{n+1}\phi_{2i}^{n+1}$$
$$+\Sigma_{s12i}^{n+1}\phi_{1i}^{n+1}\quad , \qquad (2.3.4)$$

where

$$R_i^n = \frac{-\sum_{m=1}^{6}\dfrac{D_{2i}^{n+1}D_{2m}^{n+1}}{D_{2i}^{n+1}+D_{2m}^{n+1}}\ \dfrac{(\phi_{2i}^{n}-\phi_{2m}^{n})}{\dfrac{d_m}{2}}\ \dfrac{A_m}{V_i}}{\Sigma_{a2i}^{n}\phi_{2i}^{n}}\quad . \qquad (2.3.5)$$

Substituting Eq.(2.3.4) into the fast group equation, Eq.(2.2.16), yields the finite-difference approximation for the fast group,

$$\begin{aligned}\frac{1}{v_1}\frac{\phi_{1i}^{n+1}-\phi_{1i}^{n}}{\Delta t_n} = &-\sum_{m=1}^{6}\frac{D_{1i}^{n+1}D_{1m}^{n+1}}{D_{1i}^{n+1}+D_{1m}^{n+1}}\;\frac{(\phi_{1i}^{n+1}-\phi_{1m}^{n+1})}{\frac{d_m}{2}}\;\frac{A_m}{V_i}\\ &-\Sigma_{r1i}^{n+1}\phi_{1i}^{n+1}\\ &+(1-B^n)\,[\nu\Sigma_{f1i}^{n+1}+\frac{\nu\Sigma_{f2i}^{n+1}\Sigma_{s12i}^{n+1}}{\frac{1}{v_2\Delta t_n}+(1+R_i^n)\,\Sigma_{a2i}^{n+1}}]\,\phi_{1i}^{n+1}\\ &+(1-B^n)\,[\frac{\frac{\nu\Sigma_{f2i}^{n+1}}{v_2\Delta t_n}}{\frac{1}{v_2\Delta t_n}+(1+R_i^n)\,\Sigma_{a2i}^{n+1}}]\,\phi_{2i}^{n}\\ &+\sum_{d=1}^{DG}\frac{\lambda_d}{1+\frac{\lambda_d}{2}\Delta t_n}\,C_{di}^{n}\;.\end{aligned}\qquad(2.3.6)$$

The finite-difference approximation for the precursor group is the same as Eq.(2.2.5).

The mathematical structure of above equation is almost identical to the finite-difference modified one-group equation except that the former includes the R_i^n known from the previous time step calculations. The numerical solution for $[\phi_2]$ of Eq.(2.3.4) needs neither inner nor outer iterations. Once we have the solution for $[\phi_1]$ by solving Eq.(2.3.6) then the solution for $[\phi_2]$ is straight forward. Even if the solution of Eqs.(2.3.4) and (2.3.6) reveal errors, they can be used as initial guess solutions for the two-group method described in Section 2.2.1 in order to accelerate the convergence of the iteration.

2.3.2 Transient Solution Algorithm for the Two-Group Diffusion Equations Using a Thermal Leakage-to-Absorption Ratio Acceleration Method

By employing the strategy of the acceleration scheme, the transient solution algorithm described in Section 2.2.3 is modified and summarized as follows:

1. Choose the times(t_0, t_1,..., t_t,..., t_T) which partition the transient problem.
2. Calculate the steady state fluxes $[\phi_1]^0$, $[\phi_2]^0$, and precursor densities $[C_d]^0$.
3. Obtain $[\bar{\phi}_1]^0$ and $[\bar{\phi}_2]^0$ with Eq.(2.2.28) and Eq.(2.2.32).
4. Obtain $[P]^0$ with Eq.(2.2.35).
5. Alter cross sections to correspond to transient conditions such as control rod position change and/or feedback effects at time t_{n+1}.

6a. Solve Eq.(2.3.5) for $[R]^n$.

6b. Solve Eq.(2.3.6) iteratively for $[\phi_1]^{n+1}$.

6c. Solve Eq.(2.3.4) for $[\phi_2]^{n+1}$.

6d. Using solutions obtained in step 6b and 6c as initial guess solutions, solve Eq.(2.2.20) and Eq.(2.2.21) iteratively for $[\phi_1]^{n+1}$ and $[\phi_2]^{n+1}$, respectively.

7. Obtain $[C_d]^{n+1}$ at time t_{n+1} with Eq.(2.2.5).
8. Obtain $[\bar{\phi}_1]^{n+1}$ and $[\bar{\phi}_2]^{n+1}$ with Eq.(2.2.28) and Eq.(2.2.32).
9. Obtain $[P]^{n+1}$ with Eq.(2.2.35).
10. Repeat step 5-9 for each time step until the end of the last time domain.

2.4 Thermal-hydraulic Feedback Model

The important thermal-hydraulic parameters which describe the state of the reactor core and plant are coolant mass flow, steam quality, void fraction, coolant temperature, fuel rod temperature, system pressure, and the thermodynamic properties of the coolant fluid. In nuclear reactors, the variation of output power affects the temperature of both coolant and fuel. The density of coolant in turn affects the power. Thus a thermal-hydraulic feedback calculation is linked to the neutronics model. This is an iterative process by which new nodal nuclear properties are calculated as a function of the thermal-hydraulic parameters.

A closed-channel thermal-hydraulic model is used to estimate the effects of moderator and fuel temperature and moderator density on the nodal neutron flux distribution. Each axial string of nodes is considered a thermal-hydraulic channel, and there are as many axial zones as there are planes.

A minimum departure from nuclear boiling (MDNBR) calculation has also been included in the thermal-hydraulic model and will be discussed in this section.

2.4.1 Heat Balance Equations

Our interest, at present, is not so much in accurate solutions to realistically coupled neutronic, thermal-hydraulic problems, but rather in accurate neutronic solutions to problems with feedback effects. Therefore, the use of a complicated thermal-hydraulic model such as COBRA [33] to test the transient nodal method with feedback appears unnecessary. The much simpler single lumped parameter model such as TWIGL is used as the fundamental feedback mechanism for testing the neutronics solution methods.

In a single lumped parameter model, the fuel rod cell is divided into two fuel and coolant control volume regions within which we desire to know the average fuel temperature T_f, the average coolant temperature T_c, and the average coolant enthalpy h. The equations describing the time behavior of T_f, T_c, and h in a given region are

$$V_{fi}\frac{\rho_{fi}^{n+1}c_{fi}^{n+1}T_{fi}^{n+1}-\rho_{fi}^{n}c_{fi}^{n}T_{fi}^{n}}{\Delta t_n} = q'''^{n+1}_{fi}V_{fi}-\frac{T_{fi}^{n+1}-T_{ci}^{n+1}}{\tilde{R}_i^{n+1}} , \qquad (2.4.1)$$

$$V_{ci}\frac{\rho_{ci}^{n+1}c_{ci}^{n+1}T_{ci}^{n+1}-\rho_{ci}^{n}c_{ci}^{n}T_{ci}^{n}}{\Delta t_n} = \frac{T_{fi}^{n+1}-T_{ci}^{n+1}}{\tilde{R}_i^{n+1}} \qquad (2.4.2)$$

$$-\dot{m}_c\, h_{i+\frac{1}{2}}^{n+1}+\dot{m}_c\, h_{i-\frac{1}{2}}^{n+1} ,$$

where

superscripts n = time indices

subscripts i = space indices

ρ_f = density of fuel

ρ_c = density of coolant

c_f = specific heat of fuel

c_c = specific heat of coolant

V_f = volume of fuel

V_c = volume of coolant

q_f''' = volumetric heat generation rate in fuel

$\tilde{R}$ = thermal resistance of the fuel pin cell

$\dot{m}_c$ = mass flow rate of coolant

$\dot{m}_c\ h_{i+1/2}$ = rate of heat removal due to core coolant flow

$\dot{m}_c\ h_{i-1/2}$ = rate at which the heat content of nodes located lower in the core is being swept into the node of interest due to core coolant flow.

In Eq.(2.4.2) the amount of direct heating of the coolant caused by gamma attenuation and neutron moderation is ignored.

The node average coolant enthalpy h_i at the end of the time step of length Δt_n can be obtained by a straight line interpolation:

$$h_i^{n+1} = \frac{h_{i+\frac{1}{2}}^{n+1} + h_{i-\frac{1}{2}}^{n+1}}{2} \quad . \tag{2.4.3}$$

By using Eq.(2.4.3), the Eq.(2.4.2) can be written as

$$(\rho_c c_c V_c)_i \frac{T_{ci}^{n+1} - T_{ci}^{n}}{\Delta t_n} = \frac{T_{fi}^{n+1} - T_{ci}^{n+1}}{\tilde{R}_i^{n+1}} - 2\dot{m}_c (h_i^{n+1} - h_{i-\frac{1}{2}}^{n+1}) \quad . \tag{2.4.4}$$

Equations (2.4.1) and (2.4.4) are solved for each control volume region only after the node average fluxes, $\bar{\phi}_1^{n+1}$, $\bar{\phi}_2^{n+1}$ have been obtained and used to form q_f'''. For the sake of simplicity, the Eqs.(2.4.1) and (2.4.4) are solved using a fully-implicit temporal differencing scheme, with the same temporal mesh as the neutronic equations. Following evaluation of the core-wide fuel and coolant temperatures, the coolant density is calculated assuming uniform pressure throughout the reactor core.

The power distribution is used to generate the quality of each region through summation of enthalpy from the inlet of region to the node of interest. The bulk coolant properties are then estimated through the use of the standard water property tables and various correlations. Two enthalpy regimes are considered: (a) single-phase forced convection and (b) bulk boiling of the coolant. For the first, the bulk coolant temperature is calculated from the table of enthalpy versus temperature, while for the second case, it is taken as the saturation temperature. For the bulk boiling case, the void fraction is calculated directly from the quality, and the coolant density is derived from the void content and the specific volumes of water and steam, while for the nonboiling case, the coolant density is calculated directly from the specific volume of water.

A. Steam Quality and Void Fraction

The quality, x, of the active coolant in a flow channel is calculated from

$$x = \frac{h-h_{sat}}{h_{fg}} \quad ,$$

where

h = coolant enthalpy (Btu/lbm)

h_{sat} = enthalpy of saturated water

h_{fg} = heat of vaporization.

The void fraction of a node is calculated from the average node quality using Bankoff correlation [34].

$$\alpha = 0 \qquad for\ x \leq 0$$

$$\alpha = \frac{x}{C_0\,[x + \frac{\nu_f}{\nu_g}(1-x)]} \qquad for\ x > 0$$

where

ν_f = specific volume of saturated water

ν_g = specific volume of saturated steam

C_0 = 1/[0.71 + 0.0001 p (psia)].

The density of two-phase mixture, ρ, can be related to the void fraction by

$$\rho = (1 - \alpha)\ \rho_f + \alpha\ \rho_g$$

where

ρ_f = density of saturated water

ρ_g = density of saturated steam.

B. DNBR Calculation

Fuel elements must operate safely below the critical heat flux since a heat flux in excess of the DNBR (Departure From Nucleate Boiling) may result in fuel-element cladding failure.

For the range of interest for pressurized water reactors, the W-3 correlation [35] is applied to the burnout analysis of rod bundles.

$$\begin{aligned} q''_{DNB,EU} = 10^6 * [(&2.022 - 0.0004302\ p) + (0.1722 - \\ &0.0000984\ p)*\exp\{(18.77 - 0.004129\ p)\ x\}] \\ &* [(0.1484 - 1.596x + 0.1729x|x|)\ G/10^6 + \\ &1.037] * [1.157 - 0.869x] \\ &* [0.2664 + 0.8357\ \exp(-3.151\ D_e)] \\ &* [0.8258 + 0.000794\ (h_{sat} - h_{in})]\ , \end{aligned}$$

where

$q''_{DNB,EU}$ = equivalent uniform channel DNB heat flux, Btu/hr-ft^2

p = pressure, psia (between 1000 and 2300 psia)

x = quality (between -0.15 and 0.15)

G = mass flux, lbm/hr-ft^2 (between 0.5*10^6 and 5*10^6 lbm/hr-ft$^{2)}$

h = enthalpy, Btu/lbm (inlet h $\geq$ 400 Btu/lbm)

D_e = channel equivalent diameter, in. (between 0.2 and 0.7 in).

For a nonuniform heat flux, the above correlation is modified by employing a correction factor

$q''_{DNB,N} = q''_{DNB,EU} / F$

where

$q''_{DNB,N}$ = DNB heat flux for the nonuniformly-heated channel

and

$$F = \frac{c\int_0^{l_{DNB}} q''(z)\exp[-c(l_{DNB}-z)]\,dz}{q''_{local}[1-\exp(-c\,l_{DNB})]}$$

where

c = 0.44 (1 - x_{DNB})$^{7.9}$ / (G/10^6)$^{1.72}$ in.$^{-1}$.

l_{DNB} = channel length at which DNB takes place, in.

q''_{local} = heat flux at l_{DNB}, Btu/hr-ft^2

z = variable distance from channel entrance, in.

x_{DNB} = quality at DNB.

The DNB ratio is calculated from

$$\text{DNBR} = q''_{\text{DNB,N}} \,/\, q''_{\text{local}} \;.$$

2.4.2 Treatment of Cross Section Feedback

The cross sections averaged over a volume element depend on different local core thermal-hydraulic parameters, such as moderator density, fuel temperature, and moderator temperature. Knowing the cross section values at the nominal point, the transient behavior of the cross section can then be represented as a function of the derivatives of thermal-hydraulic parameters from the nominal point cross section values. Feedback from the thermal-hydraulic equations to the neutronic equations is accomplished by assuming that all macroscopic cross sections vary linearly with moderator temperature, moderator density, and the square root of the fuel temperature. The dependence of the cross section on the thermal-hydraulic parameter is approximated by the inclusion of the first derivative of the cross section in the following manner:

$$\begin{aligned}\Sigma\,(\rho\,, T_c, T_f) &= \Sigma_0\,(\rho_0\,, T_{c0}\,, T_{f0}) \\ &\quad + \frac{\partial\Sigma}{\partial\rho}\,\Delta\rho \;+\; \frac{\partial\Sigma}{\partial T_c}\,\Delta T_c \;+\; \frac{\partial\Sigma}{\partial T_f^{1/2}}\,\Delta T_f^{\frac{1}{2}} \;,\end{aligned} \qquad (2.4.5)$$

where

ρ_0, T_{co}, T_{f0} = nominal moderator density, moderator temperature, and fuel temperature

$\Delta\rho = \rho - \rho_0$ = change in density from nominal value

$\Delta T_c = T_c - T_{c0}$ = change in moderator temperature from nominal value

$\Delta T_f^{1/2} = T_f^{1/2} - T_{f0}^{1/2}$ = change in square root of fuel temperature from nominal value

$\partial\Sigma/\partial\rho$ = change in cross section with moderator density

$\partial\Sigma/\partial T_c$ = change in cross section with moderator temperature

$\partial\Sigma/\partial T_f^{1/2}$ = change in cross section with square root of fuel temperature.

Since our objective is to provide a simple but reasonably accurate means of feedback without precisely modeling the behavior of cross sections, the linear functional form of the cross sections is assumed to be valid over the entire range of thermal-hydraulic variables. Therefore, if the partial derivative of the macroscopic cross sections are known, the thermal-hydraulic feedback model can be completely specified. The cross section derivative terms in Eq.(2.4.5) are precalculated from cross sections and are included as part of the cross section data. The ($\Delta\rho$, ΔT_c, $\Delta T_f^{1/2}$) terms are derived from a heat balance calculation for each node and are updated during each feedback pass.

The variation of fuel temperature mainly affects the fast absorption, fission, and slowing down cross sections. Then, from Eq.(2.4.5), the neutron parameters can be represented as

$$
\begin{aligned}
D_1 &= D_{10} + \left(\frac{\partial D_1}{\partial \rho}\right) [\rho - \rho_0] + \left(\frac{\partial D_1}{\partial T_c}\right) [T_c - T_{c0}] + \Delta D_1 \\
\Sigma_{a1} &= \Sigma_{a10} + \left(\frac{\partial \Sigma_{a1}}{\partial \rho}\right) [\rho - \rho_0] + \left(\frac{\partial \Sigma_{a1}}{\partial T_c}\right) [T_c - T_{c0}] \\
&\quad + \left(\frac{\partial \Sigma_{a1}}{\partial T_f^{1/2}}\right) [(T_f + 459.72)^{\frac{1}{2}} \\
&\quad - (T_{f0} + 459.72)^{\frac{1}{2}}] + \Delta \Sigma_{a1} \\
\Sigma_{a2} &= \Sigma_{a20} + \left(\frac{\partial \Sigma_{a2}}{\partial \rho}\right) [\rho - \rho_0] + \left(\frac{\partial \Sigma_{a2}}{\partial T_c}\right) [T_c - T_{c0}] \\
&\quad + \Delta \Sigma_{a2} \\
\Sigma_{s12} &= \Sigma_{s120} + \left(\frac{\partial \Sigma_{s12}}{\partial \rho}\right) [\rho - \rho_0] + \left(\frac{\partial \Sigma_{s12}}{\partial T_c}\right) [T_c - T_{c0}] \\
&\quad + \left(\frac{\partial \Sigma_{s12}}{\partial T_f^{1/2}}\right) [(T_f + 459.72)^{\frac{1}{2}} \\
&\quad - (T_{f0} + 459.72)^{\frac{1}{2}}] + \Delta \Sigma_{s12} \\
\nu \Sigma_{f1} &= \nu \Sigma_{f10} + \left(\frac{\partial \nu \Sigma_{f1}}{\partial \rho}\right) [\rho - \rho_0] + \left(\frac{\partial \nu \Sigma_{f1}}{\partial T_c}\right) [T_c - T_{c0}] \\
&\quad + \left(\frac{\partial \nu \Sigma_{f1}}{\partial T_f^{1/2}}\right) [(T_f + 459.72)^{\frac{1}{2}} \\
&\quad - (T_{f0} + 459.72)^{\frac{1}{2}}] + \Delta \nu \Sigma_{f1} \\
\nu \Sigma_{f2} &= \nu \Sigma_{f20} + \left(\frac{\partial \nu \Sigma_{f2}}{\partial \rho}\right) [\rho - \rho_0] + \left(\frac{\partial \nu \Sigma_{f2}}{\partial T_c}\right) [T_c - T_{c0}] \\
&\quad + \Delta \nu \Sigma_{f2}
\end{aligned}
\tag{2.4.6}
$$

where the last term on the right side of each expression is the specified variation of the neutron parameters that is initiated by human control intervention.

2.5 Results

The spatially- and temporally-discretized two-group finite difference diffusion equations developed in Section 2.2, the TLAR acceleration method derived in Section 2.3, and the thermal-hydraulic model detailed in Section 2.4 are incorporated into a computer code, hereafter referred to as OSUTRAN (Oregon State University Transient Code). This computer code solves two- and three-dimensional, two-group diffusion theory problems. The OSUTRAN program uses Gauss-Seidel iteration for the inner iteration and fission source overrelaxation method for the outer iteration.

OSUTRAN is written in Lahey EM/32 FORTRAN77 compiler. All computations reported in this study were performed on a COMPAQ DESKPRO 386/25 personal computer.

In order to test OSUTRAN for a transient problem, some perturbation of the critical configuration must be made to initiate a transient. This perturbation can take any of several forms.

Perhaps the simplest mechanism for including a transient in a critical reactor is to alter the position of the control rods. In the method described in Section 2.2, control rod motions are modeled as spatially-uniform changes in macroscopic cross sections within individual nodes. This restriction is necessary since the spatial coupling coefficients of Section 2.2 are obtained only for the case of uniform material properties within a node.

For the sake of simplicity, the cross sections for a partially-rodded node can be assumed to be the volume-weighted averages of the fully-rodded and fully-unrodded nodal cross sections. This approximation will be erroneous, unless the neutron flux is spatially flat within the node. In cases of where the flux is not spatially flat, the solution may result in over or under prediction of the differential control rod worth. Despite this shortcoming of the control rod model, it is incorporated into the transient neutronics model.

The thermal-hydraulic model discussed in Section 2.4 allows two additional mechanisms for inducing transients. Both the reactor coolant inlet temperature and the coolant flow rate can be altered as a function of time to induce a transient.

In this section the TLAR acceleration method is applied to a three-dimensional LWR benchmark problem and a superprompt critical transient problem. Appendix A presents the power calculation subroutine which uses the TLAR acceleration method. The results of benchmark test calculations are presented to demonstrate the overall computational efficiency of TLAR acceleration method in comparison with a conventional two-group scheme.

2.5.1 The 3-D LMW Transient Problem

The 3-D LMW (Langenbuch-Maurer-Werner) test problem [36] is a simplified LWR, shown in Figure 2.5.1. The problem is modeled with two neutron energy groups, six delayed precursor groups, and quarter-core symmetry. The reactor has two-zone core containing 77 fuel assemblies with widths of 20 cm. The core is reflected both radially and axially by 20 cm of water, and the active core height is 160 cm. Five control rods are parked in the upper axial reflector, and four control rods are inserted from the upper reflector to the axial midplane of the core.

The two-group diffusion theory parameters for each fuel assembly are

Reactor Horizontal Cross Section:

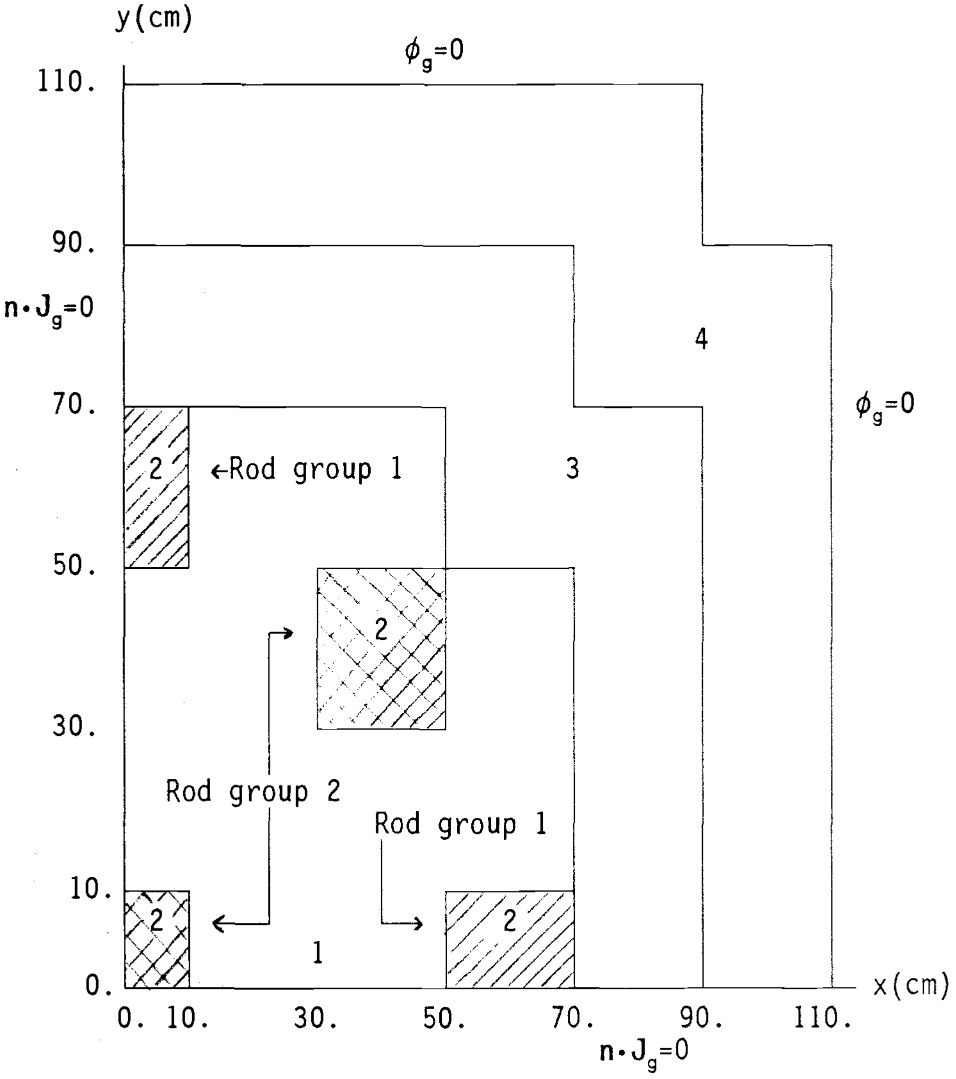


Figure 2.5.1 Geometry of the LMW Reactor

Vertical Cross Section for Initial Control Rod Positions:

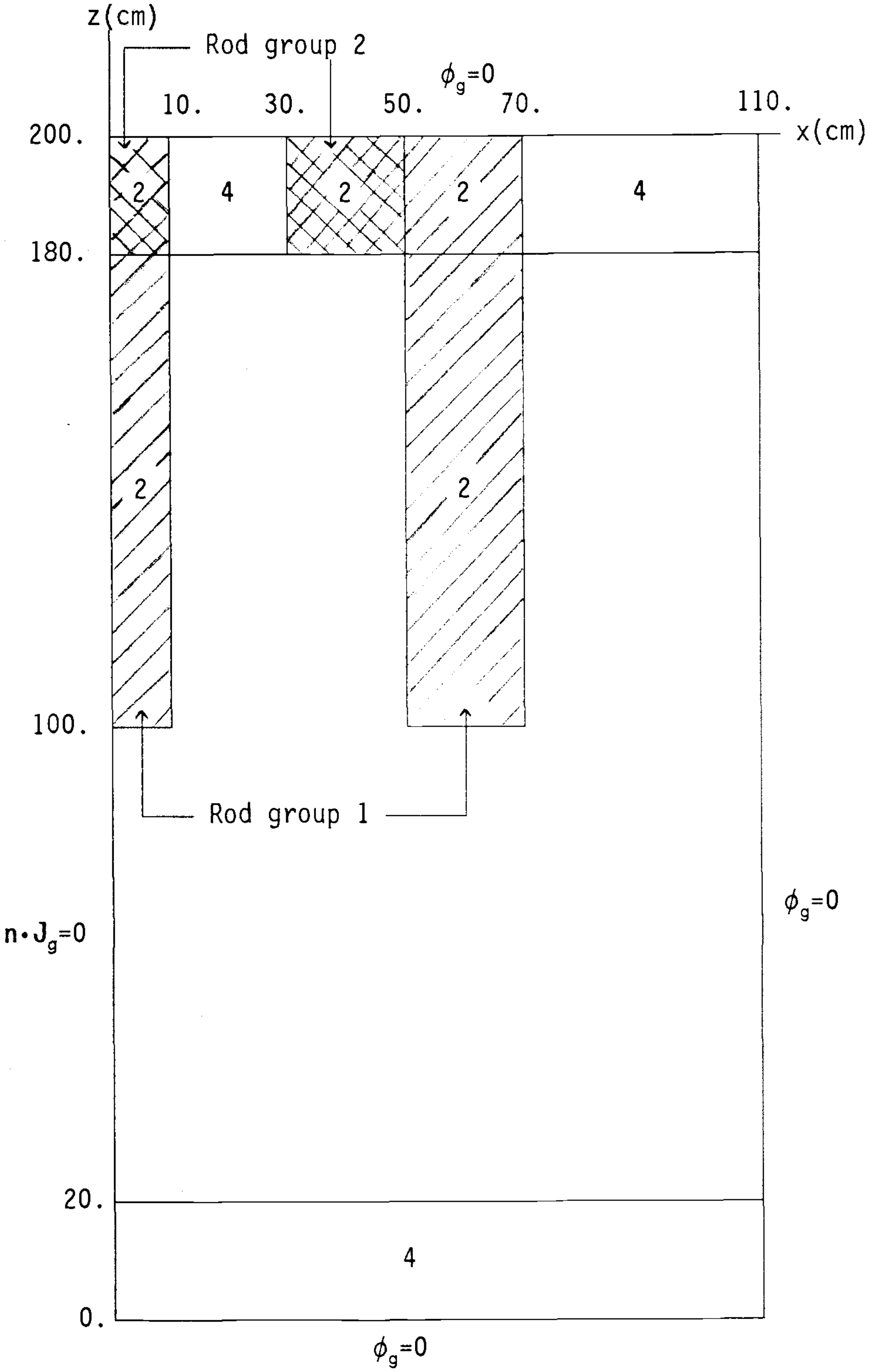


Figure 2.5.1 Geometry of the LMW Reactor (continued)

Vertical Cross Section for Final Control Rod Positions:

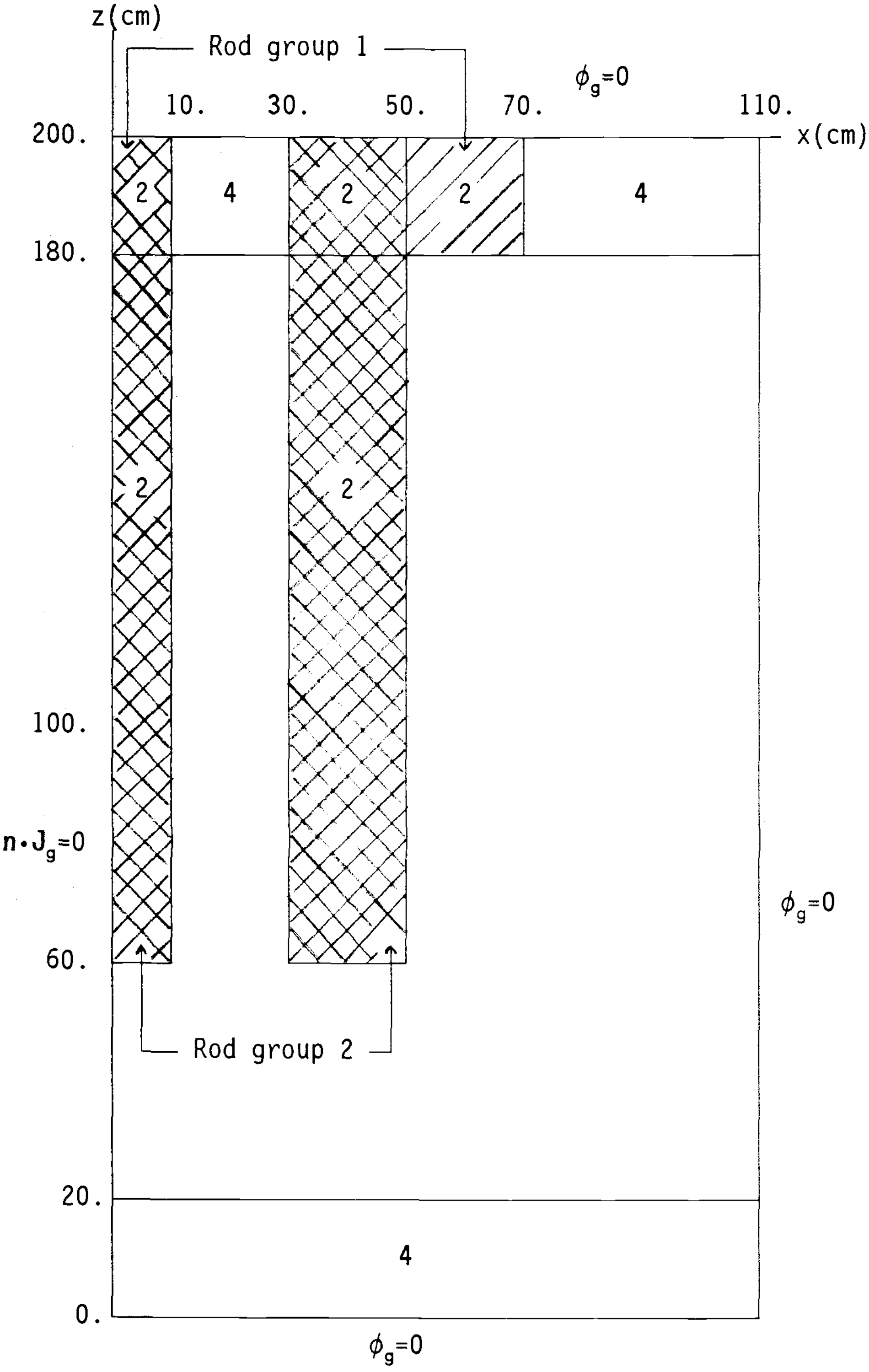


Figure 2.5.1 Geometry of the LMW Reactor (continued)

Table 2.5.1 Macroscopic Cross Sections and Other Input Data

Material Properties:

Mate-	Group	D_g	Σ_{ag}	$\Sigma_{sg'g}$	$\nu\Sigma_{fg}$
rial		(cm)	(cm^{-1})	(cm^{-1})	(cm^{-1})
1	1	1.423913	0.01040206	0.01755550	0.006477691
	2	0.356306	0.08766217	0.0	0.1127328
2	1	1.423913	0.01095206	0.01755550	0.006477691
	2	0.356306	0.09146217	0.0	0.1127328
3	1	1.425611	0.01099263	0.01717768	0.007503284
	2	0.350574	0.09925634	0.0	0.1378004
4	1	1.634227	0.002660573	0.02759693	0.0
	2	0.264002	0.04936351	0.0	0.0

ν = 2.5
v_1= 1.25 x 10^7 cm/sec
v_2= 2.5 x 10^5 cm/sec

Energy Conversion Factor:

3.204 x 10^{-11} W-sec/fission

Delayed Neutron Data:

Precursor group, d	β_d	$\lambda_d(sec^{-1})$
1	0.000247	0.0127
2	0.0013845	0.0317
3	0.001222	0.115
4	0.0026455	0.311
5	0.000832	1.40
6	0.000169	3.87

Perturbation:

Control rod group 1 removed at 3.0 cm/s, $0 \leq t \leq 26.666$ s

Control rod group 2 inserted at 3.0 cm/s, $7.5 \leq t \leq 47.5$ s summarized in Table 2.5.1. The perturbation information is also found in Table 2.5.1. The transient is initiated by withdrawing a bank of four partially-inserted control rods at a rate of 3 cm per second. Subsequently, a bank of five control rods (initially parked in the upper reflector) is inserted at a rate of 3 cm per second. The resulting transient is followed for 60 seconds.

Prior to examination of the actual transient solution accuracy and computational efficiency it is enlightening to investigate the sensitivity of the transient nodal solution to various parameters such as the number of inner iteration per outer iteration and convergence criteria.

Experience shows that the convergence criteria for inner iterations may not be too strict. If the number of inner iterations per outer iteration is smaller, the number of outer iterations is increased. However, there is no guideline for an optimal convergence criteria with which the overall computing time is minimized. In light of this experience, the general practice of performing one inner iteration per outer iteration has been adopted.

Six OSUTRAN solutions with and without the TLAR acceleration scheme to the LMW test problem are displayed in Table 2.5.2 through 2.5.4 along with the reference solution of CUBBOX [36] calculations. These calculations for quarter core geometry employed a convergence criterion of 10^{-5} for the outer iteration, mesh sizes of 10 cm for both radial and axial directions (2340 meshes), and time step sizes of 1.0, 0.5, and 0.125 seconds. The CUBBOX calculation uses mesh sizes of 10 cm for $0 \leq x \leq 10$ cm, $0 \leq y \leq 10$ cm in radial direction, and 20 cm elsewhere (350 meshes) and time step size $\Delta t = 0.125$ second.

Table 2.5.2 Core Power versus Time for the 3-D LMW Test Problem (Time Step Size Δt = 1.0 sec)

	Core Average Power Densities (% Error)[a]		
Time (sec)	With-Acceleration	Without-Acceleration	Reference[b]
0.0	150.0 (0.0)	150.0 (0.0)	150.0
1.0	151.50 (-0.58)	151.42 (-0.64)	152.39
2.0	154.61 (-0.59)	154.51 (-0.66)	155.54
5.0	168.87 (0.47)	168.65 (-0.08)	168.79
10.0	201.48 (0.18)	201.06 (-0.02)	201.11
15.0	236.83 (-1.17)	236.14 (-1.46)	239.63
20.0	257.50 (-0.97)	256.60 (-1.32)	260.03
25.0	246.68 (-0.85)	246.38 (-0.97)	248.79
30.0	209.79 (-0.70)	209.68 (-0.75)	211.26
40.0	126.19 (0.58)	126.18 (0.57)	125.46
50.0	79.20 (2.76)	79.22 (2.79)	77.07
60.0	61.09 (3.64)	61.12 (3.70)	58.94
CPU[c]	11.8	170.3	-

[a] Average Power Density in W/cc
[b] Reference Solution with Time Step Size of 0.125 sec
[c] Computing Time in Minutes on COMPAQ DESKPRO 386/25 PC

Table 2.5.3 Core Power versus Time for the 3-D LMW Test Problem (Time Step Size Δt = 0.5 sec)

	Core Average Power Densities (% Error)[a]		
Time (sec)	With-Acceleration	Without-Acceleration	Reference[b]
0.0	150.0 (0.0)	150.0 (0.0)	150.0
1.0	152.15 (-0.15)	152.04 (-0.23)	152.39
2.0	156.01 (0.30)	155.80 (0.17)	155.54
5.0	169.68 (0.53)	169.14 (0.21)	168.79
10.0	202.25 (0.57)	201.52 (0.20)	201.11
15.0	237.63 (-0.83)	236.63 (-1.25)	239.63
20.0	258.34 (-0.65)	257.10 (-1.13)	260.03
25.0	247.42 (-0.55)	246.90 (-0.76)	248.79
30.0	210.39 (-0.41)	210.08 (-0.56)	211.26
40.0	126.51 (0.83)	126.41 (0.76)	125.46
50.0	79.40 (3.02)	79.34 (2.94)	77.07
60.0	61.25 (3.92)	61.22 (3.87)	58.94
CPU[c]	22.3	266.6	-

[a] Average Power Density in W/cc
[b] Reference Solution with Time Step Size of 0.125 sec
[c] Computing Time in Minutes on COMPAQ DESKPRO 386/25 PC

Table 2.5.4 Core Power versus Time for the 3-D LMW Test Problem (Time Step Size Δt = 0.125 sec)

	Core Average Power Densities (% Error)[a]		
Time (sec)	With-Acceleration	Without-Acceleration	Reference[b]
0.0	150.0 (0.0)	150.0 (0.0)	150.0
1.0	153.58 (0.78)	153.50 (0.73)	152.39
2.0	156.96 (0.91)	156.72 (0.76)	155.54
5.0	170.44 (0.98)	170.10 (0.78)	168.79
10.0	202.95 (0.91)	202.42 (0.65)	201.11
15.0	238.37 (-0.53)	237.57 (-0.86)	239.63
20.0	258.05 (-0.76)	258.04 (-0.76)	260.03
25.0	248.17 (-0.25)	247.79 (-0.40)	248.79
30.0	211.00 (-0.12)	210.78 (-0.23)	211.26
40.0	126.87 (1.12)	126.81 (1.08)	125.46
50.0	79.62 (3.31)	79.59 (3.27)	77.07
60.0	61.42 (4.21)	61.40 (4.17)	58.94
CPU[c]	75.7	625.0	-

[a] Average Power Density in W/cc

[b] Reference Solution with Time Step Size of 0.125 sec

[c] Computing Time in Minutes on COMPAQ DESKPRO 386/25 PC

If one compares the solutions using TLAR acceleration to the solutions without TLAR acceleration, the maximum discrepancies are less than 0.33%, 0.48%, and 0.39% for time step sizes $\Delta t = 0.125$, 0.5, and 1.0 second, respectively. The discrepancies are believed to come from the nature of the "iterative" solution method. Even for the same convergence criterion (10^{-5}) the solutions which use TLAR acceleration have different histories of convergence from the solutions without TLAR acceleration. However, there is no reason to believe that one solution is superior to the other. If a smaller convergence criterion value is used, the discrepancies would be decreased further. This was considered to be unnecessary at this time.

When compared to CUBBOX solutions, the errors of the 1.0 and 0.5 second time step OSUTRAN solutions (with or without TLAR acceleration) are slightly less than those of the 0.125 second time step solutions (with or without TLAR acceleration). However, the 1.0 and 0.5 second time step solutions are not better temporally-converged than the 0.125 second time step solution. Hence, in order to compare the accuracy of the OSUTRAN solution method, the 0.125 second time step calculation is to be compared to a CUBBOX reference solution. These results indicate that the OSUTRAN solution is quite accurate during most of transient and has a maximum error in mean power density of about 4.2% at the end of transient. The reason for the apparent discrepancy seems to come mainly from control rod cusping effects. OSUTRAN models the cross sections for a partially-rodded nodes as the volume averages of the partially-rodded and partially-unrodded nodal cross sections whereas CUBBOX can model more precisely the partially-inserted control rods.

Comparisons between the solutions with and without TLAR acceleration show that when TLAR acceleration is employed, the computing time of the conventional two-group scheme can be reduced eight to fourteen times without sacrificing the computational accuracy. The TLAR acceleration scheme can be applied to not only the finite-difference method which is used for OSUTRAN but for various nodal methods in an attempt to reduce computing time.

2.5.2 Superprompt Critical Transient Problem

A superprompt critical transient from low power is induced by the rapid withdrawal of an asymmetric peripheral control rod. This problem is a quarter-core BWR kinetics problem with two neutron energy groups and two delayed neutron precursor families. The horizontal cross sections of a reactor quadrant is shown in Figure 2.5.2. The reactor has four-zone core containing fuel assemblies with widths of 15 cm. The core is reflected both radially and axially by 30 cm of water, and the active core height is 300 cm.

The transient is initiated by withdrawing control rod (R) as shown in Figure 2.5.2 at a rate of 150 cm per second. The resulting transient is followed for 3 seconds.

The feedback model is specified fully by two relations:

1. adiabatic heatup

$$\omega\,[\Sigma_{f1}(\vec{r},t)\phi_1(\vec{r},t) + \Sigma_{f2}(\vec{r},t)\phi_2(\vec{r},t)\,] = \frac{\partial}{\partial t}T(\vec{r},t)$$

Reactor Horizontal Cross Section:

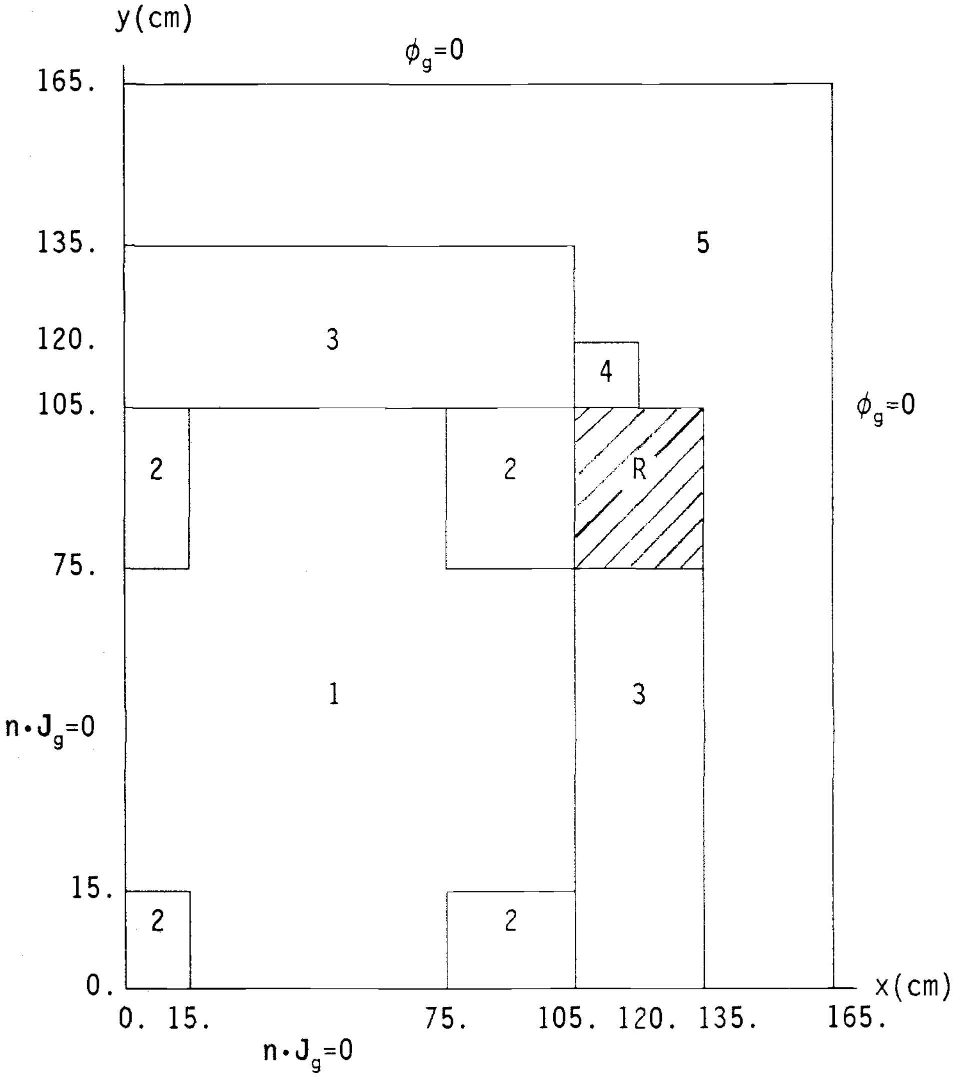


Figure 2.5.2 Geometry of the Superprompt Critical Transient Problem

2. Doppler feedback

$$\Sigma_{a1}(\vec{r},t) = \Sigma_{a1}(\vec{r},0)\,[\,1+\gamma(\sqrt{T(\vec{r},t)} \;-\sqrt{T_0}\,)]$$

where $T(\bar{r},t)$ is the fuel temperature, and ω, γ, and T_0 are known constants. A complete problem description is contained in Table 2.5.5.

Two OSUTRAN solutions with and without the TLAR acceleration scheme to this problem are displayed in Table 2.5.6. A plot of total reactor power density as a function of time (solutions with the TLAR acceleration scheme), is shown in Figure 2.5.3. These calculations employed a convergence criterion of 10^{-5} for the outer iteration, mesh sizes of 15 cm for radial and 30 cm for axial directions, respectively. Variable time step sizes have been employed to this problem. The time step sizes are defined as follows:

Time Interval	Time Step Size
$0.0 \leq t \leq 0.5$ s	25 ms
$0.5 \leq t \leq 0.6$ s	10 ms
$0.6 \leq t \leq 0.7$ s	2.5 ms
$0.7 \leq t \leq 0.8$ s	1.25 ms
$0.8 \leq t \leq 0.95$ s	1.0 ms
$0.95 \leq t \leq 1.0$ s	2.5 ms
$1.0 \leq t \leq 2.0$ s	20 ms
$2.0 \leq t \leq 3.0$ s	25 ms.

If one compares the solutions using TLAR acceleration to the solutions without TLAR acceleration, the maximum discrepancy is not greater than 0.38% for power peaks and minimum. The discrepancies for average and peak fuel temperature are less than 0.1%. These results indicate that when TLAR acceleration is employed, the computing time of the conventional two-group scheme can be reduced 9.3 times while maintaining computational accuracy.

Table 2.5.5 Two-Group Constants and Other Input Data

Material Properties:

Material	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\Sigma_{sg'g}$ (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})
1	1	1.255	0.008252	0.02533	0.004602
	2	0.211	0.1003	0.0	0.1091
2	1	1.268	0.007181	0.02767	0.004609
	2	0.1902	0.07047	0.0	0.08675
3	1	1.259	0.008002	0.02617	0.004663
	2	0.2091	0.08344	0.0	0.1021
4	1	1.259	0.008002	0.02617	0.004663
	2	0.2091	0.073324	0.0	0.1021
5	1	1.257	0.0006034	0.04754	0.0
	2	0.1592	0.01911	0.0	0.0

$\nu = 2.43$
$v_1 = 3.0 \times 10^7$ cm/sec
$v_2 = 3.0 \times 10^5$ cm/sec

Energy Conversion Factor:

3.204×10^{-11} W-sec/fission

Delayed Neutron Data:

Precursor group, d	β_d	$\lambda_d(sec^{-1})$
1	0.0054	0.0654
2	0.001087	1.35

Adiabatic Feedback Data:

$\omega = 3.83 \times 10^{-11}$ °K cm^3
$\gamma = 2.8 \times 10^{-3}$ $°K^{-1/2}$
$T_0 = 300$ °K

Transient Initial Conditions:

Mean power density at t = 0, 10^{-6} W/cc
Fuel temperature at t = 0, 300 °K

Perturbation:

Control rod (R) removed at 150 cm/s, $0 \leq t \leq 2.0$ s

Table 2.5.6 Summary of Results for the Superprompt Critical Transient Problem

	With-Acceleration	Without-Acceleration	Relative Error (%)[a]
Time to first peak (s)	0.849	0.849	.
Power at first peak (W/cc)	5736.5	5730.2	0.11
Time to first minimum (s)	0.939	0.939	.
Power at first minimum (W/cc)	103.2	103.4	-0.19
Time to second peak (s)	1.400	1.400	.
Power at second peak (W/cc)	291.1	290.0	0.38
Power at t = 3.0 s (W/cc)	69.1	69.2	-0.14
Average fuel temperature at t = 3.0 s (°K)	951.7	952.1	0.04
Peak assembly fuel temperature at t = 3.0 s (°K)	4071.4	4074.4	0.07
CPU[b]	66.5	618.1	.

[a] Relative Error Between the Solutions with and without TLAR acceleration $\left(= \frac{\text{with acceleration - without acceleration}}{\text{without acceleration}}\right)$

[b] Computing Time in minutes on COMPAQ DESKPRO 386/25 PC

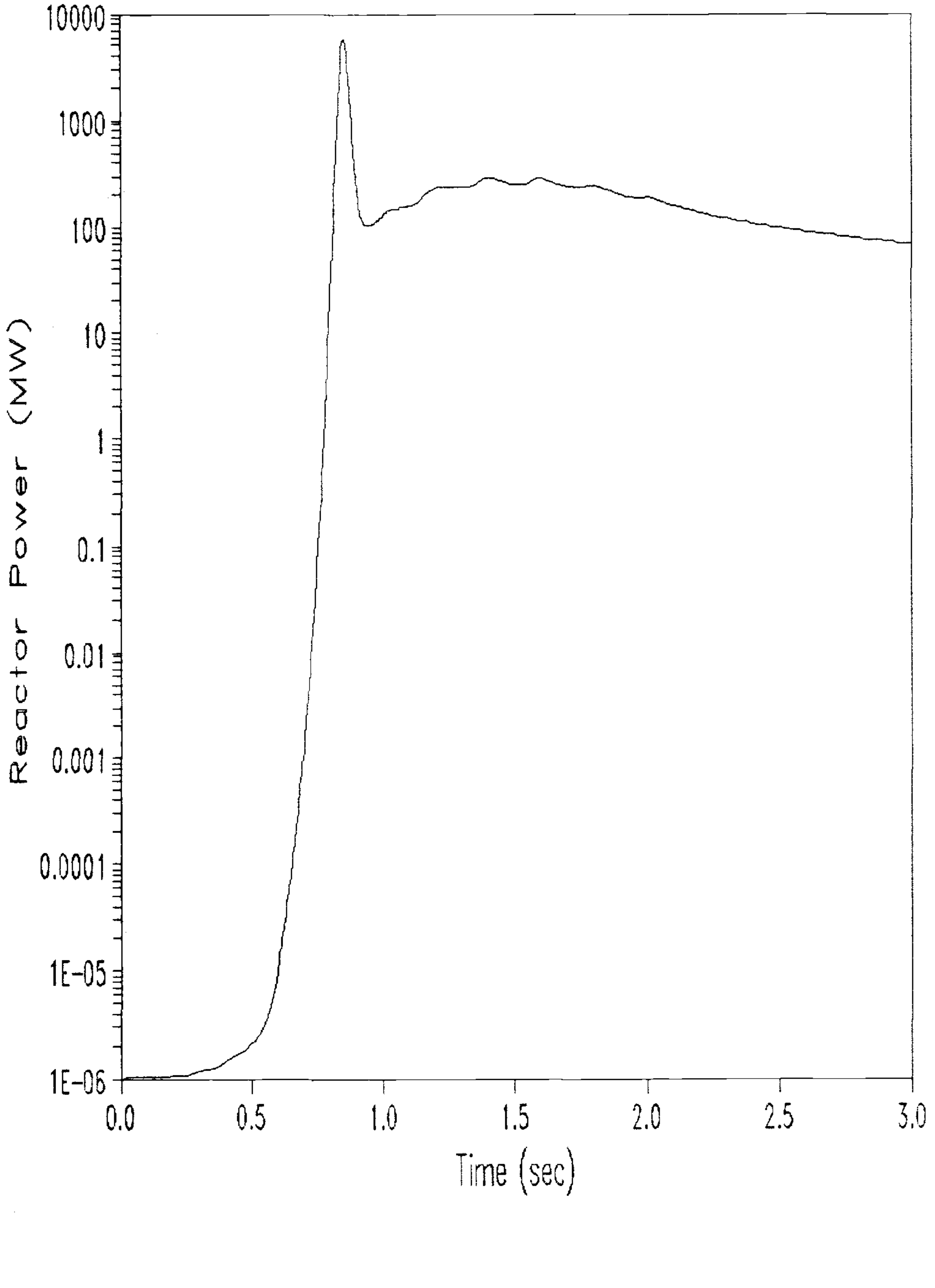


Figure 2.5.3 Total Core Power versus Time for the Superprompt Critical Transient Problem

2.6 Conclusions

The objective of this study was to develop an accurate and computationally efficient method for solving the time-dependent, multi-dimensional, two-group neutron diffusion equations with feedback. Comparisons with conventional two-group scheme indicate that the TLAR acceleration scheme is approximately one order of magnitude more computationally efficient in transient applications.

Thus, the TLAR acceleration scheme has been demonstrated to be highly efficient for multi-dimensional, two-group, transient LWR analysis. This scheme is an appealing solution tool for various nodal methods currently used in design and analysis of LWRs.

2.6.1 Recommendations for Future Improvements

During the course of this research, many interesting topics have been left unsolved. This section contains a brief description of these potential research areas.

1. In order to reduce the prediction error of a power distribution which, in particular, comes from low order difference approximations for the leakage term, a more accurate nodal scheme such as the "discontinuity factor method" would be advised. In this case discontinuity factors for each node must be obtained for each time step.
2. If the validity of TLAR acceleration scheme is proved, the TLAR acceleration scheme will be computationally more efficient for multi-group problems than two-group problems. For multi-group problems there is a need to investigate the

scope of all thermal groups to which the TLAR acceleration scheme can be effectively applied.

3 The control rod cusping problem requires additional work. Representing the space-dependence of the cross sections within a node seems overwhelmingly complicated. If one determines the equivalent homogenized parameters which properly predict control rod effects in partially-rodded nodes, the rod cusping problem could be solved without employing space-dependent cross sections.

Chapter References

1. J. J. Duderstadt and L. J. Hamilton, Nuclear Reactor Analysis, John Wiley & Sons, Inc., (1976).

2. K. Koebke, "Advances in Homogenization and Dehomogenization," International Tropical Meeting on Advances in Mathematical Method for the Solution of Nuclear Engineering Problems, Vol. 2, p. 59, Munich, 27-29 April (1981).

3. K. S. Smith, A. F. Henry, and R.A.Loretz, "Determination of Homogenized Diffusion Theory Parameters for Coarse Mesh Nodal Analysis," Proc. ANS Topl. Mtg., Advances in Reactor Physics and Shielding., Sun Valley, ID, September 14-19, 1980, p. 294, American Nuclear Society (1980).

4. M. Clark, Jr. and K. F. Hansen, Numerical Methods of Reactor Analysis, Academic Press, New York, N. Y. (1964).

5. R. Roechlich, "Current Problems in Multi-dimensional Reactor Calculations," Proc. Conf. Mathematical models and Computational Techniques for Analysis of Nuclear Systems, CONF-730414, Vol. 2, p. VII-1 (1973).

6. W. R. Cadwell, A. F. Henry, and A. J. Vigilotti, "WIGLE - A Program for the Solution of the Two-Group, Space-Time Diffusion Equations in Slab Geometry," WAPD-TM-416, Bettis Atomic Power Laboratory (1964).

7. J. B. Yasinsky, M. Natelson, and L. A. Hageman, "TWIGL - A Program to Solve the Two-Dimensional, Two-Group, Space-Time Neutron Diffusion Equations with Temperature Feedback," WAPD-TM-743, Bettis Atomic Power Laboratory (1968).

8. C. M. Kang and K. F. Hansen, "Finite Element Methods for Reactor Analysis," Nucl. Sci. Eng., 51, 456 (1973).

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

10. L. M. Grossman, J. P. Hennart, and D. Meade, "Finite Element Collocation Methods for Space-Time Reactor Dynamics," ANS Trans., 41, 311-312 (1982).

11. D. L. Delp, D. L. Fischer, J. M. Harriman, and M. J. Stedwell, "FLARE: A Three-Dimensional Boiling Water Reactor Simulator," GEAP-4598, General Electric Company (1964).

12. A. Ancona, M. Becker, D. R. Harris, A. DaC. Menezes, M. A. Robinson, and D. M. Ver Plank, "Unified Analysis of Nodal Methods," Trans. Am. Nucl. Soc., 24, 444 (1976).

13. R. A. Shober, R. N. Sims, and A. F. Henry, "Two Nodal Methods for Solving Time-Dependent Group Diffusion Equations," Nucl. Sci. Eng., 64, 582-592 (1977).

14. Y. A. Chao and J. A. Penkrot, "A nodal Method for Direct Intra-Node Distribution Calculation," Proc. Topl. Mtg., Advances in Reactor Physics, NUREG/CP-0034 (1982).

15. H. L. Dodds, Jr., "Accuracy of the Quasistatic Method for Two-Dimensional Thermal Reactor transient with Feedback," Nucl. Sci. Eng., 59, 271-281 (1976).

16. J. Devooght and E. Mund, "Generalized Quasi-static method for Nuclear Reactor Space-Time Kinetics," Nucl. Sci. Eng., 76, 10-17 (1980).

17. C. D. Wu and J. Weisman, "An Efficient Computational Technique for Light Water Reactor Core Dynamics," Nuclear Technology, 81 June (1988).

18. S. Kaplan, "Synthesis Methods in Reactor Analysis," Advan. Nucl. Sci. Technology, 3, 233 (1965).

19. R. Goldstein and L. M. Shotkin, "Use of the Prompt-Jump Approximation in Fast-Reactor Kinetics," Nucl. Sci. Eng., 38, 94-103 (1969).

20. T. Blenski, A. Gadomski, and J. Mika, "Higher Order Prompt-Jump Approximation in Reactor Kinetics," Nucl. Sci. Eng., 66, 277-283 (1978).

21. D. W. Peaceman and H. H. Rachford, Jr., "The Numerical Solution of Parabolic Differential Equations," J. Soc. Ind. Appl. Math., 3, 28 (1955).

22. A. L. Wright, K. F. Hansen, and D. R. Ferguson, "Application of Alternating Direction Implicit Methods to Space-Dependent Kinetics Equations," Nucl. Sci. Eng.,44, 239 (1971).

23. W. H. Reed and K. F. Hansen, "Alternating Direction Methods for Reactor Kinetics Equations," Nucl. Sci. Eng., 41, 431 (1970).

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

25. G. P. Bottoni, "Fast Factorization Procedure Solving the Multi-dimensional Reactor Dynamics Equations," ANS Trans., 26, 229-230 (1977).

26. W. Werner, "Mathematical Methods Applicable to Space Time Kinetics," Proc. Specialist Meeting Reactivity Effects in Large Power Reactors, Ispra, Italy, October 28-30, 1970, Commission of the European Communities, EUR 4731 f-e, Luxembourg (1972).

27. G. S. Chen and J. M. Christenson, "Solution of the Two-Dimensional Space-Time Reactor Kinetics Equation by a Locally One-Dimensional Method," ANS Trans., 50, 536-537 (1985).

28. J. B. Yasinsky and A. F. Henry, "Some Numerical Experiments Concerning Space-time Reactor Kinetics Behavior," Nucl. Sci. Eng., 22, 171 (1965).

29. K. F. Hansen and S. R. Johnson, "GAKIN - A Program for the Solution of the One-dimensional, Multigroup, Space-time Dependent Diffusion Equations," USAEC Rpt. GA-7543 (1967).

30. S. Borresen, "A Simplified, Coarse-mesh, Three Dimensional Diffusion Scheme for Calculating the Gross-Power Distribution in a Boiling Water Reactor," Nucl. Sci. Eng., 44, 37 (1971).

31. J. Chang and C. H. Kim, "A Modified Borresen's Coarse-Mesh Method for a Spatial Neutron Kinetics Problem of Light Water Reactor," NUREG/CP-0080, Vol. 1 (1986).

32. C. H. Kim and S. H. Levine, "A Modified Borresen's Coarse-Mesh Computation for a Three-Dimensional Pressurized Water Reactor Benchmark Problem," Nucl. Tech., 61, 49 (1983).

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

34. Bankoff, S. G., "A Variable Density Single-Fluid Model for Two-Phase Flow with Particular Reference to Steam-Water Flow," J. Heat Transfer, 82, 265 (1960).

35. Tong, L. S., "Prediction of Departure from Nuclear Boiling for an Axially Non-Uniform Heat Flux Distribution," J. Nuclear Energy, Part A and B, 21, 241 (1967).

36. S. Langenbuch, W. Maurer, and W. Werner, "Coarse-Mesh Nodal diffusion method for the Analysis of Space-Time Effects in Large Light Water Reactors," Nucl. Sci. Eng. 63, 437-456 (1977).

Chapter 3

AUTOMATED CONTROL ROD PROGRAMMING IN BOILING WATER REACTOR USING SEARCH TECHNIQUE

3.1 Introduction

Nuclear fuel management has been defined as the collection of practices and principles required for the planning, scheduling, refueling, and safe operations of nuclear power plants while seeking to minimize total plant and system costs through the timely procurement of nuclear fuel and related services. It can be divided into two major subareas: out-of-core management and in-core management. Out-of-core fuel management focuses major efforts on contracting and purchasing services such as conversion, enrichment, fabrication, and spent fuel disposal. On the other hand, in-core fuel management attempts to optimize nuclear fuel utilization within the reactor core in order to meet the required licensing and operational constraints and still maintain an economic advantage.

In the case of boiling water reactor (BWR) in-core fuel management, one of the most important tasks is to determine the control rod patterns that maximize the cycle length under various operational constraints. A BWR is equipped with 100 to 200 control rods that are gradually withdrawn from the core during operation to compensate for reactivity variations. Control rod patterns which include the depth distributions of these rods determines the power and burnup distribution in the core. The power and the burnup distribution control is one of the major concerns due to the economic and safety requirements of reactors. If it is possible to attain higher burnups for the fuels discharged from the core by means of a control rod programming, considerable reduction of power generation cost can be achieved in a nuclear power plant. The control rod insertion/withdrawal policy is determined to keep the actual power shape as close to the target distribution as possible. The target power distribution is uniquely determined to yield the maximum cycle energy with a given fuel loading pattern.

Various techniques have been devised and applied to solve the problems of power distribution control. In early years, control rod programs were generated by trial-and-error methods. Programs were repeatedly set up and tested using a core burnup simulator such as FLARE [1] until a core-management engineer found the results satisfactory. Such an empirical approach required lots of manpower and computing time, and sometimes produced solutions which were far from an optimum.

Many researchers have been trying to apply optimization techniques to this problem. Terney and Fenech [2] solved the problem of determining an optimal sequence of control rod patterns for a two-region cylindrical core by using dynamic programming. Wade and Terney [3] modeled the design and operation of a nuclear reactor as an optimal control problem by applying the maximum principle of a distributed parameter system to a one-dimensional core. Motoda and Kawai [4] introduced a concept of phase-space analysis in which general relationships among control rods, power distribution, burnup, and cycle length were given geometrical meaning for a two-region core. Snyder and Lewis [5] applied dynamic programming to solve a one-dimensional axial control problem of the BWR for various cost functions. Motoda [6,7] applied a method of approximation programming

(MAP) [8] to a one-dimensional multiregion slab reactor to optimize the control rod density and the fuel loading pattern simultaneously. Since these papers commonly simplify the core model, the degrees of control rod freedom, and the variations of rod density, it is difficult to obtain a control rod pattern for an actual three-dimensional core on the basis of these analyses.

It is essential that a three-dimensional model is used as a tool for core-management engineers. In recent years, a complete three-dimensional core model has been adopted for control rod programming. Various formal optimization procedures have been applied to this problem. Kawai et al. [9] applied the MAP technique for generation of a long-term control rod program algorithm in which the optimization problem is decomposed into multiple intermediate-term programs. Hayase and Motoda [10] improved the performance of the Kawai et al.'s work by implementing heuristics that were derived from accumulated experience. In this heuristic method the control rods are always split into two deep and shallow subgroups, and the rods within each subgroup are moved in the same direction and over the same distance. Tokumasu et al. [11] developed a mathematical programming method as an alternative to the MAP. This technique is constructed of a dual feasible direction (DFD) algorithm [12]. The DFD method determines the direction of control rod movement at the approximate solution to obtain a better approximated solution.

This chapter presents an efficient search technique for the generation of optimum control rod patterns that can be applied to the actual core-management of a BWR. In Section 3.2, the background of the optimum control rod positioning problem is presented. The use of the steepest-ascent hill climbing search technique as a solution method is discussed in Section 3.3. Results of test problems are included in Section 3.4. Finally, a summary of this investigation and recommendations for future research are given in Section 3.5.

3.2 Formalization for Optimum Control Rod Positioning Problem

3.2.1 Control Rod Programming

Control rod programming is a general term used for techniques in which the amount of control rod withdrawal for actual operation is determined. This program is found by considering both changes in nuclear and thermohydraulic characteristics of the core following fuel burnup and maximization of the cycle length. Control rod programming is very simple in a pressurized water reactor since the reactor usually operates with all rods essentially fully withdrawn. This is not the case in a BWR since chemical shim control is not used; therefore a substantial number of control rods are present in the core for most of core life.

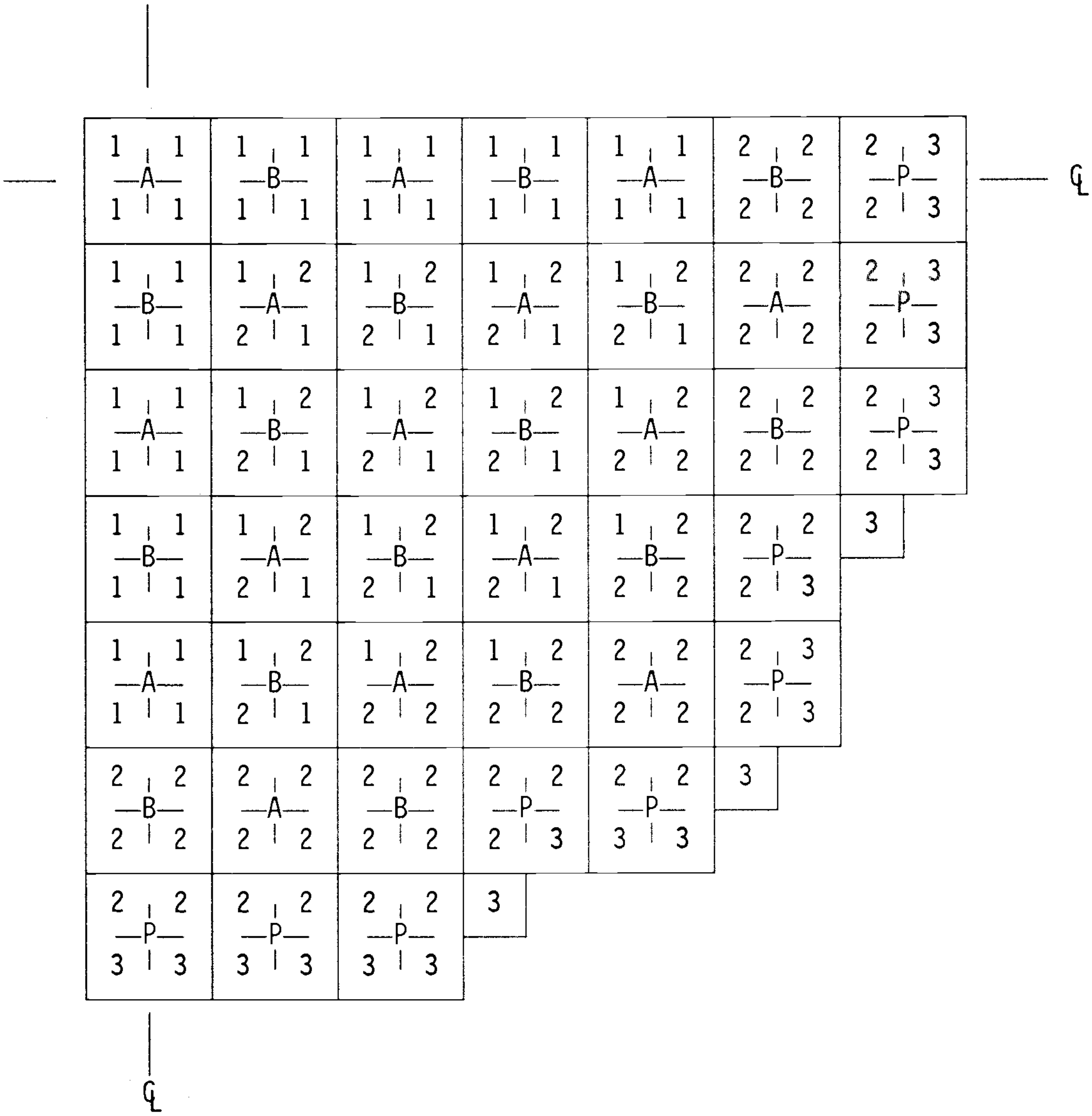
The fuel loading pattern of the first cycle of the Edwin Hatch Unit #2 reactor [13] was used as a reference for the control rod programming. Its pertinent information can be found in Table 3.2.1. It consists of 548 fuel bundles and 137 control rods.

Table 3.2.1 Design data for Edwin I. Hatch Nuclear Plant Unit #2

Nominal thermal output	2436	MW
Incore coolant flow rate	$7.7\text{x}10^7$	lbm/hr
Nominal system pressure	1035.0	psia
Coolant saturation temperature	548.8	^{o}F
Average heat flux	145060.	Btu/hr-ft^2
Core inlet enthalpy	526.9	Btu/lbm
Active coolant flow area per assembly	15.82	in^2
Active fuel length	150.	in
Assembly lattice	8 x 8	
Maximum thermal output	13.4	kW/ft
Average thermal output	5.38	kW/ft
Number of fuel bundle	548	
Number of control rods	137	

The core configuration of the reference reactor is shown in Figure 3.2.1. The control rods, which are cruciform types, are designated by a '+' sign in Figure 3.2.1. In practice, about one-third of the control rods are placed in the periphery as P group rods and are usually withdrawn except during a nuclear scram. The remaining control rods are grouped into A and B rods (designated by A and B in Figure 3.2.1). When control rods in one group are partially inserted and are effectively in control, those in the other group are completely withdrawn. Thus, if the rods in group A are inserted, the reactor is said to be operating in the "A-pattern." The "B-pattern" is defined in the same way as the "A-pattern." The core is modeled as a one-fourth core with mirror symmetry.

In control rod programming, control rod patterns are examined by dividing one cycle into a set of periods. Figure 3.2.2 illustrates various kinds of control rod programs. At each of these periods, the sequence of control rod patterns is changed from A to B or B to A. This is done to flatten the exposure distribution.



Assembly Region	Average Enrichment w/o U-235
1	1.83
2	2.33
3	0.711

Control Rod Type	Description
A	A group rods
B	B group rods
P	Periphery rods

Figure 3.2.1 Control Rod Position and Fuel Loading Pattern

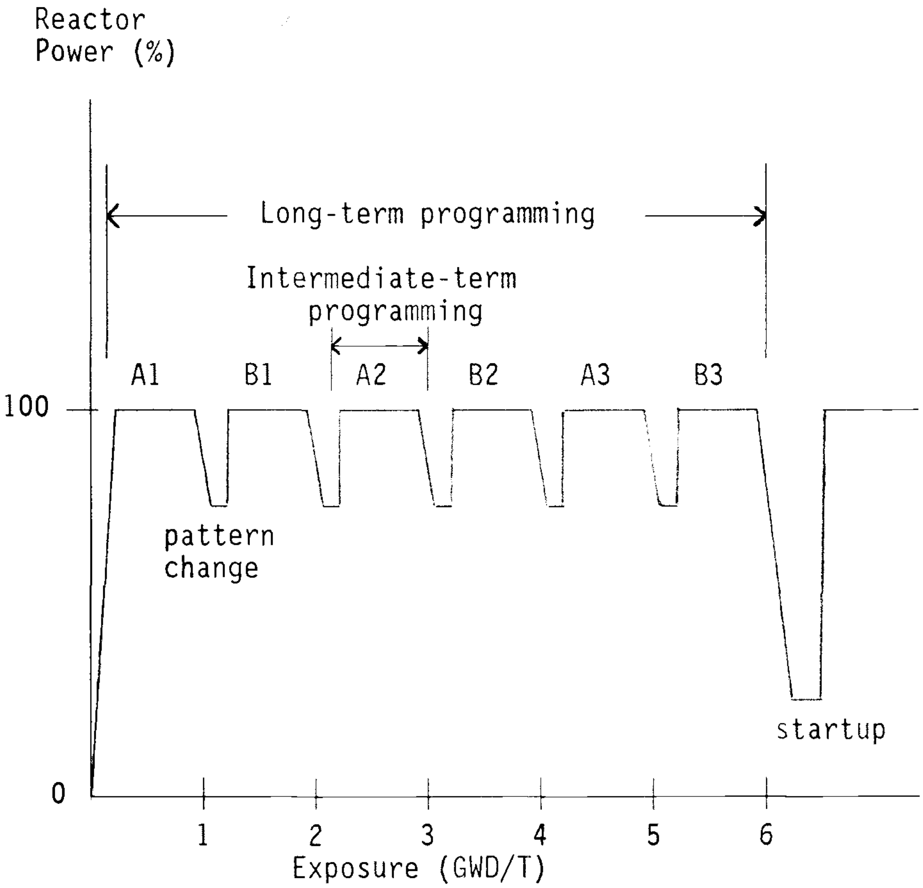
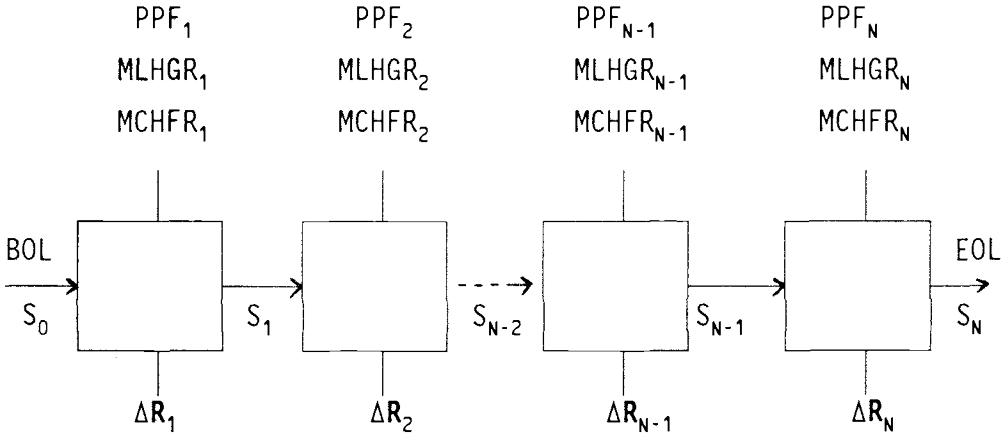


Figure 3.2.2 Classification of Control Rod Programming

During the period, the control rods are gradually withdrawn to compensate for reactivity depletion. Intermediate-term programming refers to the power distribution control over this time period. The intermediate pattern changes for one cycle are called long-term programming. The intermediate-term programming determines the actual amount of control rod withdrawal according to the operational program chosen for the long-term programming. The period for the intermediate-term programming is usually 1 to 3 months. For the long-term programming economy throughout the cycle must be considered.

In addition to the programs mentioned above, there are other kinds of rod programs for startup and pattern changes. All these short-term problems can be solved separately from the long-term problem by considering instantaneous operational constraints.

The depletion of a reactor and corresponding movement of control rods can be thought of as a multistage decision process in which the control rod is withdrawn at the end of each depletion period to make the reactor critical for the next step. The process may be visualized as in Figure 3.2.3.



PPF_N = Power peaking factor in the N^{th} stage

$MLHGR_N$ = Maximum linear heat generation ratio in the N^{th} stage

$MCHFR_N$ = Minimum critical heat flux ratio in the N^{th} stage

S_N = State at the end of the N^{th} stage

ΔR_N = Control rod motion for criticality in the N^{th} stage

Figure 3.2.3 Control Rod Programming Stages

The process begins at the beginning-of-life (BOL), at which time the state of the reactor is S_0. Then poison is removed, denoted by ΔR_1, which results in rod motions, in order to bring the reactor to criticality. In the critical configuration, flux, power, maximum linear heat generation rate (MLHGR) and minimum critical heat flux ratio (MCHFR) calculations can be made and the core reactivity depleted to the stage S_1. Then control rods are removed again (ΔR_2) to return the reactor to criticality and the procedure is repeated to the end-of-life (ELO). The problem is to choose the optimum sequence ΔR_1, ΔR_2, $\cdots$, ΔR_n, satisfying the constraints, which maximize the cycle length.

3.2.2 Optimization

The problem of optimization of the control rod pattern is formulated in order to find values of control variables such as rod positions that maximize the cycle length within the given constraints. The cycle length is a function of both state and control variables. The interrelations between them are determined by the system equations. Mathematically, the optimum control simply means that an objective function is minimized or maximized. Control rod pattern optimization is often performed using core performance indices that indirectly represent the power costs. The performance index may vary depending on the characteristics of the problem. The cycle length is the figure-of-merit [14] which is closely related to performance index for the control rod programming length. Haling [15] demonstrated that the cycle length could be extended if the actual power distribution was kept as close to the target distribution as possible. Thus the average of the squared difference sum between the actual power distribution, denoted by P, and the target power distribution, denoted by H, is chosen as the objective function in order to evaluate the effectiveness of the control rod pattern. The optimization problem is formulated as

$$\text{minimize} \quad J(\mathbf{R}) = \sum_{ijk} (P_{ijk} - H_{ijk})^2 / L \ , \qquad (3.2.1)$$

subject to nuclear and thermal-hydraulic constraints, where J(**R**) is the objective function of control rod pattern **R**, L is the total number of reactor nodes, and i, j, and k designate node numbers for x, y, and z axes, respectively.

3.2.3 Constraints

The most important task of the fuel cycle engineer is probably the evaluation of the power production capability of a reactor during each operating cycle. The evaluation of the power production capability requires the integration of many technical disciplines. A determination of acceptable fuel operating limits requires a knowledge of fuel failure mechanisms and the operating conditions that can lead to fuel failure. The relationship between these mechanisms and the primary system operating conditions is evaluated using the thermal and hydraulic characteristics of the reactor, as well as the core neutronics characteristics.

All light water reactors have specified operating limits that are established to assure integrity of the first fission product barrier, i.e., the fuel and its cladding. There are generally two operating constraints of importance to BWR's: nuclear and thermal-hydraulics constraints.

3.2.3.1 Nuclear Constraints

A reactor must be critical during steady-state operation:

$$\lambda_{target} - \varepsilon \leq \lambda(R) \leq \lambda_{target} + \varepsilon \qquad (3.2.2)$$

where

λ = effective multiplication factor,

ε = tolerance factor.

3.2.3.2 Thermal-hydraulic Constraints

Many fuel operating limits are directly related to the maximum linear power density of the fuel rod. For a given fuel rod diameter, the peak fuel temperature, the surface heat flux, the decay heat generation rate, and the stored thermal energy are proportional or approximately proportional to linear heat generation rate of the fuel rod. The linear heat generation rate (LHGR) is a direct measure of the thermal performance of a fuel rod. It must be less than a prescribed maximum linear heat generation rate (MLHGR) at all locations:

$$\mathrm{LHGR}(\mathbf{R}) \leq \mathrm{MLHGR}_{\mathrm{target}}. \tag{3.2.3}$$

A second operating limit in a BWR is the surface heat flux. If a certain value is exceeded, the fuel rod surface is enveloped in a steam blanket, the surface heat transfer coefficient is degraded, and the clad temperature increases significantly above its normal operating value. This value of heat flux is called the "critical heat flux" and it is a complex function of fuel geometry, coolant pressure, coolant heat content, coolant flow rate, and power distribution. The critical heat flux ratio (CHFR) is defined as the ratio of critical heat flux to actual heat flux. The CHFR must be greater than a prescribed minimum critical heat flux ratio (MCHFR) value at all locations:

$$\mathrm{CHFR}(\mathbf{R}) \geq \mathrm{MCHFR}_{\mathrm{target}}. \tag{3.2.4}$$

The above constraints are usually part of the technical specifications for a plant, and are therefore not subject to change without an amendment to the operating license.

3.2.3.3 Control Rod Movement Constraints

In addition to the operating limits described above, another constraint frequently occurs when some of control rods are stuck;

stuck rod position: fixed.

3.2.4 Control Variables in System Equation

The system equations for control rod programming are determined by any three-dimensional BWR simulator which has a power-void iteration feature and which gives the power distribution, LHGR, and CHFR of each fuel rod corresponding to a given control rod pattern. The core state is determined by the flow rate, heat content, and pressure of the coolant, as well as the control rod pattern. However, only the control rod pattern is chosen as the control variable for the control rod program because the other variables are always fixed at their prescribed design values during normal operation.

3.2.5 Space-Time Decomposition

The number of control rods which are inserted during steady-state operation is 49 for both the A-pattern and B-pattern in the reference reactor. It is reasonable to postulate that one-eighth core symmetry is preserved during operation. In such a case the number of rods in a quarter core reduces to 10 for the A-pattern or 8 for the B-pattern. Normally the average burnup in the cycle is 6 GWD/T, and the control rod pattern is changed at every 1 GWD/T interval. This requires about six different control rod patterns per cycle.

If the control rod pattern is solved for a given cycle, the number of unknown variables (the position values for each of the rods) is 54 (3 A-patterns per cycle times 10 rods per A-pattern plus 3 B-patterns per cycle times 8 rods per B-pattern). If we consider such a large number of unknown variables altogether, the system becomes very complex. A small amount of control rod movement affects criticality and power distribution of the core at present and future times. To relieve this complexity, the space-time problem is composed of two stages: outer loop optimization and inner loop optimization, as shown in Figure 3.2.4. The outer loop optimization modifies the time-invariant target power distribution to minimize the control rods which are left in the core at the end of cycle. The inner loop optimization determines the control rod positions which minimize the mean square error of the power distribution from the target power distribution at each burnup step. The inner loop optimization in which time dependence is eliminated reduces the overall problem to six small control rod pattern problems in which solutions are computed for three A-patterns and three B-patterns. Thus, in this study, emphasis will be placed on the inner loop optimization method.

3.2.6 Target Power Distribution

A very useful and widely applied principle of in-core fuel management and control management is the Haling principle [15]. The Haling principle states that there is a particular time-invariant power shape and control strategy to realize the consistent power and burnup distributions at the end of cycle. This power and burnup distribution is called Haling distribution. In an actual BWR core it is impossible to realize the Haling distribution, since the control rods are inserted from

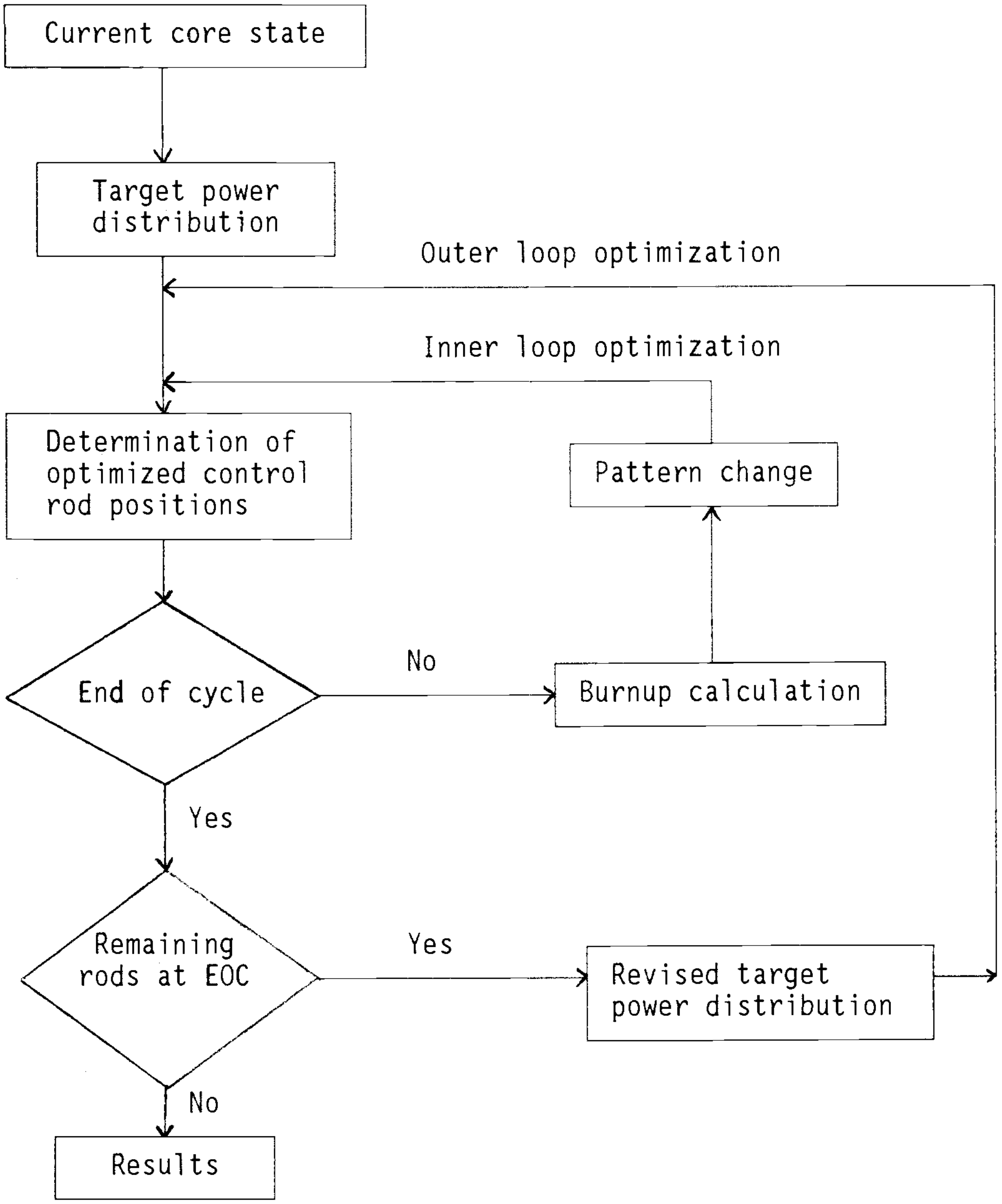


Figure 3.2.4 Flow Chart of Control Rod Programming

the bottom of the core and there is no complete freedom in a three-dimensional sense to distribute any amount of negative reactivity at any place. It has also been revealed [4,7] that the Haling principle is not always optimum but is usually near optimum. The Haling solution can, however, be used as a target power distribution when planning control rod patterning strategy.

The Haling distribution corresponding to a given initial fuel distribution can be obtained by estimating an initial power shape and iterating between power shape and exposure distribution.

3.2.7 BWR Simulator

The analysis of the burnup dependent static behavior of a reactor core plays an important role in a control rod patterning problem. A BWR simulator must have the ability to predict this behavior. For this purpose, STORM [16] will be used. The code calculates three-dimensional quarter core power distributions for a BWR, using a two-group diffusion theory model. STORM accounts for the mutual interactions between the power, fuel exposure, coolant flow, coolant void fraction, and equilibrium xenon distributions. Control rod effects are also included. The Haling solution option calculates the target power distribution for an operating cycle.

The input to STORM consists primarily of a description of the core geometry and fuel loading, control rod placement, core thermal-hydraulic parameters, and tables of void and exposure dependent cross sections of the fuel. Output includes the predicted core power distribution, effective multiplication factor, coolant flow and void fraction distribution, power peaking factor, LHGR, CHFR, and the exposure distribution at the end of each burnup step.

3.3 Steepest-Ascent Hill Climbing Method

3.3.1 Search Method

There is no definite solution to the control rod patterning problem. An acceptable control rod pattern is determined by investigating many trial solutions. The number of distinct control rod patterns a given reactor may assume is given by

$$t_N = p^N \qquad (3.3.1)$$

where

N = number of control rods in use during operation

p = the number of control rod positions in the axial direction

= notch values

t_N = number of unique control rod patterns.

When only considering the control rods which fill a quarter core (N=15 for A-pattern or N=16 for B-pattern) and the notch values (p=24) the total number of possible patterns is on the order of 24^{15} to 24^{16} !. The optimum solution could be never found from this many combinations. It is possible, however, to formulate a problem solving methodology to direct the search and constrain the number of alternative patterns which are considered. Thus the method, although no longer guaranteed to find the best answer, will find a very good answer.

Direct search in the control rod patterning problem is an iterative procedure of control rod pattern generation, evaluation, and selection that guides the solution to find the control rod pattern which best minimizes the objective function within various constraints. The direct search method employs heuristics to direct the search and constraints to limit the number of configurations considered. These heuristics and constraints applied on top of a general-purpose search method provide the intelligence of the search. The search method used here is known as the steepest-ascent hill climbing method [17] and is classified as a weak search method [18].

Steepest-ascent hill climbing is a variant of generate-and-test in which feedback from the test procedure is used to help the generator decide which direction to move in the search space. At each step of the steepest-ascent hill climbing search process, we select the most promising stage from the list of the stages which can be generated from the current state.

For the control rod patterning problem, the test function in a generate-and-test procedure is the solution of the BWR simulator; this solution requires a huge amount of computing time. However, if the test function is augmented with a heuristic function that provides a good estimate of how close a given state is to a goal state, then the test procedure can rely on the test function instead of running BWR simulator. This is particularly nice because often the computation of the heuristic function can be done with very little computing time.

The history of steepest-ascent hill climbing search as it proceeds from an initial control rod pattern to each succeeding pattern configuration may be imagined as a graph containing nodes corresponding to states (Figure 3.3.1). The nodes of this state space graph are linked together by arcs that represent the moves that transform one control rod pattern to another. Initially there is only one node. This node is expanded to generate three new nodes. The heuristic function, which, in this example, is an estimate of the minimum critical heat flux ratio of a given node, is applied to each of these new nodes. Since node B is the most promising, it is expanded next, producing two successor nodes, E and F. At the next step, F will be expanded, since it is the most promising. This process continues until a solution is found.

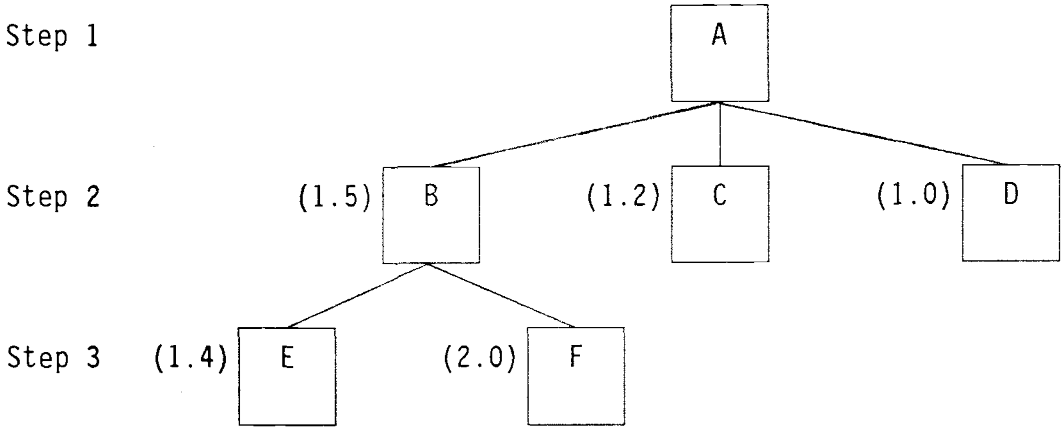


Figure 3.3.1 A Steepest-Ascent Hill Climbing Search Step

3.3.2 System Equations

Since only the control rod pattern is chosen as the control variable for the control rod program, the state of a BWR core for a given fuel distribution can be described as a function of control rod depth. The number of control rods belonging to one control rod pattern is divided into n elements and the rod depth of each element in that pattern is denoted by r_n. Thus one control rod pattern is represented by a set of rod depths r_n. For an actual core, r_n takes inter notch value of between 0 and 24, which indicates that control rod is fully withdrawn and fully inserted, respectively. Using rod depth quantities as input variables, the reactor state can be expressed as

$$\mathbf{C} = f(\mathbf{R}) \tag{3.3.2}$$

where

$\mathbf{C}$ = reactor condition vector, $(c^1, c^2, \cdots, c^\ell)$

$\mathbf{R}$ = control rod depth vector, $(r_1, r_2, \cdots, r_n)$

where the superscript ℓ stands for the parameters J, λ, LHGR, and CHFR in Eqs. (3.2.1) through (3.2.4). Once the core nominal quantities such as core flow, core inlet temperature, etc. are selected for analysis, control rod patterns are described by these input variables, and all relevant output variables are expressed by elements of $\mathbf{C}$ in Eq. (3.3.1).

3.3.3 Control Rod Pattern Generator

The control rod pattern generator applies all possible control rod moves to a particular control rod pattern and thereby generates successor control rod patterns. Unless the moves are restricted, the search space is unmanageable. A reasonable use of heuristics and constraints on the moves can result in considerable reductions in the search. By specifying one-eighth core symmetry and only one notch position change for each control rod move, all the possible successor control rod patterns can be examined within a reasonable amount of time and the search space can be greatly diminished. For example, the total number of successor control rod patterns generated at each search step is reduced to at most 3^{10}

(59059) instead of 24^{15} and 3^8 (6561) instead of 24^{16} for the A- and B-patterns, respectively.

3.3.4 Control Rod Pattern Tester

The control rod pattern tester uses heuristic strategies to select the most appropriate pattern from the list of the control rod patterns, which are created by the control rod pattern generator by evaluating the reactor condition vector. The heuristic strategies will be discussed later in this section.

The reactor condition vector which contains the performance index function ($J(\mathbf{R})$) and the constraint functions (λ, $LHGR(\mathbf{R})$, $CHFR(\mathbf{R})$) is a nonlinear function of the control rod depth. The behavior of these functions with the change in the control rod pattern **R** is obtained only by the core simulator and is not given explicitly. However, due to simulator computing time it is practically impossible to solve for these functions when the core simulator is used exclusively.

According to simulations, however, those functions vary linearly with at least one or two notch changes of each control rod depth. It can be therefore assumed that linearity exists for the small number of notch changes of multiple control rods. From this assumption the simulation for the test procedure can be converted into a linearized problem using the MAP technique. The following approximation is used for this purpose:

$$\mathbf{C} = \mathbf{C}_0 + \nabla \mathbf{C}(\mathbf{R}_0)^t \, (\mathbf{R} - \mathbf{R}_0) \qquad (3.3.3)$$

where subscript 0 refers to a current value, ∇ stands for the gradient over the rod depth vector, and superscript t denotes the transpose of a vector.

A search using sensitivity data, which is the measure of the dependence of each element of reactor condition vector on the depth change of each rod, greatly reduces the number of simulator runs needed for the test procedure. By changing the rod depth r_n one by one around a reference control rod pattern (current pattern in search space) independently, sensitivity data can be obtained using the simulator:

$$S_n^{\ell} = \Delta c^{\ell} / \Delta r_n \qquad (3.3.4)$$

where superscript ℓ stands for J, λ, LHGR, and CHFR. Since the notch change of the each control rod depth, Δr_n, is set to ±1 ('+' sign refers to rod insertion while '-' sign stands for rod withdrawal) and the total number of control rods per pattern is denoted by N, a total of 2N simulations are required for obtaining sensitivity data. If the simulation solutions of the current reference pattern are used as initial guess solutions for sensitivity calculations, the convergence of the simulation scheme can be accelerated, and the total computing time can be greatly reduced.

Using Eqs. (3.3.3) and (3.3.4), the objective function and the nuclear and thermal-hydraulic constraints can be modified as heuristic functions that guide the search by evaluating each individual state. For rod depth change Δr_n the heuristic functions are given as

$$J = \sum_{n=1}^{N} S_n^{J} \Delta r_n + J_0 \qquad (3.3.5)$$

$$\lambda = \sum_{n=1}^{N} S_n^{\lambda} \Delta r_n + \lambda_0 \qquad (3.3.6)$$

$$\mathrm{MLHGR}=\max\{\sum_{n=1}^{N} S_n^{LHGR_m}\Delta r_n + \mathrm{LHGR}_0\},\ m=1,2,\ldots,M_p \quad (3.3.7)$$

$$\mathrm{MCHFR}=\min\{\sum_{n=1}^{N} S_n^{CHFR_m}\Delta r_n + \mathrm{CHFR}_0\},\ m=1,2,\ldots,M_p \quad (3.3.8)$$

where M_p is the total number of monitoring points. These monitoring points, i.e., the number of nodes, is 3360: 140 fuel bundles (for ¼ core) x 24 axial segments (notches). However, the constraints need only be monitored at points where the thermal margin is small. This can cause considerable reduction in the test procedure. Such monitoring points are determined at a reference control rod pattern. Nodes where normalized power density is greater than half of peak value or where the CHFR is less than 3.0 are selected as monitoring points. On the average monitoring points can be reduced to half by this selection process. The critical points for LHGR and CHFR of the resulting power shape are very likely to lie in these monitoring points.

The steepest-ascent hill climbing method usually searches for the direction that shows the greatest improvements for all of the heuristic functions combined. Such improvements are the combination of the fractional improvements of each heuristic function. The weights given to individual heuristic functions are chosen in such a way that the value of the heuristic functions at a given node in the search process gives as good an estimate as possible of whether that node is on the desired path to a solution.

The steepest-ascent hill climbing is composed of four heuristic search strategies. Each strategy is activated depending on thermal-hydraulic constraints values of current reference control rod pattern denoted by $\mathbf{R}_{ref}$. If a strategy fails to generate a pattern, an alternate strategy is applied to continue to search. The alternate strategy is used as a way of dealing with problems such as local maxima, plateaus, or ridges that can happen in hill climbing. The four search strategies are summarized as follows:

1. Search strategy I

   Condition: $MLHGR(\mathbf{R}_{ref}) > MLHGR_{target}$ and $MCHFR(\mathbf{R}_{ref}) < MCHFR_{target}$.

   Search direction: The direction that shows the greatest improvements for MLHGR as long as the nuclear constraint is satisfied and the radial and axial power peaks are reduced.

   Background: As a general rule, J is reduced and MCHFR is increased as PPF, which is proportional to MLHGR, is reduced. However, reduction of PPF sometimes deteriorates MCHFR. Since MCHFR usually occurs in the upper part of the assembly at which the integrated assembly power is greatest, the steam quality is high, and the critical heat flux is small, a reduction in radial and axial power peaks tends to improve MCHFR.

   Alternate strategy: none.

Further explanation for the search direction is necessary. Using Eqs.(3.3.3) and (3.3.4) the radial and axial power distributions, denoted by p_r and p_z, respectively, are modified as heuristic functions that determine the radial and axial power peaks referred to pp_r and pp_z, respectively.

For rod depth change Δr_n the heuristic functions for the pp_r and pp_z are given as

$$pp_r = \max\{\sum_{n=1}^{N} S_n^{p_{r_{ij}}} \Delta r_n + p_{r_0}\}, \ ij = 1, 2, \ldots, IJ \qquad (3.3.9)$$

$$pp_z = \max\{\sum_{n=1}^{N} S_n^{p_{z_k}} \Delta r_n + p_{z_0}\}, \ k = 1, 2, \ldots, K \qquad (3.3.10)$$

where IJ and K are the total monitoring points in radial and axial directions, respectively.

2. Search strategy II

   Condition: $\mathrm{MLHGR}(\mathbf{R}_{ref}) < \mathrm{MLHGR}_{target}$ and $\mathrm{MCHFR}(\mathbf{R}_{ref}) < \mathrm{MCHFR}_{target}$.

   Search direction: The direction that shows the greatest improvements for MCHFR as long as the nuclear and MLHGR constraints are satisfied.

   Background: Generally, J is reduced as MCHFR is reduced.

   Alternate strategy: strategy I.

3. Search strategy III

   Condition: $\mathrm{MLHGR}(\mathbf{R}_{ref}) > \mathrm{MLHGR}_{target}$ and $\mathrm{MCHFR}(\mathbf{R}_{ref}) > \mathrm{MCHFR}_{target}$.

   Search direction: The direction that shows the greatest improvements for MLHGR as long as the nuclear and MCHFR constraints are satisfied.

   Background: Generally, J is reduced as MLHGR is reduced.

Alternate strategy: strategy II.

4. Search strategy IV

   Condition: MLHGR($\mathbf{R}_{ref}$) < MLHGR$_{target}$ and MCHFR($\mathbf{R}_{ref}$) > MCHFR$_{target}$.

   Search direction: The direction that shows the greatest improvements for J as long as all constraints are satisfied.

   Background: Since all constraints are satisfied, J is minimized.

   Alternate strategy: strategy III.

The heuristic search method based on these strategies is extensively used in guiding the search for the optimum solution. Depending on the type of strategy, the search is made in the direction of maximum improvement for a particular heuristic function as long as other heuristic functions are improved or satisfied.

3.3.5 Search Solution Algorithm

Employing the control rod pattern generate-and-test procedure which was described before, the overall optimization procedure illustrated in Figure 3.3.2 consists of the following steps:

1. Select an initial control rod pattern, $\mathbf{R} = (r_{10}, r_{20}, \cdots, r_{n0})$.
2. Calculate sensitivity data using the simulator (Eq. 3.3.4) and changing the control rod depth $r_n : r_n \leftarrow r_n + \Delta r_n$.
3. Generate new control rod patterns by changing the control rod depth $r_n : r_n \leftarrow r_n + \Delta r_n$.
4. Test patterns by applying the proper strategies discussed before.
5. Check that the selected control rod pattern is unique.
6. Shift the reference rod pattern to the new one : $\mathbf{R}_0 \leftarrow \mathbf{R}_0 + \Delta\mathbf{R}_0$.

7. Run three-dimensional simulator.

8. Repeat steps 2 to 7 until a suboptimal solution is found.

In relation to procedure 4, more explanation would be necessary. If a particular strategy fails to find any control rod pattern that satisfies the conditions listed, then the nuclear constraint is temporally relaxed and the search is made to find alternative control rod pattern with the relaxed nuclear constraint. After the alternative pattern is found, the nuclear constraint recovers its initial setting value. For Strategy I constraints on radial and axial power peaks are also relaxed if it fails with relaxed nuclear constraints.

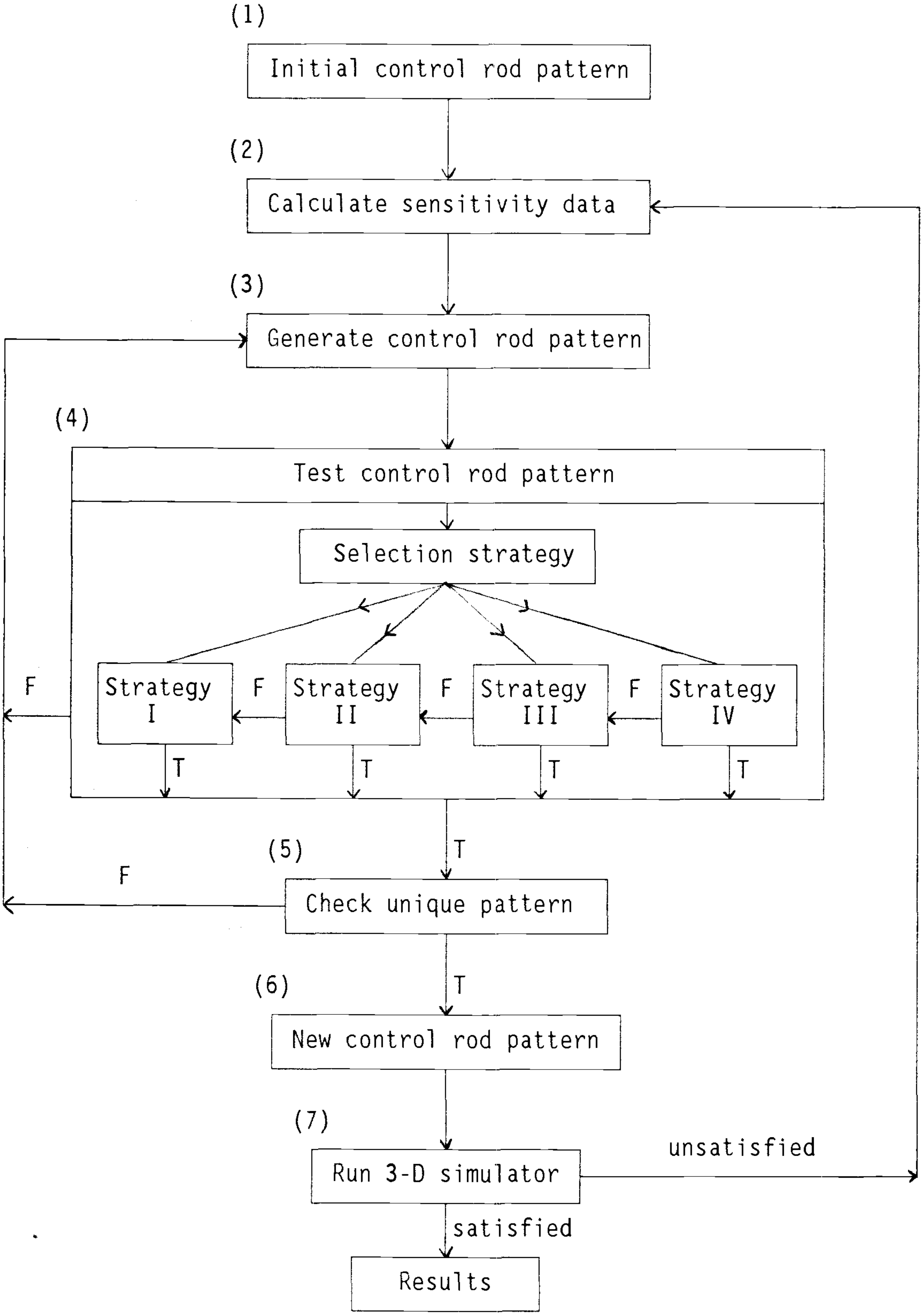


Figure 3.3.2 Flow Chart of the Control Rod Pattern Optimization Procedure

3.4 Results

A computer code, Optimum Control Rod Pattern Searcher (OCRPS), has been developed for the generation of an optimum control rod pattern for a given fuel loading pattern. This code employs heuristic search strategies discussed before and data from the reference BWR. OCRPS is written in Lahey EM/32 FORTRAN77 compiler. Appendix B presents the sensitivity data calculation, control rod pattern generation, and control rod pattern evaluation subroutines. All computations reported in this study were performed on a COMPAQ DESKPRO 386/25 personal computer.

In order to test OCRPS for a control rod problem, a guess of the control rod pattern must be made to initiate the search. This initial pattern can take any form but the choice of initial patterns affects the solution performance. The initial control rod pattern can be determined arbitrarily, but reasonable estimates from accumulated experiences can facilitate the overall convergence.

Without the benefit of any reasonable estimate from previous experience, the initial control rod pattern is selected in a way that all control rods are inserted the same distance in the core to satisfy only the nuclear constraint. This preparation for initial rod pattern can be easily done by several simulator runs.

For examining the validity of the OCRPS, test calculations are performed on the reference core along with the constraints and the design data listed in Table 3.4.1. Two different values, 0.003 and 0.005, are used for the normal and relaxed tolerance factors for nuclear constraint, respectively. Tolerance factors for radial and axial power peaking are 0.03 and 0.05, respectively.

Table 3.4.1 Constraint and Tolerance Data

Power peaking factor	2.2
MLHGR	13.4 kW/ft
MCHFR	1.9
Nuclear tolerance factor	0.003[a] or 0.005[b]
Radial power peaking tolerance factor	0.03
Axial power peaking tolerance factor	0.05

[a] Initial tolerance value
[b] Relaxed tolerance value

Using data listed above and initial guess patterns, a series of the optimum control rod pattern of six exposure steps (5, 6, 7, 8, 9, and 10 GWD/T) are illustrated in Fig. 3.4.1 through 3.4.6. Two-group cross section data were available for 0, 5, and 10 GWD/T burnups [19]. Unfortunately, cross sections for 0 GWD/T burnup were not suitable for test calculations since criticality could not be achieved with these cross section data. It would seem that the cross section data were not generated properly. However, the accuracy of cross section is not extremely important for the nature of this research.

Instead, cross sections for 5 and 10 GWD/T burnups were used for test calculations. Two-group cross sections for 6, 7, 8, and 9 GWD/T burnups were obtained by linear interpolation. Insertion depth for each rod is given for a quarter core using a conventional notch unit. The optimization is performed for a given burnup step, and no coupled effects between burnup steps are considered. The convergence of J, PPF, and MCHFR are also illustrated in Figure 3.4.1 through 3.4.6. The convergence of MLHGR is equivalent to that of PPF. The iterative search procedure with sequential improvements in the control rod pattern has been found to converge rapidly. In general, as the search goes on, the J is improved and the other two parameters are also improved and finally satisfy

5.0 GWD/T: A-pattern

```
(Guess)                                              (Optimum)

20-------20----------20                              13-------13----------23
   20          20          19                           21          20          13
20       20          19                              13       21          20
   20          20                   ======>             20          21
20       19          19              0: full out     23       20          12
   19                               24: full in         13
```

λ = 1.0027
PPF = 3.2573
MLHGR = 17.524 kW/ft
MCHFR = 1.2556
J = 0.5103

λ = 1.0039
PPF = 2.1985
MLHGR = 11.828 kW/ft
MCHFR = 2.0463
J = 0.1692

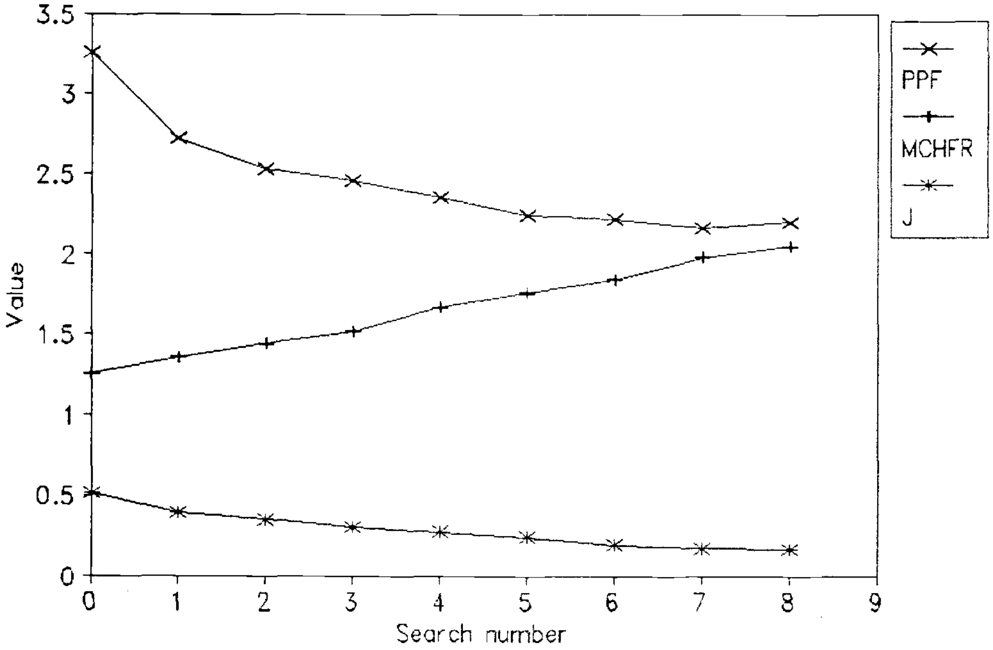


Figure 3.4.1 Example of Search Process of OCRPS (5.0 GWD/T)

6.0 GWD/T: B-pattern

```
(Guess)                                      (Optimum)

 20      20      19                           18      3      16
20   20     19                               18   21     22
   20    19     19                              21    20     9
20   19     19            ======>            3    20     19
   19    19                 0: full out         22    19
19   19                    24: full in       16    9
```

λ = 1.0008
PPF = 4.1678
MLHGR = 22.423 kW/ft
MCHFR = 1.0339
J = 0.9512

λ = 0.9952
PPF = 2.0159
MLHGR = 10.845 kW/ft
MCHFR = 2.0653
J = 0.1448

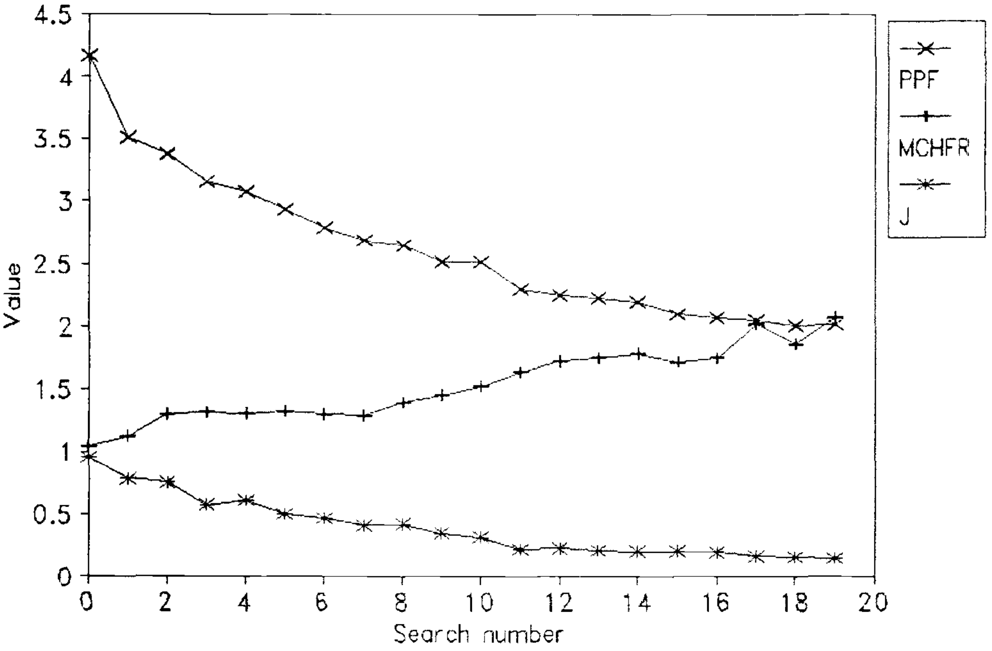


Figure 3.4.2 Example of Search Process of OCRPS (6.0 GWD/T)

7.0 GWD/T: A-pattern

```
(Guess)                                       (Optimum)

19     19        19                           19     16        23
   19       19       18                          21        8        15
19     19        19                           16     21        21
   19       19                 ====>             8         17
19     19        19          0: full out      23     21        9
   18                       24: full in          15

λ     = 1.0000                                λ     = 0.9976
PPF   = 4.2864                                PPF   = 2.1368
MLHGR = 23.061 kW/ft                          MLHGR = 11.496 kW/ft
MCHFR = 1.1252                                MCHFR = 1.9120
J     = 1.0779                                J     = 0.2046
```

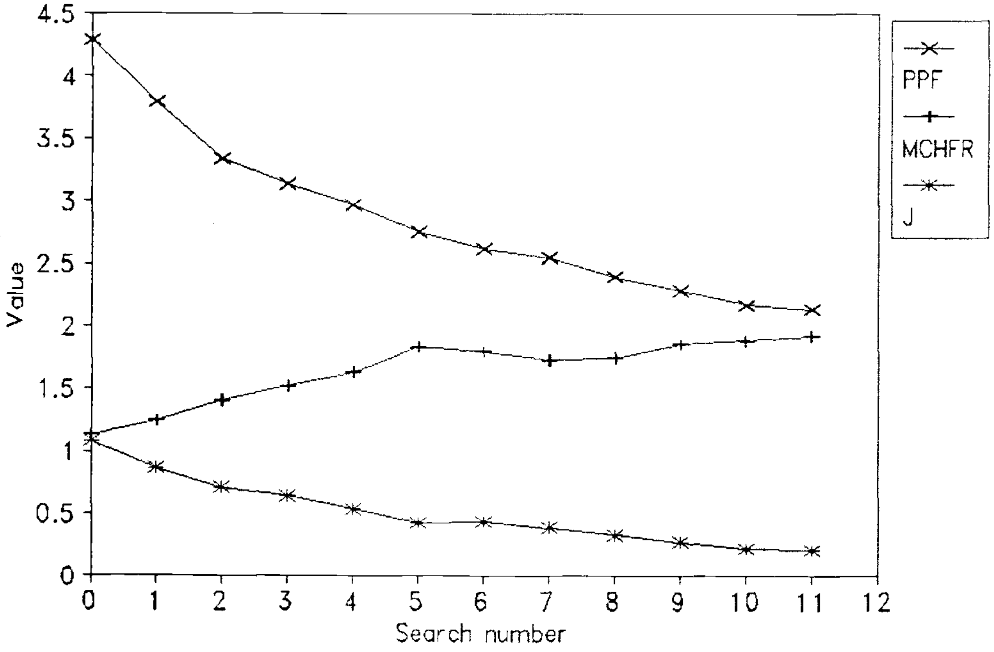
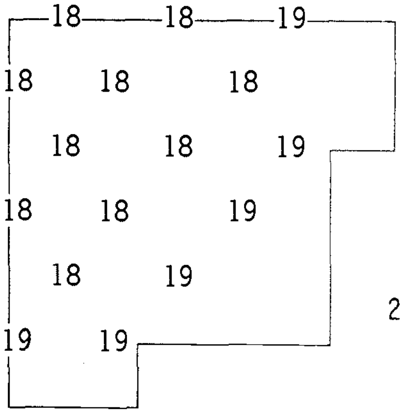


Figure 3.4.3 Example of Search Process of OCRPS (7.0 GWD/T)

8.0 GWD/T: B-pattern

(Guess)

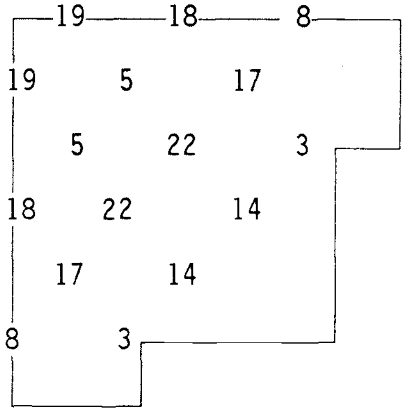


λ = 0.9987
PPF = 5.2469
MLHGR = 28.228 kW/ft
MCHFR = 0.8140
J = 1.8031

⟹

0: full out
24: full in

(Optimum)



λ = 1.0025
PPF = 2.1668
MLHGR = 11.657 kW/ft
MCHFR = 1.9298
J = 0.2110

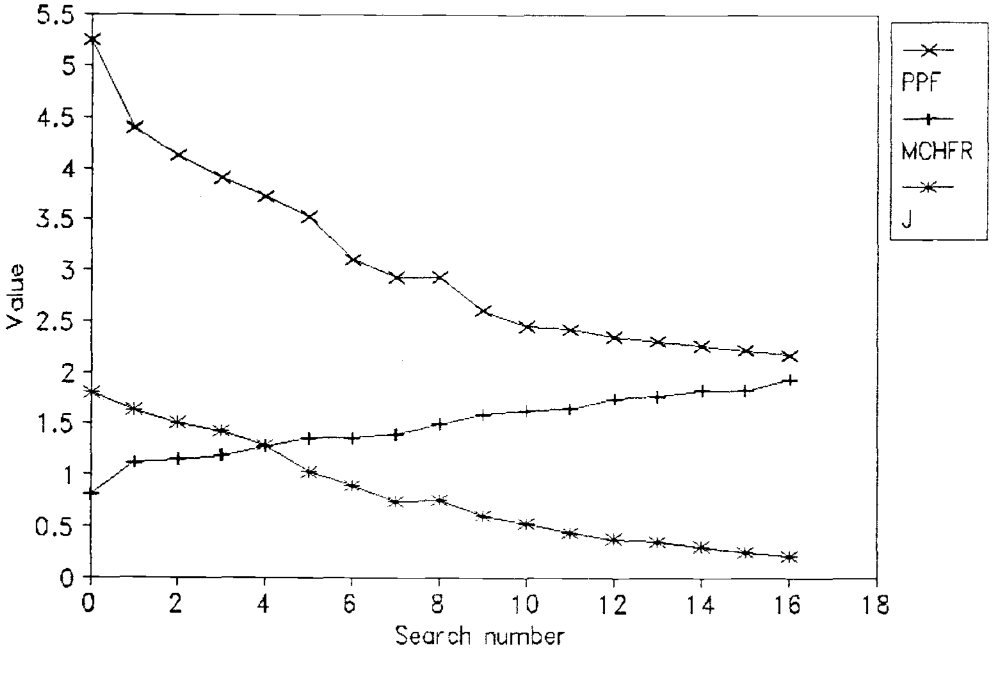
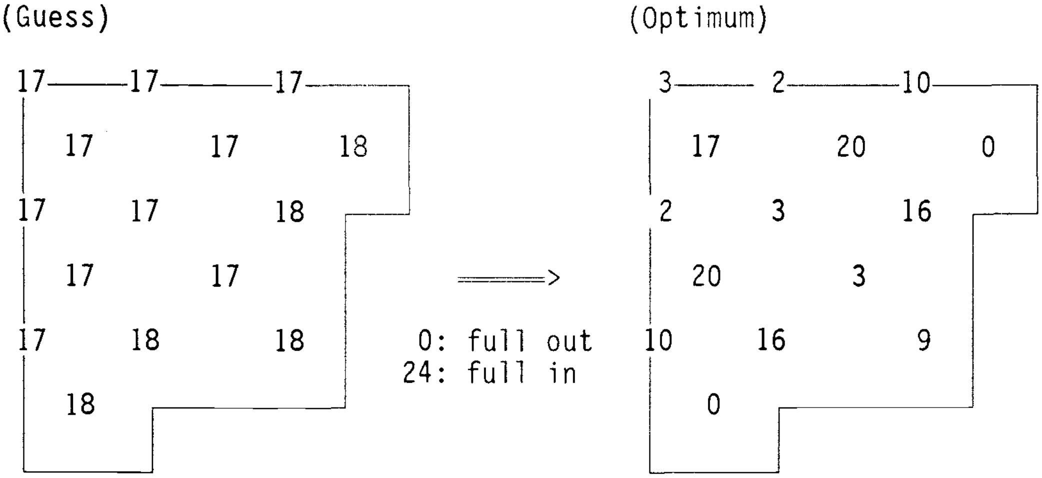


Figure 3.4.4 Example of Search Process of OCRPS (8.0 GWD/T)

9.0 GWD/T: A-pattern



```
(Guess)                                          (Optimum)

17———17————17————                                3———— 2—————10————
   17       17       18                             17       20       0
17      17       18                              2      3        16
   17       17                  ====>               20       3
17      18       18                0: full out   10     16        9
   18                             24: full in       0
```

```
λ     = 0.9995                                   λ     = 1.0045
PPF   = 4.8823                                   PPF   = 1.9449
MLHGR = 26.267 kW/ft                             MLHGR = 10.464 kW/ft
MCHFR = 0.9277                                   MCHFR = 1.9992
J     = 1.7987                                   J     = 0.1302
```

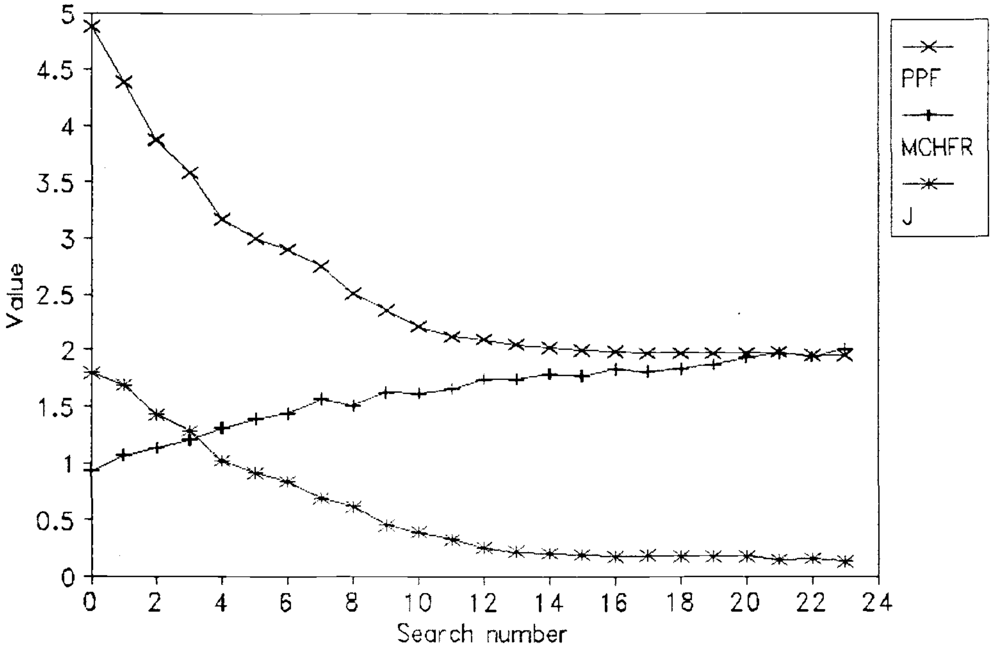
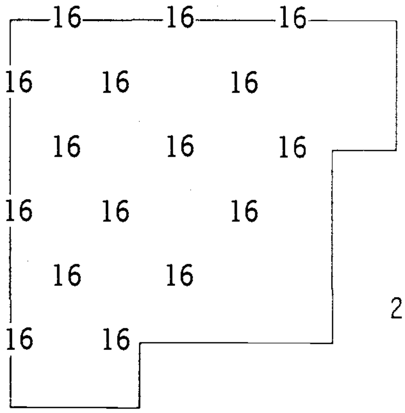


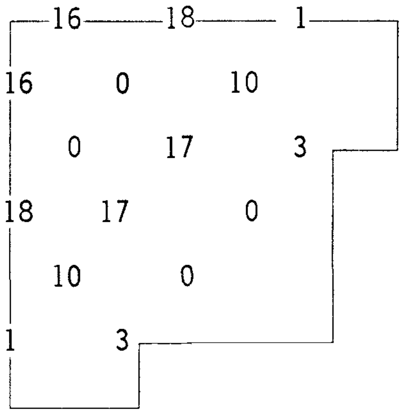
Figure 3.4.5 Example of Search Process of OCRPS (9.0 GWD/T)

10.0 GWD/T: B-pattern

(Guess)



(Optimum)



==>

0: full out
24: full in

λ	=	1.0001
PPF	=	4.5132
MLHGR	=	24.281 kW/ft
MCHFR	=	1.1017
J	=	1.9550

λ	=	1.0030
PPF	=	2.0062
MLHGR	=	10.793 kW/ft
MCHFR	=	2.3819
J	=	0.1776

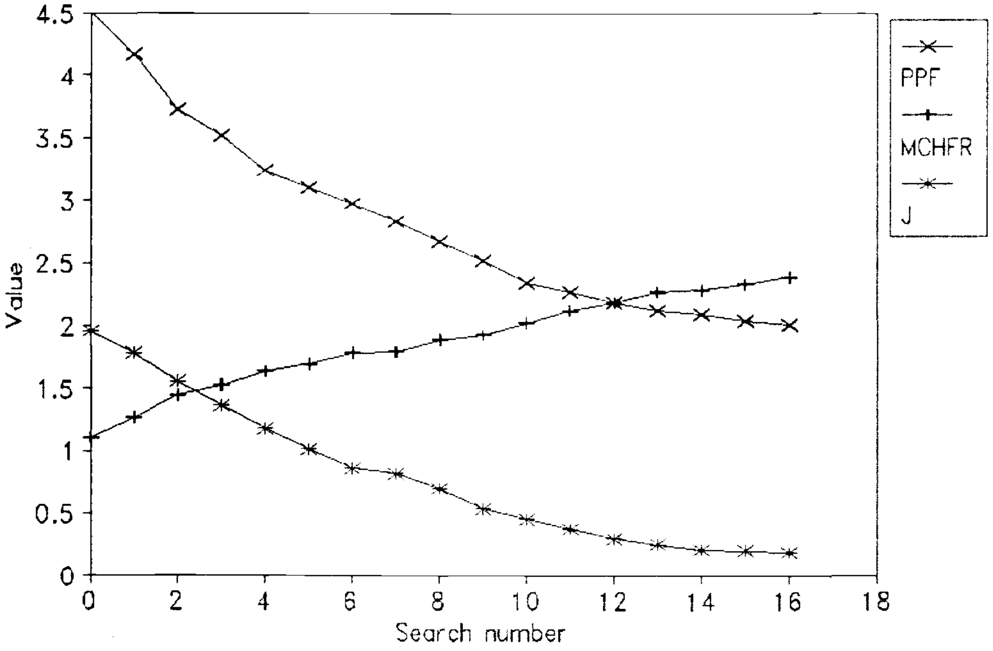


Figure 3.4.6 Example of Search Process of OCRPS (10.0 GWD/T)

constraints as expected.

The above examples show that the steepest-ascent hill climbing method using heuristic search strategies works well when started from an unfeasible initial guess pattern. Most of initial control rod patterns have extremely deteriorated J as well as PPF and MCHFR. For example the J of the initial control rod pattern for 10.0 GWD/T burnup is 1.9950, which means that on the average the difference between initial pattern power distribution and target power distribution for each node is 1.3982 ($\sqrt{1.9950}$). This method always finds a optimum solution. No oscillation or divergence of solution has been observed.

Table 3.4.2 lists λ, PPF, MLHGR, MCHFR, and J of the optimum control rod pattern for each burnup step. Table 3.4.2 also summarizes total number of search steps, the total computation time required for obtaining the optimum control rod pattern, for calculating sensitivity data, generating and testing patterns, and running the 3-D simulator.

As can be seen in Table 3.4.2, more than 76% of the overall computation time is spent on calculating sensitivity data. The total computation required for generating and testing A-patterns is much greater than B-patterns. This is due to the size of the search space.

The overall computing time for obtaining a series of optimum control rod patterns is 3112 minutes (51.9 hrs). Since one simulator run requires approximately 2.93 minutes, the overall computing time is equivalent to 1062 simulator runs.

The overall computing time (3112 minutes) is estimated to be equivalent to 41.5 minutes on a CRAY XMP/SE, which is about 70 times fast than a COMPAQ DESKPRO 386/25 personal computer. This shows that running

OCRPS on a CRAY XMP/SE can result in substantial savings in engineering man-hours.

Table 3.4.2 Summary of Results of the Optimum Control Rod Pattern

	Burnup (GWD/T)						Overall Burnup Steps
	5.0	6.0	7.0	8.0	9.0	10.0	
Control Rod Pattern	A1	B1	A2	B2	A3	B3	
λ	1.0039	0.9952	0.9976	1.0025	1.0045	1.0030	
PPF	2.1985	2.0159	2.1368	2.1668	1.9449	2.0062	
MLHGR*	11.828	10.845	11.496	11.657	10.464	10.793	
MCHFR	2.0463	2.0653	1.9120	1.9298	1.9992	2.3819	
J	0.1692	0.1448	0.2046	0.2110	0.1302	0.1776	
$\sqrt{J}$	0.4113	0.3805	0.4523	0.4593	0.3608	0.4214	
(a)	8	19	11	16	23	16	93
(b)	244.	456.	332.	335.	677.	329.	2373.
(c)	86.	13.	53.	16.	288.	9.	465.
(d)	23.	58.	31.	47.	66.	49.	274.
(e)	353.	527.	416.	398.	1031.	387.	3112.

* kW/ft

(a) Total Number of Search Steps

(b) Total Computation Time for Calculating Sensitivity Data (Min.)

(c) Total Computation Time for Generating and Testing Patterns (Min.)

(d) Total Computation Time for Simulator Runs (Min.)

(e) Overall Computing Time for Obtaining Optimum Control Rod Pattern (= (b) + (c) + (d)) (Min.)

The largest and smallest mean differences between the optimum control rod pattern power distribution and the corresponding target power distribution for each node ($\sqrt{J}$) are 0.4593 at 8.0 GWD/T burnup and 0.3608 at 9.0 GWD/T burnup, respectively. This indicates that the power distributions of the optimum patterns fit target power distributions well.

3.5 Conclusions

The objective of this study was to develop an efficient method for solving control rod positioning problem in BWRs. By using the steepest-ascent hill climbing search with heuristic strategies, a practical solution method, OCRPS, is developed.

The test calculations have demonstrated that, for the reference BWR, OCRPS shows excellent performance for finding a series of optimum control rod patterns for six burnup steps during an operating cycle. Computing costs are modest even if the initial guess patterns have extremely deteriorated core characteristics. Thus, OCRPS has proven to be an appealing solution tool for in-core fuel management of BWRs.

The report is not a final evaluation of OCRPS; evaluation of the code should continue as more control rod pattern applications are attempted.

3.5.1 Recommendations for Future Improvements

During the course of this research, many interesting topics have not been investigated. This section contains a brief description of these research areas.

1. Computation time can be reduced further. More than 76% of the time is spent on calculating sensitivity data using the 3-D simulator. Improved methods should obtain the perturbed power distribution to a small change of control rod position in less time. In particular, if the TLAR acceleration scheme can be used to calculate the perturbed power distribution, the computing time

will be greatly reduced.

2. Much of the information accumulated throughout operating experience and theoretical work should be utilized in heuristic rules for the search strategy in order to facilitate the search process by discarding non-optimum paths without exploring them.
3. An expert system which has an interactive graphics environment and an explanation capability can be developed to provide a user with a trace or display of system operation.

Chapter References

1. D. L. Delp et al., "FLARE-A Three Dimensional Boiling Water Reactor Simulator," GEAP-4598, General Electric Company (1964).

2. W. B. Terny and H. Fenech, "Control Rod Programing Optimization Using Dynamic Programing," Nucl. Sci. Eng., 39, 109 (1970).

3. W. C. Wade and W. B. Terney, "Optimal Control of Nuclear Reactor Depletion," Nucl. Sci. Eng., 45, 199 (1971).

4. H. Motoda and T. Kawai, "A Theory of Control-Rod Programming Optimization in Two-Region Reactors," Nucl. Sci. Eng., 39, 114 (1970).

5. B. Snyder and E. E. Lewis, "Optimal Control Rod Policies for an Operating Cycle of a Simulated BWR Core," Proc. 1973 Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, CONF-730414-PI, II-56, U.S. Atomic Energy Commission (1973).

6. H. Motoda, "Optimal Control Rod Programming of Light Water Reactors in Equilibrium Fuel Cycle," Nucl. Sci. Eng., 46, 88 (1971).

7. H. Motoda, "Optimization of Control Rod Programming and Loading Pattern in Multiregion Nuclear Reactor by the Method of Approximation Programming," Nucl. Sci. Eng., 49, 515 (1972).

8. R. E. Griffith and R. A. Stewart, "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems," Management Sci., 7, 379 (1961).

9. T. Kawai, H. Motoda, T. Kiguchi, and M. Ozawa, "A Method for Generating a Control Rod Program for Boiling Water Reactors," Nucl. Tech., 28, 108 (1976).

10. T. Hayase and H. Motoda, "Boiling water Control Rod Programming Using Heuristic and Mathematical Methods," Nucl. Tech., 48, 91 (1980).

11. S. Tokumasu, M. Ozawa, H. Hiranuma, and M. Yokomi, "A Mathematical Method for Boiling Water Reactor Control Rod Programming," Nucl. Tech., 71, 568 (1985).

12. O. L. Mangasarin, "Duel Feasible Direction Algorithms," Technical Summary Report No. 1173, Mathematics Research Center, University of Wisconsin (Feb. 1972).

13. "Edwin I. Hatch Nuclear Plant Unit Number 2, Final Safety Analysis Report," Section 4, Georgia Power Company.

14. John O. Mingle, "In-Core Fuel Management via Perturbation Theory," Nucl. Tech., 27, 248 (1975).

15. R. K. Haling, "Operating Strategy for Maintaining an Optimum Power Distribution throughout Life," USAEC Report TID-7072 (1964).

16. B. O. Cho, "A Comparison of Three Coarse-Mesh Nodal Methods for Boiling Water Reactor Analysis," M.S. thesis, Oregon State University (June 1988).

17. Elaine Rich, Artificial Intelligence, McGraw-Hill, Inc., 75, 1983.

18. Elaine Rich, Artificial Intelligence, McGraw-Hill, Inc., 72, 1983.

19. R. L. Hatteberg, "A Simulated Two-Group Method for Computing Boiling Water Reactor Power Distributions," M.S. thesis, Oregon State University (June 1978).

Chapter 4

A POLYNOMIAL FLUX EXPANSION METHOD FOR PIN POWER RECONSTRUCTION

4.1 Introduction

Finite difference methods are commonly used as a standard computational technique for calculation of the spatial power distribution required for the design and analysis of light water reactors. However, the real disadvantage of finite difference methods is that very fine spatial meshes are required to achieve acceptable accuracy. Thus, even with present-day high speed digital computers, the finite difference fine-mesh calculation remains prohibitively expensive for large water reactors.

In recent years, more sophisticated theoretical methods have been developed. In particular, the nodal equivalence theory devised by Koebke, K. [1] and later modified by Smith, K., et. al. [2], has been demonstrated to be at least two orders of magnitude more computationally efficient than finite difference schemes.

In many reactor calculations, global quantities such as k_{eff} and average assembly power distributions are of primary importance. In such a case, coarse-mesh nodal methods become more attractive than fine-mesh finite difference methods.

However, several reactor studies such as the determination of power peaking factors or thermal-hydraulic analysis of the hottest channels in an assembly, require detailed knowledge of pin-cell powers. Coarse-mesh nodal methods cannot be directly and efficiently applied to calculate this detailed information. A fundamental drawback of conventional coarse-mesh nodal methods is the loss of detailed flux and power distributions within the coarse-mesh nodes. In order to enlarge the scope of the nodal schemes, there are strong incentives to develop computationally efficient methods to calculate fine-mesh point fluxes and powers from a global nodal solution.

Several methods to reconstruct, from nodal solutions, the fine-mesh flux and power, have been developed by Koebke, K., et. al. [3,4,5], Smith, K., et. al. [6,7], Jonsson, A., et. al. [8], and Liu, Y. S., et. al. [9]. These methods are based on the assumption that the fine-mesh point flux (heterogeneous flux) can be expressed as the product of an intra-nodal flux distribution and an assembly form function. The difference between reconstruction methods arises primarily from the approximations used to construct the intra-nodal flux distributions and the manners in which the assembly form function is obtained.

The objective of this study is to develop an accurate and computationally efficient method for reconstructing two-dimensional fine-mesh flux shapes and power distributions from coarse-mesh nodal solutions. By employing an approximation for the spatial shape of the intra-nodal flux distributions, the least square second-order polynomial flux expansion method is developed for the reconstruction of fine-mesh flux and power distributions. The basic assumption of this method is that the reconstructed flux can be expressed as a product of an assembly form function and a second-order polynomial function which represents the intra-nodal flux distribution. The assembly form function is calculated by solving an assembly criticality calculation. The polynomial function is determined by minimizing the least-squares difference between the intra-nodal quantities obtained from the global two-dimensional coarse-mesh nodal solutions and those obtained from the evaluation of the polynomial function.

This study discusses the method and results of polynomial flux expansion for reconstruction of fine-mesh flux and power distributions in a two-dimensional core. In Section 4.2, the derivation of the two-dimensional polynomial flux expansion method is presented. Section 4.3 presents the accuracy and computational efficiency of the polynomial flux expansion method when applied to the two-dimensional PWR benchmark problem. Finally, a summary of this investigation and recommendations for future research are given in Section 4.4.

4.2 Derivation of the Two-Dimensional Flux Reconstruction Method

In this section, a method for reconstruction of flux shapes from two-dimensional reactor geometry is discussed. This method is capable of calculating fine-mesh, two-dimensional flux distribution and thus the reactor pointwise power distribution.

Within any reactor node, the reconstructed flux may be expressed as the product of a second-order polynomial function and an assembly form function. The assembly form function is obtained by a series of local fine-mesh, two-dimensional criticality calculations. The coefficients of the second-order polynomial will be obtained by the node volume averaged flux, the 4 face-integrated fluxes, and the 4 face-integrated currents. The unknown polynomial coefficients are determined by forcing the 9 intra-nodal quantities to match by a least-square approximation the corresponding values of the polynomial.

4.2.1 Global Reactor Analysis

The Boltzmann transport equation governs the neutron population behavior inside a nuclear reactor. Unfortunately, for realistic reactor configurations, an exact solution of the transport equation is prohibitively time intensive, therefore, approximations of the transport equation are normally used. For light water reactors, multigroup neutron diffusion theory is the most widely used of these approximate methods. For this model, the static multigroup neutron diffusion equation can be written as

$$-\nabla\cdot D_g(\vec{r})\,\nabla\phi_g(\vec{r}) + \Sigma_{tg}(\vec{r})\,\phi_g(\vec{r}) = \sum_{g'=1}^{G}\left[\frac{1}{\lambda}\chi_g \nu\Sigma_{fg'}(\vec{r}) + \Sigma_{sg'g}(\vec{r})\right]\phi_{g'}(\vec{r}), \qquad (4.2.1)$$

$$g=1,2,\ldots,G$$

where

ϕ_g = neutron flux in group g

D_g = diffusion coefficient for group g

Σ_{tg} = total cross section for group g

$\nu\Sigma_{fg}$ = ν-fission cross section for group g

$\Sigma_{sg'g}$ = group-transfer cross section

λ = eigenvalue

χ_g = fission neutron spectrum in group g

g, g′ = energy groups.

4.2.2 Derivation of the Two-Dimensional Polynomial Flux Expansion Method

The global reactor problem is treated in two-dimensional Cartesian geometry, where x and y represent the two coordinate directions. The reactor spatial domain, D_R, is first partitioned into large rectangles, called nodes. Each node is specified by the indices (i,j) corresponding to the center of node. The coordinates of nodal boundaries in x-direction is given by $x_{i-1/2}$ and $x_{i+1/2}$. The nodal spatial domain, D^{ij}, for node (i,j) is defined by

$$D^{ij} = \{\ x \in [x_{i-1/2},x_{i+1/2}] \text{ and } y \in [y_{j-1/2},y_{j+1/2}]\ \}. \qquad (4.2.2)$$

The basic assumption of the two dimensional second-order polynomial method is that within any nodal spatial domain D^{ij}, and for neutron group g, the reconstructed flux $\phi_g^{ij}(x,y)$ can be expressed as the product of a two-dimensional assembly form function, $\psi_g^{ij}(x,y)$ and a second-order polynomial function, $p_g^{ij}(x,y)$. Thus,

$$\phi_g^{ij}(x,y) = \psi_g^{ij}(x,y) \cdot p_g^{ij}(x,y) \qquad (4.2.3)$$

where

$\phi_g^{ij}(x,y)$ = reconstructed(local heterogeneous flux) function for group g and node (i,j)

$\psi_g^{ij}(x,y)$ = assembly(heterogeneous) form function for group g and node(i,j)

$p_g^{ij}(x,y)$ = second-order polynomial function(global homogenized flux distribution) for group g and node (i,j).

The reconstructed power distribution is then calculated from the group-wise reconstructed flux, $\phi_g^{ij}(x,y)$, and ν-fission cross sections:

$$\tilde{P}^{ij}(x,y) = \sum_{g=1}^{G} \nu\Sigma_{fg}^{ij}(x,y)\,\Phi_g^{ij}(x,y) \qquad (4.2.4)$$

where

G = total number of energy groups

$\tilde{P}^{ij}(x,y)$ = reconstructed pin power in node.

The assembly function, $\psi_g^{ij}(x,y)$, can be obtained by solving the multigroup neutron diffusion equation by a fine-mesh, two-dimensional assembly criticality calculation with appropriate boundary conditions. The usual choice for assembly calculations is zero net current at the boundary. Such fine-mesh criticality calculations can be performed with an existing finite difference computer code, for example, 2DB [10].

The polynomial function, $p_g^{ij}(x,y)$, can be written as

$$p_g^{ij}(x,y) = \sum_{m=0}^{2} \sum_{n=0}^{2-m} a_{gmn}^{ij} x^m y^n \qquad (4.2.5)$$

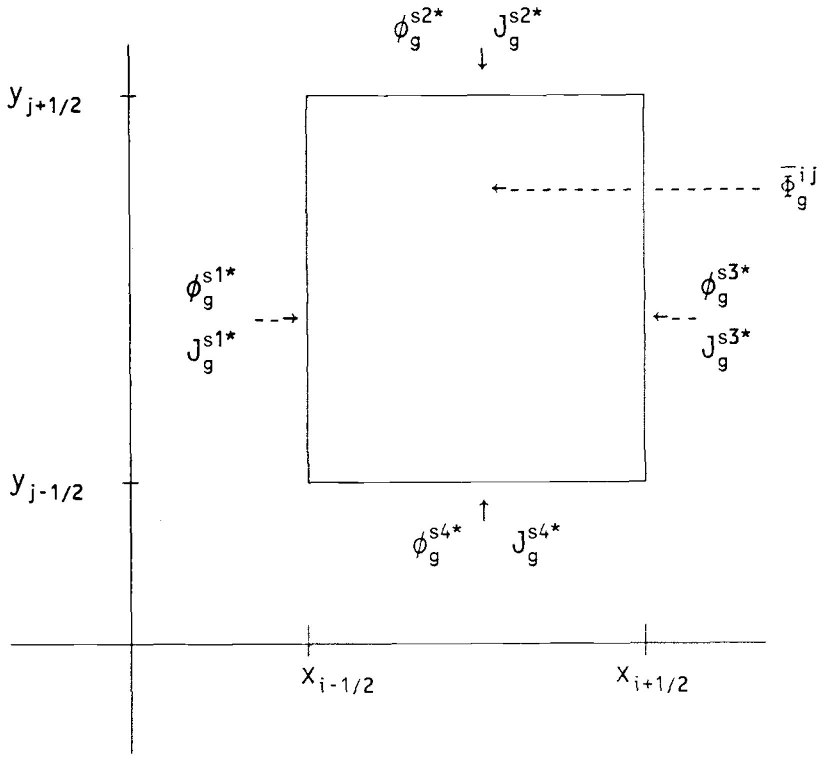
Since no separability is assumed, there are 6 independent polynomial coefficients, a_{gmn}^{ij}, for each neutron group, g, and node (i,j). Once these polynomial coefficients are obtained, the reconstructed flux, $\phi_g^{ij}(x,y)$, can be easily computed by using Eqs. (4.2.5), (4.2.4), and (4.2.3). In order to determine such coefficients, it is required to impose, for each group and node, 9 intra-nodal conditions on the polynomial function. Since the intra-nodal flux distribution, in general, can not be expressed by a finite order of polynomials, the $p_g^{ij}(x,y)$, appearing in Eq.(4.2.5) is an approximate solution for the intra-nodal flux distribution.

4.2.3 Determination of Polynomial Coefficients

When the intra-nodal flux distribution is given, the face-integrated flux, face-integrated current, and the node-averaged intra-nodal flux shown in Figure 4.2.1 can be expressed by the following equations:

$$\phi_g^{s1*} = \int_{y_{j-1/2}}^{y_{j+1/2}} \Phi_g^{ij}(x_{i-1/2}, y)\, dy \qquad (4.2.6)$$

$$\phi_g^{s2*} = \int_{x_{i-1/2}}^{x_{i+1/2}} \Phi_g^{ij}(x, y_{j+1/2})\, dx \qquad (4.2.7)$$



ϕ_g^{s1*}, ϕ_g^{s2*}, ϕ_g^{s3*}, ϕ_g^{s4*} = face-integrated fluxes for surfaces 1, 2, 3, and 4, respectively

J_g^{s1*}, J_g^{s2*}, J_g^{s3*}, J_g^{s4*} = face-integrated currents for surfaces 1, 2, 3, and 4, respectively

$\bar{\Phi}_g^{ij}$ = node-averaged intra-nodal flux

Figure 4.2.1 Nodal Quantities in a Node

$$\phi_g^{s3*} = \int_{y_{j-1/2}}^{y_{j+1/2}} \Phi_g^{ij}\,(x_{i+1/2}, y)\,dy \tag{4.2.8}$$

$$\phi_g^{s4*} = \int_{x_{i-1/2}}^{x_{i+1/2}} \Phi_g^{ij}\,(x, y_{j-1/2})\,dx \tag{4.2.9}$$

$$J_g^{s1*} = \int_{y_{j-1/2}}^{y_{j+1/2}} -D_g^{ij}\,\frac{\partial}{\partial x}\,\Phi_g^{ij}\,(x_{i-1/2}, y)\,dy \tag{4.2.10}$$

$$J_g^{s2*} = \int_{x_{i-1/2}}^{x_{i+1/2}} -D_g^{ij}\,\frac{\partial}{\partial y}\,\Phi_g^{ij}\,(x, y_{j+1/2})\,dx \tag{4.2.11}$$

$$J_g^{s3*} = \int_{y_{j-1/2}}^{y_{j+1/2}} -D_g^{ij}\,\frac{\partial}{\partial x}\,\Phi_g^{ij}\,(x_{i+1/2}, y)\,dy \tag{4.2.12}$$

$$J_g^{s4*} = \int_{x_{i-1/2}}^{x_{i+1/2}} -D_g^{ij}\,\frac{\partial}{\partial y}\,\Phi_g^{ij}\,(x, y_{j-1/2})\,dx \tag{4.2.13}$$

$$\overline{\Phi}_g^{ij*} = \frac{1}{V_{ij}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \Phi_g^{ij}\,(x, y)\,dy\,dx \tag{4.2.14}$$

where

V_{ij} = volume of node (i,j)

$\phi_g^{s\alpha*}$ = face-integrated flux (α=1,...,4)

$J_g^{s\alpha*}$ = face-integrated current (α=1,...,4)

$\Phi_g^{ij}(x,y)$ = intra-nodal flux distribution

$\bar{\Phi}_g^{ij*}$ = node-averaged intra-nodal flux.

Even if detailed information about the intra-nodal flux distribution is unknown, the face-integrated flux, $\phi_g^{s\alpha*}$, (α=1,...,4), face-integrated current, $J_g^{s\alpha*}$, (α=1,...4), and node-averaged intra-nodal flux, $\bar{\Phi}_g^{ij*}$, can be obtained from a global two-dimensional nodal solution, for example, STORM [11].

Since the polynomial function, $p_g^{ij}(x,y)$, is an approximate solution for the intra-nodal flux distribution, $\Phi_g^{ij}(x,y)$, which is unknown, the group intra-nodal quantities defined as above can be reproduced for each group and node by substituting the polynomial function, $p_g^{ij}(x,y)$, for $\Phi_g^{ij}(x,y)$ into Eq.(4.2.6) through Eq.(4.2.14).

$$\phi_g^{s1+} = \int_{y_{j-1/2}}^{y_{j+1/2}} p_g^{ij}(x_{i-1/2}, y)\, dy \tag{4.2.15}$$

$$\phi_g^{s2+} = \int_{x_{i-1/2}}^{x_{i+1/2}} p_g^{ij}(x, y_{j+1/2})\, dx \tag{4.2.16}$$

$$\phi_g^{s3+} = \int_{y_{j-1/2}}^{y_{j+1/2}} p_g^{ij}(x_{i+1/2}, y)\, dy \tag{4.2.17}$$

$$\phi_g^{s4+} = \int_{x_{i-1/2}}^{x_{i+1/2}} p_g^{ij}(x, y_{j-1/2})\, dx \tag{4.2.18}$$

$$J_g^{s1+} = \int_{y_{j-1/2}}^{y_{j+1/2}} -D_g^{ij} \frac{\partial}{\partial x} p_g^{ij}(x_{i-1/2}, y)\, dy \tag{4.2.19}$$

$$J_g^{s2+} = \int_{x_{i-1/2}}^{x_{i+1/2}} -D_g^{ij} \frac{\partial}{\partial y} p_g^{ij}(x, y_{j+1/2})\, dx \qquad (4.2.20)$$

$$J_g^{s3+} = \int_{y_{j-1/2}}^{y_{j+1/2}} -D_g^{ij} \frac{\partial}{\partial x} p_g^{ij}(x_{i+1/2}, y)\, dy \qquad (4.2.21)$$

$$J_g^{s4+} = \int_{x_{i-1/2}}^{x_{i+1/2}} -D_g^{ij} \frac{\partial}{\partial y} p_g^{ij}(x, y_{j-1/2})\, dx \qquad (4.2.22)$$

$$\overline{\Phi}_g^{ij+} = \frac{1}{V_{ij}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} p_g^{ij}(x, y)\, dy dx \qquad (4.2.23)$$

where

$\phi_g^{s\alpha+}$ = approximate face-integrated flux (α=1,...,4)

$J_g^{s\alpha+}$ = approximate face-integrated current (α=1,...,4)

$\overline{\Phi}_g^{ij+}$ = approximate node-averaged intra-nodal flux.

Since the approximate intra-nodal quantities expressed in Eq.(4.2.15) through Eq.(4.2.23) will not exactly be the same as the real intra-nodal quantities represented in Eq.(4.2.6) through Eq.(4.2.14), we define the error function as the sum of differences between the real intra-nodal quantities and the approximate intra-nodal quantities:

$$E = \sum_{k=1}^{9} [u_k^* - u_k^+]^2 \qquad (4.2.24)$$

where

E = error function

u_k^* = intra-nodal quantities defined in Eq.(4.2.6) through Eq.(4.2.14), (k=1,...,9)

u_k^+ = intra-nodal quantities defined in Eq.(4.2.15) through Eq.(4.2.23), (k=1,...,9).

The error function is to be minimized using a least-squares calculation to determine the polynomial coefficients for each node and group:

$$\frac{\partial E}{\partial a_{gmn}^{ij}} = 0 \qquad (4.2.25)$$

where m=0,1,2, n=0,1,2, and $m+n\leq 2$.

This procedure minimizes the least-squares difference between the intra-nodal quantities obtained from the global two-dimensional nodal solutions and those obtained from the evaluation of the polynomial function. For each node and group, the minimization of the error results in six linearly independent equations which can be easily solved for a_{gmn}^{ij}.

4.3 Results

In Section 4.2, the two-dimensional second-order flux expansion method for constructing flux shapes from nodal solutions was derived. In this section, the results when the second-order polynomial flux expansion method is applied to a two-dimensional, two-group, reactor benchmark problem are presented. Appendix C presents the program which calculates the intra-nodal flux distributions.

The accuracy of the flux reconstruction scheme, when applied to the two-dimensional benchmark problem is examined by comparing pin power distributions predicted by this method with those obtained by a conventional fine-mesh finite difference method.

Since the intra-nodal quantities appearing in Eq.(4.2.6) through Eq.(4.2.14) obtained by a nodal calculation are input data for reconstructing the fine-mesh flux, the accuracy of nodal solution affects the overall efficiency of reconstructing the pin power distribution. To overcome this difficulty, the input data is assumed to be obtained from the reference solution.

The numerical solution was obtained using the 2DB computer code. The fine-mesh local criticality calculations to obtain the assembly form function were also obtained using 2DB computer code.

4.3.1 The Modified EPRI-9R Benchmark Problem

The EPRI-9R problem models the baffle and reflector regions of PWR's. In the modified EPRI-9R problem, the baffle region is neglected, and the geometry of each assembly is also changed for simplicity. Each assembly consists of 10 x 10 array of homogenized pin cells. The size of each assembly and pin cell are 10 cm and 1 cm, respectively. The assembly heterogeneities, control rods or water holes, are indicated in Figure 4.3.1 by a shaded cell. If an assembly is unrodded, then all the shaded cells contain water. If an assembly is rodded, then all the shaded cells contain control rod material. The heterogeneous, two-group cross sections for the different materials used in this benchmark problem are given in Table 4.3.1.

The modified EPRI-9R problem consists of 32 fuel assemblies surrounded by a water reflector. The geometry and boundary conditions of problem are described in Figure 4.3.2.

4.3.2 The Reference 2DB Solution

For the problem benchmark, a fine-mesh solution was obtained with one mesh per fuel cell (i.e., 1.0 cm mesh spacing). The node average power densities obtained from 2DB are shown in Figure 4.3.3. The node face-integrated fluxes and node face-integrated currents were also calculated with 2DB. These have not been listed in this study due to the space required.

4.3.3 Assembly Form Function Calculation

The assembly form function calculation usually can be obtained by solving a single assembly criticality calculation with a zero current boundary condition. In addition, the assembly calculations may be improved by performing a four-assembly bundle calculation which takes into consideration the intra-nodal currents. This would require at most four geometric representations for assembly calculations.

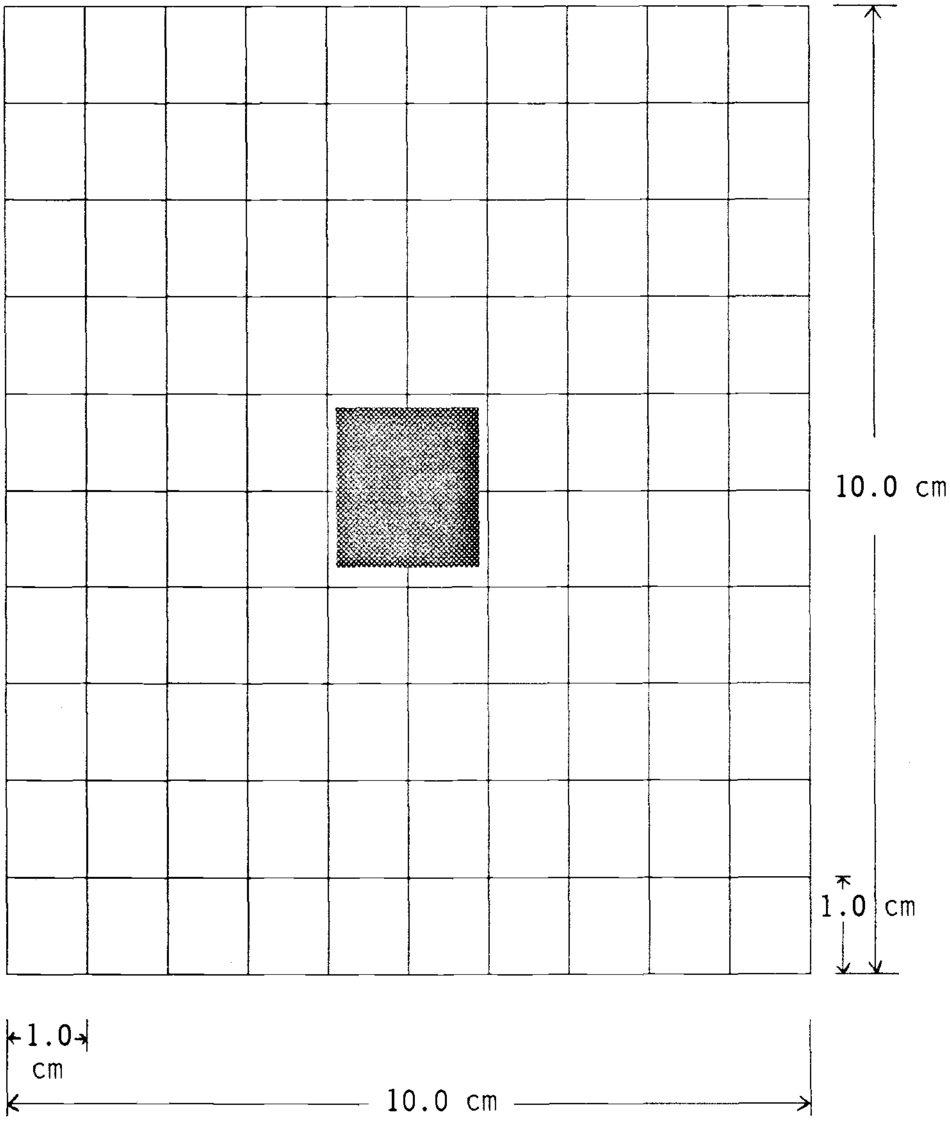
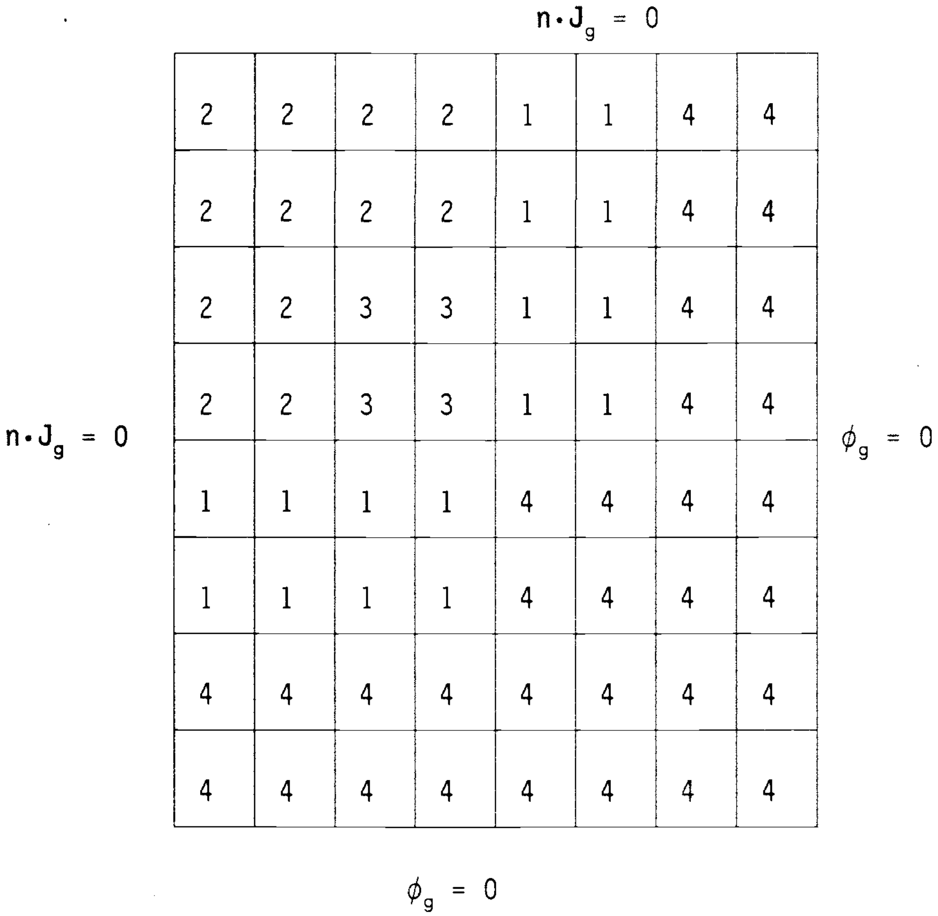


Figure 4.3.1 Geometry for Fuel Assembly

Table 4.3.1 Heterogeneous, Two-Group Cross Sections

Material	Group	D_g (cm)	Σ_{ag} (cm^{-1})	$\Sigma_{sg'g}$ (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})
Cell-1	1	1.500	0.0130	0.0200	0.0065
	2	0.400	0.1800	0.0	0.2400
Cell-2	1	1.500	0.0100	0.0200	0.0050
	2	0.400	0.1500	0.0	0.1800
water	1	1.700	0.0010	0.0350	0.0
	2	0.350	0.0500	0.0	0.0
Control Rod	1	1.113	0.0800	0.0038	0.0
	2	0.184	0.9600	0.0	0.0



2	2	2	2	1	1	4	4
2	2	2	2	1	1	4	4
2	2	3	3	1	1	4	4
2	2	3	3	1	1	4	4
1	1	1	1	4	4	4	4
1	1	1	1	4	4	4	4
4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4

Assembly Type	Fuel Composition	Non-Fuel Composition
1	Cell-1	Water
2	Cell-2	Water
3	Cell-1	Control-Rod
4		Water

Figure 4.3.2 Geometry for the Modified EPRI-9R Problem

(I,J)	I = 1	2	3	4	5	6
J = 1	1.711	1.615	1.442	1.209	1.109	0.721
2	1.615	1.508	1.310	1.085	1.004	0.655
3	1.442	1.310	1.198	0.958	0.804	0.529
4	1.209	1.085	0.958	0.727	0.609	0.379
5	1.109	1.004	0.804	0.609		
6	0.721	0.655	0.529	0.379		

Figure 4.3.3 Core Power Distribution for the Benchmark Problem

To obtain the assembly form function for assembly types 1, 2, and 3 that are not adjacent to a water reflector, the single assembly of configurations (a), (b), and (c) of Figure 4.3.4 are used, respectively. The uppermost left corner of configuration (d) of Figure 4.3.4 is used to calculate assembly form functions for assemblies neighboring a water reflector. In order to maintain consistency with the reference solution, the assembly solutions utilized 100 mesh points (10 x 10) per assembly.

4.3.4 Pin Power Reconstruction Results

Numerical results of the second-order polynomial flux expansion method for the benchmark problem is displayed in Figure 4.3.5. The first value in the figure is the node average power density obtained from the 2DB reference solution and the second number is the root-mean-square error for reconstructed pin powers between 2DB results and the current reconstruction method. The comparisons are quite good throughout most nodes and show that pin powers differ by only 1.36% root-mean-square for the all pins in the core. These results also illustrate that, for interior nodes, the current scheme predicts pin powers very accurately. The largest root-mean-square error obtained in the problem examined appeared at the core periphery node, row 5, column 4 (5,4) and is 3.464%, where the assembly power is lower and flux gradients are severe.

The pin power distributions for the three assembly positions ((1,1), (3,3), and (5,4)) are shown in Figure 4.3.6 through 4.3.8. The first, second, and third numbers in the figures are the 2DB reference solution, reconstructed pin power, and the relative errors between the two, respectively. The largest relative errors for the pins in the assembly

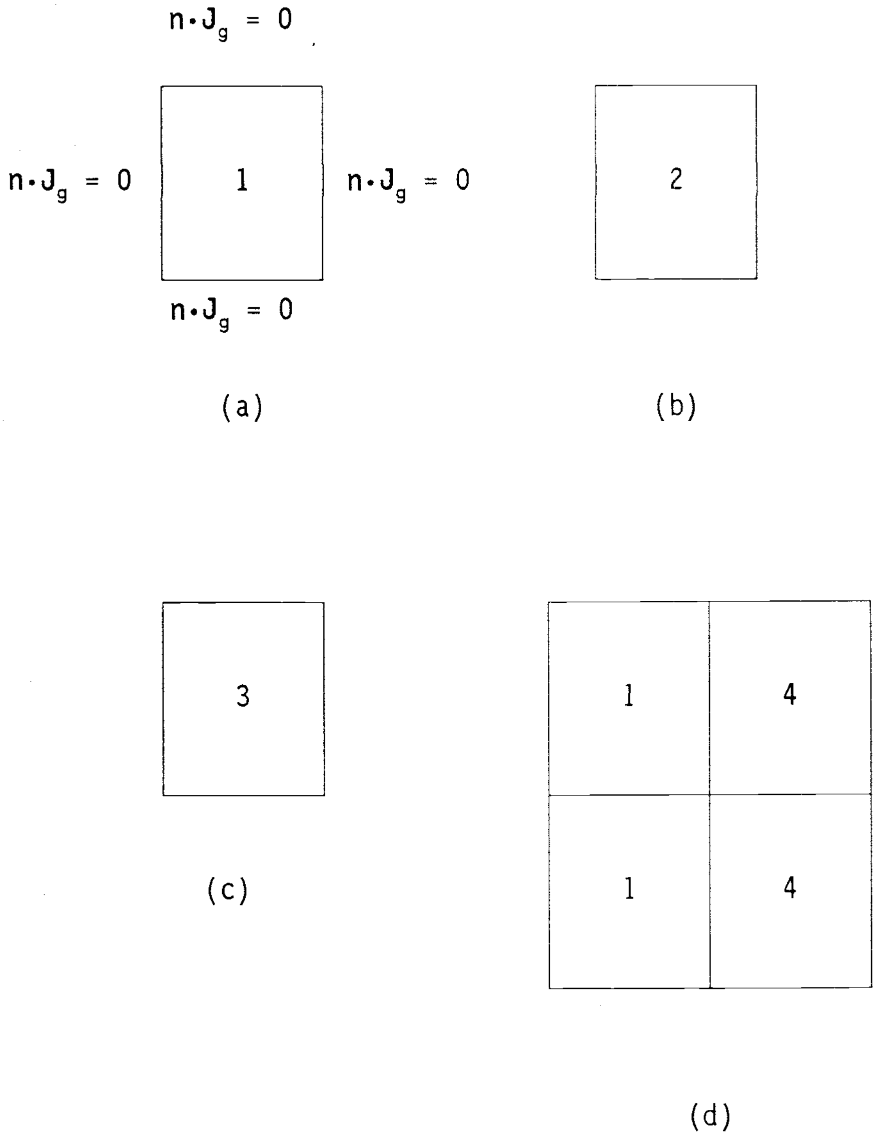


Figure 4.3.4 Assembly Form Function Calculation

(I,J)	I = 1	2	3	4	5	6
J = 1	1.711 0.206	1.615 0.209	1.442 0.228	1.209 0.830	1.109 0.920	0.721 2.206
2	1.615 0.209	1.508 0.323	1.310 1.056	1.085 1.261	1.004 1.017	0.655 2.105
3	1.442 0.228	1.310 1.056	1.198 1.302	0.958 1.111	0.804 0.808	0.529 2.328
4	1.209 0.830	1.085 1.261	0.958 1.111	0.727 1.091	0.609 3.464	0.379 2.863
5	1.109 0.920	1.004 1.017	0.804 0.808	0.609 3.464		
6	0.721 2.206	0.655 2.105	0.529 2.328	0.379 2.863		

<--- 2DB REFERENCE SOLUTION

<--- ROOT-MEAN-SQUARE ERROR(%)

Figure 4.3.5 Comparison of 2DB and Reconstructed Pin Power Distribution for the Benchmark Problem (Quarter-Core)

(I,J)

	I = 1	2	3	4	5	6	7	8	9	10
J = 1	1.775	1.774	1.775	1.777	1.778	1.773	1.762	1.749	1.737	1.729
	1.764	1.766	1.770	1.774	1.776	1.771	1.759	1.745	1.731	1.719
	0.642	0.435	0.279	0.173	0.112	0.097	0.129	0.213	0.351	0.542
2	1.774	1.774	1.779	1.788	1.794	1.789	1.772	1.753	1.738	1.727
	1.766	1.770	1.778	1.788	1.795	1.791	1.773	1.753	1.735	1.721
	0.435	0.230	0.079	-0.024	-0.082	-0.099	-0.069	0.014	0.154	0.349
3	1.775	1.779	1.794	1.819	1.845	1.840	1.803	1.768	1.743	1.729
	1.770	1.778	1.795	1.822	1.849	1.844	1.807	1.770	1.743	1.725
	0.280	0.079	-0.067	-0.164	-0.219	-0.240	-0.216	-0.135	0.008	0.209
4	1.777	1.788	1.819	1.881	1.973	1.967	1.865	1.792	1.751	1.731
	1.774	1.788	1.822	1.886	1.979	1.974	1.871	1.797	1.753	1.729
	0.174	-0.023	-0.163	-0.253	-0.308	-0.339	-0.323	-0.239	-0.089	0.121
5	1.778	1.794	1.845	1.973	0.000	0.000	1.955	1.818	1.758	1.732
	1.776	1.795	1.849	1.979	0.000	0.000	1.963	1.824	1.760	1.730
	0.114	-0.082	-0.219	-0.308	0.000	0.000	-0.398	-0.301	-0.138	0.082
6	1.773	1.789	1.840	1.967	0.000	0.000	1.950	1.813	1.753	1.727
	1.771	1.791	1.844	1.974	0.000	0.000	1.958	1.819	1.755	1.725
	0.098	-0.098	-0.240	-0.339	0.000	0.000	-0.420	-0.307	-0.133	0.095
7	1.762	1.772	1.803	1.865	1.955	1.950	1.848	1.777	1.736	1.716
	1.759	1.773	1.807	1.871	1.963	1.958	1.855	1.781	1.738	1.714
	0.131	-0.068	-0.217	-0.323	-0.398	-0.420	-0.367	-0.246	-0.068	0.163
8	1.749	1.753	1.768	1.792	1.818	1.813	1.777	1.742	1.718	1.704
	1.745	1.753	1.770	1.797	1.824	1.819	1.781	1.744	1.717	1.699
	0.214	0.015	-0.135	-0.239	-0.301	-0.307	-0.247	-0.124	0.056	0.288
9	1.737	1.738	1.743	1.751	1.758	1.753	1.736	1.718	1.703	1.693
	1.731	1.735	1.743	1.753	1.760	1.755	1.738	1.717	1.699	1.685
	0.351	0.154	0.008	-0.090	-0.139	-0.134	-0.068	0.055	0.235	0.470
10	1.729	1.727	1.729	1.731	1.732	1.727	1.716	1.704	1.693	1.685
	1.719	1.721	1.725	1.729	1.730	1.725	1.714	1.699	1.685	1.673
	0.543	0.348	0.208	0.120	0.081	0.094	0.163	0.287	0.469	0.709

a	<--- RECONSTRUCTED PIN POWER DISTRIBUTION
b	<--- 2DB REFERENCE SOLUTION
c	<--- RELATIVE ERROR(%) ([(a-b)/b]x100)

Figure 4.3.6 Comparison of 2DB and Reconstructed Pin Power Distribution for the Benchmark Problem (Assembly Position (1,1))

(I,J)

	I = 1	2	3	4	5	6	7	8	9	10
J = 1	1.629	1.587	1.542	1.496	1.453	1.419	1.392	1.367	1.341	1.311
	1.648	1.591	1.547	1.509	1.476	1.447	1.422	1.399	1.376	1.352
	-1.154	-0.252	-0.339	-0.892	-1.522	-1.949	-2.142	-2.284	-2.547	-3.022
2	1.587	1.544	1.494	1.441	1.392	1.358	1.338	1.320	1.298	1.270
	1.591	1.524	1.470	1.422	1.380	1.351	1.334	1.320	1.303	1.283
	-0.252	1.308	1.647	1.382	0.933	0.582	0.343	0.044	-0.406	-1.037
3	1.542	1.494	1.434	1.363	1.294	1.261	1.263	1.263	1.250	1.226
	1.547	1.470	1.403	1.335	1.272	1.243	1.246	1.249	1.244	1.229
	-0.341	1.646	2.230	2.074	1.669	1.460	1.392	1.068	0.504	-0.250
4	1.496	1.441	1.363	1.253	1.117	1.088	1.159	1.196	1.200	1.182
	1.509	1.422	1.335	1.230	1.103	1.074	1.138	1.179	1.191	1.183
	-0.895	1.380	2.074	1.846	1.232	1.302	1.818	1.465	0.760	-0.128
5	1.453	1.392	1.294	1.117	0.000	0.000	1.031	1.131	1.154	1.140
	1.476	1.380	1.272	1.103	0.000	0.000	1.006	1.113	1.146	1.145
	-1.525	0.931	1.668	1.232	0.000	0.000	2.549	1.648	0.670	-0.374
6	1.419	1.358	1.261	1.088	0.000	0.000	1.003	1.099	1.119	1.105
	1.447	1.351	1.243	1.074	0.000	0.000	0.976	1.083	1.116	1.115
	-1.952	0.580	1.459	1.302	0.000	0.000	2.690	1.440	0.274	-0.878
7	1.392	1.338	1.263	1.159	1.031	1.003	1.065	1.096	1.096	1.076
	1.422	1.334	1.246	1.138	1.006	0.976	1.045	1.088	1.101	1.094
	-2.144	0.341	1.391	1.818	2.549	2.690	1.848	0.732	-0.429	-1.604
8	1.367	1.320	1.263	1.196	1.131	1.099	1.096	1.090	1.074	1.048
	1.399	1.320	1.249	1.179	1.113	1.083	1.088	1.093	1.089	1.076
	-2.286	0.043	1.068	1.465	1.649	1.440	0.733	-0.251	-1.356	-2.519
9	1.341	1.298	1.250	1.200	1.154	1.119	1.096	1.074	1.049	1.020
	1.376	1.303	1.244	1.191	1.146	1.116	1.101	1.089	1.076	1.058
	-2.547	-0.406	0.504	0.761	0.672	0.275	-0.428	-1.355	-2.422	-3.581
10	1.311	1.270	1.226	1.182	1.140	1.105	1.076	1.048	1.020	0.988
	1.352	1.283	1.229	1.183	1.145	1.115	1.094	1.076	1.058	1.037
	-3.022	-1.036	-0.248	-0.125	-0.371	-0.875	-1.602	-2.517	-3.580	-4.754

a	<--- RECONSTRUCTED PIN POWER DISTRIBUTION
b	<--- 2DB REFERENCE SOLUTION
c	<--- RELATIVE ERROR(%) ([(a-b)/b]x100)

Figure 4.3.7 Comparison of 2DB and Reconstructed Pin Power Distribution for the Benchmark Problem (Assembly Position (3,3))

(I,J)

	I = 1	2	3	4	5	6	7	8	9	10
J = 1	0.720	0.713	0.704	0.691	0.676	0.660	0.647	0.650	0.690	0.805
	0.783	0.758	0.733	0.709	0.686	0.665	0.649	0.644	0.662	0.728
	-8.101	-5.905	-4.000	-2.446	-1.339	-0.699	-0.208	0.988	4.205	10.639
2	0.703	0.697	0.690	0.680	0.668	0.652	0.637	0.637	0.674	0.787
	0.766	0.742	0.720	0.699	0.679	0.659	0.642	0.639	0.666	0.756
	-8.274	-6.059	-4.204	-2.745	-1.724	-1.140	-0.795	-0.194	1.317	4.049
3	0.687	0.683	0.679	0.676	0.672	0.657	0.634	0.628	0.661	0.769
	0.749	0.726	0.708	0.694	0.682	0.663	0.639	0.632	0.661	0.761
	-8.238	-5.956	-4.057	-2.562	-1.494	-0.905	-0.699	-0.532	-0.017	1.039
4	0.672	0.670	0.672	0.684	0.708	0.693	0.644	0.623	0.648	0.752
	0.730	0.710	0.698	0.699	0.715	0.695	0.645	0.624	0.649	0.750
	-8.017	-5.647	-3.680	-2.114	-0.963	-0.299	-0.143	-0.185	-0.134	0.238
5	0.656	0.656	0.666	0.704	0.000	0.000	0.665	0.618	0.635	0.734
	0.710	0.692	0.688	0.716	0.000	0.000	0.661	0.615	0.631	0.728
	-7.617	-5.173	-3.167	-1.625	0.000	0.000	0.598	0.618	0.591	0.792
6	0.638	0.638	0.648	0.685	0.000	0.000	0.647	0.601	0.618	0.714
	0.686	0.668	0.664	0.692	0.000	0.000	0.638	0.591	0.606	0.699
	-7.022	-4.527	-2.515	-1.036	0.000	0.000	1.464	1.754	1.902	2.220
7	0.618	0.616	0.618	0.629	0.651	0.637	0.592	0.573	0.597	0.693
	0.659	0.639	0.628	0.630	0.645	0.625	0.576	0.555	0.575	0.665
	-6.200	-3.658	-1.609	-0.093	0.949	1.886	2.726	3.292	3.697	4.262
8	0.598	0.594	0.590	0.587	0.584	0.570	0.551	0.546	0.575	0.672
	0.630	0.609	0.592	0.580	0.570	0.551	0.527	0.518	0.543	0.629
	-5.133	-2.525	-0.377	1.294	2.546	3.609	4.546	5.276	5.921	6.780
9	0.578	0.573	0.566	0.558	0.548	0.534	0.522	0.523	0.555	0.651
	0.601	0.580	0.560	0.542	0.524	0.505	0.489	0.486	0.511	0.593
	-3.818	-1.125	1.156	3.017	4.500	5.724	6.757	7.631	8.509	9.683
10	0.560	0.554	0.546	0.536	0.524	0.511	0.501	0.504	0.537	0.631
	0.572	0.551	0.530	0.510	0.491	0.473	0.459	0.457	0.482	0.559
	-2.256	0.542	2.969	5.020	6.715	8.106	9.264	10.293	11.410	12.917

a	<--- RECONSTRUCTED PIN POWER DISTRIBUTION
b	<--- 2DB REFERENCE SOLUTION
c	<--- RELATIVE ERROR(%) ([(a-b)/b]x100)

Figure 4.3.8 Comparison of 2DB and Reconstructed Pin Power Distribution for the Benchmark Problem (Assembly Position (5,4))

positions (1,1), (3,3), and (5,4) are 0.709%, -4.754%, and 12.917%, respectively, and occur at the corner pins, row 10, column 10.

All computations were performed on a COMPAQ DESKPRO 386/25 personal computer. The computing times for the second-order polynomial flux expansion method and the 2DB reference solution are listed in Table 4.3.2.

Table 4.3.2 Summary of Computing Times for the Second-Order Polynomial Expansion Method and the 2DB Reference Solution

		CPU	
Second-Order Polynomial Expansion Method	Coarse-Mesh Nodal Solution	9.1 sec	1.4 Min.
	Assembly Form Function	50.1 sec	
	Polynomial Coefficients Calculations	25.6 sec (0.8 x 32)	
2DB Reference Solution			45.0 Min.

The cpu time to obtain the assembly function from 2DB local criticality calculation was about 50.0 seconds. About 9.1 seconds were required to obtain the coarse-mesh nodal solution.The pin power reconstruction calculation required approximately 0.8 second per each node. Thus the total cpu time for pin power reconstruction using the second-order polynomial flux expansion method was approximately 1.4 minutes. Since the total execution time for generating the 2DB reference was 45.0 minutes, the overall reconstruction scheme was demonstrated to be computationally about 30 times faster than a fine-mesh finite difference calculation in order to achieve comparable accuracy.

4.4 Conclusions

The objective of this study was to develop an accurate and computationally efficient method for fine-mesh flux shapes and power distributions from nodal solutions. From the results presented in Section 4.3, the least square second-order polynomial flux expansion method was shown to be quite accurate and efficient for reconstructing the reactor pointwise power when a coarse-mesh nodal solution is known. Thus, the method developed in this study to calculate pointwise power distributions has been demonstrated to be a very attractive alternative to the fine-mesh calculation for the LWR applications.

4.4.1 Recommendations for Future Improvements

During the course of this research, many interesting topics have not been investigated. This section contains a brief description of these potential research areas.

1. Additional work is needed for assemblies located at the core periphery. At the periphery of the core, large errors in the prediction of fine-mesh powers are due to either intra-nodal flux shapes assumed to be second-order polynomials or the assembly form function calculations with zero current boundary conditions or both. The second-order polynomials could be partially or totally modified to match accurately the intra-nodal flux shapes (i.e., sines, sinh's). The coefficient of these functions could be obtained by methods similar to those employed to determine the polynomial coefficients. Albedo boundary conditions can be used in the four-assembly bundle

calculations rather than zero net current boundary conditions to treat the reflector effects more precisely.

2. A scheme to improve the prediction accuracy of pointwise flux and power could be developed by employing the node corner point fluxes. These values can be calculated by improving nodal solution code.

3. The model is limited to reactor cores with two-dimensional geometry. Extension to a three-dimensional model would require extensive program modification, but would extend its applicability.

Chapter References

1. K. Koebke, "A New Approach to Homogenization and Group Condensation," Proc. IAEA Technical Committee Mtg. Homogenization Methods in Reactors Physics, IAEA-TECDOC-231, International Atomic Energy Agency, Lugano, Italy (1978).

2. K. S. Smith, A. F. Henry, and R.A.Loretz, "Determination of Homogenized Diffusion Theory Parameters for Coarse Mesh Nodal Analysis," Proc. ANS Topl. Mtg., Advances in Reactor Physics and Shielding., Sun Valley, ID, September 14-19, 1980, p. 294, American Nuclear Society (1980).

3. K. Koebke, "Advances in Homogenization and Dehomogenization," International Tropical Meeting on Advances in Mathematical Method for the Solution of Nuclear Engineering Problems, vol. 2, p. 59, Munich, 27-29 April (1981).

4. K. Koebke and M. R. Wagner, "The Determination of the Pin Power Distribution in a Reactor Core on the Basis of Nodal Coarse Mesh Calculations," Atomkernenergie(ATKE), 30, 2, 126 (1977).

5. K. Koebke and L. Hetzelt, "On the Reconstruction of Local Homogeneous Neutron Flux and Current Distributions of Light Water Reactors from Nodal Schemes," Nucl. Sci. Eng., 91, 123-131 (1985).

6. K. R. Rempe and K. S. Smith, "SIMULATE-3: Power Distributions Detector Response Modeling," ANS Trans., 54, 355-357 (1987).

7. K. S. Smith, "Application of Diffusion Theory Methods to PWR Analysis," ANS Trans., 60, 329-332 (1989).

8. A. Jonsson, S. Grill, J. R. Rec, "Nodal Imbedded Calculation for the Retrieval of Local Power Peaking from Coarse Mesh Reactor Analysis," International Tropical Meeting on Advances in Mathematical Method for the Solution of Nuclear Engineering Problems, vol. 2, p. 23, Munich, 27-29 April (1981).

9. Y. S. Liu, A. Meliksetian, and A. L. Casadei, "Pin Power Prediction in the Westinghouse Advanced Nodal Code," ANS Trans., 53, 246-247 (1986).

10. R. W. Hardie, W. W. Little, Jr., "2DB, A Two-Dimensional Diffusion Theory Burnup Code," BNWL-831, (August 1969).

11. B. O. Cho, "A Comparison of Three Coarse-Mesh Nodal Methods for Boiling Water Reactor Analysis," M.S. thesis, Oregon State University (June 1988).

REFERENCES

Ancona, A., Becker, M., Harris, D. R., Menezes, A. DaC., Robinson, M. A., and Ver Plank, D. M. "Unified Analysis of Nodal Methods," Trans. Am. Nucl. Soc., 24, 444 (1976).

Bankoff, S. G., "A Variable Density Single-Fluid Model for Two-Phase Flow with Particular Reference to Steam-Water Flow," J. Heat Transfer, 82, 265 (1960).

Blenski, T., Gadomski, A., and Mika, J. "Higher Order Prompt-Jump Approximation in Reactor Kinetics," Nucl. Sci. Eng., 66, 277-283 (1978).

Borresen, S. "A Simplified, Coarse-mesh, Three Dimensional Diffusion Scheme for Calculating the Gross-Power Distribution in a Boiling Water Reactor," Nucl. Sci. Eng., 44, 37 (1971).

Bottoni, G. P. "Fast Factorization Procedure Solving the Multidimensional Reactor Dynamics Equations," ANS Trans., 26, 229-230 (1977).

Cadwell, W. R., Henry, A. F., and Vigilotti, A. J. "WIGLE - A Program for the Solution of the Two-Group, Space-Time Diffusion Equations in Slab Geometry," WAPD-TM-416, Bettis Atomic Power Laboratory (1964).

Chang, J. and C. H. Kim, "A Modified Borresen's Coarse-Mesh Method for a Spatial Neutron Kinetics Problem of Light Water Reactor," NUREG/CP-0080, Vol. 1 (1986).

Chao, Y. A. and Penkrot, J. A. "A nodal Method for Direct Intra-Node Distribution Calculation," Proc. Topl. Mtg., Advances in Reactor Physics, NUREG/CP-0034 (1982).

Chen, G. S. and J. M. Christenson, "Solution of the Two-Dimensional Space-Time Reactor Kinetics Equation by a Locally One-Dimensional Method," ANS Trans., 50, 536-537 (1985).

Cho, B. O. "A Comparison of Three Coarse-Mesh Nodal Methods for Boiling Water Reactor Analysis," M.S. thesis, Oregon State University (June 1988).

Clark, M., Jr. and Hansen, K. F. Numerical Methods of Reactor Analysis, Academic Press, New York, N. Y. (1964).

Delp, D. L., Fischer, D. L., Harriman, J. M., and Stedwell, M. J. "FLARE: A Three-Dimensional Boiling Water Reactor Simulator," GEAP-4598, General Electric Company (1964).

Delp, D. L., et al., "FLARE-A Three Dimensional Boiling Water Reactor Simulator," GEAP-4598, General Electric Company (1964).

Devooght, J. and Mund, E. "Generalized Quasi-static method for Nuclear Reactor Space-Time Kinetics," Nucl. Sci. Eng., 76, 10-17 (1980).

Dodds, H. L., Jr. "Accuracy of the Quasistatic Method for Two-Dimensional Thermal Reactor transient with Feedback," Nucl. Sci. Eng., 59, 271-281 (1976).

Duderstadt, J. J. and Hamilton, L. J., Nuclear reactor Analysis, John Wiley & Sons, Inc., (1976).

Georgia Power Company "Edwin I. Hatch Nuclear Plant Unit Number 2, Final Safety Analysis Report," Section 4.

Goldstein, R. and Shotkin, L. M. "Use of the Prompt-Jump Approximation in Fast-Reactor Kinetics," Nucl. Sci. Eng., 38, 94-103 (1969).

Griffith, R. E. and R. A. Stewart, "A Nonlinear Programming Technique for the Optimization of Continuous Processing Systems," Management Sci., 7, 379 (1961).

Grossman, L. M., Hennart, J. P., and Meade, D. "Finite Element Collocation Methods for Space-Time Reactor Dynamics," ANS Trans., 41, 311-312 (1982).

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

Haling, R. K. "Operating Strategy for Maintaining an Optimum Power Distribution throughout Life," USAEC Report TID-7072 (1964).

Hansen, K. F. and S. R. Johnson, "GAKIN - A Program for the Solution of the One-dimensional, Multigroup, Space-time Dependent Diffusion Equations," USAEC Rpt. GA-7543 (1967).

Hardie, R. W., W. W. Little, Jr., "2DB, A Two-Dimensional Diffusion Theory Burnup Code," BNWL-831, (August 1969).

Hatteberg, R. L. "A Simulated Two-Group Method for Computing Boiling Water Reactor Power Distributions," M.S. thesis, Oregon State University (June 1978).

Hayase, T. and H. Motoda, "Boiling water Control Rod Programming Using Heuristic and Mathematical Methods," Nucl. Tech., 48, 91 (1980).

Jonsson, A., S. Grill, J. R. Rec, "Nodal Imbedded Calculation for the Retrieval of Local Power Peaking from Coarse Mesh Reactor Analysis," International Tropical Meeting on Advances in Mathematical Method for the Solution of Nuclear Engineering Problems, Vol. 2, p. 23, Munich, 27-29 April (1981).

Kang, C. M. and Hansen, K. F. "Finite Element Methods for Reactor Analysis," Nucl. Sci. Eng., 51, 456 (1973).

Kaplan, S. "Synthesis Methods in Reactor Analysis," Advan. Nucl. Sci. Technology, 3, 233 (1965).

Kawai, T., H. Motoda, T. Kiguchi, and M. Ozawa, "A Method for Generating a Control Rod Program for Boiling Water Reactors," Nucl. Tech., 28, 108 (1976).

Kim, C. H. and S. H. Levine, "A Modified Borresen's Coarse-Mesh Computation for a Three-Dimensional Pressurized Water Reactor Benchmark Problem," Nucl. Tech., 61, 49 (1983).

Koebke, K. "A New Approach to Homogenization and Group Condensation," Proc. IAEA Technical Committee Mtg. Homogenization Methods in Reactors Physics, IAEA-TECDOC-231, International Atomic Energy Agency, Lugano, Italy (1978).

Koebke, K. "Advances in Homogenization and Dehomogenization," International Tropical Meeting on Advances in Mathematical Method for the Solution of Nuclear Engineering Problems, Vol. 2, p. 59, Munich, 27-29 April (1981).

Koebke, K. and M. R. Wagner, "The Determination of the Pin Power Distribution in a Reactor Core on the Basis of Nodal Coarse Mesh Calculations," Atomkernenergie(ATKE), 30, 2, 126 (1977).

Koebke, K. and L. Hetzelt, "On the Reconstruction of Local Homogeneous Neutron Flux and Current Distributions of Light Water Reactors from Nodal Schemes," Nucl. Sci. Eng., 91, 123-131 (1985).

Langenbuch, S., W. Maurer, and W. Werner, "Coarse-Mesh Nodal diffusion method for the Analysis of Space-Time Effects in Large Light Water Reactors," Nucl. Sci. Eng. 63, 437-456 (1977).

Liu, Y. S., A. Meliksetian, and A. L. Casadei, "Pin Power Prediction in the Westinghouse Advanced Nodal Code," ANS Trans., 53, 246-247 (1986).

Mangasarin, O. L. "Duel Feasible Direction Algorithms," Technical Summary Report No. 1173, Mathematics Research Center, University of Wisconsin (Feb. 1972).

Mingle, John O. "In-Core Fuel Management via Perturbation Theory," Nucl. Tech., 27, 248 (1975).

Motoda, H. and T. Kawai, "A Theory of Control-Rod Programming Optimization in Two-Region Reactors," Nucl. Sci. Eng., 39, 114 (1970).

Motoda, H. "Optimal Control Rod Programming of Light Water Reactors in Equilibrium Fuel Cycle," Nucl. Sci. Eng., 46, 88 (1971).

Motoda, H. "Optimization of Control Rod Programming and Loading Pattern in Multiregion Nuclear Reactor by the Method of Approximation Programming," Nucl. Sci. Eng., 49, 515 (1972).

Peaceman, D. W. and Rachford, H. H., Jr. "The Numerical Solution of Parabolic Differential Equations," J. Soc. Ind. Appl. Math., 3, 28 (1955).

Reed, W. H. and K. F. Hansen, "Alternating Direction Methods for Reactor Kinetics Equations," Nucl. Sci. Eng., 41, 431 (1970).

Rempe, K. R. and K. S. Smith, "SIMULATE-3: Power Distributions Detector Response Modeling," ANS Trans., 54, 355-357 (1987).

Rich, Elaine Artificial Intelligence, McGraw-Hill, Inc., 72, 1983.

Rich, Elaine Artificial Intelligence, McGraw-Hill, Inc., 75, 1983.

Roechlich, R. "Current Problems in Multi-dimensional Reactor Calculations," Proc. Conf. Mathematical models and Computational Techniques for Analysis of Nuclear Systems, CONF-730414, Vol. 2, p. VII-1 (1973).

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

Shober, R. A., Sims, R. N., and Henry, A. F. "Two Nodal Methods for Solving Time-Dependent Group Diffusion Equations," Nucl. Sci. Eng., 64, 582-592 (1977).

Smith, K. S., A. F. Henry, and R.A.Loretz, "Determination of Homogenized Diffusion Theory Parameters for Coarse Mesh Nodal Analysis," Proc. ANS Topl. Mtg., Advances in Reactor Physics and Shielding., Sun Valley, ID, September 14-19, 1980, p. 294, American Nuclear Society (1980).

Smith, K. S. "Application of Diffusion Theory Methods to PWR Analysis," ANS Trans., 60, 329-332 (1989).

Smith, K. S., Henry, A. F., and Loretz, R. A. "Determination of Homogenized Diffusion Theory Parameters for Coarse Mesh Nodal Analysis," Proc. ANS Topl. Mtg., Advances in Reactor Physics and Shielding., Sun Valley, ID, September 14-19, 1980, p. 294, American Nuclear Society (1980).

Snyder, B. and E. E. Lewis, "Optimal Control Rod Policies for an Operating Cycle of a Simulated BWR Core," Proc. 1973 Conf. Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, CONF-730414-PI, II-56, U.S. Atomic Energy Commission (1973).

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

Terny, W. B. and H. Fenech, "Control Rod Programing Optimization Using Dynamic Programing," Nucl. Sci. Eng., 39, 109 (1970).

Tokumasu, S., M. Ozawa, H. Hiranuma, and M. Yokomi, "A Mathematical Method for Boiling Water Reactor Control Rod Programming," Nucl. Tech., 71, 568 (1985).

Tong, L. S., "Prediction of Departure from Nuclear Boiling for an Axially Non-Uniform Heat Flux Distribution," J. Nuclear Energy, Part A and B, (1967).

Wade, W. C. and W. B. Terney, "Optimal Control of Nuclear Reactor Depletion," Nucl. Sci. Eng., 45, 199 (1971).

Werner, W. "Mathematical Methods Applicable to Space Time Kinetics," Proc. Specialist Meeting Reactivity Effects in Large Power Reactors, Ispra, Italy, October 28-30, 1970, Commission of the European Communities, EUR 4731 f-e, Luxembourg (1972).

Wright, A. L., K. F. Hansen, and D. R. Ferguson, "Application of Alternating Direction Implicit Methods to Space-Dependent Kinetics Equations," Nucl. Sci. Eng.,44, 239 (1971).

Wu, C. D. and Weisman, J. "An Efficient Computational Technique for Light Water Reactor Core Dynamics," Nuclear Technology, 81 June (1988).

Yasinsky, J. B., Natelson, M., and Hageman, L. A. "TWIGL - A Program to Solve the Two-Dimensional, Two-Group, Space-Time Neutron Diffusion Equations with Temperature Feedback," WAPD-TM-743, Bettis Atomic Power Laboratory (1968).

Yasinsky, J. B. and A. F. Henry, "Some Numerical Experiments Concerning Space-time Reactor Kinetics Behavior," Nucl. Sci. Eng., 22, 171 (1965).

APPENDICES

Appendix A

The TLAR Acceleration Method Subroutine

```
      SUBROUTINE RELPWW
C     ** RELPOW CALCULATES THE RELATIVE POWER FOR EACH NODE  *********
C     ** USING THE R-FACTOR *******
C     ** TRANSIENT CASE
C  IMPLICIT, SOURCE OVER-RELAXATION
C
C
      COMMON/NUCLER/SIGA1(0:18,0:18,0:25),SIGA2(0:18,0:18,0:25)
     $     ,SIGFNU1(0:18,0:18,0:25),SIGFNU2(0:18,0:18,0:25)
     $     ,SIGS12(0:18,0:18,0:25),SIGR1(0:18,0:18,0:25)
     $     ,D1(0:18,0:18,0:25),D2(0:18,0:18,0:25)
      COMMON/COEFF/C1F(18,18,0:25),C1T(18,18,0:25)
     $            ,C2F(18,18,0:25),C2T(18,18,0:25)
     $            ,C3F(18,18,0:25),C3T(18,18,0:25)
     $            ,C4F(18,18,0:25),C4T(18,18,0:25)
     $            ,C5F(18,18,0:25),C5T(18,18,0:25)
     $            ,C6F(18,18,0:25),C6T(18,18,0:25)
     $            ,C7F(18,18,0:25),C7T(18,18,0:25)
      COMMON/CXDATA/CXA1(9,9),CXA2(9,9),CXF1(9,9),CXF2(9,9),CXD1(9,9)
     1 ,CXD2(9,9),CXSC(9,9),PPMB,XNHYD,H2ODEN,CXBMIC,CXBMC
      COMMON/DIMEN/NXA,LNXF(0:18),LNX(0:18),AREA,SIZE,MAXO,MAXI,MAXFT
     $,TESTOT,TESTIN,ALPHA,TFUEL,AA(2),CC(2),DWATER(2),EXTB2(18,2)
     $,EXTB4(18,2),SIGAW(2),SIGSLW,CXBMACR,HEIGHT,NZA
C **** MINIMUM DIMENSIONS ON PHI SHOULD BE PHI(NX+1,NY+1,NZ+1)
      COMMON/ABC/B2(18,18),B4(18,18),PHI1(0:18,0:18,0:25)
     $,PHI2(0:18,0:18,0:25),PHIA1(0:18,0:18,0:25),PHIA2(0:18,0:18,0:25)
     $,TF1(18,18,0:25),TF2(18,18,0:25),PHIOLD1(0:18,0:18,0:25)
     $,PHIOLD2(0:18,0:18,0:25)
      COMMON/CTN/ICNTRL,NOTCH(0:18,0:18),WORTH(18)
      COMMON/DELAYED/BETA(6),BTA(6),SUMBTA,RAMDA(6)
     $,CD1(18,18,0:25),CD2(18,18,0:25),CD3(18,18,0:25)
     $,CD4(18,18,0:25),CD5(18,18,0:25),CD6(18,18,0:25)
      COMMON/TRANSNT/OFT1(18,18,0:25),OFT2(18,18,0:25),SPA(18,18,0:25)
     $,ITRAN,IDG,PWRSMO,PWRSMN,VEL1,VEL2,HT,TRNTIME,TIMEMAX,PRNTIME,TTM
      COMMON/FUEL/ITYPE(0:18,0:18),EX(0:18,0:18)
     $,RPOW(18,18,25),RPOWXY(18,18)
C C=COOLANT F=FUEL OLD=OLD TIME  BASE=BASE TEMP
      REAL KF,KC,MF,LOUF,LOUC,LG,MDNBR
      COMMON/TH1/PMWTH,H(18,18,25),TC(18,18,25),TF(18,18,25)
     $,HOLD(18,18,25),TCOLD(18,18,25),TFOLD(18,18,25),QQQ(18,18,25)
     $,X(18,18,25),VOID(18,18,25),HFLX(18,18,25),MDNBR
     $,TFBASE(18,18,25),TCBASE(18,18,25),ROU(18,18,25),ROUBASE(18,18,25)
      COMMON/TH2/PSIA,PITCH,RCO,RFI,CAREA,NPINS,DHEIGHT,TFLOW,MF
      COMMON/TH3/TIN,TSAT,HIN,HSAT,HFG,AC1,AC2
      COMMON/TH4/LOUF,LOUC,KF,KC,HG,HC,LG,VF,VG
      EQUIVALENCE (NXA,NYA)
      COMMON/TEST/OD2(18,18,25),OSIGA2(18,18,25),ZZZ(18,18,25)
C ZZZ(I,J,K)=THERMAL LEAKAGE-TO-ABSORPTION RATIO
      DIMENSION PHIO1(0:18,0:18,0:25),R(0:18,0:18,0:25)
C R(I,J,K)=RESIDUAL OF ITERATION
      LOGICAL CONVSR
C
      CONVSR=.FALSE.
      TRNTIME=0.0
1234  TRNTIME=TRNTIME+HT
      TTM=TTM+HT
      CALL CSINIT
      ITOUT=0
```

```
C
C R1.5 GROUP ROUTINE
      DO 6300 K=1,NZA
      DO 6300 I=1,NYA
      DO 6300 J=1,LNX(I)
6300  C7F(I,J,K)=SIGR1(I,J,K)+C1F(I,J,K)+C2F(I,J,K)+C3F(I,J,K)
     1             +1./VEL1/HT+C4F(I,J,K)+C5F(I,J,K)+C6F(I,J,K)
     2 -(1.-SUMBTA)*(SIGFNU1(I,J,K)+SIGFNU2(I,J,K)*SIGS12(I,J,K)
     3                   /(1./VEL2/HT+(1.+ZZZ(I,J,K))*SIGA2(I,J,K)))
      TLFISL=0.
      DO 3100 K=1,NZA
      DO 3100 I=1,NYA
      DO 3100 J=1,LNXF(I)
      SPA(I,J,K)=0.
3100  SPA(I,J,K)=SPA(I,J,K)+RAMDA(1)/(1.+RAMDA(1)*HT/2.)*CD1(I,J,K)
     1                  +RAMDA(2)/(1.+RAMDA(2)*HT/2.)*CD2(I,J,K)
     2                  +RAMDA(3)/(1.+RAMDA(3)*HT/2.)*CD3(I,J,K)
     3                  +RAMDA(4)/(1.+RAMDA(4)*HT/2.)*CD4(I,J,K)
     4                  +RAMDA(5)/(1.+RAMDA(5)*HT/2.)*CD5(I,J,K)
     5                  +RAMDA(6)/(1.+RAMDA(6)*HT/2.)*CD6(I,J,K)
      DO 603 K=1,NZA
      DO 603 I=1,NYA
      DO 603 J=1,LNX(I)
      TF1(I,J,K)=SIGFNU1(I,J,K)*PHI1(I,J,K)
      TLFISL=TLFISL+TF1(I,J,K)
      OFT1(I,J,K)=PHIOLD1(I,J,K)/VEL1/HT+(1.-SUMBTA)*SIGFNU2(I,J,K)
     $ /VEL2/HT*PHIOLD2(I,J,K)/(1./VEL2/HT+(1.+ZZZ(I,J,K))*SIGA2(I,J,K))
603   CONTINUE
C
C** SOURCE OVER-RELAXATION ROUTINE
C** CALCULATE ALPHA VALUE
C
      IF(.NOT. CONVSR) THEN
           ITER=0
           OLDSUM=1.E-20
           OLDALP=1.0
           ALPHA=1.0
  33       SUM=0.
           DO 44 K=1,NZA
           DO 44 I=1,NYA
           DO 44 J=1,LNX(I)
      R(I,J,K)=(C3F(I,J,K)*PHI1(I-1,J,K)+C4F(I,J,K)*PHI1(I+1,J,K)
     1  -C7F(I,J,K)*PHI1(I,J,K)+SPA(I,J,K)+OFT1(I,J,K)
     2  +C1F(I,J,K)*PHI1(I,J-1,K)+C2F(I,J,K)*PHI1(I,J+1,K)
     3  +C5F(I,J,K)*PHI1(I,J,K-1)+C6F(I,J,K)*PHI1(I,J,K+1))/C7F(I,J,K)
      PHI1(I,J,K)=PHI1(I,J,K)+R(I,J,K)
   44      SUM=SUM+R(I,J,K)*R(I,J,K)
           RLAMDA=SQRT(SUM/OLDSUM)
           ALPHA=2./(1.+SQRT(ABS(1.-RLAMDA)))
           OLDSUM=SUM
           TEST=ABS((OLDALP-ALPHA)/ALPHA)
           OLDALP=ALPHA
           ITER=ITER+1
           IF(TEST.GT.1.E-3) GOTO 33
           CONVSR=.TRUE.
           WRITE(*,*) 'CONVSR=TRUE','   ITER=',ITER,'  ALPHA=',ALPHA
      ENDIF
C
C     *** BEGINING OF INNER AND OUTER ITERATIONS  *******************
C
191   ITIN=0
192   DO 193 K=1,NZA
      DO 193 I=1,NYA
      DO 193 J=1,LNX(I)
      R(I,J,K)=(C3F(I,J,K)*PHI1(I-1,J,K)+C4F(I,J,K)*PHI1(I+1,J,K)
     1  -C7F(I,J,K)*PHI1(I,J,K)+SPA(I,J,K)+OFT1(I,J,K)
     2  +C1F(I,J,K)*PHI1(I,J-1,K)+C2F(I,J,K)*PHI1(I,J+1,K)
     3  +C5F(I,J,K)*PHI1(I,J,K-1)+C6F(I,J,K)*PHI1(I,J,K+1))/C7F(I,J,K)
      PHI1(I,J,K)=PHI1(I,J,K)+ALPHA*R(I,J,K)
193   CONTINUE
      ITIN=ITIN+1
C      IF(ITIN.LT.MAXI) GOTO 192
      ITOUT=ITOUT+1
C
C     *** CHECK CONVERGENCE AND PRINT OUT OF DATA  *******************
C
      TLFIS=0.
195   DO 196 K=1,NZA
      DO 196 I=1,NYA
      DO 196 J=I,LNXF(I)
196   TLFIS=TLFIS+SIGFNU1(I,J,K)*PHI1(I,J,K)
      TESTO=ABS((TLFIS-TLFISL)/TLFIS)
      TLFISL=TLFIS
250   IF (ITOUT.GE.MAXO) GO TO 255
      IF (ITOUT.LE.3)   GO TO 191
      IF (TESTO-TESTOT) 260,260,191
255   WRITE(*,*) 'WARNING: MAXMIMUM OUTER ITERATIONS EXCEEDED.  CALCULATI
     $ON CONTINUES...'
260   DO 153 K=1,NZA
      DO 153 I=1,NYA
      DO 153 J=1,LNX(I)
      PHI2(I,J,K)=(1./VEL2/HT*PHIOLD2(I,J,K)+SIGS12(I,J,K)*PHI1(I,J,K))
     $ /(1./VEL2/HT+(1.+ZZZ(I,J,K))*SIGA2(I,J,K))
153   CONTINUE
C
C 2 GROUP ROUTINE
      DO 7300 K=1,NZA
      DO 7300 I=1,NYA
      DO 7300 J=1,LNX(I)
      C7F(I,J,K)=SIGR1(I,J,K)+C1F(I,J,K)+C2F(I,J,K)+C3F(I,J,K)
     1           +1./VEL1/HT+C4F(I,J,K)+C5F(I,J,K)+C6F(I,J,K)
7300  C7T(I,J,K)=SIGA2(I,J,K)+C1T(I,J,K)+C2T(I,J,K)+C3T(I,J,K)
     1           +1./VEL2/HT+C4T(I,J,K)+C5T(I,J,K)+C6T(I,J,K)
      TLFISL=0.
      DO 7603 K=1,NZA
      DO 7603 I=1,NYA
      DO 7603 J=1,LNX(I)
      TF1(I,J,K)=(SIGFNU1(I,J,K)*PHI1(I,J,K)+SIGFNU2(I,J,K)*PHI2(I,J,K))
     $           *(1.-SUMBTA)
      TLFISL=TLFISL+TF1(I,J,K)
      OFT1(I,J,K)=PHIOLD1(I,J,K)/VEL1/HT
7603  OFT2(I,J,K)=PHIOLD2(I,J,K)/VEL2/HT
C
C     *** BEGINING OF INNER AND OUTER ITERATIONS  ********************
C
      ITOUT=0
7191  ITIN=0
7192  DO 7193 K=1,NZA
      DO 7193 I=1,NYA
      DO 7193 J=1,LNX(I)
      PHI1(I,J,K)=(C3F(I,J,K)*PHI1(I-1,J,K)+C4F(I,J,K)*PHI1(I+1,J,K)
     1 +TF1(I,J,K)+OFT1(I,J,K)+SPA(I,J,K)+C1F(I,J,K)*PHI1(I,J-1,K)
     2 +C2F(I,J,K)*PHI1(I,J+1,K)+C5F(I,J,K)*PHI1(I,J,K-1)
     3 +C6F(I,J,K)*PHI1(I,J,K+1))/C7F(I,J,K)
7193  CONTINUE
      ITIN=ITIN+1
      IF(ITIN.LT.MAXI) GO TO 7192
C
      ITIN=0
      IF(ITOUT.EQ.0) GO TO 7787
```

```
C
C    ** ACCELERATION BY OVER RELAXATION *************************
C
      DO 7753 K=1,NZA
      DO 7753 I=1,NYA
      DO 7753 J=1,LNX(I)
 7753 TF2(I,J,K)=TF2(I,J,K)+ALPHA*(SIGS12(I,J,K)*PHI1(I,J,K)-TF2(I,J,K))
      GO TO 7790
 7787 DO 7755 K=1,NZA
      DO 7755 I=1,NYA
      DO 7755 J=1,LNX(I)
 7755 TF2(I,J,K)=SIGS12(I,J,K)*PHI1(I,J,K)
 7790 CONTINUE
C
 7138 DO 7153 K=1,NZA
      DO 7153 I=1,NYA
      DO 7153 J=1,LNX(I)
      PHI2(I,J,K)=(C3T(I,J,K)*PHI2(I-1,J,K)+C4T(I,J,K)*PHI2(I+1,J,K)
     1 +TF2(I,J,K)+OFT2(I,J,K)+C1T(I,J,K)*PHI2(I,J-1,K)
     2 +C2T(I,J,K)*PHI2(I,J+1,K)+C5T(I,J,K)*PHI2(I,J,K-1)
     3 +C6T(I,J,K)*PHI2(I,J,K+1))/C7T(I,J,K)
 7153 CONTINUE
      ITIN=ITIN+1
      IF (ITIN.LT.MAXI) GO TO 7138
C
      ITOUT=ITOUT+1
      IF (ITOUT.EQ.1) GO TO 7554
C
C      ***  ACCELERATION BY OVER RELAXATION  **************************
C
      DO 7552 K=1,NZA
      DO 7552 I=1,NYA
      DO 7552 J=1,LNX(I)
 7552 TF1(I,J,K)=TF1(I,J,K)+ALPHA*((SIGFNU1(I,J,K)*PHI1(I,J,K)
     1        +SIGFNU2(I,J,K)*PHI2(I,J,K))*(1.-SUMBTA)-TF1(I,J,K))
      GO TO 7558
 7554 DO 7556 K=1,NZA
      DO 7556 I=1,NYA
      DO 7556 J=1,LNX(I)
 7556 TF1(I,J,K)=(SIGFNU1(I,J,K)*PHI1(I,J,K)+SIGFNU2(I,J,K)*PHI2(I,J,K))
     1            *(1.-SUMBTA)
 7558 CONTINUE
C
C      *** CHECK CONVERGENCE AND PRINT OUT OF DATA  *******************
C
      TLFIS=0.
 7195 DO 7196 K=1,NZA
      DO 7196 I=1,NYA
      DO 7196 J=1,LNXF(I)
 7196 TLFIS=TLFIS+(SIGFNU1(I,J,K)*PHI1(I,J,K)
     1             +SIGFNU2(I,J,K)*PHI2(I,J,K))*(1.-SUMBTA)
      TESTO=ABS((TLFIS-TLFISL)/TLFIS)
      TLFISL=TLFIS
 7250 IF (ITOUT.GE.MAXO) GO TO 7255
      IF (ITOUT.LE.3)   GO TO 7191
      IF (TESTO-TESTOT) 7260,7260,7191
 7255 WRITE(*,*) 'WARNING: MAXMIMUM OUTER ITERATIONS EXCEEDED.  CALCULATI
     $ON CONTINUES...'

C
 7260 TLFIS=0.
      DO 350 K=1,NZA
      DO 350 I=1,NYA
      DO 350 J=1,LNX(I)
      RPOW(I,J,K)=PHI1(I,J,K)*SIGFNU1(I,J,K)+PHI2(I,J,K)*SIGFNU2(I,J,K)
```

```
350   TLFIS=TLFIS+RPOW(I,J,K)
C
      PWRSMN=TLFIS
      PRATIO=PWRSMN/PWRSMO
      WRITE(*,970) TRNTIME,PRATIO,ITOUT,TESTO
C
      CALL PRECUSR
C
      IF(AMOD((TRNTIME+1.E-5),PRNTIME) .GE. (HT/5.)) GO TO 1234
      WRITE(*,420)
C    AVGFIS=TOTAL POWER PER SUB-ASSEMBLY
      AVGFIS=TLFIS/TFUEL
      WRITE(*,430)
      DO 370 I=1,NYA
      DO 365 J=1,LNXF(I)
      RPOWXY(I,J)=0.
      DO 360 K=1,NZA
360   RPOWXY(I,J)=RPOWXY(I,J)+RPOW(I,J,K)
365   RPOWXY(I,J)=RPOWXY(I,J)/AVGFIS
370   WRITE(*,440) I,(RPOWXY(I,J)*PRATIO,J=1,LNXF(I))
      IF((TRNTIME+1.E-5) .LT. TIMEMAX) GO TO 1234
      RETURN
C
420   FORMAT (//,10X,' *****  RELATIVE POWER FOR I,J POSITION  *****'/)
430   FORMAT(8X,'1        2        3        4        5        6       7
     1 8       9        10       11        12       13       14       15',/)
440   FORMAT(' ',I2,2X,17F8.3)
970   FORMAT(1X,' ** TIME=',E11.6,' SEC',1X,'P(t)/PO=',E11.6,2X,
     1  'OUTERS=',I4,' POWER CNV=',E10.5,' **')
      END
```

Appendix B

Optimum Control Rod Pattern Searcher Subroutine

```
C********
C******** SENSITIVITY CALCULATION ROUTINE
C********
208   IF (OPEAKN.GT.2.2 .AND. OMCHFR.LT.1.9) ICASE=1
      IF (OPEAKN.LT.2.2 .AND. OMCHFR.LT.1.9) ICASE=2
      IF (OPEAKN.GT.2.2 .AND. OMCHFR.GT.1.9) ICASE=3
      IF (OPEAKN.LT.2.2 .AND. OMCHFR.GT.1.9) ICASE=4
      DO 535 I=1,NMAX
      DO 535 M=1,3
      SKEFF(I,M)=0.
      SZJ(I,M)=0.
      DO 536 L=1,NODES
      SCHFR(I,M,L)=0.
536   SPOWER(I,M,L)=0.

      IF (ICASE.EQ.1) THEN
      DO 537 L=1,NP
537   SPIJ(I,M,L)=0.
      DO 538 K=1,NZ
538   SPZ(I,M,K)=0.
      ENDIF

535   CONTINUE

      DO 500 M=1,2
        IF(M.EQ.1) IDELROD=1
        IF(M.EQ.2) IDELROD=-1
      DO 500 I=1,NMAX
        IF(OCTNROD(I).EQ.NZ .AND. M.EQ.1) GOTO 500
        IF(OCTNROD(I).EQ.0  .AND. M.EQ.2) GOTO 500
          DO 502 IR=1,NMAX
 502      CTNROD(IR)=OCTNROD(IR)
       CTNROD(I)=OCTNROD(I)+IDELROD
       CALL GNOTCH(GROUP,CTNROD)
        DO 303 L=1,NODES
        T(L)=OT(L)
        F(L)=OF(L)
        VOID(L)=OVOID(L)
303     POWER(L)=OPOWER(L)
      CALL STG
      CALL EDIT (5)
C      CALL EDIT (3)
      IF (NOVOID) GO TO 234
C      CALL EDIT (1)
C      CALL EDIT (4)
C      CALL EDIT (2)
      SKEFF(I,M)=KEFF-OKEFF
        PRINT*,'KEFF=',KEFF
        PRINT*,'SKEFF(I,M)=',SKEFF(I,M)
        DO 305 L=1,NODES
        SCHFR(I,M,L)=CHFR(L)-OCHFR(L)
  305   SPOWER(I,M,L)=POWER(L)-OPOWER(L)

        IF (ICASE.EQ.4) THEN
        SZJ(I,M)=0.
        DO 1306 L=1,NODES
1306    SZJ(I,M)=SZJ(I,M)+(POWER(L)-PHALING(L))*(POWER(L)-PHALING(L))
        SZJ(I,M)=SZJ(I,M)/FLOAT(NODES)-OZJ
        ENDIF
```

```
         IF (ICASE.EQ.1) THEN
         DO 306 L=1,NP
  306    SPIJ(I,M,L)=PIJ(L)-OPIJ(L)
         DO 307 K=1,NZ
  307    SPZ(I,M,L)=PZ(K)-OPZ(K)
         ENDIF

500   CONTINUE

C**************
C************** END OF SENSITIVITY CALCULATION
C**************
      CALL STRTG(GROUP,NMAX,ICASE)

234   CONTINUE
300   STOP
      END



      SUBROUTINE STRTG(GROUP,NMAX,ICASE)
      COMMON /HEADR /NTITLE(20), AVEXP, NTSTEP
      COMMON /CSINP /BUMASK(12), VMASK(3),  DFI(108),  DTI(108)
     $              ,SAFI(108), SATI(108), NSFFI(108), NSFTI(108),
     $               SDSI(108), IEXPTS   , IVDPTS    , ITSETS    ,
     $               KSFTI(108), KSFFI(108),                          A50
     $               DIMTS    , DIMEX    , DIMVD     , NOTCH(140),
     $               TABSET(3360), WORTH(3,12,3)
      INTEGER DIMTS, DIMEX, DIMVD, TABSET
      REAL    NSFFI, NSFTI, KSFFI, KSFTI
      COMMON / FUEL / FTYPE(140), BU(3360), VOID(3360)
      INTEGER FTYPE
C
      COMMON /NODAL / DF  (3360),  DT  (3360),  KSFF  (3360)
     $,              KSFT(3360),  TFR (3360),  NSF1G (3360)
     $,              C2  (3360),  C4  (3360),  C6    (3220)
     $,              C7  (3360),  SDS (3360)                          A60
     $,              CC2 (3360),  CC4 (3360),  CC6   (3220)
     $,              CC7 (3360),  NSFF(3360),  NSFT  (3360)
C
      REAL KSFF, KSFT, NSF1G, NSFF, NSFT
      COMMON /GEOM /NODES     , NP       , NDSMNP     , NX
     $,             NZ        , NXM1     , GX         ,GZ
     $,              NCOL  (13)
      COMMON /BC    / AF1, AF3, AF5, AF6, AFQ(13),
     $               BF1, BF3, BF5, BF6, BFQ(13),
     $               AT1, AT3, AT5, AT6, ATQ(13),
     $               BT1, BT3, BT5, BT6, BTQ(13)
      EQUIVALENCE (AFQ(1), AF2), (AFQ(2), AF4),
     $            (BFQ(1), BF2), (BFQ(2), BF4),
     $            (ATQ(1), AT2), (ATQ(2), AT4),
     $            (BTQ(1), BT2), (BTQ(2), BT4)
      COMMON /CONVRG/ MAXI, MAXO, MAXP, EPSO, EPSP, ALPHA, SOR
      LOGICAL SOR
C
      COMMON /FLUX /  POWER(3360), FASTA(3360), THERMA(3360), F(3360)
      COMMON /FLUX2/  KEFF, T(3360),CHFR(3360),MCHFR,IMM,JMM,KMM,LMM
     $,              PEAKN,IPP,JPP,KPP,LPP
      COMMON /PROB / INTTYP, QTRKOR, NOHAL, NOCROD, NOVOID, IEDIT(6)
     $,               NOXE, FIRST
      LOGICAL QTRKOR, NOHAL, NOCROD, NOVOID, NOXE, FIRST
      COMMON /WEIGHT/  R,  CF,  CT,  RCF,  RCT,  B1N,  B2N
      COMMON /THERMO/ HFG, XIN, AC1, AC2, AC3, AREA, AVFLUX, AVPWR,PSIA
     $,              B1(4),B2(4),B3(4),CTYPE(140),FLOW(140),TFLOW
     $,               PCTPWR
      INTEGER CTYPE
      COMMON /XENON / PO, PRSA1(108), PRSA2(108), SAXE(108)
C
      REAL MWTH, KEFF, MCHFR
      COMMON /CROD /NROOS(6,2), RODIND(140)
      INTEGER ROOINO
      CHARACTER SKIP*60,GROUP*1
      COMMON/SNS/ASMEX(24),WORD(20),YIX(3,12),PHALING(3360),OVOID(3360)
     $,OF(3360),OT(3360),OCHFR(3360),OPOWER(3360),SKEFF(10,3),SZJ(10,3)
     $,SPOWER(10,3,3360),ESPOWER(3360),SCHFR(10,3,3360),ESCHFR(3360)
     $,OPIJ(140),OPZ(24),SPIJ(10,3,140),SPZ(10,3,24),ESPIJ(140),ESPZ(24)
     $,CTNROO(10),OCTNROO(10),IDELR(10)
      INTEGER CTNROO,OCTNROD
      COMMON /PWRIJ/ PIJ(140),PEAKPIJ,IPIJ,JPIJ,LPIJ,PZ(24),PEAKPZ,KPZ
      COMMON/OLO/OKEFF,OPEAKN,OMCHFR,OZJ,OPKPZ,OPKPIJ
      integer chkpnt(3360),chkpt(140)
      INTEGER SI(10),SI1,SI2,SI3,SI4,SI5,SI6,SI7,SI8,SI9,SI10
      INTEGER TSOVLY(3  ,24)
      LOGICAL FLAG
      OATA  FLAG/.FALSE./
C
C
      IF (ICASE.EQ.1) GOTO 1111
      IF (ICASE.EQ.2) GOTO 2222
      IF (ICASE.EQ.3) GOTO 3333
      IF (ICASE.EQ.4) GOTO 4444

C**************
C************** PROMISING CONTROL ROD PATTERN GENERATION
C**************
1111  maxpnt=0
      do 1427 i=1,nodes
      if(opower(i).gt.(opeakn/2.)) then
       maxpnt=maxpnt+1
       chkpnt(maxpnt)=i
      endif
1427   continue
      maxpt=0
      do 1428 i=1,np
      if(opij(i).gt.(opkpij/2.)) then
       maxpt=maxpt+1
       chkpt(maxpt)=i
      endif
1428   continue
      NP1=NP+1
      PMAX1=1.E3
      EFFKMIN=0.997
      EFFKMAX=1.003
      ICOUNT=0
      IF(NMAX.EQ.8) GOTO 1889
      IF(NMAX.EQ.10) GOTO 1789
1889   ICOUNT=ICOUNT+1
      00 16001 I1=1,3
       ESKEFF1=OKEFF+SKEFF(1,I1)
      00 16002 I2=1,3
       ESKEFF2=ESKEFF1+SKEFF(2,I2)
      00 16003 I3=1,3
       ESKEFF3=ESKEFF2+SKEFF(3,I3)
      00 16004 I4=1,3
       ESKEFF4=ESKEFF3+SKEFF(4,I4)
      DO 16005 I5=1,3
       ESKEFF5=ESKEFF4+SKEFF(5,I5)
      00 16006 I6=1,3
       ESKEFF6=ESKEFF5+SKEFF(6,I6)
      DO 16007 I7=1,3
       ESKEFF7=ESKEFF6+SKEFF(7,I7)
      DO 16008 I8=1,3
       ESKEFF8=ESKEFF7+SKEFF(8,I8)
C
      IF (ESKEFF8.LT. EFFKMIN .OR. ESKEFF8.GT. EFFKMAX) GOTO 16008
C
      ESPZ(KPZ)=OPZ(KPZ)+SPZ(1,I1,KPZ)+SPZ(2,I2,KPZ)
     $     +SPZ(3,I3,KPZ)+SPZ(4,I4,KPZ)+SPZ(5,I5,KPZ)
     $     +SPZ(6,I6,KPZ)+SPZ(7,I7,KPZ)+SPZ(8,I8,KPZ)
      IF(ESPZ(KPZ).GT.OPKPZ) GOTO 16008
C
      ESPIJ(LPIJ)=OPIJ(LPIJ)+SPIJ(1,I1,LPIJ)+SPIJ(2,I2,LPIJ)
     $     +SPIJ(3,I3,LPIJ)+SPIJ(4,I4,LPIJ)+SPIJ(5,I5,LPIJ)
     $     +SPIJ(6,I6,LPIJ)+SPIJ(7,I7,LPIJ)+SPIJ(8,I8,LPIJ)
      IF(ESPIJ(LPIJ).GT.OPKPIJ) GOTO 16008
C
      ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
     $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
     $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
      IF(ESPOWER(LPP).GT.OPEAKN) GOTO 16008
C
      DO 16100 K=1,NZ
      ESPZ(K)=OPZ(K)+SPZ(1,I1,K)+SPZ(2,I2,K)
     $     +SPZ(3,I3,K)+SPZ(4,I4,K)+SPZ(5,I5,K)
     $     +SPZ(6,I6,K)+SPZ(7,I7,K)+SPZ(8,I8,K)
      IF(ESPZ(K).GT.OPKPZ) GOTO 16008
16100  CONTINUE
C
      DO 16102 IJ=chkpt(1),chkpt(maxpt)
      ESPIJ(IJ)=OPIJ(IJ)+SPIJ(1,I1,IJ)+SPIJ(2,I2,IJ)
     $     +SPIJ(3,I3,IJ)+SPIJ(4,I4,IJ)+SPIJ(5,I5,IJ)
     $     +SPIJ(6,I6,IJ)+SPIJ(7,I7,IJ)+SPIJ(8,I8,IJ)
      IF(ESPIJ(IJ).GT.OPKPIJ) GOTO 16008
16102  CONTINUE
C
      PMAX2=-2.0
      DO 16104 L=chkpnt(1),chkpnt(maxpnt)
      ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $     +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $     +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
       IF(ESPOWER(L).GT.PMAX2)  PMAX2=ESPOWER(L)
16104  CONTINUE
C
      IF(PMAX2.LT.OPEAKN.AND.PMAX2.LT.PMAX1) THEN
       IF(ABS(ESKEFF8-OKEFF).LE.1.E-6) GOTO 16008
       FLAG=.TRUE.
       PMAX1=PMAX2
       SI(1)=I1
       SI(2)=I2
       SI(3)=I3
       SI(4)=I4
       SI(5)=I5
       SI(6)=I6
       SI(7)=I7
       SI(8)=I8
      ENDIF
C
16008  CONTINUE
16007  CONTINUE
16006  CONTINUE
16005  CONTINUE
16004  CONTINUE
16003  CONTINUE
16002  CONTINUE
```

```
16001  CONTINUE
C
       GOTO 1575
C
C** END OF PATTERN B
C
1789    ICOUNT=ICOUNT+1
       DO 17001 I1=1,3
        ESKEFF1=OKEFF+SKEFF(1,I1)
       DO 17002 I2=1,3
        ESKEFF2=ESKEFF1+SKEFF(2,I2)
       DO 17003 I3=1,3
        ESKEFF3=ESKEFF2+SKEFF(3,I3)
       DO 17004 I4=1,3
        ESKEFF4=ESKEFF3+SKEFF(4,I4)
       DO 17005 I5=1,3
        ESKEFF5=ESKEFF4+SKEFF(5,I5)
       DO 17006 I6=1,3
        ESKEFF6=ESKEFF5+SKEFF(6,I6)
       DO 17007 I7=1,3
        ESKEFF7=ESKEFF6+SKEFF(7,I7)
       DO 17008 I8=1,3
        ESKEFF8=ESKEFF7+SKEFF(8,I8)
       DO 17009 I9=1,3
        ESKEFF9=ESKEFF8+SKEFF(9,I9)
       DO 17010 I10=1,3
        ESKEF10=ESKEFF9+SKEFF(10,I10)
C
       IF (ESKEF10.LT. EFFKMIN .OR. ESKEF10.GT. EFFKMAX) GOTO 17010
C
       ESPZ(KPZ)=OPZ(KPZ)+SPZ(1,I1,KPZ)+SPZ(2,I2,KPZ)
      $     +SPZ(3,I3,KPZ)+SPZ(4,I4,KPZ)+SPZ(5,I5,KPZ)
      $     +SPZ(6,I6,KPZ)+SPZ(7,I7,KPZ)+SPZ(8,I8,KPZ)
      $     +SPZ(9,I9,KPZ)+SPZ(10,I10,KPZ)
       IF(ESPZ(KPZ).GT.OPKPZ) GOTO 17010
C
       ESPIJ(LPIJ)=OPIJ(LPIJ)+SPIJ(1,I1,LPIJ)+SPIJ(2,I2,LPIJ)
      $     +SPIJ(3,I3,LPIJ)+SPIJ(4,I4,LPIJ)+SPIJ(5,I5,LPIJ)
      $     +SPIJ(6,I6,LPIJ)+SPIJ(7,I7,LPIJ)+SPIJ(8,I8,LPIJ)
      $     +SPIJ(9,I9,LPIJ)+SPIJ(10,I10,LPIJ)
       IF(ESPIJ(LPIJ).GT.OPKPIJ) GOTO 17010
C
       ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
      $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
      $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
      $     +SPOWER(9,I9,LPP)+SPOWER(10,I10,LPP)
       IF(ESPOWER(LPP).GT.OPEAKN) GOTO 17010
C
       DO 17100 K=1,NZ
       ESPZ(K)=OPZ(K)+SPZ(1,I1,K)+SPZ(2,I2,K)
      $     +SPZ(3,I3,K)+SPZ(4,I4,K)+SPZ(5,I5,K)
      $     +SPZ(6,I6,K)+SPZ(7,I7,K)+SPZ(8,I8,K)
      $     +SPZ(9,I9,K)+SPZ(10,I10,K)
       IF(ESPZ(K).GT.OPKPZ) GOTO 17010
17100  CONTINUE
C
       DO 17102 IJ=chkpt(1),chkpt(maxpt)
       ESPIJ(IJ)=OPIJ(IJ)+SPIJ(1,I1,IJ)+SPIJ(2,I2,IJ)
      $     +SPIJ(3,I3,IJ)+SPIJ(4,I4,IJ)+SPIJ(5,I5,IJ)
      $     +SPIJ(6,I6,IJ)+SPIJ(7,I7,IJ)+SPIJ(8,I8,IJ)
      $     +SPIJ(9,I9,IJ)+SPIJ(10,I10,IJ)
       IF(ESPIJ(IJ).GT.OPKPIJ) GOTO 17010
17102  CONTINUE
C
       PMAX2=-2.0
      DO 17104 L=chkpnt(1),chkpnt(maxpnt)
      ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $     +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $     +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
     $     +SPOWER(9,I9,L)+SPOWER(10,I10,L)
       IF(ESPOWER(L).GT.PMAX2)  PMAX2=ESPOWER(L)
17104  CONTINUE
      IF(PMAX2.LT.OPEAKN.AND.PMAX2.LT.PMAX1) THEN
       IF(ABS(ESKEF10-OKEFF).LE.1.E-6) GOTO 17010
       FLAG=.TRUE.
       PMAX1=PMAX2
       SI(1)=I1
       SI(2)=I2
       SI(3)=I3
       SI(4)=I4
       SI(5)=I5
       SI(6)=I6
       SI(7)=I7
       SI(8)=I8
       SI(9)=I9
       SI(10)=I10
      ENDIF
C
17010  CONTINUE
17009  CONTINUE
17008  CONTINUE
17007  CONTINUE
17006  CONTINUE
17005  CONTINUE
17004  CONTINUE
17003  CONTINUE
17002  CONTINUE
17001  CONTINUE
C
C** END OF PATTERN A
C
1575    DO 1253 I=1,NMAX
       IF(SI(I).EQ.1) IDELR(I)=1
       IF(SI(I).EQ.2) IDELR(I)=-1
       IF(SI(I).EQ.3) IDELR(I)=0
1253    CONTINUE
        DO 16200 IR=1,NMAX
        CTNROD(IR)=OCTNROD(IR)+IDELR(IR)
        IF(CTNROD(IR).EQ.(NZ+1)) CTNROD(IR)=NZ
        IF(CTNROD(IR).EQ.-1) CTNROD(IR)=0
16200    CONTINUE
C
       IF(FLAG) THEN
         WRITE(8,2424) GROUP,NMAX,ICASE
         WRITE(8,2426) (CTNROD(IR),IR=1,NMAX)
2424   FORMAT(A1,I10,I10)
2426     FORMAT(10I3)
       ELSE
          IF(ICOUNT.EQ.1) THEN
           EFFKMIN=0.995
           EFFKMAX=1.005
           IF(NMAX.EQ.8) GOTO 1889
           IF(NMAX.EQ.10) GOTO 1789
          ENDIF
          IF(ICOUNT.EQ.2) THEN
           OPKPIJ=OPKPIJ+0.03
           IF(NMAX.EQ.8) GOTO 1889
           IF(NMAX.EQ.10) GOTO 1789
          ENDIF
          IF(ICOUNT.EQ.3) THEN
          OPKPZ=OPKPZ+0.2
          IF(NMAX.EQ.8) GOTO 1889
          IF(NMAX.EQ.10) GOTO 1789
         ENDIF
         IF(ICOUNT.EQ.4) THEN
          WRITE(8,*) ' NO MORE SUGGESTION --- END OF HILL CLIMBING'
         ENDIF
      ENDIF
C***********
C*********** END OF CONTROL ROD PATTERN GENERATION
C***********

      GOTO 300
C**************
C************** PROMISING CONTROL ROD PATTERN GENERATION
C**************
2222  maxpnt=0
      do 2427 i=1,nodes
      if(ochfr(i).lt.(omchfr+2.) .OR. OPOWER(I).GT.(2.2/2.)) then
       maxpnt=maxpnt+1
       chkpnt(maxpnt)=i
      endif
2427   continue
      NP1=NP+1
      CHFMIN1=-1.E3
      EFFKMIN=0.997
      EFFKMAX=1.003
      ICOUNT=0
      IF(NMAX.EQ.8) GOTO 2889
      IF(NMAX.EQ.10) GOTO 2789
2889   ICOUNT=ICOUNT+1
      DO 26001 I1=1,3
       ESKEFF1=OKEFF+SKEFF(1,I1)
      DO 26002 I2=1,3
       ESKEFF2=ESKEFF1+SKEFF(2,I2)
      DO 26003 I3=1,3
       ESKEFF3=ESKEFF2+SKEFF(3,I3)
      DO 26004 I4=1,3
       ESKEFF4=ESKEFF3+SKEFF(4,I4)
      DO 26005 I5=1,3
       ESKEFF5=ESKEFF4+SKEFF(5,I5)
      DO 26006 I6=1,3
       ESKEFF6=ESKEFF5+SKEFF(6,I6)
      DO 26007 I7=1,3
       ESKEFF7=ESKEFF6+SKEFF(7,I7)
      DO 26008 I8=1,3
       ESKEFF8=ESKEFF7+SKEFF(8,I8)
C
      IF (ESKEFF8.LT. EFFKMIN .OR. ESKEFF8.GT. EFFKMAX) GOTO 26008
C
      ESCHFR(LMM)=OCHFR(LMM)+SCHFR(1,I1,LMM)+SCHFR(2,I2,LMM)
     $       +SCHFR(3,I3,LMM)+SCHFR(4,I4,LMM)+SCHFR(5,I5,LMM)
     $       +SCHFR(6,I6,LMM)+SCHFR(7,I7,LMM)+SCHFR(8,I8,LMM)
      IF(ESCHFR(LMM).LT.OMCHFR) GOTO 26008
C
      ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
     $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
     $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
      IF(ESPOWER(LPP).GT.2.2) GOTO 26008
C
      CHFMIN2=1.E4
      DO 26100 L=chkpnt(1),chkpnt(maxpnt)
      ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $     +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $     +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
```

```
        IF(ESPOWER(L).LT.2.2) THEN
          ESCHFR(L)=OCHFR(L)+SCHFR(1,I1,L)+SCHFR(2,I2,L)
     $         +SCHFR(3,I3,L)+SCHFR(4,I4,L)+SCHFR(5,I5,L)
     $         +SCHFR(6,I6,L)+SCHFR(7,I7,L)+SCHFR(8,I8,L)
          IF(ESCHFR(L).LT.CHFMIN2) CHFMIN2=ESCHFR(L)
        ENDIF
26100   CONTINUE
       IF(CHFMIN2.GT.OMCHFR.AND.CHFMIN2.GT.CHFMIN1) THEN
        IF(ABS(ESKEFF8-OKEFF).LE.1.E-6) GOTO 26008
        FLAG=.TRUE.
        CHFMIN1=CHFMIN2
        SI(1)=I1
        SI(2)=I2
        SI(3)=I3
        SI(4)=I4
        SI(5)=I5
        SI(6)=I6
        SI(7)=I7
        SI(8)=I8
       ENDIF
C
26008   CONTINUE
26007   CONTINUE
26006   CONTINUE
26005   CONTINUE
26004   CONTINUE
26003   CONTINUE
26002   CONTINUE
26001   CONTINUE
C
       GOTO 2575
C
C** END OF PATTERN B
C
2789    ICOUNT=ICOUNT+1
       DO 27001 I1=1,3
        ESKEFF1=OKEFF+SKEFF(1,I1)
       DO 27002 I2=1,3
        ESKEFF2=ESKEFF1+SKEFF(2,I2)
       DO 27003 I3=1,3
        ESKEFF3=ESKEFF2+SKEFF(3,I3)
•      DO 27004 I4=1,3
        ESKEFF4=ESKEFF3+SKEFF(4,I4)
       DO 27005 I5=1,3
        ESKEFF5=ESKEFF4+SKEFF(5,I5)
       DO 27006 I6=1,3
        ESKEFF6=ESKEFF5+SKEFF(6,I6)
       DO 27007 I7=1,3
        ESKEFF7=ESKEFF6+SKEFF(7,I7)
       DO 27008 I8=1,3
        ESKEFF8=ESKEFF7+SKEFF(8,I8)
       DO 27009 I9=1,3
        ESKEFF9=ESKEFF8+SKEFF(9,I9)
       DO 27010 I10=1,3
        ESKEF10=ESKEFF9+SKEFF(10,I10)
C
       IF (ESKEF10.LT. EFFKMIN .OR. ESKEF10.GT. EFFKMAX) GOTO 27010
C
       ESCHFR(LMM)=OCHFR(LMM)+SCHFR(1,I1,LMM)+SCHFR(2,I2,LMM)
     $        +SCHFR(3,I3,LMM)+SCHFR(4,I4,LMM)+SCHFR(5,I5,LMM)
     $        +SCHFR(6,I6,LMM)+SCHFR(7,I7,LMM)+SCHFR(8,I8,LMM)
     $        +SCHFR(9,I9,LMM)+SCHFR(10,I10,LMM)
       IF(ESCHFR(LMM).LT.OMCHFR) GOTO 27010
C
       ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
     $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
     $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
     $     +SPOWER(9,I9,LPP)+SPOWER(10,I10,LPP)
      IF(ESPOWER(LPP).GT.2.2) GOTO 27010
C
      CHFMIN2=1.E4
      DO 27100 L=chkpnt(1),chkpnt(maxpnt)
      ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $     +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $     +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
     $     +SPOWER(9,I9,L)+SPOWER(10,I10,L)
       IF(ESPOWER(L).LT.2.2) THEN
         ESCHFR(L)=OCHFR(L)+SCHFR(1,I1,L)+SCHFR(2,I2,L)
     $       +SCHFR(3,I3,L)+SCHFR(4,I4,L)+SCHFR(5,I5,L)
     $       +SCHFR(6,I6,L)+SCHFR(7,I7,L)+SCHFR(8,I8,L)
     $       +SCHFR(9,I9,L)+SCHFR(10,I10,L)
         IF(ESCHFR(L).LT.CHFMIN2) CHFMIN2=ESCHFR(L)
       ENDIF
27100  CONTINUE
      IF(CHFMIN2.GT.OMCHFR.AND.CHFMIN2.GT.CHFMIN1) THEN
       IF(ABS(ESKEF10-OKEFF).LE.1.E-6) GOTO 27010
       FLAG=.TRUE.
       CHFMIN1=CHFMIN2
       SI(1)=I1
       SI(2)=I2
       SI(3)=I3
       SI(4)=I4
       SI(5)=I5
       SI(6)=I6
       SI(7)=I7
       SI(8)=I8
       SI(9)=I9
       SI(10)=I10
      ENDIF
C
27010  CONTINUE
27009  CONTINUE
27008  CONTINUE
27007  CONTINUE
27006  CONTINUE
27005  CONTINUE
27004  CONTINUE
27003  CONTINUE
27002  CONTINUE
27001  CONTINUE
C
C** END OF PATTERN A
C
2575   DO 2253 I=1,NMAX
      IF(SI(I).EQ.1) IDELR(I)=1
      IF(SI(I).EQ.2) IDELR(I)=-1
      IF(SI(I).EQ.3) IDELR(I)=0
2253   CONTINUE
       DO 26200 IR=1,NMAX
       CTNROD(IR)=OCTNROD(IR)+IDELR(IR)
       IF(CTNROD(IR).EQ.(NZ+1)) CTNROD(IR)=NZ
       IF(CTNROD(IR).EQ.-1) CTNROD(IR)=0
26200   CONTINUE
C
      IF(FLAG) THEN
        WRITE(8,2424) GROUP,NMAX,ICASE
        WRITE(8,2426) (CTNROD(IR),IR=1,NMAX)
      ELSE
         IF(ICOUNT.EQ.1) THEN
          EFFKMIN=0.995
          EFFKMAX=1.005
          IF(NMAX.EQ.8) GOTO 2889
          IF(NMAX.EQ.10) GOTO 2789
         ELSE
          ICASE=1
          ICOUNT=0
          GOTO 1111
         ENDIF
      ENDIF
C***********
C*********** END OF CONTROL ROD PATTERN GENERATION
C***********

      GOTO 300
C
C**************
C************** PROMISING CONTROL ROD PATTERN GENERATION
C**************
3333  maxpnt=0
      do 3427 i=1,nodes
      if(ochfr(i).lt.(omchfr+2.) .OR. OPOWER(I).GT.(OPEAKN/2.)) then
       maxpnt=maxpnt+1
       chkpnt(maxpnt)=i
      endif
3427   continue
      NP1=NP+1
      PMAX1=1.E3
      EFFKMIN=0.997
      EFFKMAX=1.003
      ICOUNT=0
      IF(NMAX.EQ.8) GOTO 3889
      IF(NMAX.EQ.10) GOTO 3789
3889   ICOUNT=ICOUNT+1
      DO 36001 I1=1,3
       ESKEFF1=OKEFF+SKEFF(1,I1)
      DO 36002 I2=1,3
       ESKEFF2=ESKEFF1+SKEFF(2,I2)
      DO 36003 I3=1,3
       ESKEFF3=ESKEFF2+SKEFF(3,I3)
      DO 36004 I4=1,3
       ESKEFF4=ESKEFF3+SKEFF(4,I4)
      DO 36005 I5=1,3
       ESKEFF5=ESKEFF4+SKEFF(5,I5)
      DO 36006 I6=1,3
       ESKEFF6=ESKEFF5+SKEFF(6,I6)
      DO 36007 I7=1,3
       ESKEFF7=ESKEFF6+SKEFF(7,I7)
      DO 36008 I8=1,3
       ESKEFF8=ESKEFF7+SKEFF(8,I8)
C
      IF (ESKEFF8.LT. EFFKMIN .OR. ESKEFF8.GT. EFFKMAX) GOTO 36008
C
      ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
     $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
     $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
      IF(ESPOWER(LPP).GT.OPEAKN) GOTO 36008
C
      ESCHFR(LMM)=OCHFR(LMM)+SCHFR(1,I1,LMM)+SCHFR(2,I2,LMM)
     $       +SCHFR(3,I3,LMM)+SCHFR(4,I4,LMM)+SCHFR(5,I5,LMM)
     $       +SCHFR(6,I6,LMM)+SCHFR(7,I7,LMM)+SCHFR(8,I8,LMM)
      IF(ESCHFR(LMM).LT.1.9) GOTO 36008
C
      PMAX2=-2.0
      DO 36100 L=chkpnt(1),chkpnt(maxpnt)
      ESCHFR(L)=OCHFR(L)+SCHFR(1,I1,L)+SCHFR(2,I2,L)
     $       +SCHFR(3,I3,L)+SCHFR(4,I4,L)+SCHFR(5,I5,L)
     $       +SCHFR(6,I6,L)+SCHFR(7,I7,L)+SCHFR(8,I8,L)
       IF(ESCHFR(L).GT.1.9) THEN
          ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $       +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $       +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
         IF(ESPOWER(L).GT.PMAX2) PMAX2=ESPOWER(L)
       ENDIF
36100  CONTINUE
      IF(PMAX2.LT.OPEAKN .AND. PMAX2.LT.PMAX1) THEN
       IF(ABS(ESKEFF8-OKEFF).LE.1.E-6) GOTO 36008
       FLAG=.TRUE.
       PMAX1=PMAX2
       SI(1)=I1
       SI(2)=I2
       SI(3)=I3
       SI(4)=I4
       SI(5)=I5
       SI(6)=I6
       SI(7)=I7
       SI(8)=I8
      ENDIF
C
36008  CONTINUE
36007  CONTINUE
36006  CONTINUE
36005  CONTINUE
36004  CONTINUE
36003  CONTINUE
36002  CONTINUE
36001  CONTINUE
C
      GOTO 3575
C
C** END OF PATTERN B
C
3789   ICOUNT=ICOUNT+1
      DO 37001 I1=1,3
       ESKEFF1=OKEFF+SKEFF(1,I1)
      DO 37002 I2=1,3
       ESKEFF2=ESKEFF1+SKEFF(2,I2)
      DO 37003 I3=1,3
       ESKEFF3=ESKEFF2+SKEFF(3,I3)
      DO 37004 I4=1,3
       ESKEFF4=ESKEFF3+SKEFF(4,I4)
      DO 37005 I5=1,3
       ESKEFF5=ESKEFF4+SKEFF(5,I5)
      DO 37006 I6=1,3
       ESKEFF6=ESKEFF5+SKEFF(6,I6)
      DO 37007 I7=1,3
       ESKEFF7=ESKEFF6+SKEFF(7,I7)
      DO 37008 I8=1,3
       ESKEFF8=ESKEFF7+SKEFF(8,I8)
      DO 37009 I9=1,3
       ESKEFF9=ESKEFF8+SKEFF(9,I9)
      DO 37010 I10=1,3
       ESKEF10=ESKEFF9+SKEFF(10,I10)
C
      IF (ESKEF10.LT. EFFKMIN .OR. ESKEF10.GT. EFFKMAX) GOTO 37010
C
      ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
     $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
     $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
     $     +SPOWER(9,I9,LPP)+SPOWER(10,I10,LPP)
      IF(ESPOWER(LPP).GT.2.2) GOTO 37010
```

```
C
      ESCHFR(LMM)=OCHFR(LMM)+SCHFR(1,I1,LMM)+SCHFR(2,I2,LMM)
     $        +SCHFR(3,I3,LMM)+SCHFR(4,I4,LMM)+SCHFR(5,I5,LMM)
     $        +SCHFR(6,I6,LMM)+SCHFR(7,I7,LMM)+SCHFR(8,I8,LMM)
     $        +SCHFR(9,I9,LMM)+SCHFR(10,I10,LMM)
      IF(ESCHFR(LMM).LT.OMCHFR) GOTO 37010
C
      PMAX2=-2.0
      DO 37100 L=chkpnt(1),chkpnt(maxpnt)
      ESCHFR(L)=OCHFR(L)+SCHFR(1,I1,L)+SCHFR(2,I2,L)
     $        +SCHFR(3,I3,L)+SCHFR(4,I4,L)+SCHFR(5,I5,L)
     $        +SCHFR(6,I6,L)+SCHFR(7,I7,L)+SCHFR(8,I8,L)
     $        +SCHFR(9,I9,L)+SCHFR(10,I10,L)
       IF(ESCHFR(L).GT.1.9) THEN
          ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $        +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $        +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
     $        +SPOWER(9,I9,L)+SPOWER(10,I10,L)
         IF(ESPOWER(L).GT.PMAX2) PMAX2=ESPOWER(L)
       ENDIF
37100  CONTINUE
      IF(PMAX2.LT.OPEAKN .AND. PMAX2.LT.PMAX1) THEN
       IF(ABS(ESKEFF8-OKEFF).LE.1.E-6) GOTO 37010
       FLAG=.TRUE.
       PMAX1=PMAX2
       SI(1)=I1
       SI(2)=I2
       SI(3)=I3
       SI(4)=I4
       SI(5)=I5
       SI(6)=I6
       SI(7)=I7
       SI(8)=I8
       SI(9)=I9
       SI(10)=I10
      ENDIF
C
37010  CONTINUE
37009  CONTINUE
37008  CONTINUE
37007  CONTINUE
37006  CONTINUE
37005  CONTINUE
37004  CONTINUE
37003  CONTINUE
37002  CONTINUE
37001  CONTINUE
C
C** END OF PATTERN A
C
3575   DO 3253 I=1,NMAX
      IF(SI(I).EQ.1) IDELR(I)=1
      IF(SI(I).EQ.2) IDELR(I)=-1
      IF(SI(I).EQ.3) IDELR(I)=0
3253   CONTINUE
       DO 36200 IR=1,NMAX
       CTNROD(IR)=OCTNROD(IR)+IDELR(IR)
       IF(CTNROD(IR).EQ.(NZ+1)) CTNROD(IR)=NZ
       IF(CTNROD(IR).EQ.-1) CTNROD(IR)=0
36200   CONTINUE
C
      IF(FLAG) THEN
        WRITE(8,2424) GROUP,NMAX,ICASE
        WRITE(8,2426) (CTNROD(IR),IR=1,NMAX)
      ELSE
```

```
         IF(ICOUNT.EQ.1) THEN
          EFFKMIN=0.995
          EFFKMAX=1.005
          IF(NMAX.EQ.8) GOTO 3889
          IF(NMAX.EQ.10) GOTO 3789
         ELSE
          ICASE=2
          ICOUNT=0
          GOTO 2222
         ENDIF
      ENDIF
C***********
C*********** END OF CONTROL ROD PATTERN GENERATION
C***********
      GOTO 300
C**************
C************** PROMISING CONTROL ROD PATTERN GENERATION
C**************
4444  maxpnt=0
      do 4427 i=1,nodes
      if(ochfr(i).lt.(omchfr+2.) .OR. OPOWER(I).GT.(OPEAKN/2.)) then
       maxpnt=maxpnt+1
       chkpnt(maxpnt)=i
      endif
4427   continue
      NP1=NP+1
      ZJMIN=1.E3
      EFFKMIN=0.997
      EFFKMAX=1.003
      ICOUNT=0
      IF(NMAX.EQ.8) GOTO 4889
      IF(NMAX.EQ.10) GOTO 4789
4889   ICOUNT=ICOUNT+1
      DO 46001 I1=1,3
       ESKEFF1=OKEFF+SKEFF(1,I1)
      DO 46002 I2=1,3
       ESKEFF2=ESKEFF1+SKEFF(2,I2)
      DO 46003 I3=1,3
       ESKEFF3=ESKEFF2+SKEFF(3,I3)
      DO 46004 I4=1,3
       ESKEFF4=ESKEFF3+SKEFF(4,I4)
      DO 46005 I5=1,3
       ESKEFF5=ESKEFF4+SKEFF(5,I5)
      DO 46006 I6=1,3
       ESKEFF6=ESKEFF5+SKEFF(6,I6)
      DO 46007 I7=1,3
       ESKEFF7=ESKEFF6+SKEFF(7,I7)
      DO 46008 I8=1,3
       ESKEFF8=ESKEFF7+SKEFF(8,I8)
C
      IF (ESKEFF8.LT. EFFKMIN .OR. ESKEFF8.GT. EFFKMAX) GOTO 46008
C
      ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
     $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
     $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
      IF(ESPOWER(LPP).GT.2.2) GOTO 46008
C
      ESCHFR(LMM)=OCHFR(LMM)+SCHFR(1,I1,LMM)+SCHFR(2,I2,LMM)
     $       +SCHFR(3,I3,LMM)+SCHFR(4,I4,LMM)+SCHFR(5,I5,LMM)
     $       +SCHFR(6,I6,LMM)+SCHFR(7,I7,LMM)+SCHFR(8,I8,LMM)
      IF(ESCHFR(LMM).LT.1.9) GOTO 46008
C

      DO 46100 L=chkpnt(1),chkpnt(maxpnt)
      ESCHFR(L)=OCHFR(L)+SCHFR(1,I1,L)+SCHFR(2,I2,L)
     $       +SCHFR(3,I3,L)+SCHFR(4,I4,L)+SCHFR(5,I5,L)
     $       +SCHFR(6,I6,L)+SCHFR(7,I7,L)+SCHFR(8,I8,L)
      IF(ESCHFR(L).LT.1.9) GOTO 46008
      ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $       +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $       +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
      IF(ESPOWER(L).GT.2.2) GOTO 46008
46100  CONTINUE

        ESZJ=OZJ+SZJ(1,I1)+SZJ(2,I2)
     $       +SZJ(3,I3)+SZJ(4,I4)+SZJ(5,I5)
     $       +SZJ(6,I6)+SZJ(7,I7)+SZJ(8,I8)
      IF(ESZJ.LT.OZJ .AND. ESZJ.LT.ZJMIN) THEN
       IF(ABS(ESKEFF8-OKEFF).LE.1.E-6) GOTO 46008
       FLAG=.TRUE.
       ZJMIN=ESZJ
       SI(1)=I1
       SI(2)=I2
       SI(3)=I3
       SI(4)=I4
       SI(5)=I5
       SI(6)=I6
       SI(7)=I7
       SI(8)=I8
      ENDIF
C
46008  CONTINUE
46007  CONTINUE
46006  CONTINUE
46005  CONTINUE
46004  CONTINUE
46003  CONTINUE
46002  CONTINUE
46001  CONTINUE
C
      GOTO 4575
C
C** END OF PATTERN B
C
4789   ICOUNT=ICOUNT+1
      DO 47001 I1=1,3
       ESKEFF1=OKEFF+SKEFF(1,I1)
      DO 47002 I2=1,3
       ESKEFF2=ESKEFF1+SKEFF(2,I2)
      DO 47003 I3=1,3
       ESKEFF3=ESKEFF2+SKEFF(3,I3)
      DO 47004 I4=1,3
       ESKEFF4=ESKEFF3+SKEFF(4,I4)
      DO 47005 I5=1,3
       ESKEFF5=ESKEFF4+SKEFF(5,I5)
      DO 47006 I6=1,3
       ESKEFF6=ESKEFF5+SKEFF(6,I6)
      DO 47007 I7=1,3
       ESKEFF7=ESKEFF6+SKEFF(7,I7)
      DO 47008 I8=1,3
       ESKEFF8=ESKEFF7+SKEFF(8,I8)
      DO 47009 I9=1,3
       ESKEFF9=ESKEFF8+SKEFF(9,I9)
      DO 47010 I10=1,3
       ESKEF10=ESKEFF9+SKEFF(10,I10)
C
      IF (ESKEF10.LT. EFFKMIN .OR. ESKEF10.GT. EFFKMAX) GOTO 47010
C
      ESPOWER(LPP)=OPOWER(LPP)+SPOWER(1,I1,LPP)+SPOWER(2,I2,LPP)
     $     +SPOWER(3,I3,LPP)+SPOWER(4,I4,LPP)+SPOWER(5,I5,LPP)
     $     +SPOWER(6,I6,LPP)+SPOWER(7,I7,LPP)+SPOWER(8,I8,LPP)
     $     +SPOWER(9,I9,LPP)+SPOWER(10,I10,LPP)
      IF(ESPOWER(LPP).GT.2.2) GOTO 47010
C
      ESCHFR(LMM)=OCHFR(LMM)+SCHFR(1,I1,LMM)+SCHFR(2,I2,LMM)
     $       +SCHFR(3,I3,LMM)+SCHFR(4,I4,LMM)+SCHFR(5,I5,LMM)
     $       +SCHFR(6,I6,LMM)+SCHFR(7,I7,LMM)+SCHFR(8,I8,LMM)
     $       +SCHFR(9,I9,LMM)+SCHFR(10,I10,LMM)
      IF(ESCHFR(LMM).LT.1.9) GOTO 47010
C

      DO 47100 L=chkpnt(1),chkpnt(maxpnt)
      ESCHFR(L)=OCHFR(L)+SCHFR(1,I1,L)+SCHFR(2,I2,L)
     $       +SCHFR(3,I3,L)+SCHFR(4,I4,L)+SCHFR(5,I5,L)
     $       +SCHFR(6,I6,L)+SCHFR(7,I7,L)+SCHFR(8,I8,L)
     $       +SCHFR(9,I9,L)+SCHFR(10,I10,L)
      IF(ESCHFR(L).LT.1.9) GOTO 47010
      ESPOWER(L)=OPOWER(L)+SPOWER(1,I1,L)+SPOWER(2,I2,L)
     $       +SPOWER(3,I3,L)+SPOWER(4,I4,L)+SPOWER(5,I5,L)
     $       +SPOWER(6,I6,L)+SPOWER(7,I7,L)+SPOWER(8,I8,L)
     $       +SPOWER(9,I9,L)+SPOWER(10,I10,L)
      IF(ESPOWER(L).GT.2.2) GOTO 47010
47100  CONTINUE

        ESZJ=OZJ+SZJ(1,I1)+SZJ(2,I2)
     $       +SZJ(3,I3)+SZJ(4,I4)+SZJ(5,I5)
     $       +SZJ(6,I6)+SZJ(7,I7)+SZJ(8,I8)
     $       +SZJ(9,I9)+SZJ(10,I10)
      IF(ESZJ.LT.OZJ .AND. ESZJ.LT.ZJMIN) THEN
       IF(ABS(ESKEFF8-OKEFF).LE.1.E-6) GOTO 47010
       FLAG=.TRUE.
       ZJMIN=ESZJ
       SI(1)=I1
       SI(2)=I2
       SI(3)=I3
       SI(4)=I4
       SI(5)=I5
       SI(6)=I6
       SI(7)=I7
       SI(8)=I8
       SI(9)=I9
       SI(10)=I10
      ENDIF
C
47010  CONTINUE
47009  CONTINUE
47008  CONTINUE
47007  CONTINUE
47006  CONTINUE
47005  CONTINUE
47004  CONTINUE
47003  CONTINUE
47002  CONTINUE
47001  CONTINUE
C
C** END OF PATTERN A
C
4575   DO 4253 I=1,NMAX
      IF(SI(I).EQ.1) IDELR(I)=1
      IF(SI(I).EQ.2) IDELR(I)=-1
      IF(SI(I).EQ.3) IDELR(I)=0
4253   CONTINUE
       DO 46200 IR=1,NMAX
       CTNROD(IR)=OCTNROD(IR)+IDELR(IR)
       IF(CTNROD(IR).EQ.(NZ+1)) CTNROD(IR)=NZ
       IF(CTNROD(IR).EQ.-1) CTNROD(IR)=0
46200   CONTINUE
C
      IF(FLAG) THEN
        WRITE(8,2424) GROUP,NMAX,ICASE
        WRITE(8,2426) (CTNROD(IR),IR=1,NMAX)
      ELSE
         IF(ICOUNT.EQ.1) THEN
          EFFKMIN=0.995
          EFFKMAX=1.005
          IF(NMAX.EQ.8) GOTO 4889
          IF(NMAX.EQ.10) GOTO 4789
         ELSE
          ICASE=3
          ICOUNT=0
          GOTO 3333
         ENDIF
      ENDIF
C***********
C*********** END OF CONTROL ROD PATTERN GENERATION
C***********
      GOTO 300
300   RETURN
      END
```

Appendix C

Intra-nodal Flux Distributions Calculation Subroutine

```
      PROGRAM PHILSQ
      IMPLICIT REAL*8 (A-H,O-Z)
C LEAST SQUARE FITTING
      EXTERNAL FUNCT
      DIMENSION A(6,7),U(9), Z(6),C(9,6),G(6),PHI(10,10)
     $          ,UC(9),CN(10,10),DX(10),DY(10)
      DATA DX/0.5,8*1.0,0.5/, DY/0.5,8*1.0,0.5/

      OPEN(6,FILE='FAST')
      OPEN(7,FILE='THERMAL')
      OPEN(9,FILE='PHILSQ3.OUT')
C
C P,Q = NODAL POSITION OF X, AND Y WHICH IS CLOSEST TO THE ORIGIN
C
      READ(*,*) INODE,JNODE
      READ(*,*) P,Q,W
      WRITE(9,*) ' READ (P,Q) , AND WIDTH ?  ', P,Q,W
      P=P+0.5
      Q=Q+0.5
      W=W-1.0
       ICALLS=1
77     IF(ICALLS.GE.3) GOTO 888
       IF(ICALLS.EQ.1) THEN
        NOUT=6
        P=1.E4
        Q=1.E4
       ENDIF
       IF(ICALLS.EQ.2) THEN
        NOUT=7
        P=1.E6
        Q=1.E6
       ENDIF
      READ(*,*) D
      WRITE(9,*) ' READ DIFFUSION COEFFICIENTS ?  ', D
      READ(*,*) U(1),U(4),U(2),U(3),U(5),U(8),U(6),U(7),U(9)
      READ(*,120) ((CN(I,J),J=1,10),I=1,10)
C
C RECALCULATE U(1) --- U(9)
C
      U(1)=0.
      DO 31 JY=1,10
31    U(1)=U(1)+CN(1,JY)*DY(JY)
      U(2)=0.
      DO 32 IX=1,10
32    U(2)=U(2)+CN(IX,10)*DX(IX)
      U(3)=0.
      DO 33 JY=1,10
33    U(3)=U(3)+CN(10,JY)*DY(JY)
      U(4)=0.
      DO 34 IX=1,10
34    U(4)=U(4)+CN(IX,1)*DX(IX)

      U(5)=0.
      DO 35 JY=1,10
35    U(5)=U(5)+(-D)*(CN(2,JY)-CN(1,JY))*DY(JY)
      U(6)=0.
      DO 36 IX=1,10
36    U(6)=U(6)+(-D)*(CN(IX,10)-CN(IX,9))*DX(IX)
      U(7)=0.
      DO 37 JY=1,10
37    U(7)=U(7)+(-D)*(CN(10,JY)-CN(9,JY))*DY(JY)
      U(8)=0.
      DO 38 IX=1,10
38    U(8)=U(8)+(-D)*(CN(IX,2)-CN(IX,1))*DX(IX)

      U(9)=0.
      DO 39 IX=1,10
      DO 39 JY=1,10
39    U(9)=U(9)+CN(IX,JY)/100.
C
C SET UP MATRIX ELEMENTS
C
      C(1,1)=W
      C(1,2)=1./2.*((Q+W)**2-Q**2)
      C(1,3)=P*W
      C(1,4)=1./2.*P*((Q+W)**2-Q**2)
      C(1,5)=1./3.*((Q+W)**3-Q**3)
      C(1,6)=P*P*W
      C(2,1)=W
      C(2,2)=W*(Q+W)
      C(2,3)=1./2.*((P+W)**2-P**2)
      C(2,4)=1./2.*(Q+W)*((P+W)**2-P**2)
      C(2,5)=(Q+W)**2*W
      C(2,6)=1./3.*((P+W)**3-P**3)
      C(3,1)=W
      C(3,2)=1./2.*((Q+W)**2-Q**2)
      C(3,3)=(P+W)*W
      C(3,4)=1./2.*(P+W)*((Q+W)**2-Q**2)
      C(3,5)=1./3.*((Q+W)**3-Q**3)
      C(3,6)=(P+W)**2*W
      C(4,1)=W
      C(4,2)=W*Q
      C(4,3)=1./2.*((P+W)**2-P**2)
      C(4,4)=1./2.*Q*((P+W)**2-P**2)
      C(4,5)=Q**2*W
      C(4,6)=1./3.*((P+W)**3-P**3)
      C(5,1)=0.
      C(5,2)=0.
      C(5,3)=-D*W
      C(5,4)=-1./2.*D*((Q+W)**2-Q**2)
      C(5,5)=0.
      C(5,6)=-2.*D*P*W
      C(6,1)=0.
      C(6,2)=-D*W
      C(6,3)=0.
      C(6,4)=-1./2.*D*((P+W)**2-P**2)
      C(6,5)=-2.*D*(Q+W)*W
      C(6,6)=0.
      C(7,1)=0.
      C(7,2)=0.
      C(7,3)=-D*W
      C(7,4)=-1./2.*D*((Q+W)**2-Q**2)
      C(7,5)=0.
      C(7,6)=-2.*D*(P+W)*W
      C(8,1)=0.
      C(8,2)=-D*W
      C(8,3)=0.
      C(8,4)=-1./2.*D*((P+W)**2-P**2)
      C(8,5)=-2.*D*Q*W
      C(8,6)=0.
      C(9,1)=1.
      C(9,2)=1./2./W*((Q+W)**2-Q**2)
      C(9,3)=1./2./W*((P+W)**2-P**2)
      C(9,4)=1./4./W/W*((P+W)**2-P**2)*((Q+W)**2-Q**2)
      C(9,5)=1./3./W*((Q+W)**3-Q**3)
      C(9,6)=1./3./W*((P+W)**3-P**3)
C
C
      WRITE(9,*) '  INTG. SURFACE FLUXES:   ',U(1),U(4),U(2),U(3)
      WRITE(9,*) '  INTG. SURFACE CURRENTS:',U(5),U(8),U(6),U(7)
      WRITE(9,*) '  NODE AVERAGE FLUX:      ',U(9)

      DO 135 J1=1,6
      DO 135 J2=1,6
       A(J1,J2)=0.
      DO 135 I=1,9
135   A(J1,J2)=A(J1,J2)+C(I,J1)*C(I,J2)
      DO 136 II=1,6
       A(II,7)=0.
      DO 136 I=1,9
136   A(II,7)=A(II,7)+C(I,II)*U(I)

C END OF SETUP
C
C
      DO 200 I=1,6
200   WRITE(9,130) ( A(I,J), J=1,7 )
130   FORMAT(4(/,8E14.5))
C
      CALL SOLVE(6,A,Z)
C
      WRITE(9,*) 'CALCULATED LSQ COEFFICIENTS A00,A01,A10,A11,A02,A20'
      WRITE(9,120) ( Z(I), I=1,6 )
120   FORMAT((10E13.6))
      DO 50 I=1,10
       XX=P+W/9.*FLOAT(I-1)
      DO 50 J=1,10
       YY=Q+W/9.*FLOAT(J-1)
50    PHI(I,J)=FUNCT(XX,YY,Z)
C
      WRITE(9,*) '  '
      WRITE(9,*) '  '
      WRITE(NOUT,*) INODE,JNODE
      DO 350 I=1,10
      WRITE(NOUT,120) (PHI(I,J),J=1,10)
      WRITE(9,120) (PHI(I,J),J=1,10)
350   WRITE(9,*) '  '
C
      DO 1567 I=1,9
       UC(I)=0.
      DO 1567 J=1,6
1567   UC(I)=UC(I)+C(I,J)*Z(J)
      WRITE(9,*) '                              KNOWN        CALCULATED  ERR(%)'
      WRITE(9,*) 'INTG SURFACE-1 FLUX: ',U(1),UC(1),(1.-UC(1)/U(1))*100.
      WRITE(9,*) 'INTG SURFACE-2 FLUX: ',U(4),UC(4),(1.-UC(4)/U(4))*100.
      WRITE(9,*) 'INTG SURFACE-3 FLUX: ',U(2),UC(2),(1.-UC(2)/U(2))*100.
      WRITE(9,*) 'INTG SURFACE-4 FLUX: ',U(3),UC(3),(1.-UC(3)/U(3))*100.
      IF(ABS(U(5)).LE.1.E-10) THEN
       WRITE(9,*) 'INTG SURFACE-1 CURRENT:',U(5),UC(5),UC(5)
      ELSE
       WRITE(9,*) 'INTG SURFACE-1 CURRENT:',U(5),UC(5)
     $                                    ,(1.-UC(5)/U(5))*100.
      ENDIF
      IF(ABS(U(8)).LE.1.E-10) THEN
       WRITE(9,*) 'INTG SURFACE-2 CURRENT:',U(8),UC(8),UC(8)
      ELSE
```

```
       WRITE(9,*) 'INTG SURFACE-2 CURRENT:',U(8),UC(8)
     $                                      ,(1.-UC(8)/U(8))*100.
      ENDIF
      IF(ABS(U(6)).LE.1.E-10) THEN
       WRITE(9,*) 'INTG SURFACE-3 CURRENT:',U(6),UC(6),UC(6)
      ELSE
       WRITE(9,*) 'INTG SURFACE-3 CURRENT:',U(6),UC(6)
     $                                      ,(1.-UC(6)/U(6))*100.
      ENDIF
      IF(ABS(U(7)).LE.1.E-10) THEN
       WRITE(9,*) 'INTG SURFACE-4 CURRENT:',U(7),UC(7),UC(7)
      ELSE
       WRITE(9,*) 'INTG SURFACE-4 CURRENT:',U(7),UC(7)
     $                                      ,(1.-UC(7)/U(7))*100.
      ENDIF
      WRITE(9,*) 'NODE AVERAGE FLUX:      ',U(9),UC(9)
     $                                     ,(1.-UC(9)/U(9))*100.
      ICALLS=ICALLS+1
C      READ(*,65) S1,S2,S3,S4,S5,S6,S7,S8,S9,S10
65    FORMAT(E13.6,/,E13.6,/,E13.6,/,E13.6,/,E13.6
     $    ,/,E13.6,/,E13.6,/,E13.6,/,E13.6,/,E13.6)
      GOTO 77
888   STOP
      END
C
C
C
      SUBROUTINE SOLVE(N,A,Z)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(6,7),Z(6),INDEX(6)
      NN=N+1
      DO 333 I=1,N
      Z(I)=0.
      INDEX(I)=0.
  333 CONTINUE
      NP=NN
      DET=1.0
      DO 100 K=1,N
      KP=K+1
      M=NP-K
      D=0.
      L=K
      DO 30 I=K,N
      IF(ABS(A(I,K))-D) 30,30,20
   20 L=I
      INDEX(K)=L
      D=ABS(A(L,K))
   30 CONTINUE
      IF(L-K) 40,55,40
   40 DO 50 J=1,NN
      D=A(L,J)
      A(L,J)=A(K,J)
      A(K,J)=D
   50 CONTINUE
   55 CONTINUE
      WRITE(9,900) A(K,K)
   32 IF(A(K,K).EQ.0.) GOTO 200
   33 DET=DET*A(K,K)
      DO 80 J=KP,NP
      A(K,J)=A(K,J)/A(K,K)
      DO 70 I=KP,N
      A(I,J)=A(I,J)-A(K,J)*A(I,K)
   70 CONTINUE
   80 CONTINUE
  100 CONTINUE
```

```
      Z(N)=A(N,NP)
      DO 120 I=2,N
      J=NP-I
      JP=J+1
      SUM=0.
      DO 150 M=JP,N
      SUM=SUM-A(J,M)*Z(M)
  150 CONTINUE
      Z(J)=A(J,NP)+SUM
  120 CONTINUE
      GOTO 210
  200 CONTINUE
      WRITE(9,901)
  210 CONTINUE
      WRITE(9,970) DET
970   FORMAT(/,' DET= ',E13.6)
  900 FORMAT(' PIVOT = ',1PE12.5)
  901 FORMAT(' THE MATRIX IS SINGULAR')
  902 FORMAT(' THE MATRIX AFTER ELIMINATION')
      RETURN
      END
C
      FUNCTION FUNCT(X,Y,Z)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION Z(6)
      FUNCT=Z(1)+Z(2)*Y+Z(3)*X+Z(4)*X*Y+Z(5)*Y*Y+Z(6)*X*X
C
      RETURN
      END
```