

A Quasi-Static Polynomial Nodal Method for Nuclear Reactor Analysis

by

Jess C. Gehin

S.M., Nuclear Engineering, Massachusetts Institute of Technology
(1990)

B.S., Nuclear Engineering, Kansas State University
(1988)

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

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1992

© Jess C. Gehin, MCMXCII. All Rights Reserved.

The author hereby grants to MIT permission to reproduce and
to distribute copies of this thesis document in whole or in part.

Author _____
Department of Nuclear Engineering
August 18, 1992

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

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

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

ARCHIVES

FEB 11 1993

LIBRARIES

A Quasi-Static Polynomial Nodal Method for Nuclear Reactor Analysis

by

Jess C. Gehin

Submitted to the Department of Nuclear Engineering
on August 18, 1992, in partial fulfillment of the
requirements for the degree of
DOCTOR OF PHILOSOPHY

Abstract

Modern nodal methods are currently available which can accurately and efficiently solve the static and transient neutron diffusion equations. Most of the methods, however, are limited to two energy groups for practical application. The objective of this research is the development of a static and transient, multidimensional nodal method which allows more than two energy groups and uses a non-linear iterative method for efficient solution of the nodal equations.

For both the static and transient methods, finite-difference equations which are corrected by the use of discontinuity factors are derived. The discontinuity factors are computed from a polynomial nodal method using a non-linear iteration technique. The polynomial nodal method is based upon a quartic approximation and utilizes a quadratic transverse-leakage approximation. The solution of the time-dependent equations is performed by the use of a quasi-static method in which the node-averaged fluxes are factored into shape and amplitude functions. Since the shape function generally changes more slowly than the amplitude function it can be computed less frequently, providing a substantial computational savings. The amplitude function is obtained by solving point kinetics equations for which the parameters are determined by precise mathematical expressions based on the nodal model.

The application of the quasi-static polynomial method to several benchmark problems demonstrates that the accuracy is consistent with that of other nodal methods. The use of the quasi-static method is shown to substantially reduce the computation time over the traditional fully-implicit time-integration method. Problems involving thermal-hydraulic feedback are accurately, and efficiently, solved by performing several reactivity/thermal-hydraulic updates per shape calculation.

Thesis Supervisor: Allan F. Henry

Title: Professor, Department of Nuclear Engineering

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude and appreciation to Professor Allan F. Henry for his unmeasurable guidance and support throughout this project and my education at M.I.T.

Further, I would like to thank my thesis reader, Professor David D. Lanning, for his comments and suggestions.

During my stay at M.I.T. I have made many friends who have made my graduate education more enjoyable. These people include Mark Byers, Jonathan Witter, Santiago Parra, and Chris Owens.

Finally, I would like to thank my wife Ann for the love she has provided and the sacrifices that she has made in order for me to complete my degree. I am looking forward to our future together.

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

CONTENTS

Abstract	2
Acknowledgements	3
Table of Contents	4
List of Figures	10
List of Tables	12
Chapter 1 Introduction and Background	13
1.1 Overview	13
1.2 Background	14
1.3 Research Objectives	16
1.4 Thesis Organization	17
Chapter 2 Derivation of the Static Nodal Equations	18
2.1 Introduction	18
2.2 Notation and the Nodal Balance Equation	18
2.3 Corrected Finite-Difference Coupling Equations	21
2.3.1 Boundary Conditions	25
2.3.2 Evaluation of the Discontinuity Factors	26
2.4 Polynomial Coupling Equations	27
2.4.1 The Transverse-Integration Procedure	27
2.4.2 The Polynomial Expansion	29
2.4.3 The Two-Node Problem	30
2.4.4 The Weighted Residual Procedure	31
2.4.5 Expansion Coefficient Solution	35
2.4.6 Boundary Conditions	37

2.5	The Non-Linear Iteration Procedure	38
2.6	Summary	40
Chapter 3 Derivation of the Transient Nodal Equations		42
3.1	Introduction	42
3.2	Notation	43
3.3	The Time-Dependent, Corrected Finite-Difference Equations	44
3.4	The Time-Dependent Polynomial Nodal Equations	46
3.5	Time-Integration of the Corrected Finite-Difference Equation	48
3.6	The Quasi-Static Method	50
3.6.1	The Amplitude Function Equation	51
3.6.2	Shape Function Equation	53
3.6.3	Choice of Weight Function	54
3.7	Thermal-Hydraulic and Feedback Models	56
3.7.1	The WIGL Model	56
3.7.2	The Cabral-IPM Model	58
3.7.3	Cross Section Feedback	58
3.8	Transient Control Mechanisms	59
3.9	Summary	60
Chapter 4 Static and Transient Numerical Solution Methods		62
4.1	Introduction	62
4.2	Static Solution Methods	62
4.2.1	Numerical Properties	63
4.2.2	Discontinuity Factor Iterations	64
4.2.3	Outer Iterations	65
4.2.4	Inner Iterations	69
4.2.5	General Iterative Strategy	72
4.2.6	Criticality Search Problems	74
4.2.7	Source Problems	74
4.2.8	Mathematical Adjoint Problems	75

4.3	Transient Solution Methods	76
4.3.1	Numerical Properties	76
4.3.2	Iterative Solution of the Transient Equations	77
4.3.3	Frequency Estimation	78
4.3.4	Solution of the Point Kinetics Equations	79
4.3.5	General Transient Computational Procedure	81
4.4	Summary	83
Chapter 5 Application of the Transient Nodal Method		84
5.1	Introduction	84
5.2	Forward to Transient Problems	84
5.2.1	Computer Code	85
5.2.2	Transverse-Leakage Approximations	86
5.2.3	Power Distribution Errors	86
5.2.4	Execution Times	87
5.3	The 2-D TWIGL Seed-Blanket Reactor Problems	88
5.3.1	The Static Solution	88
5.3.2	The Step Transient	92
5.3.3	The Ramp Transient	94
5.4	The 3-D LMW Operational Transient	96
5.4.1	The 3-D LMW Problem Without Feedback	98
5.4.2	The 3-D LMW Problem with Thermal-Hydraulic Feedback . .	103
5.5	The LRA BWR Transient Problems	108
5.5.1	The 2-D LRA Problem	110
5.5.2	The 3-D LRA Problem	112
5.6	The PWR Operational Transient	117
5.7	The PWR Coolant Inlet-Temperature Transient	125
5.8	Summary	129
Chapter 6 Summary, Conclusions and Recommendations		130
6.1	Overview of the Investigation	130

6.2	Conclusions	131
6.3	Recommendations for Future Research	132
6.3.1	Diagonal Dominance Required by the Iterative Methods . . .	132
6.3.2	Application to Multi-Group Analyses	133
6.3.3	Study of the Thermal-Hydraulic/Neutron Coupling	134
6.3.4	Study of the Time Dependence of the Discontinuity Factors .	134
References		135
Appendix A The Quadratic Transverse Leakage Moments and Coefficients		139
A.1	The Quadratic Transverse-Leakage Approximation	140
A.2	LHS-Biased Quadratic Transverse-Leakage Approximation	142
A.3	RHS-Biased Quadratic Transverse-Leakage Approximation	144
A.4	The Flat Transverse-Leakage Approximation	146
Appendix B Problem Specifications		147
B.1	The TWIGL 2-D Seed-Blanket Reactor Kinetics Problem	148
B.2	The LMW LWR Transient Problem	150
B.3	The LRA BWR Kinetics Benchmark Problem	154
B.4	The PWR Transient Problems	158
Appendix C Selected Results of Problem Analyses		165

LIST OF FIGURES

2-1	Diagram showing the the surface and node labeling conventions. . . .	22
2-2	Diagram showing the orientation of the two-node problem.	30
2-3	A flow diagram of the non-linear iteration procedure for the static problem.	39
3-1	Diagram showing the subdivision of the time steps in the quasi-static method.	51
4-1	Flow diagram of the quasi-static transient solution procedure.	82
5-1	The group 2, x-direction transversely-integrated fluxes ($j = 1$) for the TWIGL problem.	90
5-2	The group 2, x-direction, quadratic transversely-integrated currents ($j = 1$) for the TWIGL problem.	90
5-3	The group 2, x-direction, cubic transversely-integrated currents ($j = 1$) for the TWIGL problem.	91
5-4	The group 2, x-direction, quartic transversely-integrated currents ($j = 1$) for the TWIGL problem.	91
5-5	Power density vs. time for the 3-D LMW problem without feedback. .	101
5-6	Reactivity vs. time for the 3-D LMW problem without feedback. . . .	101
5-7	Power density vs. time for the 3-D LMW problem without feedback demonstrating the cusping correction.	104
5-8	Reactivity vs. time for the 3-D LMW problem without feedback demonstrating the cusping correction.	104
5-9	Total power vs. time for the 3-D LMW problem with feedback.	106
5-10	Reactivity vs. time for the 3-D LMW problem with feedback.	106

5-11	The 3-D LMW transient with feedback using 5 second shape and reactivity/thermal-hydraulic steps.	107
5-12	Power density vs. time for the 3-D LMW problem with feedback demonstrating the cusping correction.	109
5-13	Reactivity vs. time for the 3-D LMW problem with feedback, demonstrating the cusping correction.	109
5-14	Power vs. time for the 2-D LRA transient problem.	114
5-15	Fuel temperature vs. time for the 2-D LRA transient problem.	114
5-16	Power vs. time for the 3-D LRA transient problem.	119
5-17	Fuel temperature vs. time for the 3-D LRA transient problem.	119
5-18	Control rod motions for the PWR operational transient.	121
5-19	Power vs. time for the PWR operational transient demonstrating the temporal convergence of the solution.	123
5-20	Reactivity vs. time for the PWR operational transient demonstrating the temporal convergence of the solution.	123
5-21	Power vs. time for the PWR operational transient, large time-step quasi-static solution.	124
5-22	Reactivity vs. time for the PWR operational transient, large time-step quasi-static solution.	124
5-23	Power vs. time for the PWR coolant inlet-temperature transient demonstrating the temporal convergence.	127
5-24	Reactivity vs. time for the PWR coolant inlet-temperature transient demonstrating the temporal convergence.	127
5-25	Power vs. time for the PWR coolant inlet-temperature transient, large time-step quasi-static solution.	128
5-26	Reactivity vs. time for the PWR coolant inlet-temperature transient, large time-step quasi-static solution.	128
C-1	3-D LMW problem without feedback, comparison of initial static solutions.	166

C-2	3-D LMW problem with feedback, comparison of initial static solutions.	167
C-3	2-D LRA problem, comparison of initial static solutions.	168
C-4	2-D LRA transient problem, normalized power distributions and fuel temperatures.	169
C-5	3-D LRA problem, comparison of initial static solutions.	176
C-6	PWR operational transient, comparison of initial static solutions. . .	184
C-7	PWR coolant inlet-temperature transient, comparison of initial static solutions.	185

LIST OF TABLES

5.1	LINPACK MFLOPS ratings of several computer systems used for the comparison of execution times.	88
5.2	A summary of the CONQUEST static results for the 2-D TWIGL seed-blanket test problem.	89
5.3	Reactor power vs. time for the 2-D TWIGL step transient problem.	92
5.4	Reactor power vs. time for the 2-D TWIGL step transient problem with various number of steps per discontinuity factor update.	93
5.5	Reactor power and reactivity vs. time for the TWIGL step problem for point kinetics (PK) and quasi-static (QS) methods with adjoint and unity weighting.	94
5.6	Reactor power vs. time for the 2-D TWIGL step transient for several nodal codes ($\Delta t_n = 10$ ms).	95
5.7	Reactor power vs. time for the TWIGL ramp problem with various number of steps per discontinuity factor update ($\Delta t_n = 5$ ms).	95
5.8	Reactor power and reactivity vs. time for the TWIGL ramp problem for point kinetics (PK) and quasi-static (QS) methods with adjoint and unity weighting.	96
5.9	Reactor power vs. time for the TWIGL ramp problem for various shape-update time steps.	97
5.10	Reactor power vs. time for the 2-D TWIGL ramp transient for several nodal codes ($\Delta t_n = 5$ ms).	97
5.11	A summary of the CONQUEST static results for the 3-D LMW problem without feedback.	98

5.12	A comparison of the CONQUEST and QUANDRY 3-D LMW static calculations without feedback.	99
5.13	Mean power density (W/cm^3) vs. time for the 3-D LMW problem without feedback and different shape-update time steps.	100
5.14	A comparison of the node and assembly errors in the normalized power densities.	100
5.15	A comparison of the mean power density (W/cm^3) for solutions of the 3-D LMW Test Problem without feedback for several nodal codes. . .	102
5.16	A summary of the CONQUEST static results for the 3-D LMW problem with feedback.	105
5.17	Total power vs. time for the 3-D LMW problem with feedback and different shape-update time steps.	105
5.18	Total power vs. time for the 3-D LMW problem with feedback, CONQUEST and QUANDRY solutions without cusping correction. . . .	108
5.19	A summary of the 2-D LRA problem static results for 15×15 cm nodes for several nodal codes.	110
5.20	A summary of the CONQUEST results for 2-D LRA transient using fully-implicit time differencing.	113
5.21	A summary of the CONQUEST results for 2-D LRA transient using the quasi-static method.	113
5.22	A comparison of nodal solutions to the 2-D LRA transient problem. .	115
5.23	A summary of the CONQUEST 3-D LRA problem static results. . . .	116
5.24	A comparison the coarse mesh 3-D LRA problem static results. . . .	116
5.25	A comparison of nodal solutions to the 3-D LRA transient problem. .	118
5.26	A summary of the results of the static calculation for the PWR Operational Transient.	122
5.27	A summary of the CONQUEST and QUANDRY static calculations for the PWR coolant inlet-temperature problem.	126

Chapter 1

INTRODUCTION AND BACKGROUND

1.1 Overview

The design and operation of nuclear reactors requires detailed and accurate knowledge of the spatial and temporal behavior of the core power distribution and neutron interaction rates for all possible core conditions and configurations. This not only includes the intended operational conditions but many hypothetical accident scenarios. Since experimental analysis is impractical because of cost and safety concerns, we must rely on calculational methods.

The traditional method for reactor core calculations is few-group diffusion theory using finite-difference solution techniques. In the application of this method there is a natural subdivision of effort which occurs: obtaining the few-group constants versus solving the multidimensional finite-difference equations. In a typical application, the few-group constants are first obtained by using a simple representation of the spatial dependence in such a manner that a single fuel assembly may be considered. The neutron energy spectrum, however, is represented in considerable detail and is used to collapse the required parameters to a small number of energy groups (typically four or less). Once these parameters are obtained, a fine-mesh finite-difference calculation can be performed and the power distribution and reaction rates can be evaluated.

In the early days of reactor design these two tasks each represented considerable effort and contributed significant errors to the results. The methods of obtaining the few-group constants did not have the flexibility to give parameters which could exactly reproduce reaction rates. Thus, ad hoc corrections were sometimes required to obtain acceptable results. The difficulty with the finite-difference method is not that acceptable accuracy cannot be obtained, but that the calculation effort required

is quite large for multidimensional and transient calculations (a three-dimensional calculation, for example, may require several million finite-difference mesh points).

Fortunately, research over the last 15 years has led to significant advances in both areas. The difficulties in obtaining the few-group constants have been solved by introducing additional group parameters which are determined with the other few-group constants from the spectrum calculation. Generalized Equivalence Theory [S-1] is an example of one method which has been successfully applied to reactor calculations. The calculational expense in performing multidimensional analysis has been significantly reduced by the development of nodal methods. Modern nodal methods provide the accuracy of fine-mesh finite-difference methods while using large, homogeneous nodes. The calculation effort, measured in terms of computer execution time, is two orders of magnitude lower for nodal methods when compared to finite-difference calculations of the same accuracy.

This thesis research is primarily concerned with the development of an advanced nodal method which can be efficiently used for transient reactor analysis. Special properties of the Generalized Equivalence Theory, however, will be exploited to accomplish this task.

1.2 Background

Modern nodal methods which provide accurate solutions to the static and transient multidimensional diffusion equations have been in existence for 15 years. The major distinction of "modern" nodal methods is that mathematically systematic, rather than empirical, formulations are used to obtain the inter-node coupling. As a result, modern nodal methods converge to the exact solution of the diffusion equation as the mesh spacings are reduced.

The inter-node coupling equations of these methods are typically obtained by integration of the neutron diffusion equation over the directions transverse to each coordinate axis. This results in the reduction of the multidimensional equation into a set of coupled one-dimensional equations. The methods by which these one-dimensional equations are solved represent the different classes of nodal methods. One approach

is to solve these one-dimensional equations analytically [S-2, L-1]. This method has proven to be highly successful, but because of its complicated nature, is limited to two energy groups for practical application. Another approach is to approximate the solution of the one-dimensional equations by a low order polynomial [F-1, L-2, S-3]. The polynomial approximation leads to simpler expressions but maintains accuracy that is comparable to that of the analytic methods. As a result of these simple polynomial expressions, there are no practical restrictions on the number of energy groups.

One recent development in nodal methods is in the manner in which the nodal equations are solved. Smith [S-4] has introduced an efficient non-linear iteration method in which Generalized Equivalence Theory is used to force the finite-difference method to match a more accurate nodal model. Applications of this method to the static [S-4] and transient Analytic Nodal Method [H-1], as well as static polynomial methods [S-3, S-7, Z-1], have been performed.

Many of the nodal methods that have been developed have been applied to the transient as well as the static diffusion equation. A wide variety of temporal solution methods have been applied. The QUANDRY computer code, based on the Analytic Nodal Method, uses a simple linear difference method to approximate the time derivatives [S-2]. The polynomial based code IQSBOX, uses a similar fully-implicit time difference but with a frequency transform technique [H-2] to reduce the truncation error. An alternating direction explicit-implicit technique has been employed in the QUABOX/CUBBOX code [L-2]. All of these methods provide accurate results if sufficiently small time steps are taken. A decrease in time step, however, can lead to a significant increase in computational effort since a full spatial calculation is required at each time step.

The number of spatial calculations that must be performed can be reduced by the use of flux factorization methods. In these methods the space-time dependent neutron flux is expressed as the product of a space and energy dependent shape function and a space and energy independent amplitude function. Under many circumstances the shape function varies more slowly in time than the amplitude function. Therefore, accurate results may be obtained by using a larger time step for the shape calcu-

lations than for the amplitude calculations. These methods are called *quasi-static* schemes since the shape function may be approximated using essentially a static calculation [H-3]. Very few transients, however, can be treated accurately using shapes obtained from static, criticality calculations

A more accurate approach is to use a time-dependent equation to perform the shape calculations [O-1]. In this procedure, the flux factorization is substituted into the time-dependent neutron diffusion equation resulting in expressions for the shape and amplitude functions. A large time step is employed for the solution of the shape function while a small time step is used for the solution of the amplitude function. Although transient equations are used for both the shape and amplitude computation, the method is still referred to as a quasi-static method.

The quasi-static method was first applied to the finite-difference diffusion equations and resulted in a large reduction of computation time over more direct treatments. More recently, Taiwo [T-1] and Kao [K-1] have applied the quasi-static procedure to the Analytic Nodal Method. Again, a significant reduction in computation time resulted.

1.3 Research Objectives

The objective of this research is the development of an efficient polynomial nodal method for the solution of the multidimensional, few-group, static and transient neutron diffusion equations. The polynomial method will allow a variable order approximation ranging from quadratic to quartic as well as the finite-difference method. The non-linear procedure discussed above will be implemented for both static and transient calculations to reduce storage requirements, increase efficiency, and add flexibility to the method. No restrictions will be placed on the number or structure of the energy groups and up-scattering in energy will be permitted. Steady-state and time-dependent extraneous neutron sources will also be modelled.

The actual development of the static method closely follows that of Zerkle [Z-1]. The solution method of the nodal equations, has been modified to increase the computational efficiency. This method maintains the accuracy which Zerkle demonstrated

because the fundamental equations are unchanged.

The quasi-static method will be used to solve the time-dependent problem. As discussed in the previous section, the efficiency of the quasi-static method has been demonstrated for both finite-difference and nodal methods. The objective of this application is to show the efficiency of the quasi-static method in combination with the non-linear iteration procedure. The quasi-static method also adds to the flexibility of this nodal method by allowing varying levels of sophistication of the time-dependent solution procedures ranging from point kinetics to full space-time treatment.

Finally, two different thermal-hydraulics models will be incorporated to allow feedback effects to be analyzed. One model is very simple but captures the proper behavior of the reactor. The other thermal-hydraulics model, however, is quite sophisticated including subcooled boiling, two-phase flow, and a two-node fuel conduction model. Again, the flexibility of the nodal method is increased by allowing different levels of sophistication in the thermal hydraulic modelling of the reactor system.

1.4 Thesis Organization

In Chapter 2 the complete mathematical derivation of the static polynomial method is presented. First, a corrected finite-difference method which incorporates equivalence parameters is developed along with rigorous mathematical definitions of those equivalence parameters. Then, a polynomial nodal method is derived and the non-linear iteration procedure is introduced.

In Chapter 3 a similar derivation of the corrected finite-difference and polynomial methods for the transient neutron diffusion equation. The quasi-static method is introduced and complete specification of the required point kinetics parameters is given. The thermal-hydraulics and cross section feedback models are also discussed.

The numerical properties and solution methods for the static and transient equations are presented in Chapter 4. The nodal method is then used to solve several static and transient problems with the results being presented in Chapter 5.

Finally, Chapter 6 presents a summary and conclusions of this research. Recommendations for further research are also made.

Chapter 2

DERIVATION OF THE STATIC NODAL EQUATIONS

2.1 Introduction

In this chapter the static nodal equations will be derived from the few-group diffusion equations. First, corrected finite-difference equations are obtained which are rigorous in the sense that they can reproduce the results of any reference calculation if appropriate equivalence parameters are supplied. Next, a polynomial nodal method which will produce accurate results for assembly sized nodes will be introduced. The complete nodal method is obtained when the corrected finite-difference equations are forced to match the polynomial nodal method.

2.2 Notation and the Nodal Balance Equation

The starting point for our derivation is the few-group, steady-state diffusion equations in P_1 form [H-4]

$$\nabla \cdot \mathbf{J}_g(\mathbf{r}) + \Sigma_{tg}(\mathbf{r})\phi_g(\mathbf{r}) = \sum_{g'=1}^G \left[\frac{1}{\lambda} \chi_g \nu \Sigma_{fg'}(\mathbf{r}) + \Sigma_{gg'}(\mathbf{r}) \right] \phi_{g'}(\mathbf{r}) + q_g(\mathbf{r}), \quad (2.1a)$$

$$\mathbf{J}_g(\mathbf{r}) = -D_g(\mathbf{r})\nabla\phi_g(\mathbf{r}) \quad g = 1, 2, \dots, G. \quad (2.1b)$$

Where

$\mathbf{J}_g(\mathbf{r})$ = net neutron current in group g ($\text{cm}^{-2}\text{s}^{-1}$),

$\phi_g(\mathbf{r})$ = scalar neutron flux in group g ($\text{cm}^{-2}\text{s}^{-1}$),

λ = reactor eigenvalue,

$\Sigma_{tg}(\mathbf{r})$ = macroscopic total cross section for group g (cm^{-1}),

- χ_g = fission spectrum for group g ,
- $\nu\Sigma_{fg}(\mathbf{r})$ = mean number of neutrons emitted per fission times the macroscopic fission cross section for group g (cm^{-1}).
- $\Sigma_{gg'}(\mathbf{r})$ = macroscopic transfer cross section from group g' to g (cm^{-1}).
- $q_g(\mathbf{r})$ = extraneous neutron source in group g ($\text{cm}^{-3}\text{s}^{-1}$),
- $D_g(\mathbf{r})$ = diffusion coefficient for group g (cm),
- G = total number of energy groups.

Note that this equation encompasses both eigenvalue and source problems. When eigenvalue problems are considered, the source, $q_g(\mathbf{r})$, is zero and when source problems are considered, the eigenvalue, λ , is either unity or a value which forces the system to be subcritical.

Equations (2.1a) and (2.1b) are obtained by integrating the continuous-energy diffusion equation over discrete energy groups. The group parameters (cross sections and diffusion coefficients) are obtained by a spectrum-weighted average over the energy group. Recall that few-group cross sections use a pre-determined spectrum while a multi-group approach uses an arbitrary spectrum [H-4]. As their names suggest, few-group calculations typically require a smaller number of energy groups than multi-group calculations. In general it is impossible to obtain group constants which can exactly replicate the solution to the Boltzmann transport equation. Later in the derivation more parameters will be introduced to overcome this difficulty.

Three-dimensional Cartesian geometry will be used for the nodal method to avoid complexity and allow modelling of a wide variety of reactor systems. Note that polynomial nodal methods for other geometries are currently being developed at M.I.T. [B-1, S-5, D-1]. In this derivation a generalized coordinate system will be used in which the coordinates are represented by u , v , and w . The spatial domain of the reactor is divided into a regular rectangular array of right parallelepipeds consisting of nuclearly homogenous materials and with grid indices defined by u_l , v_m , and

w_n where

$$l, m, n = \begin{cases} i = 1, 2, \dots, I & u = x, y, z \\ j = 1, 2, \dots, J & v \neq u \\ k = 1, 2, \dots, K & w \neq u, v. \end{cases}$$

The node (i, j, k) is defined by

$$\begin{aligned} x &\in [x_i, x_{i+1}], \\ y &\in [y_j, y_{j+1}], \\ z &\in [z_k, z_{k+1}]. \end{aligned}$$

The node widths are

$$h_u^l \equiv u_{l+1} - u_l; \quad u = x, y, z,$$

and the node volume is

$$V^{ijk} \equiv h_x^i h_y^j h_z^k.$$

In Cartesian geometry Eq. (2.1a) and (2.1b) are

$$\frac{\partial}{\partial x} J_{gx}(x, y, z) + \frac{\partial}{\partial y} J_{gy}(x, y, z) + \frac{\partial}{\partial z} J_{gz}(x, y, z) + \Sigma_{tg}(x, y, z) \phi_g(x, y, z) = \quad (2.2a)$$

$$\sum_{g'=1}^G \left[\frac{1}{\lambda} \chi_{g\nu} \Sigma_{fg'}(x, y, z) + \Sigma_{gg'}(x, y, z) \right] \phi_{g'}(x, y, z) + q_g(x, y, z),$$

$$J_{gu}(x, y, z) = -D_g(x, y, z) \frac{\partial}{\partial u} \phi_g(x, y, z), \quad u = x, y, z. \quad (2.2b)$$

The first step in the derivation of the nodal method is to integrate Eq. (2.2a) over the volume of node (i, j, k) and then divide by V^{ijk} to obtain the *nodal balance equation*

$$\begin{aligned} \frac{1}{h_x^i} [J_{gx}^{jk}(x_{i+1}) - J_{gx}^{jk}(x_i)] + \frac{1}{h_y^j} [J_{gy}^{ik}(y_{j+1}) - J_{gy}^{ik}(y_j)] + \frac{1}{h_z^k} [J_{gz}^{ij}(z_{k+1}) - J_{gz}^{ij}(z_k)] \\ + \Sigma_{tg}^{ijk} \bar{\phi}_g^{ijk} = \sum_{g'=1}^G \left[\frac{1}{\lambda} \chi_{g\nu} \Sigma_{fg'}^{ijk} \bar{\phi}_{g'}^{ijk} + \Sigma_{gg'}^{ijk} \right] \bar{\phi}_{g'}^{ijk} + \bar{q}_g^{ijk}, \end{aligned} \quad (2.3)$$

where

$$\bar{\phi}_g^{ijk} \equiv \frac{1}{V^{ijk}} \int_{x_i}^{x_{i+1}} dx \int_{y_j}^{y_{j+1}} dy \int_{z_k}^{z_{k+1}} dz \phi_g(x, y, z), \quad (2.4)$$

$$\bar{q}_g^{ijk} \equiv \frac{1}{V^{ijk}} \int_{x_i}^{x_{i+1}} dx \int_{y_j}^{y_{j+1}} dy \int_{z_k}^{z_{k+1}} dz q_g(x, y, z), \quad (2.5)$$

$$J_{gu}^{mn}(u) \equiv \frac{1}{h_v^m h_w^n} \int_{v_m}^{v_{m+1}} dv \int_{w_n}^{w_{n+1}} dw J_{gu}(u, v, w), \quad \begin{array}{l} u = x, y, z \\ v \neq u \\ w \neq u, v \end{array} \quad (2.6)$$

and the cross sections represent averages over the node. Despite the fact that this equation has been obtained without approximation, it is incomplete since it relates several unknown quantities, namely the node-averaged fluxes and surface-averaged net currents. This represents seven unknowns for each node and energy group. Therefore, we must provide six additional relations between the node-averaged fluxes and the surface-averaged currents to obtain a complete system of equations. These additional equations are called *coupling equations*. Two different approaches will be considered for obtaining the coupling equations: a corrected finite-difference method and a polynomial nodal method.

2.3 Corrected Finite-Difference Coupling Equations

One method of obtaining the additional required relations is the finite-difference approximation. We begin by integrating the second P_1 equation, Eq. (2.2b), over the node and dividing by the node volume

$$J_{gu}^{mn}(u) = -\frac{D_g^{lmn}}{h_v^m h_w^n} \frac{d}{du} \int_{v_m}^{v_{m+1}} dv \int_{w_n}^{w_{n+1}} dw \phi_g(u, v, w), \quad \begin{array}{l} u = x, y, z, \\ u \in [u_l, u_{l+1}]. \end{array} \quad (2.7)$$

We can now approximate the derivative in this equation as a simple difference as follows

$$J_{gu}^{mn}(u_l) \approx -D_g^{lmn} \frac{\bar{\phi}_g^{lmn} - \phi_{gu}^{mn}(u_l^+)}{h_u^l/2}. \quad (2.8)$$

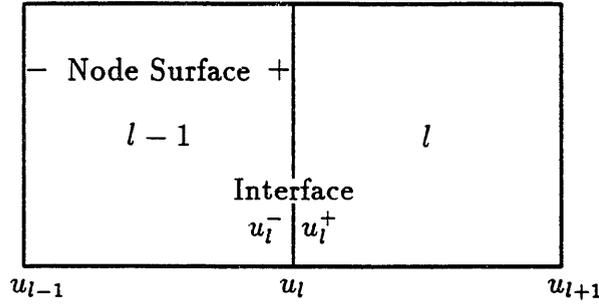


Figure 2-1: Diagram showing the the surface and node labeling conventions.

Where the surface-averaged flux is defined by

$$\phi_{gu}^{mn}(u) \equiv \frac{1}{h_v^m h_w^n} \int_{v_m}^{v_{m+1}} dv \int_{w_n}^{w_{n+1}} dw \phi_g(u, v, w), \quad (2.9)$$

and u_l^+ indicates the positive side of the interface as shown in Figure 2-1.

Note that this differencing is also equivalent to assuming that the flux in the node varies linearly from the node surface, where it has the value $\phi_{gu}^{mn}(u_l^+)$, to the node center, where it has the value $\bar{\phi}_g^{lmn}$. A similar procedure can be performed for the adjacent node to obtain another relation for net current at the interface

$$J_{gu}^{mn}(u_l) \approx -D_g^{lmn} \frac{\phi_{gu}^{mn}(u_l^-) - \bar{\phi}_g^{lmn}}{h_u^{l-1}/2}. \quad (2.10)$$

These equations, however, are only accurate for small mesh spacings and can lead to large errors when entire assemblies are used as nodes. For LWRs, in fact, finite-difference methods typically require about one mesh per fuel pin to obtain spatially-converged results.

This difficulty is overcome by the introduction of correction factors which force Eqs. (2.8) and (2.10) to be formally exact [H-5]. This procedure is performed by multiplying the surface-averaged fluxes in Eqs. (2.8) and (2.10) by correction factors

to give the true surface-averaged flux at that interface

$$\begin{aligned}\phi_{gu}^{mn}(u_l) &= f_{gu-}^{lmn} \phi_{gu}^{mn}(u_l^+) \\ &= f_{gu+}^{l-1,mn} \phi_{gu}^{mn}(u_l^-).\end{aligned}\tag{2.11}$$

Since the correction factors, f_{gu-}^{lmn} and $f_{gu+}^{l-1,mn}$, are generally not equal, the surface-averaged fluxes appearing in Eqs. (2.8) and (2.10) must be discontinuous. For this reason, these correction factors are called *discontinuity factors*. Inserting the correction given in Eq. (2.11) into the Eqs.(2.8) and (2.10) results in the following equations, which, when used with reference values for the net currents and fluxes, also serve as definitions for the discontinuity factors

$$\begin{aligned}J_{gu}^{mn}(u_l) &\equiv -D_g^{lmn} \frac{\bar{\phi}_g^{lmn} - \phi_{gu}^{mn}(u_l)/f_{gu-}^{lmn}}{h_u^l/2} \\ &\equiv -D_g^{l-1,mn} \frac{\phi_{gu}^{mn}(u_l)/f_{gu+}^{l-1,mn} - \bar{\phi}_g^{l-1,mn}}{h_u^{l-1}/2}.\end{aligned}\tag{2.12}$$

The context in which we have introduced the discontinuity factors is for the correction of the spatial difference errors. Their original purpose in General Equivalence Theory, however, was to correct for errors made in treating heterogeneous regions as having a homogenous composition [S-1]. Further, they can be used to correct for the diffusion theory approximation and for errors in obtaining diffusion coefficients. Throughout the remainder of the corrected finite-difference derivation we shall treat these discontinuity factors as being corrections for all errors (spatial, homogenization, and diffusion theory).

Now we can obtain the final relations necessary in obtaining a complete set of nodal equations. By using the continuity condition given in Eq.(2.11) we can eliminate the surface-averaged fluxes from Eq. (2.12) to obtain the following expression relating the surface-averaged net current to the node-averaged fluxes

$$J_{gu}^{mn}(u_l) = - \left[\frac{h_u^l}{2D_g^{lmn}} \frac{f_{gu-}^{lmn}}{f_{gu+}^{l-1,mn}} + \frac{h_u^{l-1}}{2D_g^{l-1,mn}} \right]^{-1} \left(\frac{f_{gu-}^{lmn}}{f_{gu+}^{l-1,mn}} \bar{\phi}_g^{lmn} - \bar{\phi}_g^{l-1,mn} \right).\tag{2.13}$$

This corrected finite-difference coupling equation has three important properties. First, the equation is formally exact since the discontinuity factors correct for the deficiencies of Eqs. (2.8) and (2.10). Second, the discontinuity factors in the coupling equation appear as a ratio. Thus, in the application of the method we need not store the factors individually but as ratios. Finally, if the ratios of the discontinuity factors at the interfaces are unity, we obtain the mesh-centered finite-difference equations.

Similar manipulations of the equations can be performed for the node interface u_{l+1} to obtain the following coupling equation

$$J_{gu}^{mn}(u_{l+1}) = \left[\frac{h_u^l}{2D_g^{lmn}} \frac{f_{gu+}^{lmn}}{f_{gu-}^{l+1,mn}} + \frac{h_u^{l+1}}{2D_g^{l+1,mn}} \right]^{-1} \left(\frac{f_{gu+}^{lmn}}{f_{gu-}^{l+1,mn}} \bar{\phi}_g^{lmn} - \bar{\phi}_g^{l+1,mn} \right). \quad (2.14)$$

Substituting Eqs. (2.13) and (2.14) into the nodal balance equation, Eq. (2.3), results in a nodal equation which has the finite-difference form

$$\begin{aligned} & \frac{1}{h_x^i} \left[\frac{h_x^i}{2D_g^{ijk}} \frac{f_{gx-}^{ijk}}{f_{gx+}^{i-1,jk}} + \frac{h_x^{i-1}}{2D_g^{i-1,jk}} \right]^{-1} \left(\frac{f_{gx-}^{ijk}}{f_{gx+}^{i-1,jk}} \bar{\phi}_g^{ijk} - \bar{\phi}_g^{i-1,jk} \right) \\ & + \frac{1}{h_x^i} \left[\frac{h_x^i}{2D_g^{ijk}} \frac{f_{gx+}^{ijk}}{f_{gx-}^{i+1,jk}} + \frac{h_x^{i+1}}{2D_g^{i+1,jk}} \right]^{-1} \left(\frac{f_{gx+}^{ijk}}{f_{gx-}^{i+1,jk}} \bar{\phi}_g^{ijk} - \bar{\phi}_g^{i+1,jk} \right) \\ & + \frac{1}{h_y^j} \left[\frac{h_y^j}{2D_g^{ijk}} \frac{f_{gy-}^{ijk}}{f_{gy+}^{i,j-1,k}} + \frac{h_y^{j-1}}{2D_g^{i,j-1,k}} \right]^{-1} \left(\frac{f_{gy-}^{ijk}}{f_{gy+}^{i,j-1,k}} \bar{\phi}_g^{ijk} - \bar{\phi}_g^{i,j-1,k} \right) \\ & + \frac{1}{h_y^j} \left[\frac{h_y^j}{2D_g^{ijk}} \frac{f_{gy+}^{ijk}}{f_{gy-}^{i,j+1,k}} + \frac{h_y^{j+1}}{2D_g^{i,j+1,k}} \right]^{-1} \left(\frac{f_{gy+}^{ijk}}{f_{gy-}^{i,j+1,k}} \bar{\phi}_g^{ijk} - \bar{\phi}_g^{i,j+1,k} \right) \\ & + \frac{1}{h_z^k} \left[\frac{h_z^k}{2D_g^{ijk}} \frac{f_{gz-}^{ijk}}{f_{gz+}^{i,j,k-1}} + \frac{h_z^{k-1}}{2D_g^{i,j,k-1}} \right]^{-1} \left(\frac{f_{gz-}^{ijk}}{f_{gz+}^{i,j,k-1}} \bar{\phi}_g^{ijk} - \bar{\phi}_g^{i,j,k-1} \right) \\ & + \frac{1}{h_z^k} \left[\frac{h_z^k}{2D_g^{ijk}} \frac{f_{gz+}^{ijk}}{f_{gz-}^{i,j,k+1}} + \frac{h_z^{k+1}}{2D_g^{i,j,k+1}} \right]^{-1} \left(\frac{f_{gz+}^{ijk}}{f_{gz-}^{i,j,k+1}} \bar{\phi}_g^{ijk} - \bar{\phi}_g^{i,j,k+1} \right) \\ & + \sum_{g'=1}^G \bar{\phi}_g^{ijk} = \sum_{g'=1}^G \left[\frac{1}{\lambda} \chi_{g'} \nu \Sigma_{fg'}^{ijk} + \Sigma_{gg'}^{ijk} \right] \bar{\phi}_{g'}^{ijk} + q_g^{ijk}. \end{aligned} \quad (2.15)$$

This equation can be written more compactly using matrix notation to collapse all of the spatial dependence

$$\mathbf{N}_g \phi_g = \frac{1}{\lambda} \sum_{g'=1}^G \mathbf{F}_{gg'} \phi_{g'} + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'} \phi_{g'} + \mathbf{q}_g \quad (2.16)$$

where

- \mathbf{N}_g = A seven-stripe $N \times N$ matrix containing the coupling terms for group g ,
the total cross section and the in-group scattering terms,
- ϕ_g = A column vector of length N containing the fluxes for group g ,
- $\Sigma_{gg'}$ = A diagonal $N \times N$ matrix containing $\{\Sigma_{gg'}^{lmn}\}$,
- $\mathbf{F}_{gg'}$ = A diagonal $N \times N$ matrix containing $\{\chi_{g\nu} \Sigma_{fg'}^{lmn}\}$,
- \mathbf{q}_g = A column vector of length N containing the extraneous source terms,
- N = The total number of nodes = $I \times J \times K$.

An even more compact form is obtained by collapsing this equation into the following super-matrix equation

$$\mathbf{A} \Phi = \frac{1}{\lambda} \mathbf{M} \Phi + \mathbf{q}, \quad (2.17)$$

where

- \mathbf{A} = An $NG \times NG$ matrix containing $\{\mathbf{N}_g \delta_{gg'} - \Sigma_{gg'}\}$,
- Φ = A vector of length NG of the fluxes, $\text{col}\{\phi_g\}$,
- \mathbf{M} = An $NG \times NG$ matrix containing $\{\mathbf{F}_{gg'}\}$,
- \mathbf{q} = A vector of length NG representing the extraneous source, $\text{col}\{\mathbf{q}_g\}$.

These matrix forms will be useful in examining the numerical properties and solution methods for the static problems in Chapter 4.

2.3.1 Boundary Conditions

The same boundary conditions which are used by Zerkle [Z-1] will be applied. The boundary conditions are represented by the following equation

$$\phi_{gu}^{mn}(u_s) = \Gamma_{gu\pm}^{mn} J_{gu}^{mn}(u_s) \hat{\ell} \cdot \hat{\mathbf{n}} \quad (2.18)$$

where

$$\begin{aligned}
\phi_{gu}^{mn}(u_s) &= \text{Surface-averaged flux at boundary,} \\
J_{gu}^{mn}(u_s) &= \text{Surface-averaged current at boundary,} \\
u_s &= \text{External boundary,} \\
\hat{\ell} &= \text{Unit vector in the positive direction of the coordinate axis,} \\
\hat{n} &= \text{Unit normal vector of external boundary,} \\
\Gamma_{gu\pm}^{mn} &= \text{Boundary condition factor having the following values:} \\
&\quad \Gamma_{gu\pm}^{mn} = 0 \quad \text{zero flux} \\
&\quad \Gamma_{gu\pm}^{mn} = 2 \quad \text{zero incoming current} \\
&\quad \Gamma_{gu\pm}^{mn} = \infty \quad \text{zero current} \\
&\quad \Gamma_{gu\pm}^{mn} = 2 + \frac{4}{\alpha_g^{-1} - 1} \quad \text{albedo where } \alpha_g = \frac{J_{gu}^{\text{in}}(u_s)}{J_{gu}^{\text{out}}(u_s)}.
\end{aligned}$$

The expression for the current at the external surfaces required in Eq.(2.3) is obtained by combining Eqs. (2.12) and (2.18) to eliminate the surface-averaged fluxes. The resulting expressions for a lower and an upper surface are given by

$$J_{gu}^{mn}(u_s) = - \left[\frac{\Gamma_{gu-}^{mn}}{f_{gu-}^{lmn}} + \frac{h_u^l}{2D_g^{lmn}} \right]^{-1} \bar{\phi}_g^{lmn}, \quad (2.19)$$

and,

$$J_{gu}^{mn}(u_s) = \left[\frac{\Gamma_{gu+}^{mn}}{f_{gu+}^{lmn}} + \frac{h_u^l}{2D_g^{lmn}} \right]^{-1} \bar{\phi}_g^{lmn}, \quad (2.20)$$

respectively.

2.3.2 Evaluation of the Discontinuity Factors

The discontinuity factors used in this method can be obtained from any reference solution by treating Eq. (2.13) as a defining relation for the ratios of the discontinuity factors about an interface. By rearranging this equation we obtain the following equation

$$\frac{f_{gu+}^{l-1,mn}}{f_{gu-}^{lmn}} = \frac{\bar{\phi}_g^{lmn} + \frac{h_u^l}{2D_g^{lmn}} J_{gu}^{mn}(u_l)}{\bar{\phi}_g^{l-1,mn} - \frac{h_u^{l-1}}{2D_g^{l-1,mn}} J_{gu}^{mn}(u_l)}. \quad (2.21)$$

At the lower boundary, Eq. (2.19) can be rearranged to give

$$\frac{\Gamma_{gu-}^{mn}}{f_{gu-}^{lmn}} = - \left(\frac{\bar{\phi}_g^{lmn}}{J_{gu}^{mn}(u_s)} + \frac{h_u^l}{2D_g^{lmn}} \right), \quad (2.22)$$

and at the upper boundary, manipulating Eq. (2.20) gives

$$\frac{\Gamma_{gu+}^{mn}}{f_{gu+}^{lmn}} = \left(\frac{\bar{\phi}_g^{lmn}}{J_{gu}^{mn}(u_s)} - \frac{h_u^l}{2D_g^{lmn}} \right). \quad (2.23)$$

Note that when a zero flux boundary condition is applied, Γ_g^{mn} is zero but the ratio $\Gamma_g^{mn}/f_{gu-}^{lmn}$ may not be. By introducing the discontinuity factor we simply apply different boundary conditions which result in the correct leakage at the boundaries.

2.4 Polynomial Coupling Equations

As previously mentioned, the nodal coupling equations obtained by the finite-difference approximation are not accurate for assembly-sized nodes. In this section we shall obtain more accurate coupling relations by assuming that the flux can be represented by quartic polynomials. We begin our derivation of these higher-order relations by applying the *transverse-integration procedure*. Our goal in applying this procedure is to reduce the three-dimensional equation, Eq. (2.2a), to three one-dimensional equations. This is reasonable since it is generally easier to solve one-dimensional equations than three-dimensional equations. As will be seen, these one-dimensional equations will be coupled by leakage terms and will require an approximation for practical solution.

2.4.1 The Transverse-Integration Procedure

Three coupled, one-dimensional equations are obtained by integrating the neutron diffusion equations in the directions transverse to the direction of interest. This is accomplished by operating on Eq. (2.2a) with

$$\frac{1}{h_v^m h_w^n} \int_{v_m}^{v_{m+1}} dv \int_{w_n}^{w_{n+1}} dw.$$

Thus, we can obtain a one-dimensional equation in the direction u by integrating Eqs. (2.2a) and (2.2b) over a node in directions v and w . The result is

$$\begin{aligned} \frac{d}{du} J_{gu}^{mn}(u) + \Sigma_{tg}^{lmn} \phi_{gu}^{mn}(u) = \sum_{g'=1}^G \left[\frac{1}{\lambda} \chi_g \nu \Sigma_{fg'}^{lmn} + \Sigma_{gg'}^{lmn} \right] \phi_{gu'}^{mn}(u) \\ - S_{gu}^{mn}(u) + q_g^{mn}(u), \end{aligned} \quad (2.24a)$$

$$J_{gu}^{mn}(u) = -D_g^{lmn} \frac{d}{du} \phi_{gu}^{mn}(u), \quad u = x, y, z, \quad u \in [u_l, u_{l+1}] \quad (2.24b)$$

where

$$\begin{aligned} \phi_{gu}^{mn}(u) &\equiv \frac{1}{h_v^m h_w^n} \int_{v_m}^{v_{m+1}} dv \int_{w_n}^{w_{n+1}} dw \phi_g(u, v, w), \\ J_{gu}^{mn}(u) &\equiv \frac{1}{h_v^m h_w^n} \int_{v_m}^{v_{m+1}} dv \int_{w_n}^{w_{n+1}} dw J_g(u, v, w), \\ S_{gu}^{mn}(u) &\equiv \frac{1}{h_v^m} L_{gv}^{mn}(u) + \frac{1}{h_w^n} L_{gw}^{mn}(u), \\ L_{gv}^{mn}(u) &\equiv \frac{1}{h_v^m h_w^n} \int_{w_n}^{w_{n+1}} dw [J_{gv}(u, v_{m+1}, w) - J_{gv}(u, v_m, w)], \\ L_{gw}^{mn}(u) &\equiv \frac{1}{h_v^m h_w^n} \int_{v_m}^{v_{m+1}} dv [J_{gw}(u, v, w_{n+1}) - J_{gw}(u, v, w_n)]. \end{aligned}$$

The transversely-integrated equations (2.24a) and (2.24b) can be combined to obtain a system of ordinary, second-order, inhomogenous differential equations with constant coefficients. If these equations are solved analytically we obtain the Analytic Nodal Method developed by Smith [S-2]. The resulting solution, however, is rather complicated and for practical application is limited to two energy groups.

An alternate approach is to assume that the transversely-integrated fluxes have a polynomial form and to apply a weighted residual procedure to determine the polynomial coefficients [F-1]. If the transversely-integrated flux can be adequately represented by a low order polynomial, relatively simple expressions result. Furthermore, since the equations for each energy group can be treated individually, generalization to more energy groups is straightforward. For these reasons, the polynomial expan-

sion procedure along with a weighted residual procedure for determining the higher order expansion coefficients has been adopted.

2.4.2 The Polynomial Expansion

The transversely-integrated flux is approximated by a truncated polynomial

$$\phi_{gu}^{mn}(u) \approx \sum_{p=1}^P a_{gup}^{lmn} f_p\left(\frac{u - u_l}{h_u^l}\right), \quad u \in [u_l, u_{l+1}]. \quad (2.25)$$

Previous applications of polynomial methods [F-1] have shown that at least a fourth-order polynomial is required to obtain acceptable results for light water reactor applications. Further approximations, yet to be discussed, limit the accuracy such that using polynomials higher than fourth-order is not warranted. Thus, in this method we shall use a quartic polynomial approximation. For this case the basis functions are defined by [F-1, Z-1]

$$f_0(\xi) = 1, \quad (2.26a)$$

$$f_1(\xi) = \xi - \frac{1}{2}, \quad (2.26b)$$

$$f_2(\xi) = 3\xi^2 - 3\xi + \frac{1}{2}, \quad (2.26c)$$

$$f_3(\xi) = \xi(1 - \xi)\left(\xi - \frac{1}{2}\right), \quad (2.26d)$$

$$f_4(\xi) = \xi(1 - \xi)\left(\xi^2 - \xi + \frac{1}{5}\right). \quad (2.26e)$$

These polynomials have been chosen such that

$$\int_0^1 f_p(\xi) d\xi = \begin{cases} 1 & p = 0 \\ 0 & p = 1, 2, 3, 4 \end{cases}. \quad (2.27)$$

In addition, the higher order basis functions are required to satisfy

$$f_p(0) = f_p(1) = 0, \quad p = 3, 4. \quad (2.28)$$

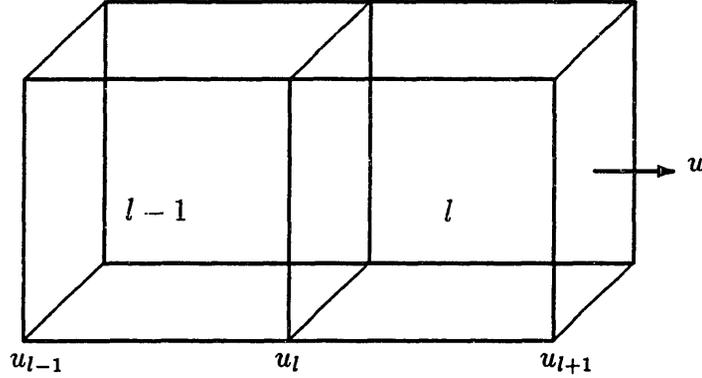


Figure 2-2: Diagram showing the orientation of the two-node problem.

This constraint on the higher order expansion functions is convenient since it leads to expressions which relate the first three expansion coefficients only to the node-averaged and surface-averaged fluxes, not the higher-order expansion coefficients.

Using the polynomial expression for the transversely-integrated flux, we can evaluate several key quantities in terms of these quartic polynomial expansion coefficients:

$$\bar{\phi}_g^{lmn} = a_{gu0}^{lmn}, \quad (2.29a)$$

$$\phi_{gu}^{mn}(u_{l+1}^-) = a_{gu0}^{lmn} + \frac{1}{2}a_{gu1}^{lmn} + \frac{1}{2}a_{gu2}^{lmn}, \quad (2.29b)$$

$$\phi_{gu}^{mn}(u_l^+) = a_{gu0}^{lmn} - \frac{1}{2}a_{gu1}^{lmn} + \frac{1}{2}a_{gu2}^{lmn}, \quad (2.29c)$$

$$J_{gu}^{mn}(u_l) = -\frac{D_g^{lmn}}{h_u^l} \left[a_{gu1}^{lmn} - 3a_{gu2}^{lmn} - \frac{1}{2}a_{gu3}^{lmn} + \frac{1}{5}a_{gu4}^{lmn} \right], \quad (2.29d)$$

$$J_{gu}^{mn}(u_{l+1}) = -\frac{D_g^{lmn}}{h_u^l} \left[a_{gu1}^{lmn} + 3a_{gu2}^{lmn} - \frac{1}{2}a_{gu3}^{lmn} - \frac{1}{5}a_{gu4}^{lmn} \right]. \quad (2.29e)$$

2.4.3 The Two-Node Problem

The determination of the expansion coefficients is made by solving the two-node problem shown in Figure 2-2 [H-6]. Our goal in solving this two-node problem is the determination of the surface-averaged current at the interface of the two nodes in terms of the node-averaged fluxes. This will result in a more accurate coupling relation than the finite-difference coupling expression (Eq. (2.13) with unity discontinuity

factor ratios).

For this two-node problem we have five unknown expansion coefficients for each node and energy group. As Eq. (2.29a) shows, the first expansion coefficient is the node-averaged flux, leaving four unknown expansion coefficients for each node and group. Thus, eight equations are required for each energy group to completely specify the polynomial approximation in the two nodes. The equations which will be used are:

1. A nodal balance equation for each node, (2)
2. Continuity of current at the interface, (1)
3. "Discontinuity" of flux at the interface, (1)
4. Two weighted residual equations for each node. (4)

The numbers in parenthesis indicate the number of equations to be obtained from each condition.

2.4.4 The Weighted Residual Procedure

Two equations for each node in the two-node problem are provided by using a weighted residual procedure. Since the truncated polynomial cannot match the exact solution of the transversely-integrated diffusion equation we require it to satisfy this equation in a weighted-integral sense. The weight functions can be chosen arbitrarily, but two different methods are typically used: Galerkin weighting, where the polynomials are weighted by themselves; and moments weighting, where polynomials of increasing order are used successively as weight functions. Previous applications of polynomial nodal methods have shown that moments weighting is superior [F-1].

We begin the weighted residual procedure by multiplying Eq. (2.24a) by a weight function $w_p(u)$ and integrate over the node. The resulting equation is

$$\left\langle w_p(u), \frac{d}{du} J_{gu}^{mn}(u) \right\rangle + \frac{D_g^{lmn}}{(h_u^l)^2} \sum_{g'=1}^G (\kappa_u^2)_{gg'}^{lmn} \phi_{g'up}^{lmn} + S_{gup}^{lmn} = -q_{gup}^{lmn} \quad (2.30)$$

where the brackets indicate inner products as in the following definitions

$$\phi_{gup}^{lmn} \equiv \langle w_p(u), \phi_{gu}^{lmn} \rangle \equiv \frac{1}{h_u^l} \int_{u_l}^{u_{l+1}} w_p(u) \phi_{gu}^{lmn}(u) du, \quad (2.31a)$$

$$S_{gup}^{lmn} \equiv \langle w_p(u), S_{gu}^{lmn}(u) \rangle, \quad (2.31b)$$

$$q_{gup}^{lmn} \equiv \langle w_p(u), q_{gu}^{lmn}(u) \rangle, \quad (2.31c)$$

and

$$(\kappa_u^2)_{gg'}^{lmn} \equiv \frac{(h_u^l)^2}{D_g^{lmn}} \left[\Sigma_{tg}^{lmn} \delta_{gg'} - \Sigma_{gg'}^{lmn} - \frac{1}{\lambda} \chi_g \nu \Sigma_{fg'}^{lmn} \right]. \quad (2.31d)$$

For moments weighting, the weight functions are given by

$$w_1(u) = f_1\left(\frac{u - u_l}{h_u^l}\right) = \left(\frac{u - u_l}{h_u^l}\right) - \frac{1}{2}, \quad (2.32a)$$

$$w_2(u) = f_2\left(\frac{u - u_l}{h_u^l}\right) = 3 \left(\frac{u - u_l}{h_u^l}\right)^2 - 3 \left(\frac{u - u_l}{h_u^l}\right) + \frac{1}{2}. \quad (2.32b)$$

After substitution of the polynomial approximation into Eq. (2.31a) and performing the necessary integrations, we find that the first and second *flux moments* are

$$\phi_{gu1}^{lmn} = \frac{1}{12} a_{gu1}^{lmn} + \frac{1}{120} a_{gu3}^{lmn}, \quad (2.33a)$$

$$\phi_{gu2}^{lmn} = \frac{1}{20} a_{gu2}^{lmn} + \frac{1}{700} a_{gu4}^{lmn}. \quad (2.33b)$$

In a similar fashion, the first and second *current-derivative moments* are obtained by substituting the polynomial approximation into Eq. (2.24b) and evaluating the inner products resulting in

$$\left\langle w_1(u), \frac{d}{du} J_{gu}^{lmn}(u) \right\rangle = \frac{1}{2} \frac{D_g^{lmn}}{(h_u^l)^2} a_{gu3}^{lmn} \quad (2.34a)$$

$$\left\langle w_2(u), \frac{d}{du} J_{gu}^{lmn}(u) \right\rangle = \frac{1}{5} \frac{D_g^{lmn}}{(h_u^l)^2} a_{gu4}^{lmn} \quad (2.34b)$$

The evaluation of the *transverse-leakage moments* requires more information since we do not know how the transverse leakage varies within the node in the u -direction.

The most common and accurate manner of treating this spatial dependence is the quadratic transverse-leakage approximation [B-2]. In this approximation, the u -directed transverse leakage is expanded in a quadratic polynomial which preserves the node-averaged transverse leakages in the node of interest and its two neighbors in the u -direction. This polynomial, however, is used only for the central node. The quadratic transverse leakage has the form

$$S_{gu}^{mn}(u) = \bar{S}_{gu}^{lmn} + \left(\bar{S}_{gu}^{l-1,mn} - \bar{S}_{gu}^{lmn} \right) \rho_{u_l}^{l-1}(u) + \left(\bar{S}_{gu}^{l+1,mn} - \bar{S}_{gu}^{lmn} \right) \rho_{u_l}^{l+1}(u), \quad (2.35)$$

where the u -direction node-averaged transverse leakage for node (l, m, n) is given by

$$\bar{S}_{gu}^{lmn} \equiv \frac{1}{h_u^l} \int_{u_l}^{u_{l+1}} S_{gu}^{mn}(u) du, \quad (2.36)$$

and the quadratic polynomials are

$$\begin{aligned} \rho_{u_l}^{l-1}(u) &= a_{u_l}^- + b_{u_l}^- \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^- \left(\frac{u - u_l}{h_u^l} \right)^2, \\ \rho_{u_l}^{l+1}(u) &= a_{u_l}^+ + b_{u_l}^+ \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^+ \left(\frac{u - u_l}{h_u^l} \right)^2. \end{aligned}$$

The coefficients of these polynomials are obtained by requiring the quadratic to preserve the leakage in the three adjacent nodes. The resulting coefficients, $a_{u_l}^\pm$, $b_{u_l}^\pm$, and $c_{u_l}^\pm$, depend only on the node widths. The complete specifications of these coefficients and the transverse-leakage moments are given in Appendix A. The resulting evaluation of the transverse-leakage moments is

$$\begin{aligned} S_{gu1}^{lmn} &= \frac{1}{12} \left[(b_{u_l}^- + c_{u_l}^-) \bar{S}_{gu}^{l-1,mn} - (b_{u_l}^- + b_{u_l}^+ + c_{u_l}^- + c_{u_l}^+) \bar{S}_{gu}^{lmn} \right. \\ &\quad \left. + (b_{u_l}^+ + c_{u_l}^+) \bar{S}_{gu}^{l+1,mn} \right], \end{aligned} \quad (2.37a)$$

$$S_{gu2}^{lmn} = \frac{1}{60} \left[c_{u_l}^- \bar{S}_{gu}^{l-1,mn} - (c_{u_l}^- + c_{u_l}^+) \bar{S}_{gu}^{lmn} + c_{u_l}^+ \bar{S}_{gu}^{l+1,mn} \right]. \quad (2.37b)$$

The remaining term in Eq. (2.30) that must be evaluated is the extraneous neutron

source moment q_{gup}^{lmn} . Since we have assumed that the source is homogenous within the node, the source moments are zero

$$q_{gup}^{lmn} = 0, \quad p = 1, 2. \quad (2.38)$$

The two required weighted residual equations are obtained by substitution of Eqs. (2.33a) and (2.34a) into (2.30) and Eqs. (2.33b) and (2.34b) into (2.30) to give

$$\frac{1}{2}a_{gu3}^{lmn} + \frac{1}{12}\sum_{g'=1}^G (\kappa_u^2)_{gg'}^{lmn} \left[a_{g'u1}^{lmn} + \frac{1}{10}a_{g'u3}^{lmn} \right] = -\frac{(h_u^l)^2}{D_g^{lmn}} S_{gu1}^{lmn}, \quad (2.39a)$$

$$\frac{1}{5}a_{gu4}^{lmn} + \frac{1}{20}\sum_{g'=1}^G (\kappa_u^2)_{gg'}^{lmn} \left[a_{g'u2}^{lmn} + \frac{1}{35}a_{g'u4}^{lmn} \right] = -\frac{(h_u^l)^2}{D_g^{lmn}} S_{gu2}^{lmn}. \quad (2.39b)$$

Before continuing, a discussion of the errors introduced by the quadratic leakage approximation, Eq. (2.35), and the choice of the fourth-order polynomial is pertinent. The error introduced by the quadratic leakage approximation is expected to be of the same order as a third-order polynomial approximation for the transversely-integrated flux since the leakages are related to the net currents which are in turn related to the spatial derivative of the flux. Hence, the leakages for a third order polynomial approximation have a quadratic form. Since the leakages are typically smaller than the net currents, errors in the leakages should be smaller and less significant than the errors in the net currents. Therefore, the use of a quartic polynomial approximation seems to be a reasonable combination with the quadratic transverse-leakage approximation.

Empirical evidence also supports this choice. A convergence analysis of a two-dimensional, homogenous, bare core problem performed by Zerkle [Z-1] shows that the spatial discretization errors in k_{eff} (i.e. λ) is $O(h^4)$ for both the cubic and quartic nodal methods. A subsequent one-dimensional analysis, however, indicates that in the absence of the transverse-leakage approximation the quartic approximation has a truncation error of $O(h^6)$ while the cubic polynomial error remains $O(h^4)$. The comparison of the one-dimensional to the two-dimensional results indicates that the

quadratic transverse-leakage approximation must introduce an error of $O(h^4)$ and, for this model problem, is more significant than the error introduced by the quartic polynomial approximation.

2.4.5 Expansion Coefficient Solution

The expansion for the two-node problem has eight unknown expansion coefficients per group. The complete set of equations for a given interface is:

1. A nodal balance equation for node $l - 1$:

$$6a_{gu2}^{l-1,mn} - \frac{2}{5}a_{gu4}^{l-1,mn} = \sum_{g'=1}^G (\kappa_u^2)_{gg'}^{l-1,mn} \bar{\phi}_{g'}^{l-1,mn} + \frac{(h_u^{l-1})^2}{D_g^{l-1,mn}} [\bar{S}_{gu}^{l-1,mn} + q_g^{l-1,mn}]. \quad (2.40)$$

This is an alternate form of the nodal balance represented by Eq. (2.3) and is obtained by using the weight function $w_0 = 1$ in Eq. (2.30).

2. The first moment equation for node $l - 1$:

$$\frac{1}{2}a_{gu3}^{l-1,mn} + \frac{1}{12} \sum_{g'=1}^G (\kappa_u^2)_{gg'}^{l-1,mn} \left[a_{g'u1}^{l-1,mn} + \frac{1}{10}a_{g'u3}^{l-1,mn} \right] = -\frac{(h_u^{l-1})^2}{D_g^{l-1,mn}} \bar{S}_{gu1}^{l-1,mn}. \quad (2.41)$$

3. The second moment equation for node $l - 1$:

$$\frac{1}{5}a_{gu4}^{l-1,mn} + \frac{1}{20} \sum_{g'=1}^G (\kappa_u^2)_{gg'}^{l-1,mn} \left[a_{g'u2}^{l-1,mn} + \frac{1}{10}a_{g'u4}^{l-1,mn} \right] = -\frac{(h_u^{l-1})^2}{D_g^{l-1,mn}} \bar{S}_{gu2}^{l-1,mn}. \quad (2.42)$$

4. A nodal balance equation for node l :

$$6a_{gu2}^{lmn} - \frac{2}{5}a_{gu4}^{lmn} = \sum_{g'=1}^G (\kappa_u^2)_{gg'}^{lmn} \bar{\phi}_{g'}^{lmn} + \frac{(h_u^l)^2}{D_g^{lmn}} [\bar{S}_{gu}^{lmn} + q_g^{lmn}]. \quad (2.43)$$

5. The first moment equation for node l :

$$\frac{1}{2}a_{gu3}^{lmn} + \frac{1}{12} \sum_{g'=1}^G (\kappa_u^2)_{gg'}^{lmn} \left[a_{g'u1}^{lmn} + \frac{1}{10}a_{g'u3}^{lmn} \right] = -\frac{(h_u^l)^2}{D_g^{lmn}} \bar{S}_{gu1}^{lmn}. \quad (2.44)$$

6. The second moment equation for node l :

$$\frac{1}{5}a_{gu4}^{lmn} + \frac{1}{20} \sum_{g'}^G (\kappa_u^2)_{gg'}^{lmn} \left[a_{g'u2}^{lmn} + \frac{1}{10} a_{g'u4}^{lmn} \right] = -\frac{(h_u^l)^2}{D_g^{lmn}} \bar{S}_{gu2}^{lmn}. \quad (2.45)$$

7. Discontinuity in flux at the node interface:

$$\left(\bar{\phi}_g^{l-1,mn} + \frac{1}{2} a_{gu1}^{l-1,mn} \frac{1}{2} a_{gu2}^{l-1,mn} \right) f_{gu+}^{l-1,mn} = \left(\bar{\phi}_g^{lmn} - \frac{1}{2} a_{gu1}^{lmn} + \frac{1}{2} a_{gu2}^{lmn} \right) f_{gu-}^{lmn}. \quad (2.46)$$

Note that the discontinuity factors which occur in this equation, $f_{gu+}^{l-1,mn}$ and f_{gu-}^{lmn} , are different from those introduced into the finite difference equations, $f_{gu+}^{l-1,mn}$ and f_{gu-}^{lmn} . As for the finite difference equations, these discontinuity factors can be used to correct for spatial, homogenization and diffusion theory errors. Since the quartic polynomial approximation leads to small spatial errors, their primary purpose is to provide homogenization correction. Generally, $f_{gu+}^{l-1,mn}$ and f_{gu-}^{lmn} will be constant throughout the entire calculation.

8. Continuity of the net current at the node interface:

$$-\frac{D_g^{l-1,mn}}{h_u^{l-1}} \left(a_{gu1}^{l-1,mn} + 3a_{gu2}^{l-1,mn} - \frac{1}{2} a_{gu3}^{l-1,mn} - \frac{1}{5} a_{gu4}^{l-1,mn} \right) = -\frac{D_g^{lmn}}{h_u^l} \left(a_{gu1}^{lmn} - 3a_{gu2}^{lmn} - \frac{1}{2} a_{gu3}^{l-1,mn} + \frac{1}{5} a_{gu4}^{l-1,mn} \right). \quad (2.47)$$

These equations represent a $8G \times 8G$ coupled system of equations which would be very time consuming to solve, especially for a large number of energy groups. With further manipulation, however, we can simplify the solution procedure. Note that the nodal balance equation and the second moment equation for each node involve only the even expansion coefficients and are not coupled to the other node. Thus, by solving the nodal balance equation, Eq. (2.43), for a_{gu2}^{lmn} and substituting into the second moment equation, Eq. (2.45), we can obtain a_{gu2}^{lmn} and a_{gu4}^{lmn} with one $G \times G$

solution. Next, the continuity conditions, Eqs. (2.46) and (2.47), can be used to obtain $a_{gu1}^{l-1,mn}$ and a_{gu1}^{lmn} in terms of $a_{gu3}^{l-1,mn}$ and a_{gu3}^{lmn} . These expressions can then be substituted into the first moment equations, Eqs. (2.41) and (2.44), to obtain a $2G \times 2G$ equation for $a_{gu3}^{l-1,mn}$ and a_{gu3}^{lmn} . Thus, the $8G \times 8G$ problem can be reduced to one $G \times G$ and one $2G \times 2G$ problems per interface.

This solution method differs from that used by Zerkle [Z-1]. In Zerkle's method, the a_{gu1}^{lmn} and a_{gu2}^{lmn} expansion coefficients are written in terms of the surface-averaged fluxes using Eqs. (2.29b) and (2.29c) and surface-averaged currents using (2.29d) and (2.29e). The continuity conditions are then applied resulting in an expression for the net current at the interface. This equation also involves the leakages for the two neighboring nodes which are eliminated using the nodal balance equation. The nodal method based on this solution method has been found to require more iterations than the method developed here since it requires the use of more information from the previous iteration.

The coupling relations for the polynomial nodal method are simply the surface-averaged net-current expressions given in Eqs. (2.29d) and (2.29e). Because of their complicated nature, they cannot be easily combined with the nodal balance equation to obtain a single nodal equation, as in the corrected finite-difference method. Further, since the expansion coefficients depend upon the node-averaged fluxes, the system of equations which must be solved is quite large, including the nodal balance equation, the polynomial coupling equations and the expansion coefficient equations. The solution procedure is simplified by the use of an iterative procedure.

2.4.6 Boundary Conditions

The boundary conditions applied to the polynomial equations are a generalization of those applied to the finite-difference equations, Eq.(2.18), and are given by

$$\phi_{gu}^{mn}(u_s) = \hat{l} \cdot \mathbf{n} \sum_{g'=1}^G \Gamma_{gg',u\pm}^{lmn} J_{g'u}^{lmn}(u_s). \quad (2.48)$$

Substituting the polynomial expansion approximations of the surface-averaged flux and current results in following equation at the lower surface

$$f_{gu-}^{lmn} \left[\bar{\phi}_g^{lmn} - \frac{1}{2} a_{gu1}^{lmn} + \frac{1}{2} \right] = - \sum_{g'=1}^G \Gamma_{gg',u-}^{mn} \frac{D_g^{lmn}}{h_u^l} \left[a_{gu1}^{lmn} - 3a_{gu2}^{lmn} - \frac{1}{2} a_{gu3}^{lmn} + \frac{1}{5} a_{gu4}^{lmn} \right], \quad (2.49)$$

and at the upper surface

$$f_{gu+}^{lmn} \left[\bar{\phi}_g^{lmn} + \frac{1}{2} a_{gu1}^{lmn} + \frac{1}{2} \right] = - \sum_{g'=1}^G \Gamma_{gg',u+}^{mn} \frac{D_g^{lmn}}{h_u^l} \left[a_{gu1}^{lmn} + 3a_{gu2}^{lmn} - \frac{1}{2} a_{gu3}^{lmn} - \frac{1}{5} a_{gu4}^{lmn} \right]. \quad (2.50)$$

These equations are combined with the other expansion coefficient equations given in Section 2.4.5 to obtain the complete set of equations for nodes at the boundaries.

Note that if the off-diagonal elements of $\Gamma_{gg',u\pm}^{mn}$ are zero, we obtain the same form as in Eq. (2.18). This general form of boundary condition can be used to model reflectors leading to a significant reduction in the number of unknowns. For example, an infinite reflector in two energy groups is obtained by using the following values [P-1]

$$\begin{aligned} \Gamma_{11,u}^{mn} &= L_1/D_1 \\ \Gamma_{12,u}^{mn} &= 0 \\ \Gamma_{21,u}^{mn} &= (L_1 - L_2) \Sigma_{21} / \left(D_1 D_2 \left[\frac{1}{L_2^2} - \frac{1}{L_1^2} \right] \right) \\ \Gamma_{22,u}^{mn} &= L_2/D_2 \end{aligned}$$

where D_g is the group g reflector diffusion coefficient and L_g is the group g reflector diffusion length ($\sqrt{D_g/\Sigma_g}$).

2.5 The Non-Linear Iteration Procedure

Rather than solving all of the polynomial nodal equations as a complete set, we can use a non-linear iteration procedure which was first proposed by Smith [S-4]. The method takes advantage of the property of the corrected finite-difference scheme in which any reference solution can be exactly reproduced if appropriate discontinuity factors are provided. A flow diagram of the non-linear iteration scheme is presented in Figure 2-3. Beginning with an initial guess for the discontinuity factor ratios, the

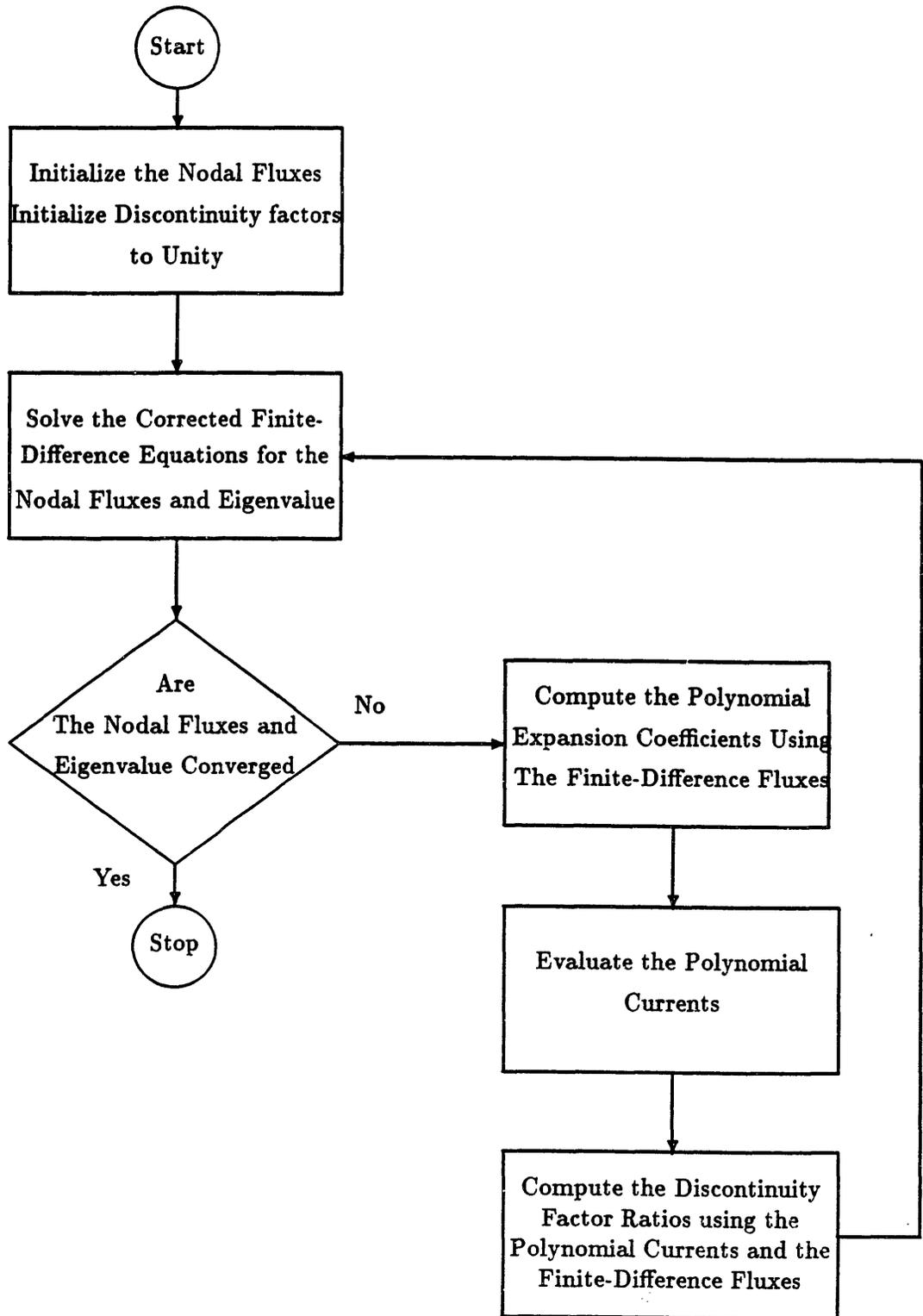


Figure 2-3: A flow diagram of the non-linear iteration procedure for the static problem.

corrected finite difference relations can be used to compute the node-averaged fluxes. These fluxes are then used in the polynomial equations to compute the expansion coefficients and, hence the net currents at the interface. From these polynomial net currents and the corrected finite-difference fluxes, discontinuity factor ratios may be computed. These discontinuity factors are used in the corrected finite-difference equations and the process is repeated until the node-averaged fluxes converge. The uniqueness of the solution of the nodal equations guarantees that, if this method converges, it will converge to the solution of the polynomial nodal equations.

This method has several advantages over solving all of the polynomial nodal equations together. First, a significant reduction in storage is obtained since none of the expansion coefficients and polynomial coupling terms are needed during the corrected finite-difference calculation. Also, since the finite-difference equations involve only the nodal fluxes, no iterations are necessary for the currents and expansion coefficients during the global flux solution procedure. Finally, if the solution can be obtained in a relatively small number of discontinuity factor iterations, a reduction in computation time will result because of the decrease in the number of times the currents must be calculated.

2.6 Summary

In this chapter the derivation of the static nodal method was presented. Starting with the space-dependent, few-group diffusion equations we obtained an equation which represented a neutron balance over an arbitrary node. Since this equation related several unknown quantities, additional equations were required. One method for obtaining the additional relations is a polynomial nodal method. However, since the set of global equations resulting from the application of this polynomial method is quite complicated, a non-linear iteration procedure was introduced in which the polynomial method is used locally to obtain correction factors for the finite-difference-like equations. The global calculation of the node-averaged fluxes is then performed by the solution of these corrected finite-difference equations.

In Chapter 3 the polynomial nodal method will be extended to time dependent

problems and a time-integration method will be presented. Numerical methods and considerations for the solution of the static and transient nodal equations are given in Chapter 4. Applications of the static and transient methods are presented in Chapter 5.

Chapter 3

DERIVATION OF THE TRANSIENT NODAL EQUATIONS

3.1 Introduction

In Chapter 2 the details of the static nodal method were outlined. A similar procedure will be presented in this chapter for the transient method. First, the corrected finite-difference formulation will be presented. Applying the finite-difference approximation to the time-dependent neutron diffusion equation and the introduction of discontinuity factors results in a system of first-order ordinary differential equations. A temporal differencing scheme will be presented to advance the solution in time. The polynomial method will be applied to give equations for the expansion coefficients and polynomial currents to compute the discontinuity factor ratios. These discontinuity factor ratios are generally time dependent and will be updated using the same non-linear iteration procedure applied for the static solution.

This chapter also presents the application of the quasi-static method. In this method we assume that the node-averaged fluxes are the product of a space and energy dependent shape function and a space and energy independent amplitude function. Since the shape function generally varies slowly in time, large time steps can be employed in its solution and it can be computed less frequently than the amplitude function.

3.2 Notation

The derivation begins with the time-dependent, few-group diffusion equations in P_1 form

$$\begin{aligned} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\mathbf{r}, t) &= -\nabla \cdot \mathbf{J}_g(\mathbf{r}, t) - \Sigma_{tg}(\mathbf{r}, t) \phi_g(\mathbf{r}, t) \\ &+ \sum_{g'=1}^G \left[(1 - \beta) \chi_{pg} \nu \Sigma_{fg'}(\mathbf{r}, t) + \Sigma_{gg'}(\mathbf{r}, t) \right] \phi_{g'}(\mathbf{r}, t) \\ &+ \sum_{d=1}^D \chi_{dg} \lambda_d c_d(\mathbf{r}, t) + q_g(\mathbf{r}, t), \quad g = 1, 2, \dots, G, \end{aligned} \quad (3.1a)$$

$$\mathbf{J}_g(\mathbf{r}, t) = -D_g(\mathbf{r}, t) \nabla \phi_g(\mathbf{r}, t), \quad (3.1b)$$

$$\frac{\partial}{\partial t} c_d(\mathbf{r}, t) = \beta_d \sum_{g'=1}^G \nu \Sigma_{fg'}(\mathbf{r}, t) \phi_{g'}(\mathbf{r}, t) - \lambda_d c_d(\mathbf{r}, t), \quad (3.1c)$$

$$d = 1, 2, \dots, D.$$

Where in addition to the terms defined for the static equations in Chapter 2,

- c_d = density of delayed neutron precursors in family d (cm^{-3}),
- χ_{pg} = prompt fission spectrum for group g ,
- χ_{dg} = fission spectrum for precursor family d , group g ,
- λ_d = decay constant for delayed neutron precursor family d (s^{-1}),
- β_d = fractional yield of delayed neutrons in family d ,
- β = total fractional yield of delayed neutrons, $\beta = \sum_{d=1}^D \beta_d$,
- v_g = neutron speed for group g (cm s^{-1}),
- G = total number of energy groups,
- D = total number of delayed precursor families.

In a manner similar to the derivation of the static nodal balance equations, we obtain the time-dependent nodal balance equations by integrating Eqs. (3.1a) and

(3.1c) over an arbitrary node (i, j, k) and dividing by the nodal volume to obtain

$$\begin{aligned} \frac{1}{v_g} \frac{\partial}{\partial t} \bar{\phi}_g^{ijk}(t) = & -\frac{1}{h_x^i} [J_{gx}^{jk}(x_{i+1}, t) - J_{gx}^{jk}(x_i, t)] - \frac{1}{h_y^i} [J_{gy}^{ik}(y_{j+1}, t) - J_{gy}^{ik}(y_j, t)] \\ & -\frac{1}{h_z^i} [J_{gz}^{ij}(z_{k+1}, t) - J_{gz}^{ij}(z_k, t)] - \Sigma_{tg}^{ijk}(t) \bar{\phi}_g^{ijk}(t) \\ & + \sum_{g'=1}^G [\chi_{g'} \nu \Sigma_{fg'}^{ijk}(t) + \Sigma_{gg'}^{ijk}(t)] \bar{\phi}_{g'}^{ijk}(t) + \sum_{d=1}^D \chi_{dg} \lambda_d \bar{c}_d^{ijk}(t) + \bar{q}_g^{ijk}(t), \end{aligned} \quad (3.2a)$$

$$\frac{\partial}{\partial t} \bar{c}_d^{ijk}(t) = \beta_d \sum_{g'=1}^G \nu \Sigma_{fg'}^{ijk}(t) \bar{\phi}_{g'}^{ijk}(t) - \lambda_d \bar{c}_d^{ijk}(t). \quad (3.2b)$$

As in the static case, the fluxes and cross sections represent averages over the node. In addition, the node-averaged precursor concentration is defined by

$$\bar{c}_d^{ijk} \equiv \frac{1}{V^{ijk}} \int_{x_i}^{x_{i+1}} dx \int_{y_j}^{y_{j+1}} dy \int_{z_k}^{z_{k+1}} dz c_d(x, y, z). \quad (3.3)$$

Since Eq. (3.2a) relates several unknown quantities (the node-averaged fluxes and face-averaged currents) additional coupling equations are required.

3.3 The Time-Dependent, Corrected Finite-Difference Equations

The derivation of the time-dependent finite-difference coupling equations is exactly the same as for the static case since no time derivatives appear in the second P_1 equation, Eq. (3.1b). Applying the finite-difference approximation and introducing discontinuity factors results in the following expression for the surface-averaged current at the node interface u_l

$$\begin{aligned} J_{gu}^{mn}(u_l, t) = & - \left[\frac{h_u^l}{2D_g^{lmn}(t)} \frac{f_{gu-}^{lmn}(t)}{f_{gu+}^{l-1, mn}(t)} + \frac{h_u^{l-1}}{2D_g^{l-1, mn}(t)} \right]^{-1} \\ & \times \left(\frac{f_{gu-}^{lmn}(t)}{f_{gu+}^{l-1, mn}(t)} \bar{\phi}_g^{lmn}(t) - \bar{\phi}_g^{l-1, mn}(t) \right). \end{aligned} \quad (3.4)$$

Note that the discontinuity factor ratios are now time varying. A similar expression for the surface-averaged current can be derived for the u_{l+1} interface. These expressions can be substituted into the time-dependent nodal balance equation and written in matrix form as

$$\begin{aligned} \mathbf{V}_g^{-1} \frac{d}{dt} \phi_g(t) = & -\mathbf{N}_g(t) \phi_g(t) + \sum_{g'=1}^G (1-\beta) \mathbf{F}_{p,gg'}(t) \phi_{g'}(t) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'}(t) \phi_{g'}(t) \\ & + \sum_{d=1}^D \lambda_d \mathbf{c}_{dg}(t) + \mathbf{q}(t), \quad g = 1, 2, \dots, G \end{aligned} \quad (3.5a)$$

$$\frac{d}{dt} \mathbf{c}_{dg}(t) = \beta_d \sum_{g'=1}^G \mathbf{F}_{d,gg'}(t) \phi_{g'}(t) - \lambda_d \mathbf{c}_{dg}(t), \quad d = 1, 2, \dots, D. \quad (3.5b)$$

In addition to the vectors and matrices defined for the static corrected finite-difference equations we now define

$$\begin{aligned} \mathbf{V}_g &= \text{An } N \times N \text{ diagonal matrix of the group speeds, } \{v_g\}, \\ \mathbf{F}_{p,gg'} &= \text{An } N \times N \text{ diagonal matrix of } \{\chi_{pg} \nu \Sigma_{f_g}^{ijk}\}, \\ \mathbf{c}_{dg} &= \text{A column vector of length } N \text{ of } \{\chi_{dg} c_d^{ijk}\}, \\ \mathbf{F}_{d,gg'} &= \text{An } N \times N \text{ diagonal matrix of } \{\chi_{dg} \nu \Sigma_{f_g}^{ijk}\}. \end{aligned}$$

These equations can be written in super-matrix form as

$$\mathbf{V}^{-1} \frac{d}{dt} \Phi(t) = [\mathbf{M}_p(t) - \mathbf{L}(t)] \Phi(t) + \sum_{d=1}^D \lambda_d \mathbf{c}_d(t) + \mathbf{q}(t). \quad (3.6a)$$

$$\frac{d}{dt} \mathbf{c}_d(t) = \mathbf{M}_d(t) \Phi(t) - \lambda_d \mathbf{c}_d(t), \quad d = 1, 2, \dots, D. \quad (3.6b)$$

Where in addition to the terms defined for the static expression,

$$\begin{aligned} \mathbf{V} &= \text{An } NG \times NG \text{ diagonal matrix of the group speeds, } \{\mathbf{V}_g\}, \\ \mathbf{L} &= \text{An } NG \times NG \text{ matrix of } \{\mathbf{N}_g \delta_{gg'} - \Sigma_{gg'}(t)\}, \\ \mathbf{M}_p &= \text{An } NG \times NG \text{ matrix of prompt fission production terms, } \{\mathbf{F}_{p,gg'}\}, \\ \mathbf{M}_d &= \text{An } NG \times NG \text{ matrix of delayed fission production terms, } \{\mathbf{F}_{d,gg'}\}, \\ \mathbf{c}_d &= \text{A column vector of length } NG \text{ of the spectrum-weighted} \\ &\quad \text{precursor densities, } \{\mathbf{c}_{dg}\}. \end{aligned}$$

Equations (3.6a) and (3.6b) represent a system of ordinary differential equations which is a semi-discrete representation of the neutron diffusion equation.

3.4 The Time-Dependent Polynomial Nodal Equations

When the polynomial nodal method is applied to the transient diffusion equations, the expansion coefficients become time dependent. The same procedure is used to determine the expansion coefficients as in the static case. First, the transverse integration procedure is used to obtain one-dimensional equations. Integrating equation (3.1a) in directions v and w results in the following one-dimensional equation for the u -direction for node (l, m, n)

$$\begin{aligned} \frac{\partial}{\partial t} \phi_{gu}^{mn}(u, t) = & - \frac{\partial}{\partial u} J_{gu}^{mn}(u, t) - \Sigma_{tg}^{lmn}(t) \phi_{gu}^{mn}(u, t) \\ & + \sum_{g'=1}^G \left[(1 - \beta) \chi_{pg} \nu \Sigma_{fg'}^{lmn}(t) + \Sigma_{gg'}^{lmn}(t) \right] \phi_{g'u}^{mn}(u, t) \\ & + \sum_{d=1}^D \lambda_d \chi_{dg} c_{du}^{mn}(u, t) + q_g(t), \quad g = 1, 2, \dots, G, \end{aligned} \quad (3.7a)$$

$$\frac{\partial}{\partial t} c_{du}^{mn}(u, t) = \beta_d \sum_{g'=1}^G \nu \Sigma_{fg'}^{lmn}(t) \phi_{g'u}^{mn}(u, t) - \lambda_d c_{du}^{mn}(u, t), \quad (3.7b)$$

$$d = 1, 2, \dots, D.$$

The time derivatives in these equations present a difficulty. They require the equations to be differenced in time and past values of the expansion coefficients to be saved from one time step to the next. Also, the transverse-integration procedure has lead us to equations for the precursor densities which vary spatially in the u -direction. Simplification is obtained if we introduce dynamic frequencies defined by the following expressions

$$\omega_g^{lmn}(t) \equiv \frac{1}{\phi_{gu}^{mn}(u, t)} \frac{\partial}{\partial t} \phi_{gu}^{mn}(u, t), \quad (3.8)$$

and

$$\omega_d^{lmn}(t) \equiv \frac{1}{c_{du}^{mn}(u,t)} \frac{\partial}{\partial t} c_{du}^{mn}(u,t). \quad (3.9)$$

Note that within a given node these frequencies vary neither spatially nor directionally. A method of estimating these frequencies is given in Chapter 4. The introduction of the “omega” into the precursor equation allows us to solve for the transversely-integrated precursor densities in terms of the transversely-integrated fluxes

$$c_{du}^{mn}(u,t) = \frac{\beta_d}{(\omega_d^{lmn}(t) + \lambda_d)} \sum_{g'=1}^G \nu \Sigma_{fg'}^{lmn}(t) \phi_{g'u}^{mn}(u,t), \quad u \in [u_l, u_{u+l}]. \quad (3.10)$$

This equation can now be substituted into the transversely-integrated flux equation to obtain

$$\begin{aligned} \frac{\partial}{\partial u} J_{gu}^{mn}(u,t) + \bar{\Sigma}_{tg}^{lmn}(t) \phi_{gu}^{mn}(u,t) &= \sum_{g'=1}^G \left[\bar{\chi}_g^{lmn}(t) \nu \Sigma_{fg'}^{lmn}(t) + \Sigma_{gg'}^{lmn}(t) \right] \phi_{g'u}^{mn}(u,t) \\ &+ \sum_{d=1}^D \lambda_d \chi_{dg} c_{du}^{mn}(u,t) - S_{gu}^{mn}(u,t) + q_g(t), \quad g = 1, 2, \dots, G, \end{aligned} \quad (3.11)$$

where we have introduced an effective total cross section, $\bar{\Sigma}_{tg}^{lmn}(t)$, and an effective fission spectrum, $\bar{\chi}_g^{lmn}(t)$, defined by

$$\begin{aligned} \bar{\Sigma}_{tg}^{lmn}(t) &\equiv \Sigma_{tg}^{lmn}(t) + \frac{\omega_g^{lmn}(t)}{v_g}, \\ \bar{\chi}_g^{lmn}(t) &\equiv (1 - \beta) \chi_{pg} + \sum_{d=1}^D \frac{\lambda_d \beta_d \chi_{dg}}{\omega_d^{lmn}(t) + \lambda_d}. \end{aligned}$$

As a result of introducing the dynamic frequencies we now have an equation which closely resembles the transversely-integrated equation for the static case, Eq. (2.24a). Therefore, the static equations for the expansion coefficient may be applied with these modified terms and with the cross sections being functions of time. In addition, the non-linear iterative update of the discontinuity factor ratios remains unchanged. Once the expansion coefficients are obtained, the polynomial current expressions, Eqs. (2.29d) and (2.29e), are applied and the discontinuity factors are computed. This

procedure gives the time-dependent discontinuity factors required for the corrected finite-difference equations.

3.5 Time-Integration of the Corrected Finite-Difference Equation

The derivations of the previous sections have resulted in spatially discretized, time-dependent ordinary differential equations. In addition, the introduction of dynamic frequency terms into the transversely-integrated equations has eliminated the time derivatives from the polynomial nodal equations. The only remaining time derivatives appear in the corrected finite-difference equations. The method that will be used to solve the time-dependent, corrected finite-difference equations will be direct integration of the precursor equation and theta-differencing of the flux equation.

The time domain is represented by discrete points at which we desire the solution

$$t = t_0, t_1, t_2, \dots$$

and the time intervals are defined as

$$\Delta t_n \equiv t_{n+1} - t_n.$$

First, consider the precursor equation, Eq. (3.6b). We can directly integrate this equation from t_n to t_{n+1} to obtain

$$\mathbf{c}_d^{(n+1)} = e^{-\lambda_d \Delta t_n} \mathbf{c}_d^{(n)} + e^{-\lambda_d \Delta t_n} \int_{t_n}^{t_{n+1}} e^{\lambda_d(t-t_n)} \mathbf{M}_d(t) \Phi(t) dt, \quad (3.12)$$

$$d = 1, 2, \dots, D.$$

The superscript represents the time at which the quantities are evaluated, for example

$$\mathbf{c}_d^{(n+1)} = \mathbf{c}_d(t_{n+1}).$$

Now we assume that $\mathbf{M}_d(t) \Phi(t)$ varies in a linear fashion across the time step. With

this assumption the integral in Eq. (3.12) can be evaluated to give

$$\mathbf{c}_d^{(n+1)} = k_{1,d} \mathbf{c}_d^{(n)} + k_{2,d} \frac{\beta_d}{\lambda_d} \mathbf{M}_d^{(n+1)} \Phi^{(n+1)} - k_{3,d} \frac{\beta_d}{\lambda_d} \mathbf{M}_d^{(n)} \Phi^{(n)}, \quad (3.13)$$

where

$$\begin{aligned} k_{1,d} &\equiv e^{-\lambda_d \Delta t_n}, \\ k_{2,d} &\equiv 1 - \frac{1 - e^{-\lambda_d \Delta t_n}}{\lambda_d \Delta t_n}, \\ k_{3,d} &\equiv e^{-\lambda_d \Delta t_n} - \frac{1 - e^{-\lambda_d \Delta t_n}}{\lambda_d \Delta t_n}. \end{aligned}$$

If the entire integrand of Eq. (3.12) were assumed to vary linearly, we would simply be approximating the integral by the trapezoidal rule, which has second-order global truncation error. The direct integration procedure is more accurate because of the inclusion of the exponential in the integrand. The truncation error, however, is still second order. Note that in computer applications $k_{1,d}$, $k_{2,d}$, and $k_{3,d}$ should be evaluated using Taylor series expansions to avoid round-off errors for small Δt_n .

Now we are prepared to difference the flux equation. The theta method [V-1] mixes the old and new values on the right-hand side of Eq. (3.6a) resulting in

$$\begin{aligned} \mathbf{V}^{-1} \frac{\Phi^{(n+1)} - \Phi^{(n)}}{\Delta t_n} &= \theta \left\{ \left[\mathbf{M}_p^{(n+1)} - \mathbf{L}^{(n+1)} \right] \Phi^{(n+1)} + \sum_{d=1}^D \lambda_d \mathbf{c}_d^{(n+1)} + \mathbf{q}^{(n+1)} \right\} \\ &+ (1 - \theta) \left\{ \left[\mathbf{M}_p^{(n)} - \mathbf{L}^{(n)} \right] \Phi^{(n)} + \sum_{d=1}^D \lambda_d \mathbf{c}_d^{(n)} + \mathbf{q}^{(n)} \right\}. \end{aligned} \quad (3.14)$$

The value of θ can be chosen to give the standard time-differencing schemes:

$$\begin{aligned} \theta = 0 & \quad \text{Forward Difference (or Fully Explicit),} \\ \theta = \frac{1}{2} & \quad \text{Trapezoidal Rule (or Crank-Nicholson),} \\ \theta = 1 & \quad \text{Backward Difference (or Fully Implicit).} \end{aligned}$$

More consideration of the choice of θ will be given in Chapter 4.

The value of the precursor densities at the new time step appears in Eq.(3.14). This term may be eliminated by using the result of the direct integration of the precursor equation. Thus, substituting Eq. (3.13) into (3.14) and subsequent rearranging,

we obtain the following equation for the fluxes

$$\begin{aligned} \left\{ \frac{1}{\Delta t_n} \mathbf{V}^{-1} + \theta \left[\mathbf{L}^{(n+1)} - \mathbf{M}_p^{(n+1)} - \sum_{d=1}^D k_{2,d} \mathbf{M}_d^{(n+1)} \right] \right\} \Phi^{(n+1)} = \\ \left\{ \frac{1}{\Delta t_n} \mathbf{V}^{-1} + (1 - \theta) \left(\mathbf{M}_p^{(n)} - \mathbf{L}^{(n)} \right) - \theta \sum_{d=1}^D k_{3,d} \mathbf{M}_d^{(n)} \right\} \Phi^{(n)} \quad (3.15) \\ + \sum_{d=1}^D \lambda_d [1 - \theta (1 - k_{1,d})] \mathbf{c}_d^{(n)} + \theta \mathbf{q}^{(n+1)} + (1 - \theta) \mathbf{q}^{(n)}. \end{aligned}$$

With the appropriate definitions we can write this equation in the following supermatrix form

$$\mathbf{A}^{(n+1)} \Phi^{(n+1)} = \mathbf{s}^{(n)}. \quad (3.16)$$

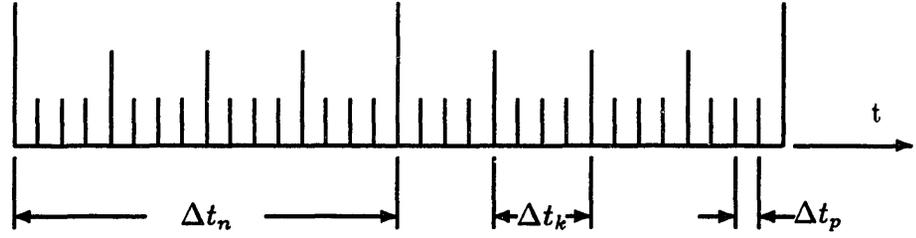
Equation (3.16) shows that for each time step a matrix inversion must be performed to obtain the node-averaged fluxes at the new time step. For large problems, iterative methods will be required. The solution procedure is discussed in Chapter 4.

3.6 The Quasi-Static Method

The temporal differencing scheme outlined in the previous section gives accurate results only when the time step, Δt_n , is small. Since each time step requires a matrix inversion, this leads to substantial computational effort. One method for reducing the amount of work required is to factor the flux into a shape function and an amplitude function, as in the derivation of the point kinetics equations [H-4]. The shape function is defined for each node and energy-group, while the amplitude function is a space and energy independent quantity. It has been empirically observed that the shape function varies more slowly than the amplitude function. Thus, the shape calculation can be performed less frequently than the amplitude function calculation.

The subdivision of the time steps for the quasi-static method is presented in Figure 3-1. The shape function is computed using the largest time step, Δt_n , while the point kinetics parameters are updated with a smaller time step, Δt_k , to reflect changes in reactor compositions and conditions. The finest time division, Δt_p , is used to obtain an accurate solution of the amplitude function equations.

Quasi-Static Method Time Steps



Δt_n = Shape Function Time Step
 Δt_k = Reactivity Calculation Time Step
 Δt_p = Amplitude Function Time Step

Figure 3-1: Diagram showing the subdivision of the time steps in the quasi-static method.

3.6.1 The Amplitude Function Equation

In order to apply the quasi-static method, an equation must be derived for the shape function and the amplitude function. We begin by factoring the node-averaged fluxes

$$\Phi(t) = \mathbf{S}(t)\mathbf{T}(t), \quad (3.17)$$

where the amplitude function is defined by

$$\mathbf{T}(t) \equiv \mathbf{w}^T \mathbf{V}^{-1} \Phi(t). \quad (3.18)$$

Note that an arbitrary, time-independent weight function, \mathbf{w} , has been introduced. We can now substitute the factorization (3.17) into the time-dependent nodal equation, Eq. (3.6a), and multiply by the weight function to obtain

$$\begin{aligned}
 \mathbf{w}^T \mathbf{V}^{-1} \frac{d}{dt} \mathbf{S}(t)\mathbf{T}(t) &= \mathbf{w}^T [\mathbf{M}_p(t) - \mathbf{L}(t)] \mathbf{S}(t)\mathbf{T}(t) \\
 &+ \sum_{d=1}^D \lambda_d \mathbf{w}^T \mathbf{c}_d(t) + \mathbf{w}^T \mathbf{q}(t).
 \end{aligned} \quad (3.19)$$

In the derivation of the equation for the amplitude function we shall make use of the following fact

$$T(t) = \mathbf{w}^T \mathbf{V}^{-1} \Phi(t) = (\mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t)) T(t),$$

$$\therefore \mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t) = 1. \quad (3.20)$$

Thus, the shape function maintains a constant normalization. As a result of this normalization, the time derivative term can be written as

$$\mathbf{w}^T \mathbf{V}^{-1} \frac{d}{dt} \mathbf{S}(t) T(t) = \frac{d}{dt} (\mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t)) T(t) = (\mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t)) \frac{d}{dt} T(t). \quad (3.21)$$

The amplitude function equation is then obtained by adding and subtracting the term $\mathbf{w}^T (\sum_{d=1}^D \mathbf{M}_d(t)) \mathbf{S}(t)$ from the right hand side of Eq. (3.19) and by dividing by $\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)$ where $\mathbf{M}(t)$ is the total fission operator

$$\mathbf{M}(t) \equiv \mathbf{M}_p(t) + \sum_{d=1}^D \mathbf{M}_d(t).$$

The resulting equation for the amplitude function is

$$\begin{aligned} \frac{\mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)} \frac{d}{dt} T(t) &= \frac{\mathbf{w}^T [\mathbf{M}(t) - \mathbf{L}(t)] \mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)} T(t) - \frac{\mathbf{w}^T (\sum_{d=1}^D \mathbf{M}_d(t)) \mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)} T(t) \\ &+ \sum_{d=1}^D \lambda_d \frac{\mathbf{w}^T \mathbf{c}_d(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)} + \frac{\mathbf{w}^T \mathbf{q}(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)}. \end{aligned} \quad (3.22)$$

With some manipulation, this equation can be written as

$$\frac{d}{dt} T(t) = \frac{\rho(t) - \beta(t)}{\Lambda(t)} T(t) + \sum_{d=1}^D \lambda_d C_d(t) + q(t), \quad (3.23)$$

where

$$\rho(t) \equiv \frac{\mathbf{w}^T [\mathbf{M}(t) - \mathbf{L}(t)] \mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)}, \quad (3.24a)$$

$$\beta_d(t) \equiv \frac{\mathbf{w}^T \beta_d \mathbf{M}_d(t) \mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)}, \quad \beta(t) \equiv \sum_{d=1}^D \beta_d(t) \quad (3.24b)$$

$$\Lambda(t) \equiv \frac{\mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}(t) \mathbf{S}(t)}, \quad (3.24c)$$

$$q(t) \equiv \frac{\mathbf{w}^T \mathbf{q}(t)}{\mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t)}, \quad (3.24d)$$

$$C_d(t) \equiv \frac{\mathbf{w}^T \mathbf{c}_d(t)}{\mathbf{w}^T \mathbf{V}^{-1} \mathbf{S}(t)}. \quad (3.24e)$$

Equation (3.23) is simply the point kinetics equation and Eqs. (3.24a), (3.24b) and (3.24c) are the reactivity, effective delayed neutron fractions and prompt neutron life time, respectively. We now have precise, mathematical definitions for all of the point kinetics parameters as functions of the variables and operators of the nodal model.

The precursor equation can be handled in a similar manner. Substituting the flux factorization, multiplying by an arbitrary weight function, and using the definitions above, we obtain

$$\frac{d}{dt} C_d(t) = \frac{\beta_d(t)}{\Lambda(t)} T(t) - \lambda_d C_d(t), \quad d = 1, 2, \dots, D. \quad (3.25)$$

3.6.2 Shape Function Equation

An equation for the shape function results from substitution of the factorization (3.17) into Eq. (3.6a)

$$\mathbf{V}^{-1} \frac{d}{dt} (\mathbf{S}(t) T(t)) = [\mathbf{M}(t) - \mathbf{L}(t)] \mathbf{S}(t) T(t) + \sum_{d=1}^D \lambda_d \mathbf{c}_d(t) + \mathbf{q}(t). \quad (3.26)$$

After expanding the time derivative and dividing by the amplitude function we obtain

$$\mathbf{V}^{-1} \frac{d}{dt} \mathbf{S}(t) = \left[\mathbf{M}(t) - \mathbf{L}(t) - \mathbf{V}^{-1} \frac{1}{T(t)} \frac{d}{dt} T(t) \right] \mathbf{S}(t) + \sum_{d=1}^D \lambda_d \frac{\mathbf{c}_d(t)}{T(t)} + \frac{\mathbf{q}(t)}{T(t)}. \quad (3.27)$$

Note that the amplitude function is still present in this equation. Thus, the calculation of the shape function at a given time requires that we know the amplitude function at that time.

Alternatively, we can solve the time dependent nodal equations, Eq. (3.6a) for node-averaged fluxes and apply the normalization, Eq. (3.20), to obtain the shape

function. The major advantage of using this latter procedure is that the amplitude function need not be known to obtain the shape function. Therefore, we can compute the final shape function and use linear interpolation to obtain intermediate shapes. Kao [K-1] has shown that this leads to a significant increase in the accuracy of the quasi-static method for large time steps.

The quasi-static method which we are applying is actually known as the *Improved Quasi-Static Method* [O-1, K-1]. The distinguishing feature of the improved quasi-static method is that the time derivative in the shape function equation is approximated by a time-differencing method and is not neglected as in the traditional quasi-static method. Ott and Meneley [O-1] have demonstrated that the quasi-static method is sufficiently accurate for fast reactors, but not for thermal reactors. They further showed that the improved quasi-static method provides good results for both types of reactors.

3.6.3 Choice of Weight Function

Recall that an arbitrary weight function was introduced into the amplitude function and the definitions of the point kinetics parameters. Although any weight function may be used, careful selection will lead to a weight function which yields more accurate reactivity values when the shape function is not exact. This becomes more apparent when we consider the perturbation formula for reactivity. Suppose that all the cross sections and the shape function are perturbed from their initial steady-state values

$$\begin{aligned}
 \mathbf{L}(t) &= \mathbf{L}_0 + \delta\mathbf{L}(t), \\
 \mathbf{M}(t) &= \mathbf{M}_0 + \delta\mathbf{M}(t), \\
 \mathbf{S}(t) &= \mathbf{S}_0 + \delta\mathbf{S}(t).
 \end{aligned}
 \tag{3.28}$$

Substituting these perturbations into the reactivity expression (3.24a) gives

$$\rho(t) = \frac{\mathbf{w}^T [\mathbf{M}_0 - \mathbf{L}_0] \mathbf{S}_0 + \mathbf{w}^T [\delta\mathbf{M}(t) - \delta\mathbf{L}(t)] \mathbf{S}_0 + \mathbf{w}^T [\mathbf{M}_0 - \mathbf{L}_0] \delta\mathbf{S}(t) + \mathbf{w}^T [\delta\mathbf{M}(t) - \delta\mathbf{L}(t)] \delta\mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}_0 \mathbf{S}_0 + \mathbf{w}^T \delta\mathbf{M}(t) \mathbf{S}_0 + \mathbf{w}^T \mathbf{M}_0 \delta\mathbf{S}(t) + \mathbf{w}^T \delta\mathbf{M}(t) \delta\mathbf{S}(t)}. \quad (3.29)$$

Since we assume that the reactor is initially at steady-state, the first term in the numerator of Eq. (3.29) is zero. Further, we can neglect the second order terms in the numerator and the denominator since they are small. Also, the second term in the denominator can be neglected in comparison with the first term in the denominator. The resulting perturbation formula for reactivity is

$$\rho(t) \approx \frac{\mathbf{w}^T [\delta\mathbf{M}(t) - \delta\mathbf{L}(t)] \mathbf{S}_0 + \mathbf{w}^T [\mathbf{L}_0 + \mathbf{M}_0] \delta\mathbf{S}(t)}{\mathbf{w}^T \mathbf{M}_0 \mathbf{S}_0 + \mathbf{w}^T \delta\mathbf{M}(t) \mathbf{S}_0} \quad (3.30)$$

From this equation we can see that, unless first order errors are to be incurred, the calculation of reactivity requires us to know the perturbation in the shape function, $\delta\mathbf{S}(t)$. From a perturbation theory point of view we would like to compute the reactivity using only the steady-state solution. Thus, we must choose a non-trivial weight function such that

$$\mathbf{w}^T [\mathbf{M}_0 - \mathbf{L}_0] \delta\mathbf{S}(t) = 0. \quad (3.31)$$

Transposing this equation gives

$$\delta\mathbf{S}^T(t) [\mathbf{M}_0 - \mathbf{L}_0]^T \mathbf{w} = 0. \quad (3.32)$$

Therefore, if we are to avoid first order errors in reactivity, we must choose the weight function such that

$$[\mathbf{M}_0 - \mathbf{L}_0]^T \mathbf{w} = 0. \quad (3.33)$$

Thus the desired weight function is simply the adjoint of the static nodal equations. Note, however, that the loss operator \mathbf{L}_0 contains the unperturbed discontinuity factors obtained from the forward calculation. The evaluation of $\delta\mathbf{L}(t)$ during a transient requires that we know both the perturbations in cross sections due to feedback or op-

erator actions and the changes in discontinuity factors resulting from changes in the flux shape. For standard applications of first order perturbation theory, these latter changes will not be known.

While this situation may represent a problem for perturbation theory, it is not serious for the quasi-static method, since the shape function is being updated periodically. In fact, since any weight function can be used if the exact shape function is known, using the adjoint which will make the first order variations in shape function vanish from the reactivity expression is not essential. Using the adjoint as a weight function is most beneficial when applied to transients involving instantaneous changes in cross sections.

3.7 Thermal-Hydraulic and Feedback Models

The realistic and accurate analysis of a nuclear reactor core requires the consideration of thermal-hydraulic feedback effects. Two different methods have been incorporated to allow flexibility in thermal-hydraulic modelling.

3.7.1 The WIGL Model

In the previous investigations of finite-difference and nodal methods with thermal-hydraulic feedback the WIGL [V-1] lumped heat capacity model was used to provide a reasonable representation of the core thermal-hydraulic behavior. As a result, many reactor problems with thermal-hydraulics based on the WIGL model have been analyzed [S-2, J-2]. Comparison of results for these problems, therefore, requires the implementation of this model.

The primary quantities of interest in the WIGL model are the average fuel temperature and the average coolant temperature within a node. The following equations for these temperatures are obtained by performing a time-dependent energy balance

over a node assuming that no boiling occurs

$$\rho_f V_f^{ijk} C_f \frac{d\bar{T}_f^{ijk}}{dt} = (1-r) (q''')^{ijk} V_f^{ijk} + V_c^{ijk} \left[\frac{1}{A_H U} + \frac{1}{A_H h_0} \left(\frac{W}{W_0} \right)^{.8} \right]^{-1} (\bar{T}_f^{ijk} - \bar{T}_c^{ijk}) \quad (3.34a)$$

$$V_c^{ijk} \left(\frac{\partial \rho_c H}{\partial \bar{T}_c} \right) \frac{d\bar{T}_c^{ijk}}{dt} = V_c^{ijk} \left[\frac{1}{A_H U} + \frac{1}{A_H h_0} \left(\frac{W}{W_0} \right)^{.8} \right]^{-1} (\bar{T}_f^{ijk} - \bar{T}_c^{ijk}) \quad (3.34b)$$

$$+ 2W_r^{ijk} C_c (T_b^{ijk} - \bar{T}_c^{ijk}) + r (q''')^{ijk} V_f^{ijk}$$

$$T_b^{ijk} = 2\bar{T}_c^{ij,k-1} - T_b^{ij,k-1}, \quad (3.34c)$$

where

- \bar{T}_f^{ijk} = average fuel temperature in node (i,j,k) (K)
- \bar{T}_c^{ijk} = average coolant temperature in node (i,j,k) (K)
- \bar{T}_b^{ijk} = inlet (bottom) coolant temperature of node (i,j,k) (K)
- t = time (s)
- ρ_f = fuel density (g/cm³)
- ρ_c = coolant density (g/cm³)
- C_f = specific heat of the fuel (erg/g K)
- C_c = specific heat of the coolant (erg/g K)
- r = fraction of fission power deposited directly into the coolant
- $(q''')^{ijk}$ = volumetric energy generation rate in node (i,j,k) (erg/cm³)
- V_c^{ijk} = volume of coolant in node (i,j,k) (cm³)
- V_f^{ijk} = volume of fuel in node (i,j,k) (cm³)
- A_H = total heat transfer area/coolant volume within a node (cm⁻¹)
- h_0 = convective heat transfer coef. at initial flow rate (erg/s cm² K)
- W_0 = initial total coolant mass flow rate (g/s)
- W = total coolant mass flow rate (g/s)
- W_r^{ijk} = coolant mass flow rate in node (i,j,k) (g/s)
- $\left(\frac{\partial \rho_c H}{\partial \bar{T}_c} \right)$ = energy required to raise the temperature of a unit volume of coolant one temperature unit (erg/cm³ K).

Note that C-G-S units have been retained to maintain consistency with earlier implementations. For steady-state calculations the time derivatives in Eqs. (3.34a) and (3.34b) are set to zero. Transient applications use a fully-implicit time integration

method with the same time step that is used for the neutronic calculations.

3.7.2 The Cabral-IPM Model

A more comprehensive thermal-hydraulics model has recently been developed at M.I.T. by Cabral [C-1] and implemented in the IPM Code. Aviles [A-1] combined this model with a transient nodal method to investigate the spatially-dependent digital control of reactors. Because of the complicated nature of this thermal-hydraulics model, only a summary of the key features will be presented. A complete description can be found in references C-1 and A-1.

The fuel model uses a two-node control volume approach to model conduction accurately. Variations in fuel conductivity and gap conductance from thermal effects, fuel relocation, and cracking are accommodated. The fluid model uses one channel per assembly and does not allow cross flow. Subcooled nucleate boiling and two-phase flow models are included. The reactor core is assumed to be at constant pressure.

The subroutines required for the implementation of this method were extracted from Aviles' POPSICLE code [A-1].

3.7.3 Cross Section Feedback

For the purposes of this investigation, the cross section feedback from the thermal-hydraulics behavior will be accomplished by assuming that all macroscopic cross sections (and inverse diffusion coefficients) are linear functions of the node averaged fuel temperature, coolant temperature and coolant density

$$\begin{aligned} \Sigma_{\alpha}^{ijk}(\bar{T}_f, \bar{T}_c, \rho_c) = & \Sigma_{\alpha}^{ijk}(\bar{T}_{f0}, \bar{T}_{c0}, \rho_{c0}) + \left(\frac{\partial \Sigma_{\alpha}}{\partial \bar{T}_f} \right) (\bar{T}_f^{ijk} - \bar{T}_{f0}) \\ & + \left(\frac{\partial \Sigma_{\alpha}}{\partial \bar{T}_c} \right) (\bar{T}_c^{ijk} - \bar{T}_{c0}) + \left(\frac{\partial \Sigma_{\alpha}}{\partial \rho_c} \right) (\rho_c^{ijk} - \rho_{c0}), \end{aligned} \quad (3.35)$$

where \bar{T}_{f0} , \bar{T}_{c0} and ρ_{c0} represent the reference conditions. The linear functional dependence can, in general, accurately represent the actual cross sections over limited ranges of the temperatures and densities. For the purposes of this investigation this

linear assumption will be sufficient. Actual design calculations, however, may require that Eq. (3.35) be replaced by table look-up or polynomial fitting procedures to obtain a more accurate cross section representation.

3.8 Transient Control Mechanisms

Before the transient is initiated, the reactor is assumed to be at steady-state. The steady-state condition is typically obtained by dividing $\nu\Sigma_{fg}$ by k_{eff} for eigenvalue problems or by solving the static source equations for extraneous source problems. In order to initiate a transient, a perturbation in the reactor conditions is required. This perturbation can be the result of control rod motions, extraneous source changes, or thermal-hydraulic changes.

Nodal methods model control rod motions as spatially uniform changes in the macroscopic cross sections. The simplest method of obtaining these cross sections is by a volume-weighted average of the rodded and unrodded nodal cross sections. Since the neutron flux within the node is not spatially flat, this procedure will introduce a modelling error. These errors result in a cusp-like behavior of the flux versus time as the control rod moves through a node. An elaborate correction scheme was developed and incorporated into QUANDRY to reduce the error caused by this cusping effect [J-1].

In order to reduce the cusping effect, a simple correction model has been developed. If we knew the average flux in the rodded and unrodded portions of the node we could flux weight the corresponding rodded and unrodded cross sections to obtain a new homogenized nodal cross section

$$\bar{\Sigma}_{\alpha g}^{ijk} = \frac{h_k^{nr} \phi_g^{nr} \Sigma_{\alpha g}^{nr} + h_k^r \phi_g^r \Sigma_{\alpha g}^r}{h_k^{nr} \phi_g^{nr} + h_k^r \phi_g^r}, \quad (3.36)$$

where superscripts r and nr represent the rodded and unrodded portions of the node, respectively. The nodal method, however, computes just the average flux of the entire node, not of the two regions in which we are interested. In the case of a strongly absorbing rod, the neutron flux makes a very sharp change at the rod tip, and varies

more slowly away from the tip. Based on this observation we can approximate the flux in the unrodded portion of the node as being the average of the node-averaged flux of the partially rodded node and its lower neighbor. Likewise, the flux in the rodded portion can be approximated as the average of the flux in the partially rodded node and its upper neighbor. This results in

$$\phi_g^{nr} = \frac{h_k^{nr} \phi_g^{ijk} + h_{k-1} \phi_g^{ij,k-1}}{h_k^{nr} + h_{k-1}}, \quad (3.37a)$$

and

$$\phi_g^r = \frac{h_k^r \phi_g^{ijk} + h_{k+1} \phi_g^{ij,k+1}}{h_k^r + h_{k+1}}. \quad (3.37b)$$

Equations (3.37a) and (3.37b) are substituted into Eq. (3.36) to obtain flux-weighted nodal cross sections. Despite of the simplicity of this method, it has been found to perform as well as more elaborate models. This will be demonstrated in Chapter 5.

Another method by which the state of the reactor can be perturbed is through a change in the extraneous neutron source. The addition or removal of neutron sources changes the neutron population present in the core, and therefore, initiates a transient.

Finally, changes in the thermal-hydraulic conditions of the core can induce a transient through the cross section feedback. Two primary mechanisms are permitted by the thermal hydraulics models: changes in the coolant inlet-temperature; and changes in the core flow rate.

3.9 Summary

In this chapter the derivation of the transient nodal method was presented. The derivation began by demonstrating that the spatial dependence can be treated in the same manner as for the static equations. First, the time-dependent corrected finite-difference equations were obtained. Next, the transient polynomial nodal equations were derived. By introducing dynamic frequencies, the time derivatives in the nodal equations were eliminated and the similarity with the static nodal equations was demonstrated. As a result of the similarity with the static equations, the same non-

linear iterative solution procedure can be used to solve the nodal equations.

Next, the time-differencing schemes were presented. The theta method was applied to obtain equations which can be used to advance the nodal fluxes in time. The quasi-static method was then introduced to provide accurate results when large time steps are used. Equations for the amplitude and shape functions, which result from the application of the quasi-static method, were obtained.

Finally, two thermal-hydraulic feedback models and the transient control mechanisms were discussed. A method of reducing the cusping errors caused by the treatment of partially rodded nodes was presented.

The numerical solution methods for the static and transient equations are given in Chapter 4. The nodal method will then be applied to several problems in Chapter 5 to evaluate the accuracy and efficiency of the method.

Chapter 4

STATIC AND TRANSIENT NUMERICAL SOLUTION METHODS

4.1 Introduction

The derivation of the static equations was presented in Chapter 2 and the derivation of the transient equations was presented in Chapter 3. A non-linear iteration procedure was introduced in which the node-averaged fluxes are computed using finite-difference-like equations that are continuously updated with a more accurate polynomial nodal method. In this chapter, the detailed methods of solving the corrected finite-difference equations are given.

First, the numerical properties of the static equations are considered. The solution methods for eigenvalue problems, criticality searches, and fixed source problems are presented. Next the transient equations are examined. The solution method for the flux and shape function along with the point kinetics equations are presented.

4.2 Static Solution Methods

The static equations for which a solution is sought are Eq. (2.15) along with the polynomial nodal Eqs. (2.40) through (2.47). These two sets of equations are coupled through the application of the discontinuity factors. A general solution method for these equations involving a non-linear iteration scheme was outlined in Chapter 2. However, the method of solving the corrected finite-difference equations was not specified. These methods and the iteration optimizations are addressed in this section.

4.2.1 Numerical Properties

Recall that the corrected finite difference equation in matrix form is

$$\mathbf{A}(\Phi, \lambda)\Phi = \frac{1}{\lambda}\mathbf{M}\Phi + \mathbf{q}, \quad (2.17)$$

where the dependence of the matrix \mathbf{A} on the node-averaged fluxes and the eigenvalue has been explicitly indicated. This equation shows the non-linearity of the nodal method caused by the introduction of the discontinuity factors. Recall that if all the discontinuity factors are unity, Eq. (2.17) reduces to the mesh-centered finite difference neutron diffusion equations and the non-linearity vanishes.

For eigenvalue problems, the source term is zero resulting in the following equation

$$\mathbf{A}(\Phi, \lambda)\Phi = \frac{1}{\lambda}\mathbf{M}\Phi. \quad (4.1)$$

In general, \mathbf{A} can only be guaranteed to be real and irreducible. Because of the introduction of the discontinuity factors, we can no longer be certain of the diagonal dominance or symmetry of this matrix. However, if the discontinuity factors introduced for homogenization are unity, then the discontinuity factors used to correct the finite-difference equations approach unity as the node size becomes infinitely small. In this limit, our nodal method reduces to the finite-difference form of the diffusion equation and \mathbf{A} will have the following properties [V-2]:

1. \mathbf{A} is real,
2. the diagonal elements of \mathbf{A} are positive,
3. the off-diagonal elements of \mathbf{A} are non-positive,
4. \mathbf{A} is diagonally dominant,
5. \mathbf{A} is irreducible.

A matrix which has these properties is called an S-matrix and its inverse, \mathbf{A}^{-1} , exists

and has all positive elements [N-1]. Therefore, Eq. (4.1) may be written as

$$\Phi = \frac{1}{\lambda} \mathbf{R} \Phi, \quad (4.2)$$

where

$$\mathbf{R} \equiv \mathbf{A}^{-1} \mathbf{M}.$$

Since \mathbf{M} is a non-symmetric matrix with non-negative diagonal elements, \mathbf{R} is a non-symmetric, non-negative matrix.

4.2.2 Discontinuity Factor Iterations

The top iteration level in the non-linear iteration scheme consists of the discontinuity factor updating. In this iteration the corrected finite-difference equations are solved for the node-averaged fluxes, which requires two additional levels of iteration as discussed below. Then, using these fluxes, the polynomial expansion coefficients are computed and the surface-averaged net-currents are evaluated. The net currents and fluxes are then used to compute the discontinuity factors required by the corrected finite-difference equations. We then return to the solution of the finite-difference-like equations and repeat the process until convergence is achieved.

No theory exists which can be used to determine the convergence conditions of this iterative process. The fact that the equations are rigorously derived guarantees that if the method converges, it must converge to the solution of the polynomial nodal equations. Applications of this process have shown that the method converges quite rapidly, typically requiring between 5 and 10 iterations for LWRs with assembly-sized nodes.

The order in which the solution process is carried out is theoretically important. By having the top iteration level be the discontinuity factor updating, all discontinuity factors are constant during the solution of the corrected finite-difference equations. In addition, cross section updating in response to changing thermal-hydraulic conditions is performed with the discontinuity factor updates. As a result, the non-linearity is eliminated from the corrected finite-difference solution thereby allowing standard

numerical methods and convergence analyses to be applied.

4.2.3 Outer Iterations

The solution of Eq. (4.2) which we seek is the one with the largest eigenvalue. This solution can be found using the power method [N-1], which can be written as

$$\Phi^{(p+1)} = \frac{1}{\lambda^{(p)}} \mathbf{R} \Phi^{(p)}, \quad (4.3a)$$

$$\lambda^{(p+1)} = \lambda^{(p)} \frac{\langle \mathbf{w}, \Phi^{(p+1)} \rangle}{\langle \mathbf{w}, \Phi^{(p)} \rangle}, \quad p = 1, 2, \dots, \infty \quad (4.3b)$$

where p is the iteration number and \mathbf{w} is a weighting vector. According to the Perron-Frobenius theorem, an irreducible matrix having non-negative elements (such as the matrix \mathbf{R}) has a unique, positive eigenvalue greater in magnitude than the modulus of any other eigenvalue of the matrix. Using this theorem and the properties of the \mathbf{R} matrix given in Section 4.2.1, it is possible to show that the power method outlined in Eqs. (4.3a) and (4.3b) converges to the largest eigenvalue and its corresponding unique positive eigenvector [N-1].

The selection of the weighting vector is arbitrary, but does affect the convergence. One common choice is to set the elements of \mathbf{w} to unity such that the inner products in Eq.(4.3b) perform a simple summation of the elements of $\Phi^{(p+1)}$ and $\Phi^{(p)}$. Another choice for the weighting vector is for \mathbf{w} to be a vector of $\{\nu \Sigma_{fg}^{ijk}\}$ such that inner products perform summations over the fission source. This represents a more physical approach of determining the eigenvalue by taking the ratio of the neutron production in the current "generation" to the previous "generation", where a "generation" now represents an iteration. This weighting vector choice also has the benefit of only using information in the fueled regions which leads to a more stable iteration procedure and possibly faster convergence.

An alternate weighting vector choice based on a minimization procedure is the Rayleigh quotient [S-8]. In this method the weighting vector is chosen to be the flux (or fission source) vector from the previous iteration. The eigenvalue calculated

with this method can be shown to converge two times faster than the power method with unity weighting [N-1].

While the weighting vector does have an impact on the rate of convergence of the power method, the asymptotic convergence rate is primarily determined by the ratio of the moduli of the two largest eigenvalues

$$d \equiv \frac{|\lambda_1|}{|\lambda_0|}, \quad (4.4)$$

where λ_0 and λ_1 are the eigenvalues with the largest and second largest moduli, respectively. This ratio is called the *dominance ratio* and for most problems of interest, is so close to unity that the power method converges very slowly. One acceleration method which is particularly well suited for nodal methods is Wielandt's fractional iteration or eigenvalue shifting [W-2]. In Wielandt's method we move a portion of the right-hand side of Eq. (4.1) to the left-hand side as follows

$$\left[\mathbf{A} - \frac{1}{\lambda'} \mathbf{M} \right] \Phi = \frac{1}{\Lambda} \mathbf{M} \Phi, \quad (4.5)$$

where

$$\frac{1}{\Lambda} = \frac{1}{\lambda} - \frac{1}{\lambda'}.$$

The eigenvalue shift, λ' , is arbitrarily selected but is subject to certain restrictions which will be discussed later.

Equation (4.5) represents an eigenvalue problem which can be solved by the power method resulting in the following equations

$$\Phi^{(p+1)} = \frac{1}{\Lambda^{(p)}} \left[\mathbf{L} - \frac{1}{\lambda'} \mathbf{M} \right]^{-1} \Phi^{(p)}, \quad (4.6a)$$

$$\Lambda^{(p+1)} = \Lambda^{(p)} \frac{\langle \mathbf{w}, \Phi^{(p+1)} \rangle}{\langle \mathbf{w}, \Phi^{(p)} \rangle}, \quad (4.6b)$$

$$\lambda^{(p+1)} = \frac{\Lambda^{(p+1)} \lambda'}{\Lambda^{(p+1)} + \lambda'}. \quad (4.6c)$$

The matrix inversion indicated in Eq. (4.6a) is usually performed using an inner

iteration procedure and is strongly affected by the eigenvalue shifting procedure.

The eigenvector of this eigenvalue problem which is associated with the largest eigenvalue, Λ , is identical to the eigenvector associated with the largest eigenvalue of the unshifted equations [W-2]. The dominance ratio of the new eigenvalue problem is given by

$$d' = \frac{\frac{1}{\lambda_0} - \frac{1}{\lambda'}}{\frac{1}{\lambda_1} - \frac{1}{\lambda'}}. \quad (4.7)$$

If λ' is chosen such that its modulus exceeds λ_0 , the dominance ratio, d' , is less than unity and less than the unshifted dominance ratio, d . Choosing λ' to be the converged static eigenvalue λ_0 gives a dominance ratio of zero but makes the coefficient matrix of Eq. (4.5) singular. Choosing the eigenvalue shift to be infinite results in the unaccelerated power method of Eqs. (4.3a) and (4.3b).

In general, we will want the eigenvalue shift to change during the the problem to ensure optimum performance and to reflect discontinuity factor changes. A common procedure is to let the eigenvalue shift be the current estimate of the eigenvalue plus an arbitrary positive constant

$$\lambda' = \lambda^{(p)} + \delta\lambda. \quad (4.8)$$

This positive offset will ensure that the power method will converge to the correct eigenvalue and eigenvector while preventing singularity of the coefficient matrix. Recall that λ' must exceed λ_0 for the power method to converge to the proper eigenvalue. If $\lambda^{(p)}$ is a low estimate of λ_0 during the solution procedure and $\delta\lambda$ is small, convergence to the correct eigenvalue may not be obtained. This difficulty can generally be avoided by performing a few initial unaccelerated iterations to obtain a reasonable guess for λ_0 .

The shift parameter, $\delta\lambda$, must be carefully chosen to ensure that the solution procedure is optimal. There are two competing factors which must be considered. First, the convergence rate of the outer iteration procedure is maximized when the shift factor is small. Second, the convergence rate of the inner iteration procedure, which is required to update the eigenvector, is minimized when the shift factor is

small (recall that the coefficient matrix becomes singular as $\lambda' \rightarrow \lambda_0$). Thus, some acceptable, intermediate value for $\delta\lambda$ must be obtained. Several optimization studies have been performed [S-2, Z-1] and for LWRs optimal values of the shift factor range from 0.02 to 0.05.

As can be seen from Eq. (4.5) the eigenvalue shifting procedure results in a coefficient matrix which closely resembles a problem with strong up-scattering since \mathbf{M} is generally a full matrix. Thus, the implementation of Wielandt's method generally requires that all energy groups be solved simultaneously. Sutton, however, has developed a modified method which allows a group-wise solution procedure to be used [S-9]. In Sutton's method a matrix of "spectrum ratios" defined by

$$\mathbf{A}_{g'g} \equiv [\phi_g \mathbf{I}]^{-1} \phi_{g'} \quad (4.9)$$

is introduced. Using this definition, the shifted eigenvalue problem, Eq. (4.5), can be written using the group matrix equations of Section 2.3 as

$$\left[\mathbf{N}_g - \frac{1}{\lambda'} \sum_{g'=1}^G \mathbf{F}_{gg'} \mathbf{A}_{g'g} \right] \phi_g = \frac{1}{\Lambda} \sum_{g'=1}^G \mathbf{F}_{gg'} \phi_{g'} + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'} \phi_{g'}. \quad (4.10)$$

Applying the power method to Eq. (4.10) results in the following equations

$$\phi_g^{(p+1)} = \left[\mathbf{N}_g - \frac{1}{\lambda'} \sum_{g'=1}^G \mathbf{F}_{gg'} \mathbf{A}_{g'g}^{(p)} \right]^{-1} \times \left\{ \frac{1}{\Lambda^{(p)}} \sum_{g'=1}^G \mathbf{F}_{gg'} \phi_{g'}^{(p)} + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'} \phi_{g'}^{(p)} \right\}, \quad (4.11a)$$

$$\Lambda^{(p+1)} = \Lambda^{(p)} \frac{\langle \mathbf{w}, \phi_g^{(p+1)} \rangle}{\langle \mathbf{w}, \phi_g^{(p)} \rangle}, \quad (4.11b)$$

$$\lambda^{(p+1)} = \frac{\Lambda^{(p+1)} \lambda'}{\Lambda^{(p+1)} + \lambda'}. \quad (4.11c)$$

Note that the spectrum matrix $\mathbf{A}_{g'g}$ is updated at each outer iteration. Sutton has shown that this group-wise Wielandt procedure converges in nearly the same number of iterations as the simultaneous group solution procedure. This can be attributed to

the quick convergence of the spectrum ratios $\phi_{g'}/\phi_g$.

4.2.4 Inner Iterations

At each outer iteration the matrix $\left[\mathbf{N}_g - \frac{1}{\lambda'} \sum_{g'=1}^G \mathbf{F}_{gg'} \mathbf{A}_{g'g} \right]$ must be inverted for each energy group. Since this matrix may be quite large, iterative methods are best suited. The method that will be used is the cyclic Chebyshev semi-iterative (CCSI) method [V-2, H-6]. The CCSI method is a variant of the successive over-relaxation (SOR) method in which the relaxation parameter is changed during the iterations resulting in better average rates of convergence than SOR.

During the inner iterations we are solving equations of the form $\mathbf{A}\mathbf{u} = \mathbf{b}$ for each energy group, where

$$\mathbf{A} \equiv \mathbf{N}_g^{(p)} - \frac{1}{\lambda'} \sum_{g'=1}^G \mathbf{F}_{gg'} \mathbf{A}_{g'g}^{(p)}, \quad (4.12a)$$

$$\mathbf{b} \equiv \frac{1}{\Lambda^{(p)}} \sum_{g'=1}^G \mathbf{F}_{gg'} \phi_g^{(p)} + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'} \phi_g^{(p)}, \quad (4.12b)$$

$$\mathbf{u} \equiv \phi_g^{(p)}. \quad (4.12c)$$

In the CCSI method we partition the mesh into a red/black checkerboard pattern such that the matrix \mathbf{A} and the vectors \mathbf{u} and \mathbf{b} can be written as

$$\mathbf{A} = \begin{bmatrix} \mathbf{D}_R & \mathbf{H}_B \\ \mathbf{H}_R & \mathbf{D}_B \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_R \\ \mathbf{b}_B \end{bmatrix},$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_R \\ \mathbf{u}_B \end{bmatrix}.$$

Using these definitions, the matrix equation which is to be solved becomes

$$\begin{bmatrix} D_R & H_B \\ H_R & D_B \end{bmatrix} \begin{bmatrix} u_R \\ u_B \end{bmatrix} = \begin{bmatrix} b_R \\ b_B \end{bmatrix}. \quad (4.13)$$

If we define the following submatrices

$$F_R \equiv -D_R^{-1}H_R,$$

$$F_B \equiv -D_B^{-1}H_B,$$

$$c_R \equiv D_R^{-1}b_R,$$

$$c_B \equiv D_B^{-1}b_B,$$

Equation (4.13) can be written as

$$\begin{bmatrix} I & -F_R \\ -F_B & I \end{bmatrix} \begin{bmatrix} u_R \\ u_B \end{bmatrix} = \begin{bmatrix} c_R \\ c_B \end{bmatrix}. \quad (4.14)$$

Applying the Jacobi iterative method to Eq. (4.14) gives

$$u^{(r+1)} = Bu^{(r)} + c,$$

where r is the iteration index and B is the Jacobi iteration matrix defined by

$$B \equiv \begin{bmatrix} 0 & F_R \\ F_B & 0 \end{bmatrix}.$$

The CCSI method is obtained by applying Chebyshev acceleration to the red/black partitioned Jacobi method giving the following equations

$$u_R^{(r+1)} = \omega_R^{(r+1)} (F_R u_B^{(r)} + c_R) + (1 - \omega_R^{(r+1)}) u_R^{(r)}, \quad (4.15a)$$

$$u_B^{(r+1)} = \omega_B^{(r+1)} (F_B u_R^{(r+1)} + c_B) + (1 - \omega_B^{(r+1)}) u_B^{(r)}. \quad (4.15b)$$

The initial relaxation parameters are given by

$$\omega_R^{(1)} = 1,$$

$$\omega_B^{(1)} = \frac{1}{1 - \frac{1}{2}\rho^2},$$

and for subsequent iterations by

$$\omega_R^{(r+1)} = \frac{1}{1 - \frac{1}{4}\rho^2\omega_B^{(r)}}, \quad r \geq 1$$

$$\omega_B^{(r+1)} = \frac{1}{1 - \frac{1}{4}\rho^2\omega_R^{(r+1)}}, \quad r \geq 1.$$

Asymptotically, the two relaxation parameters become the same as the relaxation factor of SOR

$$\omega_b = \frac{2}{1 - \sqrt{1 - \rho^2}} = \omega_R^{(\infty)} = \omega_B^{(\infty)}. \quad (4.16)$$

The relaxation parameters depend upon the spectral radius of the Jacobi iteration matrix, ρ . The significance of the spectral radius in iterative methods is that the error is asymptotically reduced by a factor proportional to ρ in each iteration. Thus, if the spectral radius is close to unity, a large number of iterations will be required.

The spectral radius may be estimated by performing a series of Gauss-Seidel iterations ($\omega_R^{(r)} = \omega_B^{(r)} = 1$) and computing

$$(\rho_g^2)^{(r+1)} = \frac{\|\phi_g^{(r+1)} - \phi_g^{(r)}\|}{\|\phi_g^{(r)} - \phi_g^{(r-1)}\|}. \quad (4.17)$$

Any consistent vector norm may be used, but the L2 norm has been found to perform well. This method is essentially the power method with the eigenvector being the error vector (approximated by $\phi_g^{(r+1)} - \phi_g^{(r)}$) and the eigenvalue being the spectral radius. This procedure for calculating the spectral radius, however, converges slowly when the spectral radius is close to unity. A method by which the spectral radius can be computed in fewer iterations is to apply a constant value for $\omega_R^{(r)} = \omega_B^{(r)} = \omega_0$ which is less than the asymptotic value, ω_b , and use Eq. (4.17) to estimate the resulting

spectral radius which will be labeled $(\tilde{\rho}_g^2)^{(r)}$. The Jacobi spectral radius is related to $(\tilde{\rho}_g^2)^{(r)}$ by the following expression [W-2]

$$(\rho_g^2)^{(r)} = \frac{(\tilde{\rho}_g^2)^{(r)} + \omega_0 - 1}{\omega_0 \sqrt{(\tilde{\rho}_g^2)^{(r)}}}. \quad (4.18)$$

The value of ω_0 can be estimated from past experience with problems of a similar nature to the one being solved. For LWRs a value of $\omega_0 \sim 1.4$ is recommended.

Generally, it is not necessary to completely converge the flux vector in each set of inner iterations since the fission source terms are from the previous outer iteration. The approach that will be taken is to perform a fixed number of inner iterations per outer to achieve some desired error reduction. If the error reduction, ϵ , is defined to be the ratio of the vector norm of the error in the node-averaged fluxes after n iterations to the vector norm of the error in the average fluxes prior to the first iteration, it can be shown that [V-2]

$$\epsilon > \frac{2(\omega_b - 1)^n}{1 + (\omega_b - 1)^{2n}}, \quad (4.19)$$

where ω_b is the asymptotic relaxation factor defined in Eq. (4.16). Thus, the number of inner iterations to be performed for each energy group, n_g , is given by

$$n_g = \frac{\log \left(\frac{1}{\epsilon} - \sqrt{\frac{1}{\epsilon^2} - 1} \right)}{\log ((\omega_b)_g - 1)} \approx \frac{\log (\epsilon/2)}{\log ((\omega_b)_g - 1)}, \quad (4.20)$$

where $(\omega_b)_g$ is the asymptotic relaxation factor for energy group g . A parametric analysis performed by Smith [S-2] has found error reduction values in the range 0.1 to 0.4 provide acceptable results.

4.2.5 General Iterative Strategy

The following algorithm is proposed for the solution of the static equations:

1. An initial flat guess is made for the flux distribution and one unaccelerated outer iteration is performed. Subsequent outer iterations are performed with a constant shift which is known to be larger than the converged eigenvalue.

A value of 1.5 has been found to be effective for problems with a converged eigenvalue near 1.0. The inner iterations are performed using the Gauss-Seidel method. These outer iterations proceed until the eigenvalue reaches a coarse convergence ($\sim 10^{-2}$)

$$|\lambda^{(p+1)} - \lambda^{(p)}| < \epsilon_1.$$

2. Next the eigenvalue shift is applied

$$\lambda' = \lambda^{(p)} + \delta\lambda,$$

where $\delta\lambda$ is typically 0.02 – 0.05 for LWRs. Inner iterations are then performed to compute the spectral radius. These iterations continue until

$$|(\rho_g^2)^{(r+1)} - (\rho_g^2)^{(r)}| < \epsilon_2$$

is satisfied or until a pre-specified number of iterations is reached. A value of 10^{-2} to 10^{-3} for ϵ_2 and maximum number of iteration of 10–20 should be sufficient.

3. Now the iterative procedure with full acceleration may be performed. The iterations proceed until

$$|\lambda^{(p+1)} - \lambda^{(p)}| < \epsilon_3,$$

$$\text{maximum over all nodes } \left\{ \frac{|P^i - P_{\text{ref}}^i|}{P_{\text{ref}}^i} \right\} < \epsilon_4,$$

where P_i represents the power in node i . An eigenvalue convergence of 10^{-6} and a power convergence of 10^{-4} – 10^{-5} is generally sufficient.

This procedure has been used to solve several static problems. The results of some of these calculations are given in Chapter 5.

4.2.6 Criticality Search Problems

Another type of static problem which is typically encountered in reactor core calculations is the criticality search. An example of such an application is an operating PWR where the boron concentration is changed to offset reactivity changes from fuel depletion. We shall assume that the cross sectional dependence upon the boron concentration can be represented homogeneously throughout the core. Furthermore, we shall assume that this cross section dependence can be represented as a linear function

$$\Sigma_{\alpha g}^{ijk}(c) = \Sigma_{\alpha g}^{ijk}(c_0) + \left(\frac{\partial \Sigma_{\alpha g}}{\partial c} \right) (c - c_0), \quad (4.21)$$

where c represents the boron concentration having a reference value of c_0 . In a criticality search, we simply adjust the boron concentration during the discontinuity factor iterations until the eigenvalue becomes unity. Assuming a linear relation between the eigenvalue and the boron concentration, the new concentration at each iteration may be estimated from

$$c^{(p+1)} = \frac{c^{(p)} - c^{(p-1)}}{\lambda^{(p)} - \lambda^{(p-1)}} (1 - \lambda^{(p)}) + c^{(p)}. \quad (4.22)$$

Note that the linear relation between the boron concentration and the cross sections in Eq. (4.21) may be easily replaced by a higher order formula or a table look-up procedure when the data are available.

4.2.7 Source Problems

The outer-inner iteration procedure outlined above can also be used for the efficient solution of source problems. The problem which we want to solve is written in Eq. (2.16) in matrix energy group form

$$\mathbf{N}_g \phi_g = \frac{1}{\lambda} \sum_{g'=1}^G \mathbf{F}_{gg'} \phi_{g'} + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'} \phi_{g'} + \mathbf{q}_g. \quad (2.16)$$

By setting the eigenvalue shift, λ' , and the eigenvalue, λ , to unity, the Wielandt scheme can be used to move the fission terms over to the left-hand side

$$\left[\mathbf{N}_g - \sum_{g'=1}^G \mathbf{F}_{gg'} \mathbf{A}_{g'g'} \right] \phi_g = \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'} \phi_{g'} + \mathbf{q}_g, \quad (4.23)$$

The outer iterations are used to update the spectrum ratios $\mathbf{A}_{g'g}$ while the inner iterations perform the inversion of the coefficient matrix. The resulting outer iterations may be expressed as

$$\phi_g^{(p+1)} = \left[\mathbf{N}_g - \sum_{g'=1}^G \mathbf{F}_{gg'} \mathbf{A}_{g'g}^{(p)} \right]^{-1} \times \left\{ \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{gg'} \phi_{g'}^{(p)} + \mathbf{q}_g \right\}. \quad (4.24)$$

where p is the outer iteration index. The inner iterations are the same as for the eigenvalue problem but with an additional source term.

4.2.8 Mathematical Adjoint Problems

The application of the quasi-static method discussed in Chapter 3 requires the computation of the mathematical adjoint defined by Eq. (3.33). This equation represents an eigenvalue problem which can be solved using the same iteration procedure as described above for the forward eigenvalue problem. The discontinuity factors in this adjoint calculation, however, are held constant and are equal to the discontinuity factors from the solution of the forward problem. Therefore, no discontinuity factor updating is required. The transposition of the matrices is accomplished as follows:

1. The scattering matrix is transposed in energy.
2. The coupling matrix is transposed in energy and the discontinuity factors are transposed about the node interfaces.
3. $\{\nu \Sigma_{fg}\}$ is swapped with $\{\chi_g\}$ and both quantities are transposed in energy.

Since down-scatter in energy is dominant and the transposition of the scattering matrix reverses the direction of scatter, the energy groups are solved starting with

group G and proceeding through group 1, the opposite order of the forward problem.

The initial guess for the eigenvalue and adjoint flux vector is that of the forward problem. Since the adjoint problem has the same eigenvalue as the forward problem, the eigenvalue shift can be held constant throughout the solution. Applications have shown, however, that the eigenvalue shift factor must be larger for adjoint problems than for the forward solutions, typically $\delta\lambda = 0.5$ to 1.5 .

4.3 Transient Solution Methods

Now that the numerical properties and solution methods for the static equations have been discussed, we may focus on the transient equations. In Chapter 3 the transient, corrected finite-difference equations were developed and time-differenced using the theta method. The polynomial equations were also obtained and the use of the non-linear iteration procedure was discussed. In this section the properties and solution method of the transient, corrected finite-difference equations will be examined. In addition, the solution of the point kinetics equations required for the application of the quasi-static method is discussed.

4.3.1 Numerical Properties

After applying the nodal approximations, a system of spatially discretized, time-dependent ordinary differential equations was obtained, Eqs. (3.6a) and (3.6b). The properties of the spatial discretization remain the same as the properties presented for the static equations in Section 4.2.1. The properties of the semi-discrete equations and the time integration method remain to be discussed.

In order for a space-time solution scheme to be useful it must be stable. The issue of stability is the major determining factor in the choice of θ which was introduced in Eq. (3.14). Recall that $\theta = 0$ is the forward difference method, $\theta = \frac{1}{2}$ is the trapezoidal rule (or the Crank-Nicholson Method) and $\theta = 1$ is the backward difference method. It can be shown that the theta method is unconditionally stable only for values of $\theta \geq \frac{1}{2}$ [L-3].

For values of θ less than $\frac{1}{2}$, restrictions on the time step size are required to ensure

stability. Typically, these restrictions limit the time step to be on the same scale as the fastest varying quantities in the system of equations. The system in which we are solving, however, has quantities which have vastly different time constants. The behavior of the neutron flux can have time constants on the order of $1/(v_g \Sigma_{ag})$ which may be smaller than 10^{-8} seconds for fast neutrons [S-10]. The delayed neutron precursors, on the other hand, have decay constants ranging from hundredths of seconds to several seconds. Such systems of differential equations which have widely varying time scales such as this are said to be *stiff*. With the conditionally stable methods we are forced to follow all short-lived transient modes of the neutron behavior, even if they are not of interest. This translates to the requirement of using a prohibitively small time step.

Therefore, we are left to consider only values of $\theta \geq \frac{1}{2}$. Without other considerations, $\theta = \frac{1}{2}$ (the Crank-Nicholson method) is the best choice because it is the most accurate. The Crank-Nicholson method, however, exhibits a slowly decaying oscillatory behavior for stiff systems if moderately large time steps are used. Therefore, the most appropriate value for general use is $\theta = 1$, the backward-difference method.

4.3.2 Iterative Solution of the Transient Equations

In Chapter 3 the system of time difference equations was written in a super-matrix form repeated here for convenience

$$\mathbf{A}^{(n+1)} \boldsymbol{\phi}^{(n+1)} = \mathbf{s}^{(n)}. \quad (3.16)$$

This form shows that a large linear system must be solved. A two-level outer-inner iteration approach will be used for the solution of this equation. The outer iterations will be responsible for the solution of the energy group structure while the inner iterations will solve the spatially-differenced equations. This method is similar to that of the static calculation, except that the outer iterations are not used to compute an eigenvalue.

The spatial equations which must be inverted in each time step have a structure

which is identical to that of the matrix inverted in each inner iteration of the static calculation. Therefore, the same cyclic Chebyshev semi-iterative procedure used for the static inner iterations may be applied. Each set of inner iterations proceeds until some user specified error reduction is achieved. The number of iterations, however, is not determined a priori using the method of Section 4.2.4 since significant variations in convergence rates occur during the calculation.

The outer iterations also use Chebyshev-accelerated iterative methods. For two energy groups the equations have a cyclic nature such that CCSI may be used, as for the inner iterations. For more than two energy groups, however, the iteration matrix loses its cyclic properties requiring that the normal, rather than cyclic, Chebyshev method [V-2] be used. The outer iterations are performed until some user specified convergence rate is achieved.

The Chebyshev procedures require the knowledge of the Jacobi spectral radii. Unlike the static calculation, the Jacobi spectral radius must be re-estimated during the transient calculation because of variations in material properties, time step sizes and the state of the reactor. This is especially true of the spectral radius of the outer iteration matrix. In order to facilitate these changing conditions the transient problem is split into time domains in which the time step size remains constant. The procedure outlined in Section 4.2.4 is used to estimate the spectral radii at the beginning of each time domain.

For very large time steps the Chebyshev method used for the outer iterations results in slow convergence because the spectral radius is very close to unity. For this reason, a direct inversion method for the energy group solution has also been incorporated into the inner iteration procedure. In two group applications, the direct solution of the group equations has proven to be more efficient.

4.3.3 Frequency Estimation

In the derivation of the time-dependent polynomial nodal equations in Section 3.4 dynamic frequencies were introduced to eliminate the time derivatives. The frequen-

cies at time step n are assumed to be given by the following expressions

$$(\omega_g^{ijk})^{(n)} = \frac{1}{\Delta t_{n-1}} \ln \left(\frac{(\bar{\phi}_g^{ijk})^{(n)}}{(\bar{\phi}_g^{ijk})^{(n-1)}} \right), \quad (4.25a)$$

$$(\omega_d^{ijk})^{(n)} = \frac{1}{\Delta t_{n-1}} \ln \left(\frac{(c_d^{ijk})^{(n)}}{(c_d^{ijk})^{(n-1)}} \right). \quad (4.25b)$$

Under most circumstances these frequencies play only a minor role. However, in large reflector regions, the estimation of the frequencies using the above equations may lead to instabilities. This occurs because the fluxes in these reflector regions are relatively small and may vary significantly throughout the transient calculation. In order to avoid this problem the change of the dynamic frequencies from one time step to the next should be limited. Typically, a limitation on the maximum change of 25 s^{-1} is sufficient.

4.3.4 Solution of the Point Kinetics Equations

The application of the quasi-static method the point kinetics equations, Eqs. (3.23) and (3.25), must be solved. The precursor equations are solved using direct integration and the amplitude equation is solved using the theta method, the same methods which are used for the spatially dependent equations. The result of the direct integration procedure for the precursor equation, assuming that all of the point kinetics parameters are time dependent, is

$$C_d^{(p+1)} = k_{1,d} C_d^{(p)} + k_{2,d} \frac{\beta_d^{(p+1)}}{\lambda_d \Lambda^{(p+1)}} T^{(p+1)} - k_{3,d} \frac{\beta_d^{(p)}}{\lambda_d \Lambda^{(p)}} T^{(p)}, \quad (4.26)$$

where the values of $k_{1,d}$, $k_{2,d}$ and $k_{3,d}$ are the same as for the spatially-dependent precursor equations in Section 3.5 and p indicates the point kinetics time step. Theta

differencing of the amplitude equation gives

$$\begin{aligned} \frac{T^{(p+1)} - T^{(p)}}{\Delta t_p} = & \theta \left[\frac{\rho^{(p+1)} - \beta^{(p)}}{\Lambda^{(p+1)}} T^{(p+1)} + \sum_{d=1}^D \lambda_d C_d^{(p+1)} \right] \\ & + (1 - \theta) \left[\frac{\rho^{(p)} - \beta^{(p)}}{\Lambda^{(p)}} T^{(p)} + \sum_{d=1}^D \lambda_d C_d^{(p)} \right]. \end{aligned} \quad (4.27)$$

Substitution of Eq. (4.26) into (4.27) and subsequent manipulation gives the following equation which can be used to advance the amplitude function in time

$$\begin{aligned} & \left\{ 1 - \theta \frac{\Delta t_p}{\Lambda^{(p+1)}} \left[\rho^{(p+1)} - \beta^{(p+1)} + \sum_{d=1}^D k_{2,d} \beta_d^{(p+1)} \right] \right\} T^{(p+1)} \\ & = \left\{ 1 + \frac{\Delta t_p}{\Lambda^{(p+1)}} \left[(1 - \theta)(\rho^{(p)} - \beta^{(p)}) - \theta \sum_{d=1}^D k_{3,d} \beta_d^{(p)} \right] \right\} T^{(p)} \\ & + \Delta t_p \sum_{d=1}^D (1 + \theta(k_{1,d} - 1)) C_d^{(p)} + \theta q^{(p+1)} (1 - \theta) q^{(p)}. \end{aligned} \quad (4.28)$$

The same considerations in the choice of the theta presented for the spatially dependent equations also apply. Thus, the general recommended value is $\theta = 1$.

Since the time step size required for accurate solution of the amplitude function may vary significantly throughout a transient calculation, an adaptive procedure is used. The most straightforward adaptive technique is step doubling [P-2]. In this method each time step is performed twice: once with a time step of Δt_p resulting in an amplitude of $T_1^{(p+1)}$ and again with two steps of $\Delta t_p/2$ giving $T_2^{(p+1)}$. The truncation error, ϵ , can then be estimated using the following relation[D-2]

$$\epsilon \approx \frac{T_2^{(p+1)} - T_1^{(p+1)}}{2^m - 1}, \quad (4.29)$$

where

$$m = \begin{cases} 1 & \theta \neq \frac{1}{2} \\ 2 & \theta = \frac{1}{2}. \end{cases}$$

If the truncation error is smaller than some user specified value, ϵ_{user} , then the next

time step is estimated using

$$\Delta t_{p+1} = .9 \Delta t_p \left| \frac{\epsilon}{\epsilon_{\text{user}}} \right|^{\frac{1}{1+m}}. \quad (4.30)$$

If the truncation error, however, is larger than ϵ_{user} then Eq. (4.30) is used to estimate a time step which is used to repeat the current time step. A reasonable value of ϵ_{user} is in the range of 10^{-4} to 10^{-5} .

4.3.5 General Transient Computational Procedure

The transient solution procedure discussed above and in Chapter 3 are outlined in Figure 4-1. Before the transient calculation is performed, the initial steady-state conditions of the reactor are obtained. The weight function is obtained by solving the adjoint equations. During the transient calculation, the shape function is first computed using a large time step, Δt_n , using the thermal-hydraulic conditions at the beginning of the time step (the shape function is relatively insensitive to the thermal-hydraulic conditions). We then adopt smaller time steps, Δt_k and, starting at t_n , make cross section adjustments required by control rod motions. The point kinetics parameters are then computed using Eqs. (3.24a) through (3.24d). The shape function and discontinuity factors used in the calculation of the point kinetics parameters are obtained by a linear interpolation of the values at t_n and t_{n+1} . The amplitude function and effective precursor densities are then obtained by solving the point kinetics equation using the adaptive time stepping procedure. Finally, the thermal-hydraulic calculation is performed using nodal powers computed from the node-averaged fluxes which are evaluated as the product of the interpolated shape function and the amplitude function. The reactivity/thermal-hydraulic steps are repeated until time t_{n+1} is reached.

For transients involving feedback, several reactivity/thermal-hydraulics time steps are typically required per shape computation. The reactivity/thermal-hydraulics time step size is primarily limited by the tandem nature in which the neutronic and thermal hydraulic equations are solved. When feedback is not involved, however, only

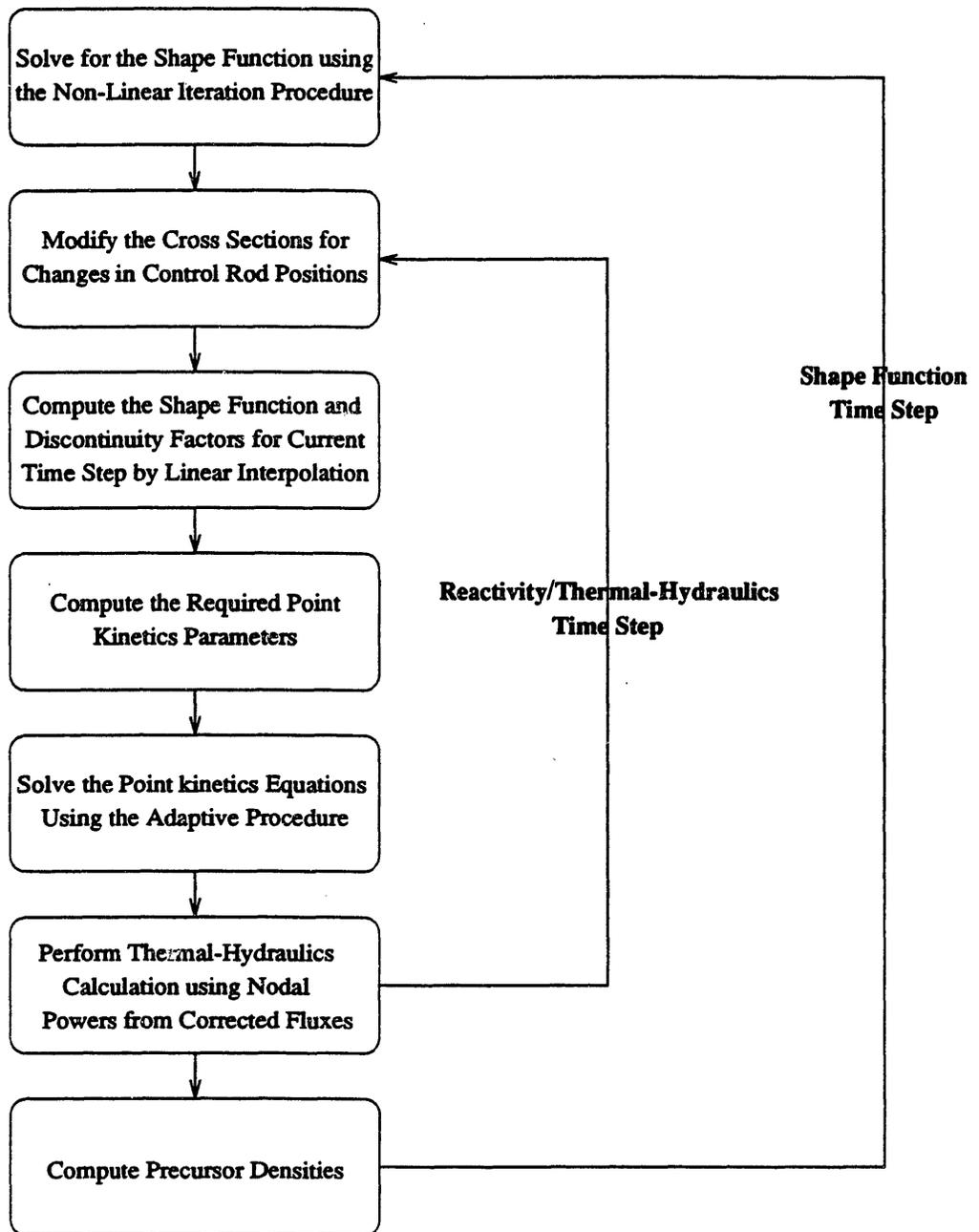


Figure 4-1: Flow diagram of the quasi-static transient solution procedure.

one reactivity/thermal-hydraulic time step is required per shape computation (i.e. $\Delta t_k = \Delta t_n$).

4.4 Summary

In this chapter the complete specification of the solution procedures for the static and transient equations were presented. The static equations are solved using a outer-inner iteration procedure. The power method, accelerated by Wielandt's fractional iteration, is used for the outer iterations and the cyclic Chebyshev semi-iterative method is applied to perform the inner iterations. A discussion of the solution procedures for other types of static problems was also presented.

The transient equations are solved using the quasi-static method in which the shape function calculation also employs an outer-iteration procedure. In both levels of iteration Chebyshev accelerated methods are used, but, for a small number of energy groups, a direct solution method for the outer iterations is applied. Finally, an adaptive procedure for solving the point kinetics equations was presented and the complete transient solution procedure was outlined

Chapter 5

APPLICATION OF THE TRANSIENT NODAL METHOD

5.1 Introduction

In Chapter 2 and Chapter 3 the static and transient nodal methods were derived. Solution methods for both cases were presented in Chapter 4. In this chapter, the methods will be applied to several transient problems to determine the efficiency and accuracy of the nodal method. Static results will be presented only as initial condition data since the results of other problems are virtually identical to the results of Zerkle [Z-1] which are based upon a similar polynomial method. The improvement in efficiency from the alternate expansion coefficient solution procedure, however, will be demonstrated.

Three of the problems which will be solved are widely-used benchmark problems for transient nodal calculations. Since the reactors modelled by these benchmarks are generally very simple in nature, the results of two additional problems based on a more realistic reactor configuration are presented. Note that many other static and transient problems have been analyzed to ensure the functionality of all options presented by this method.

5.2 Forward to Transient Problems

Before the discussion of the computational results, a few essential items remain to be discussed.

5.2.1 Computer Code

The static and transient methods presented in Chapter 2 and Chapter 3 along with the solution procedures of Chapter 4 are incorporated into a computer code which has been named CONQUEST (COde for Nodal QUasi-Static Theory). This computer code solves two- and three-dimensional, few-group, static and transient problems with and without extraneous neutron sources.

CONQUEST is written in standard FORTRAN 77, except for a few system dependent routines which return the system time and date. These references to system dependent routines are isolated and may be easily replaced or disabled without any impact on the actual code execution. CONQUEST has been compiled and executed without difficulty on the following machines:

IBM PC (80386 processor)

DEC VS3100

IBM RS/6000-320

Cray XMP 416

All computations are performed in single precision in order to minimize execution times and storage requirements.

When using CONQUEST, problems may be solved with either a polynomial method or a mesh-center finite-difference method. The polynomial method allows a quadratic, cubic or quartic approximation. Non-uniform mesh spacings and irregular geometries (jagged boundaries) are allowed as well as a diagonal symmetry option. There is no limit on the number or structure (including up-scattering) of the neutron energy groups. The code allows the use of homogenization parameters consisting of cross sections and discontinuity factors.

The implementation of the quasi-static method allows the user to specify when shape updates are to be performed and the number of reactivity and thermal-hydraulic calculations per shape update. The adaptive procedure used to solve the point kinetics equations provides an accurate solution without any user input. A point kinetics option, in which no shape updates are performed, is also available.

5.2.2 Transverse-Leakage Approximations

In Chapter 2 the use of the quadratic transverse leakage approximation was discussed. In this approximation the transverse leakage is expanded as a quadratic polynomial which preserves the node-averaged transverse leakage in the three adjacent nodes. This does not present a problem for nodes in the reactor interior or at boundaries of symmetry. However, nodes at the reactor surface do not have the third adjacent node required to perform the quadratic leakage expansion. Therefore, the transverse-leakage expansion for nodes on the reactor surface is performed using the three nodes which are closest to the surface. This approximation gives good results for problems with relatively thin reflectors or fuel-bearing compositions on the surface of the reactor.

For problems with large reflectors, a quadratic transverse-leakage approximation in the core and a flat transverse leakage in the reflector has been found to give good results. The reason for this is that the leakages deep within the reflector tend to be small and only have small effect on the core power distribution. In fact, approximating the transverse leakage in large reflectors as a quadratic polynomial has resulted in stability problems which are not present when the flat approximation is used.

5.2.3 Power Distribution Errors

The solutions to problems presented in this chapter are compared to reference solutions. The normalized power densities of the reference solutions and the errors in the CONQUEST solutions are presented in Appendix C. However, for purposes of summarizing these results, tables containing the maximum and average node and assembly errors in the normalized power densities are presented in this chapter. The maximum error in the node power density is defined to be

$$\epsilon_{\max} \equiv \text{maximum over all nodes} \left\{ \frac{|P^i - P_{\text{ref}}^i|}{P_{\text{ref}}^i} \right\},$$

where P^i represents the power density in node i and P_{ref}^i represents the reference power density in node i . The average error in the nodal power density is defined to

be

$$\bar{\epsilon} \equiv \frac{1}{V_{\text{core}}} \sum_i \frac{|P^i - P_{\text{ref}}^i|}{P_{\text{ref}}^i} V_i,$$

where V_i is the volume of node i and V_{core} is the total volume of the reactor core. All power densities are normalized such that the mean reactor power density is unity.

In the static calculations, the convergence criteria on the nodal power of 10^{-5} has been used for all 2-D calculations and 10^{-4} for all 3-D calculations. For all transient calculations convergence in the average change in nodal power of 10^{-3} has been used (except for the 3-D LRA problem where 10^{-4} was used to reduce instabilities).

5.2.4 Execution Times

The execution times of computer codes are commonly used to compare their relative performance. Direct comparisons of execution times, however, are often difficult because the calculation speed of different computer systems vary widely. In order to establish rough comparisons between computers the LINPACK benchmark [D-3] is commonly used. This benchmark measures the single and double precision floating-point performance of a computer system, in terms of millions of floating-point operations per second (MFLOPS) by solving a linear system of equations of order 100 using the LINPACK LU decomposition routines. The LINPACK benchmark is intended to represent the typical computational mix found in many engineering calculations. The LINPACK MFLOPS ratings given in Table 5.1 are used in this chapter for comparisons of execution times.

All CONQUEST calculations have been performed on a DEC VAXstation 3100 M38 in single precision. Therefore, for the purpose of comparison, all execution times have been converted to single precision DEC VS3100 M38 execution times by using the ratios of the LINPACK MFLOPS ratings. The conversion between quarter-core and eighth-core symmetries has been performed by using the ratio of the number of nodes in each symmetry. The resulting execution times should be considered approximate, but should be sufficient to determine whether large differences in execution times exist.

Table 5.1: LINPACK MFLOPS ratings of several computer systems used for the comparison of execution times.

Computer System	Precision	MFLOPS
CDC CYBER 176 [D-3]	S	4.6
IBM 370/195 [D-3]	S	3.3
SGI 4D/310 (1 proc) 25 MHz [Z-1]	D	2.8
CDC CYBER 175 [D-3]	S	2.4
IBM 370/168 [D-3]	S	1.2
IBM 360/91 [S-2]	S	~ 1.2
AMDAHL 470 V/6 [D-3]	S	1.1
DEC VS3100 M38	S	0.78
CDC 7600 [D-3]	S	0.48

5.3 The 2-D TWIGL Seed-Blanket Reactor Problems

This problem was proposed by Hageman and Yasinsky [H-7] and solutions were originally obtained using the finite-difference code TWIGL. This benchmark is a two-dimensional model of a 160 cm square, unreflected seed-blanket reactor using two neutron energy groups and one delayed precursor group. A complete description of this problem is given in Appendix B. Two different transients are initiated by a step and ramp perturbation of the corner seed assembly. The small size of this transient problem allows a study of different calculation procedures.

5.3.1 The Static Solution

The static solution to this problem was obtained with two different mesh spacings to investigate the spacial convergence of the quartic polynomial approximation. The mesh structures are denoted as “coarse” and “fine” and are defined as follows:

<u>Region</u>	<u>Coarse Mesh</u>	<u>Fine Mesh</u>
$0 \leq x, y \leq 24$ cm	12 cm	8 cm
$24 \leq x, y \leq 56$ cm	16 cm	8 cm
$56 \leq x, y \leq 80$ cm	12 cm	8 cm

Table 5.2: A summary of the CONQUEST static results for the 2-D TWIGL seed-blanket test problem.

	Mesh	
	Coarse	Fine
Nodes (1/8 core)	21	45
D.F. Updates ^a	10	11
Outer Iterations	22	25
Eigenvalue ^b	0.91312	0.91320
ϵ_{\max} (%)	+0.46	+0.04
$\bar{\epsilon}$ (%)	0.25	0.01
CPU Time ^c (sec)	1.2	3.6

^aMaximum of 3 outer its. per discontinuity factor update

^bReference Eigenvalue: 0.91321

^cDEC VS3100 M38

The static results for these two mesh structures are presented in Table 5.2. The reference for the calculation is a QUANDRY calculation using the fine mesh which is nearly spatially converged [S-2]. The errors in the eigenvalue and power densities for the coarse mesh are quite small. Thus, the coarse mesh will be used for all subsequent transient analyses. This problem was also solved with an earlier version of CONQUEST based on the alternate expansion coefficient solution procedure of Zerkle [Z-1]. For the coarse mesh calculation, 22 discontinuity factor updates were required nearly doubling the required execution time.

The nature of the polynomial approximations can be more directly examined by plotting the transversely-integrated fluxes and currents for the coarse mesh. The group 2, x-direction, transversely-integrated fluxes are shown in Figure 5-1 and currents in Figures 5-2, 5-3, and 5-4 for the quadratic, cubic, and quartic approximations, respectively. These values are for the first row of nodes $0 \leq y \leq 12$ cm (referred to as the $j = 1$ nodes). The reference curve in these figures was obtained by using an x-direction mesh spacing of 3 cm while maintaining the coarse mesh structure in the y-direction. The flux profiles show that, in general, the lower order polynomials do not have the flexibility required to closely follow the correct shape. The quartic

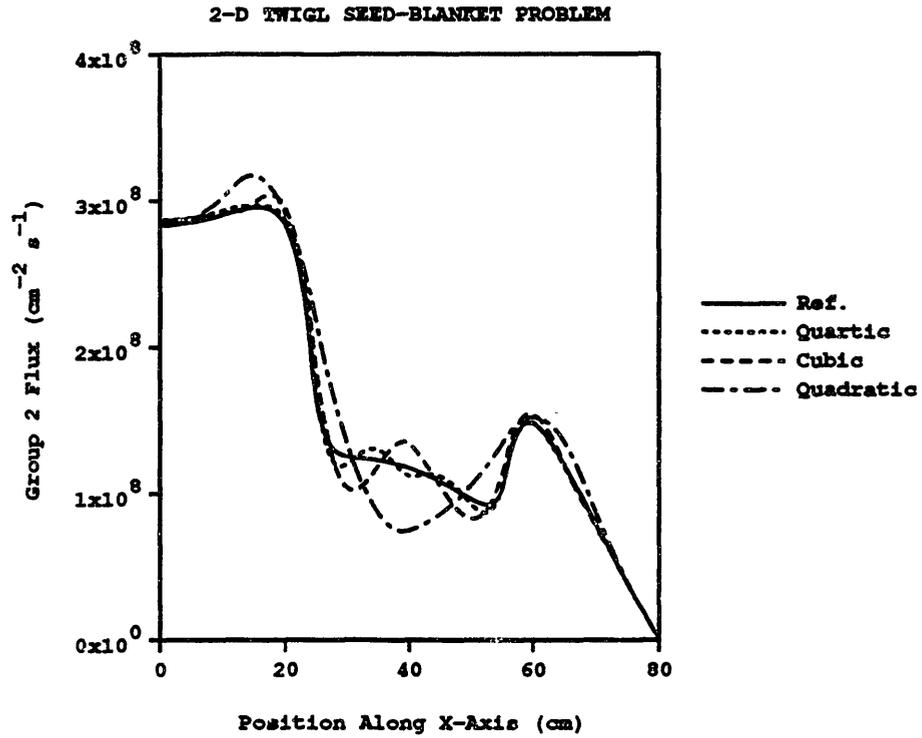


Figure 5-1: The group 2, x-direction transversely-integrated fluxes ($j = 1$) for the TWIGL problem.

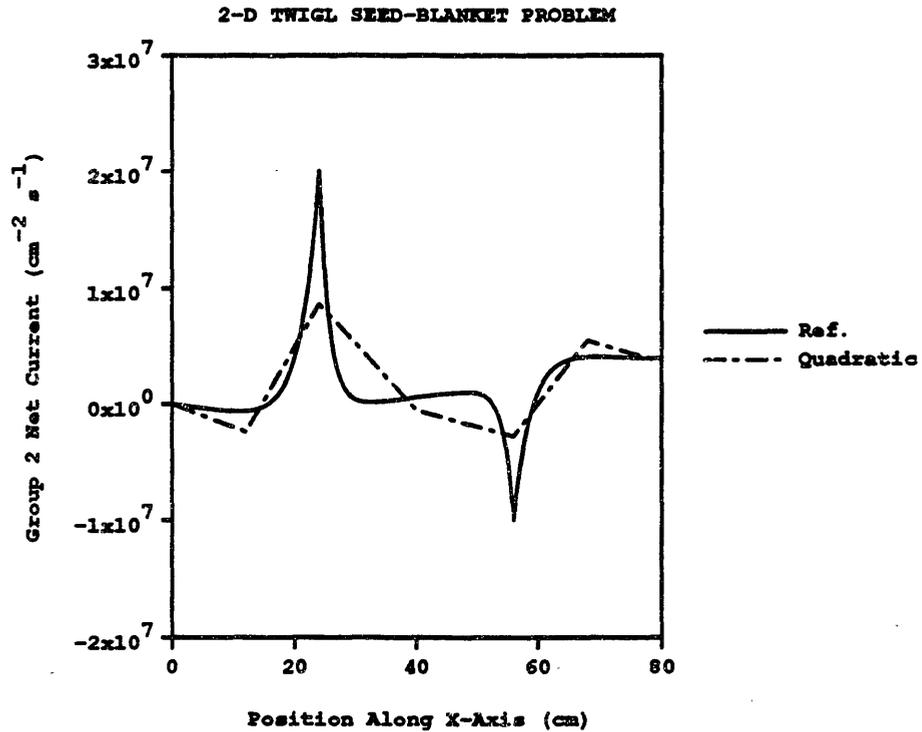


Figure 5-2: The group 2, x-direction, quadratic transversely-integrated currents ($j = 1$) for the TWIGL problem.

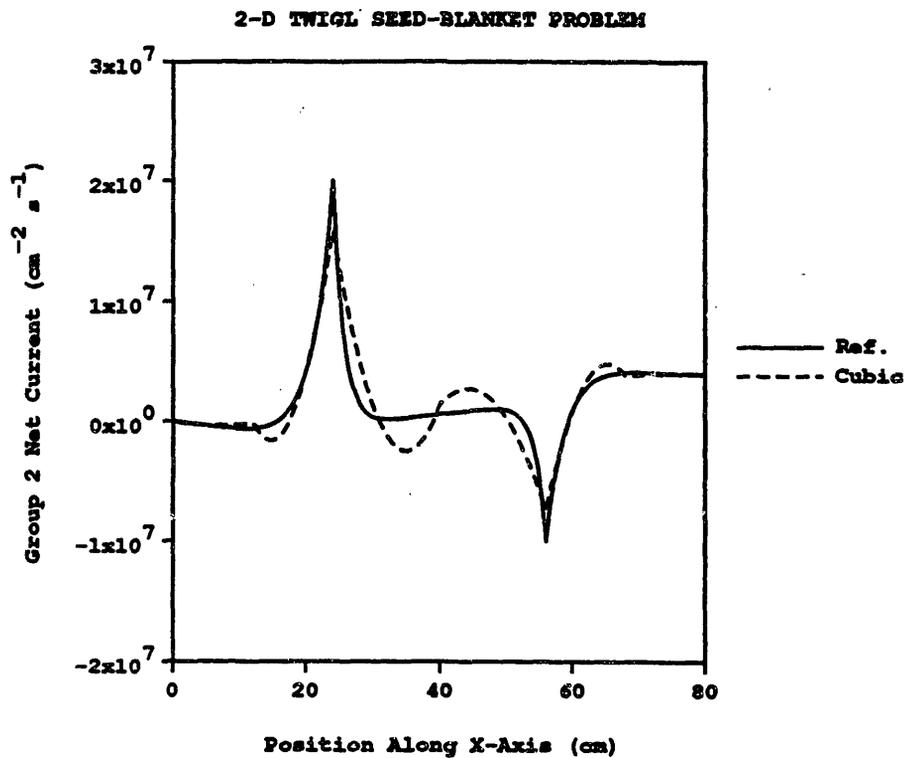


Figure 5-3: The group 2, x-direction, cubic transversely-integrated currents ($j = 1$) for the TWIGL problem.

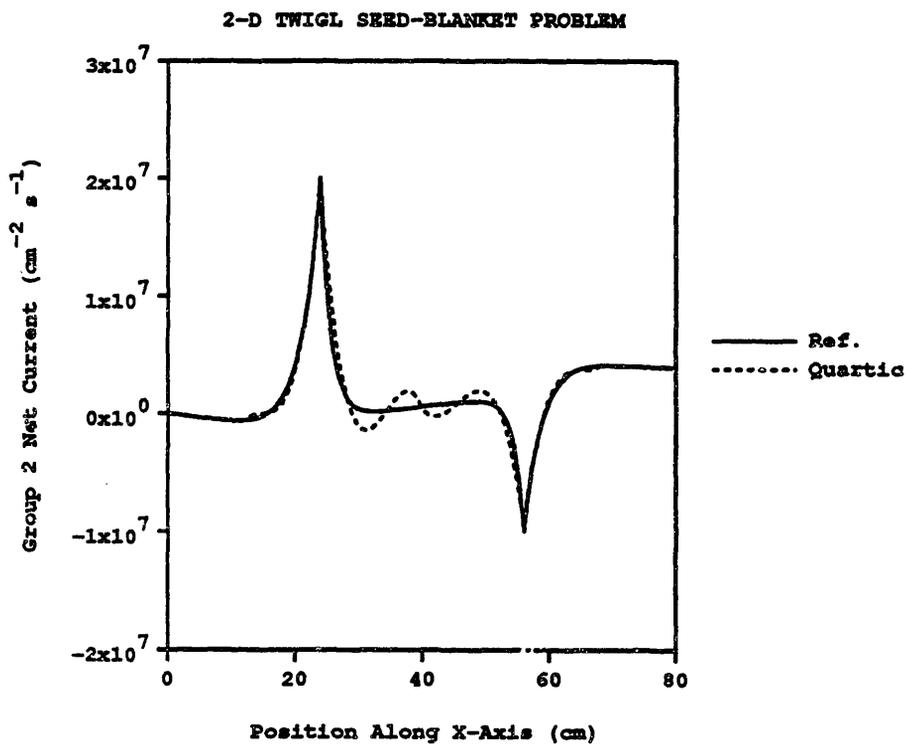


Figure 5-4: The group 2, x-direction, quartic transversely-integrated currents ($j = 1$) for the TWIGL problem.

Table 5.3: Reactor power vs. time for the 2-D TWIGL step transient problem.

Time (s)	Shape Update Step, Δt_n			
	0.01 sec	0.05 sec	0.1 sec	0.5 sec
0.0	1.000	1.000	1.000	1.000
0.1	2.065	2.060	2.066	2.040
0.2	2.078	2.078	2.078	2.064
0.3	2.095	2.095	2.094	2.088
0.4	2.113	2.112	2.113	2.113
0.5	2.130	2.130	2.131	2.138
CPU Time ^a (sec)	5.9	3.3	1.9	1.0

^aDEC VS3100 M38

polynomial, however, can provide enough flexibility to match the correct shape. Note that in the relatively flat region $20 < x < 60$ cm that the cubic and quartic fluxes oscillate about the reference flux in such a manner that accurate node-averaged fluxes are obtained.

The net current plots demonstrate the increased accuracy in the leakages of the higher order approximations. Figure 5-2, shows that the quadratic approximation for the flux gives a linear approximation for the currents and leads to significant errors at the nodal interfaces. The quartic approximation, on the other hand, closely matches the relatively large currents at the nodal interfaces. If accurate nodal leakages are obtain, the nodal reaction rates and the power densities will be accurate.

5.3.2 The Step Transient

The first of the two TWIGL transient problems is a step transient in which the thermal absorption cross section in the corner seed assembly is reduced by 2.3 % in an instantaneous fashion. The results are presented in Table 5.3 for several different shape-update steps. As this table shows, there is an initial prompt-jump after which the power rises relatively slowly. The actual shape change in this calculation, however, is small, as indicated by the accuracy of the one shape-update calculation.

Table 5.4: Reactor power vs. time for the 2-D TWIGL step transient problem with various number of steps per discontinuity factor update.

Time (s)	Number of Time Steps per D.F. Update				
	1	2	5	10	None
0	1.000	1.000	1.000	1.000	1.000
0.1	2.060	2.060	2.065	2.077	2.143
0.2	2.078	2.078	2.078	2.083	2.161
0.3	2.095	2.095	2.095	2.097	2.181
0.4	2.112	2.112	2.113	2.113	2.201
0.5	2.130	2.130	2.131	2.131	2.220
CPU Time ^a (sec)	9.5	7.0	5.9	5.4	4.9

^aDEC VS3100 M38

In these calculations the discontinuity factors were updated at every shape-update time step. If the flux shape changes slowly, however, the discontinuity factors will also change slowly and it may not be necessary to update the discontinuity factors at every time step. A parameter study with several time steps per shape update is presented in Table 5.4. The data in this table show that several time steps may be taken between discontinuity factor updates. This can be partially attributed to the small shape changes in this problem. Note that the last column in this table represents a transient calculation which is performed using finite-difference which is corrected with the initial static discontinuity factors. The relatively large error in this calculation indicates that constant discontinuity factors are not sufficient and discontinuity factor updates must be performed to reflect the changes in the flux shape.

The choice of the weight function was also examined for this transient. Table 5.5 shows point kinetics and quasi-static power and reactivity versus time for calculations performed with adjoint and unity weighting. The point kinetics method results in accurate results for this transient only if the adjoint flux is used for the weight function. The quasi-static method, however, obtains reasonable answers for both adjoint and unity weighting. Note, however, that the adjoint weight function gives a better initial

Table 5.5: Reactor power and reactivity vs. time for the TWIGL step problem for point kinetics (PK) and quasi-static (QS) methods with adjoint and unity weighting.

Time (s)	PK Adjoint		PK Unity		QS Adjoint		QS Unity	
	Power	React. $\times 10^3$	Power	React. $\times 10^3$	Power	React. $\times 10^3$	Power	React. $\times 10^3$
0.0	1.000	3.7874	1.000	2.6708	1.000	3.7874	1.000	2.6708
0.1	2.033	3.7874	1.559	2.6708	2.066	3.8406	2.045	3.9180
0.2	2.050	3.7874	1.565	2.6708	2.078	3.8254	2.071	3.8212
0.3	2.067	3.7874	1.573	2.6708	2.094	3.8241	2.074	3.8022
0.4	2.084	3.7874	1.580	2.6708	2.113	3.8275	2.105	3.8314
0.5	2.100	3.7874	1.587	2.6708	2.130	3.8266	2.129	3.8413

estimate of the reactivity. From this analysis we see that the use of the adjoint as a weight function is not as important for the quasi-static method as for point kinetics since the shape function is periodically updated.

A comparison of the results with two other nodal codes QUANDRY [S-2] and 2DTD [A-3] is presented along with execution times in Table 5.6. This table shows that CONQUEST gives answers which are just as accurate as other nodal methods with comparable execution times. Note that the execution times have been adjusted to approximate equivalent values for quarter-core calculations on a DEC VS3100 M38. The quasi-static method gives a significant reduction in computation time by allowing much larger shape-update steps.

5.3.3 The Ramp Transient

In this transient the perturbation consists of a 2.3 % linear decrease in the thermal absorption cross section of the corner seed assembly over 0.2 seconds. For the small time step calculation (5 ms), a study of the discontinuity factor update frequency is presented in Table 5.7. The results are very similar to the step transient. For this transient the discontinuity factors change slowly enough that they only need to be updated every 5 to 10 time steps. In addition, the calculation was performed using the point kinetics and quasi-static methods with adjoint and unity weighting. The

Table 5.6: Reactor power vs. time for the 2-D TWIGL step transient for several nodal codes ($\Delta t_n = 10$ ms).

Time (sec)	2DTD [A-3]	QUANDRY [S-2]	CONQUEST	Ref. [S-2]
0	1.000	1.000	1.000	1.000
0.1	2.051	2.064	2.060	2.061
0.2	2.068	2.076	2.078	2.078
0.3	2.085	2.095	2.095	2.095
0.4	2.102	2.112	2.112	2.113
0.5	2.119	2.130	2.130	2.131
CPU Time ^a (sec)	8.2	6.3	5.9	—

^aDEC VS3100 M38

Table 5.7: Reactor power vs. time for the TWIGL ramp problem with various number of steps per discontinuity factor update ($\Delta t_n = 5$ ms).

Time (sec)	Number of Time Steps per D.F. Update				
	1	2	5	10	None
0	1.000	1.000	1.000	1.000	1.000
0.1	1.309	1.310	1.311	1.313	1.323
0.2	1.961	1.962	1.966	1.973	2.026
0.3	2.074	2.074	2.074	2.074	2.157
0.4	2.091	2.091	2.091	2.091	2.177
0.5	2.109	2.109	2.109	2.109	2.196
CPU Time ^a (sec)	18.7	14.7	11.9	10.3	9.8

^aDEC VS3100 M38

Table 5.8: Reactor power and reactivity vs. time for the TWIGL ramp problem for point kinetics (PK) and quasi-static (QS) methods with adjoint and unity weighting.

Time	PK Adjoint		PK Unity		QS Adjoint		QS Unity	
	Power	React. $\times 10^3$	Power	React. $\times 10^3$	Power	React. $\times 10^3$	Power	React. $\times 10^3$
0.0	1.000	0.0	1.000	0.0	1.000	0.0	1.000	0.0
0.1	1.311	1.8937	1.204	1.3354	1.314	1.9031	1.322	1.9268
0.2	1.939	3.7874	1.531	2.6709	1.978	3.8623	1.997	3.8826
0.3	2.047	3.7874	1.565	2.6708	2.077	3.8277	2.077	3.8347
0.4	2.063	3.7874	1.571	2.6708	2.091	3.8243	2.084	3.8197
0.5	2.080	3.7874	1.579	2.6708	2.109	3.8264	2.107	3.8332

results of these calculations are presented in Table 5.8. Again, the adjoint weight function is very important for the point kinetics but not as important for the quasi-static method. Note that unlike the step transient, an initial estimate of the reactivity is not required because of the ramp nature of the transient.

Various time steps were used to solve this problem with discontinuity factor updates being performed every time step. The results are presented in Table 5.9. As for the step transient, accurate results can be obtained with 0.1 second shape-update time steps.

The results of the calculations are presented in Table 5.10 along with those of other nodal methods. The polynomial nodal method, without the use of the quasi-static method, gives accurate results with competitive execution times. The quasi-static method, in CONQUEST as well as in other nodal codes, leads to a significant reduction in execution time without degradation of accuracy.

5.4 The 3-D LMW Operational Transient

The 3-D LMW (Langenbuch-Maurer-Werner) LWR transient problem [L-2, S-2] is a highly simplified LWR described in Appendix B. The reactor is modelled with two neutron energy groups and six precursor groups. The transient involves the

Table 5.9: Reactor power vs. time for the TWIGL ramp problem for various shape-update time steps.

Time (s)	Shape Update Step, Δt_n (sec)			
	0.005	0.05	0.1	0.5
0.0	1.000	1.000	1.000	1.000
0.1	1.309	1.319	1.314	1.376
0.2	1.961	1.972	1.978	1.958
0.3	2.074	2.073	2.077	2.076
0.4	2.091	2.091	2.091	2.100
0.5	2.109	2.109	2.109	2.125
CPU Time ^a (sec)	18.7	3.7	2.1	1.1

^aDEC VS3100 M38

Table 5.10: Reactor power vs. time for the 2-D TWIGL ramp transient for several nodal codes ($\Delta t_n = 5$ ms).

Time (sec)	2DTD [A-3]	QUANDRY [S-2]	CONQUEST	Ref. [S-2]
0	1.000	1.000	1.000	1.000
0.1	1.305	1.305	1.311	1.307
0.2	1.951	1.954	1.966	1.957
0.3	2.064	2.074	2.074	2.074
0.4	2.081	2.092	2.091	2.096
0.5	2.098	2.109	2.109	2.109
CPU Time ^a (sec)	15.5	18.0	18.4	—

^aDEC VS3100 M38

Table 5.11: A summary of the CONQUEST static results for the 3-D LMW problem without feedback.

	Node Size		
	20 × 20 × 20 cm	10 × 10 × 10 cm	5 × 5 × 10 cm
Nodes (1/8 core)	200	1480	5800
D.F. Updates	10	8	15
Outer Iterations	23	24	49
Eigenvalue	0.999655	0.999677	0.999672
$\epsilon_{\max}(\text{node},\%)$	-1.20	+0.16	ref.
$\bar{\epsilon}(\text{node},\%)$	0.29	0.02	ref.
$\epsilon_{\max}(\text{assembly},\%)$	-0.68	-0.05	ref.
$\bar{\epsilon}(\text{assembly},\%)$	0.15	0.01	ref.
CPU Time ^a (sec)	15.8	84.9	637.

^aDEC VS3100 M38

withdrawal of a bank of four partially-inserted control rods and the subsequent insertion of a bank of five control rods. These complicated control rod motions lead to significant shape changes and large cusping effects and are a good test for the quasi-static method. This problem has been solved both with and without thermal-hydraulic feedback.

5.4.1 The 3-D LMW Problem Without Feedback

The static calculation required for the initial conditions was performed with three different node spacings and eighth-core symmetry. The results are summarized in Table 5.11 and the normalized assembly power density comparison is presented in Appendix C. The reference for the calculation is a CONQUEST calculation with 5 cm radial meshes and 10 cm axial meshes. The maximum error in the assembly-sized mesh occurs in a low power node on the core/reflector interface.

A comparison of the CONQUEST and QUANDRY calculation is given in Table 5.12. Note that the reference for each calculation is a spatially converged, fine mesh calculation performed with each corresponding code. This comparison shows

Table 5.12: A comparison of the CONQUEST and QUANDRY 3-D LMW static calculations without feedback.

	CONQUEST	QUANDRY [S-2]
Outer Iterations	23	17
Eigenvalue	0.999655	0.99974
$\epsilon_{\max}(\text{node},\%)$	-1.20	+0.98
$\epsilon_{\max}(\text{assembly},\%)$	-0.68	+0.28
$\bar{\epsilon}(\text{assembly},\%)$	0.15	0.12
CPU Time ^a (sec)	15.8	17.2 ^b

^aDEC VS3100 M38

^bActual computation time: 11.2 sec., IBM 370/168

that the polynomial nodal method gives accuracy which is comparable to the analytic nodal method with similar calculation times.

The transient calculation was performed using 20 cm nodes in all directions and eighth-core symmetry. Quasi-static calculations were performed with 1/4 second, 1 second, and 5 second shape update and reactivity calculation time steps. In the 1/4 second and 1 second cases the discontinuity factors were updated every four steps and in the 5 second case, every step. The cusping correction presented in Section 3.8 was used in all calculations. The results of these calculations are presented numerically in Table 5.13 and graphically in Figures 5-5 and 5-6. These results show that excellent agreement in power density and reactivity is obtained with all time steps. The execution speed for the largest time step case is nearly a factor of 9 times quicker than the reference calculation resulting in a faster than real time calculation.

In order to investigate the local accuracy of the quasi-static method, a comparison of the nodal power densities was performed. A summary of the comparison is given in Table 5.14. The maximum errors in the nodal power densities occur in nodes in which the control rods are moving. This error in nodes in which rods are moving can be expected since, for the $\Delta t_n = 5$ sec calculation, the rod completely traverses the node between shape updates. The remaining nodes have very small errors as indicated by the small node-averaged, assembly-maximum, and assembly averaged errors.

Table 5.13: Mean power density (W/cm^3) vs. time for the 3-D LMW problem without feedback and different shape-update time steps.

Time	Time Step Size, Δt_n		
	5 sec	1 sec	1/4 sec ^a
0.0	150.0	150.0	150.0
5.0	169.4 (+0.1%)	169.2 (0.0%)	169.2
10.0	201.8 (0.0%)	201.6 (-0.1%)	201.8
20.0	258.9 (-0.2%)	259.7 (+0.2%)	259.3
30.0	207.9 (+0.0%)	207.9 (+0.0%)	207.8
40.0	122.8 (+0.4%)	122.6 (+0.3%)	122.3
50.0	76.3 (+0.7%)	75.9 (+0.1%)	75.8
60.0	58.2 (0.0%)	58.2 (0.0%)	58.2
CPU Time ^b (sec)	50.	115.	434.

^aReference

^bDEC VS3100 M38

Table 5.14: A comparison of the node and assembly errors in the normalized power densities.

Time	$\Delta t_n = 1$ sec		$\Delta t_n = 5$ sec	
	Node Error ^a max/avg	Assembly Error max/avg	Node Error max/avg	Assembly Error max/avg
0.0	0.0/0.0 %	0.0/0.0 %	0.0/0.0 %	0.0/0.0 %
5.0	+0.8/0.2 %	+0.3/0.1 %	+2.5/0.2 %	+0.4/0.1 %
10.0	+1.2/0.1 %	+0.2/0.0 %	+2.9/0.2 %	+0.8/0.1 %
20.0	-1.0/0.1 %	-0.1/0.0 %	+3.4/0.2 %	-0.4/0.1 %
30.0	-1.3/0.1 %	-0.2/0.0 %	+3.9/0.3 %	-0.7/0.1 %
40.0	+1.3/0.1 %	-0.1/0.0 %	-3.3/0.2 %	-0.8/0.1 %
50.0	+0.2/0.0 %	-0.1/0.0 %	-3.0/0.2 %	-0.4/0.1 %
60.0	+0.4/0.1 %	+0.1/0.0 %	+0.6/0.0 %	+0.1/0.0 %

^aReference: Quasi-static calculation with $\Delta t_n = 1/4$ sec

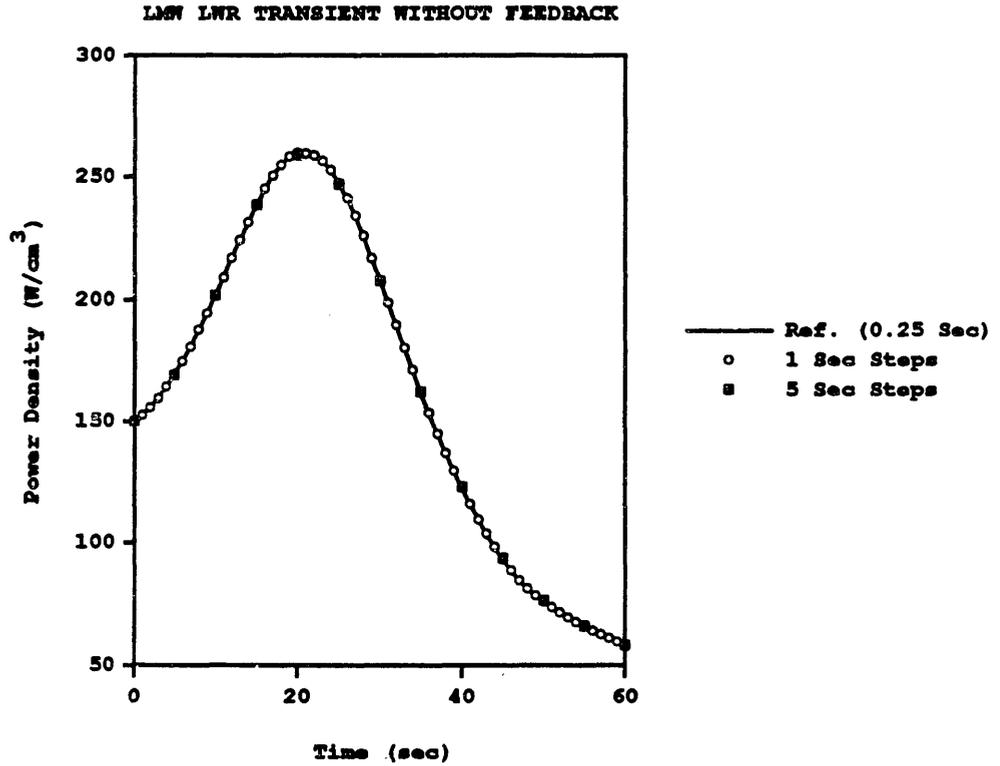


Figure 5-5: Power density vs. time for the 3-D LMW problem without feedback.

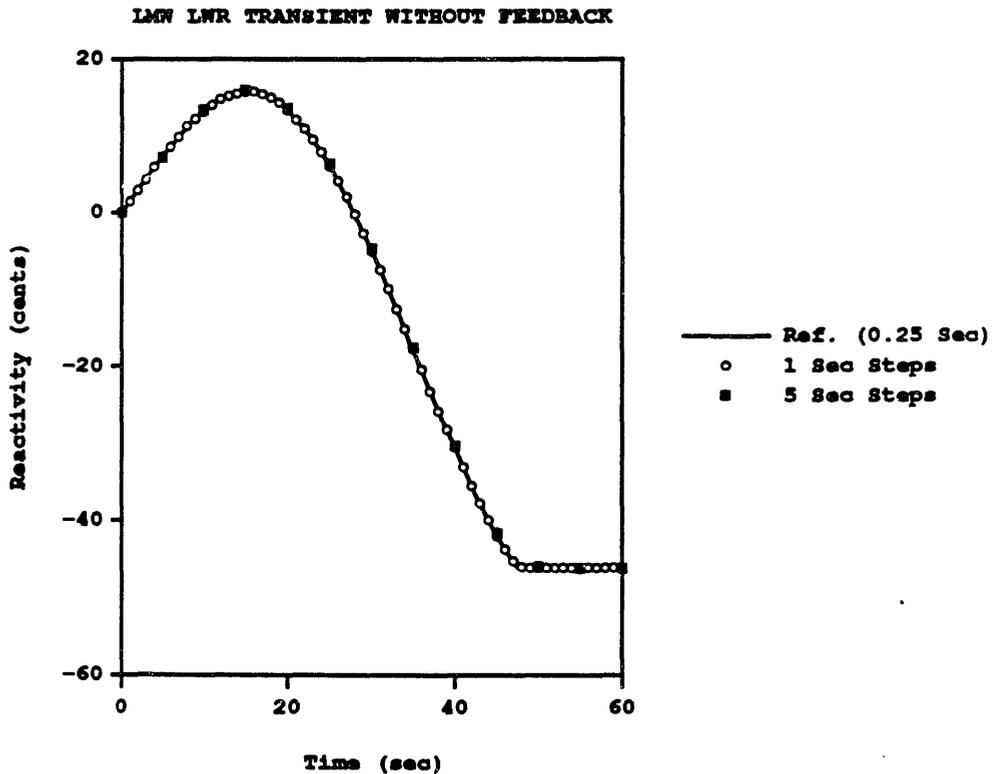


Figure 5-6: Reactivity vs. time for the 3-D LMW problem without feedback.

Table 5.15: A comparison of the mean power density (W/cm^3) for solutions of the 3-D LMW Test Problem without feedback for several nodal codes.

Time (sec)	QUABOX [L-2]	CUBBOX [L-2]	QUANDRY [S-2]	CONQUEST	Ref. ^a
0.0	150.0	150.0	150.0	150.0	150.0
5.0	168.7(-0.4%)	168.8(-0.4%)	169.1(-0.2%)	169.2(-0.1%)	169.4
10.0	200.2(-0.9%)	201.1(-0.5%)	202.0(0.0%)	201.8(-0.1%)	202.0
20.0	260.5(0.0%)	260.0(-0.2%)	262.2(+0.6%)	259.3(-0.5%)	260.5
30.0	213.6(+1.8%)	211.3(+0.7%)	210.8(+0.4%)	207.8(-1.0%)	209.9
40.0	127.5(+2.9%)	125.5(+1.3%)	123.0(-0.7%)	122.3(-1.3%)	123.9
50.0	78.6(+2.7%)	77.1(+0.8%)	75.7(-1.0%)	75.8(-0.9%)	76.5
60.0	60.3(+2.9%)	58.9(+0.5%)	57.9(-1.2%)	58.2(-0.7%)	58.6

^aRichardson extrapolation of CUBBOX results [S-2]

A comparison of the calculational results with QUABOX [L-2], CUBBOX [L-2], and QUANDRY [S-2] is presented in Table 5.15. The QUABOX and CUBBOX solutions use a time step of 1/8 second. The QUANDRY solution employs a 10 cm axial mesh (to minimize cusping effects) and a time step of 1/4 second. The CONQUEST solution is a quasi-static calculation with 1/4 second time steps. The reference has been obtained by a Richardson extrapolation of CUBBOX solutions but is not considered to be more accurate than one percent [S-2]. This comparison indicates that the CONQUEST, QUANDRY, and CUBBOX solutions have a maximum error in mean power density of about 1 % while the QUABOX solution has a maximum error of nearly 3 %. The execution times are available only for a calculation with a time step of 1 second. The calculation times for CONQUEST, QUANDRY, QUABOX, CUBBOX are 115, 86, 108, and 69 seconds, respectively. Note that the execution time reported for CONQUEST is for a quasi-static calculation which contains considerable time for reactivity calculation and point kinetics solution with a shape-update at every time step. An additional, fully-implicit calculation without the quasi-static option was performed resulting in an execution time of 83 seconds. This indicates that roughly 25 % of the calculation time is devoted to the quasi-static option.

The mean power density and reactivity versus time for calculations with and

without the cusping correction using 1/4 second time steps are given in Figures 5-7 and 5-8. The volume-averaging of the cross sections results in an over-prediction of the rod's "worth" as the rod traverses the node. As a result, the reactivity and the power are under-predicted. This leads to a maximum error in the mean power density of about -5 %. The calculation employing the cusping correction, however, displays no discernible cusping effects, even in the reactivity which is generally very sensitive to the control rod cusping.

The 3-D LMW problem results indicate that the quasi-static polynomial method can provide accurate results using large node spacings and shape-update time steps. The impact of thermal-hydraulic feedback will be examined next.

5.4.2 The 3-D LMW Problem with Thermal-Hydraulic Feedback

The 3-D LMW problem has been combined with WIGL thermal-hydraulic parameters which are representative of an operating PWR [S-2] and are given with the problem description in Appendix B. An examination of the steady-state calculation, the transient calculation, and the control rod cusping effects will be presented.

The static calculations were performed with the same node spacings as for the problem without feedback and the results are summarized in Table 5.16 (the assembly power densities are given in Appendix C). The reference is the CONQUEST calculation with 5 cm radial mesh and 10 cm axial mesh. Note that CONQUEST requires nearly the same number of discontinuity factor updates and outer iterations as the non-feedback problem. The nodal and assembly errors are also nearly the same as the non-feedback problem. Calculation of the fuel and coolant temperatures increases the calculation time by 20 to 30 %.

The transient calculations were performed using three different shape-update steps: 1/4 second, 1 second, and 5 seconds with one, two, and five reactivity/thermal-hydraulic updates per shape update, respectively. In addition, the discontinuity factors were updated every 4 time steps for the 1/4 second and 1 second time step calculation and every time step for the 5 second time step calculation. The results of the calculations are presented in Table 5.17 and Figures 5-9 and 5-10. As for the

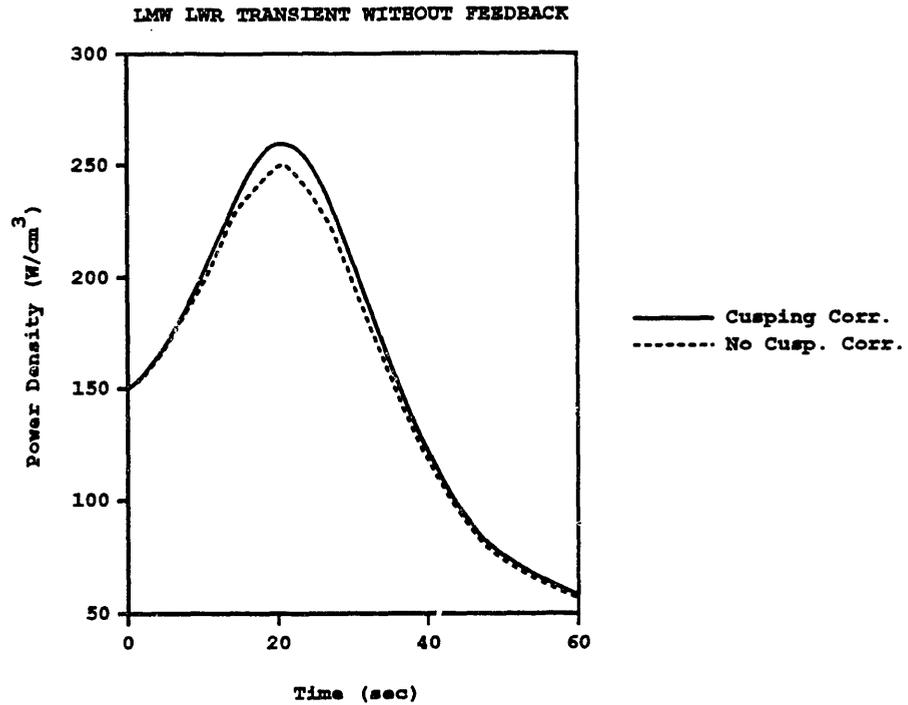


Figure 5-7: Power density vs. time for the 3-D LMW problem without feedback demonstrating the cusping correction.

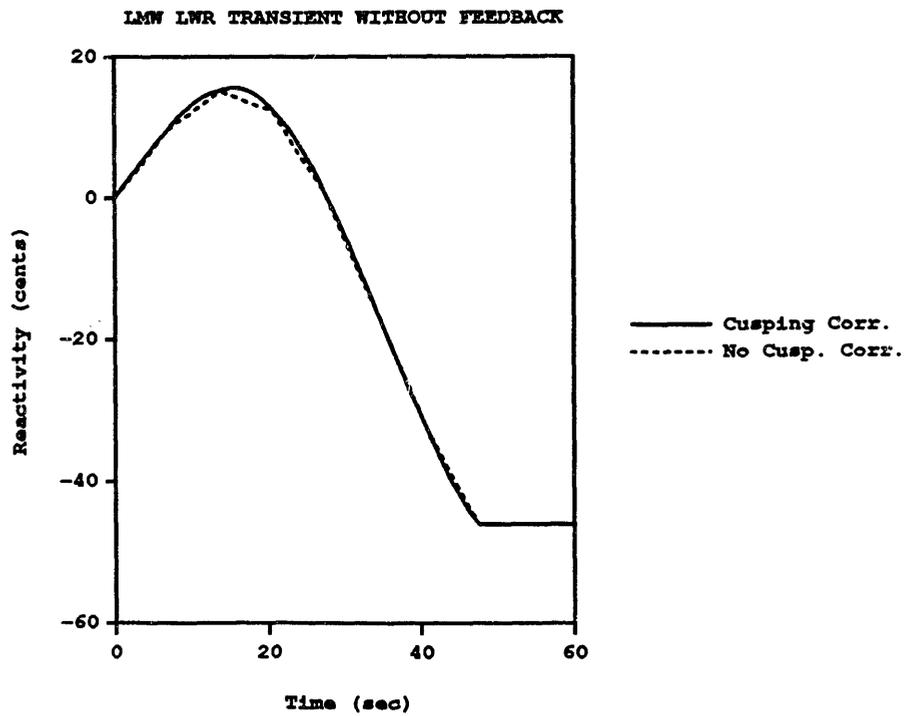


Figure 5-8: Reactivity vs. time for the 3-D LMW problem without feedback demonstrating the cusping correction.

Table 5.16: A summary of the CONQUEST static results for the 3-D LMW problem with feedback.

	Node Size		
	20 × 20 × 20 cm	10 × 10 × 10 cm	5 × 5 × 10 cm
Nodes (1/8 Core)	200	1480	5800
D.F. Updates	8	9	15
Outer Iterations	24	28	48
Eigenvalue	0.983420	0.983194	0.983166
$\epsilon_{\max}(\text{node}, \%)$	-1.24	+0.18	ref.
$\bar{\epsilon}(\text{node}, \%)$	0.33	0.03	ref.
$\epsilon_{\max}(\text{assembly}, \%)$	-0.61	-0.07	ref.
$\bar{\epsilon}(\text{assembly}, \%)$	0.15	0.01	ref.
CPU Time ^a (sec)	18.9	132.8	888.

^aDEC VS3100 M38

Table 5.17: Total power vs. time for the 3-D LMW problem with feedback and different shape-update time steps.

Time	Time Step Size, Δt_n		
	5 sec	1 sec	1/4 sec ^a
0.0	184.8	184.8	184.8
5.0	192.5 (+0.3%)	192.0 (0.0%)	192.0
10.0	195.9 (+0.3%)	195.3 (0.0%)	195.2
20.0	192.8 (-0.1%)	192.9 (0.0%)	192.9
30.0	179.6 (-0.2%)	179.6 (-0.2%)	179.9
40.0	163.1 (-0.3%)	163.3 (-0.1%)	163.5
50.0	156.8 (+0.3%)	156.2 (-0.1%)	156.4
60.0	155.8 (-0.1%)	155.9 (-0.1%)	156.0
CPU Time ^b (sec)	125.	292.	741.

^aReference

^bDEC VS3100 M38

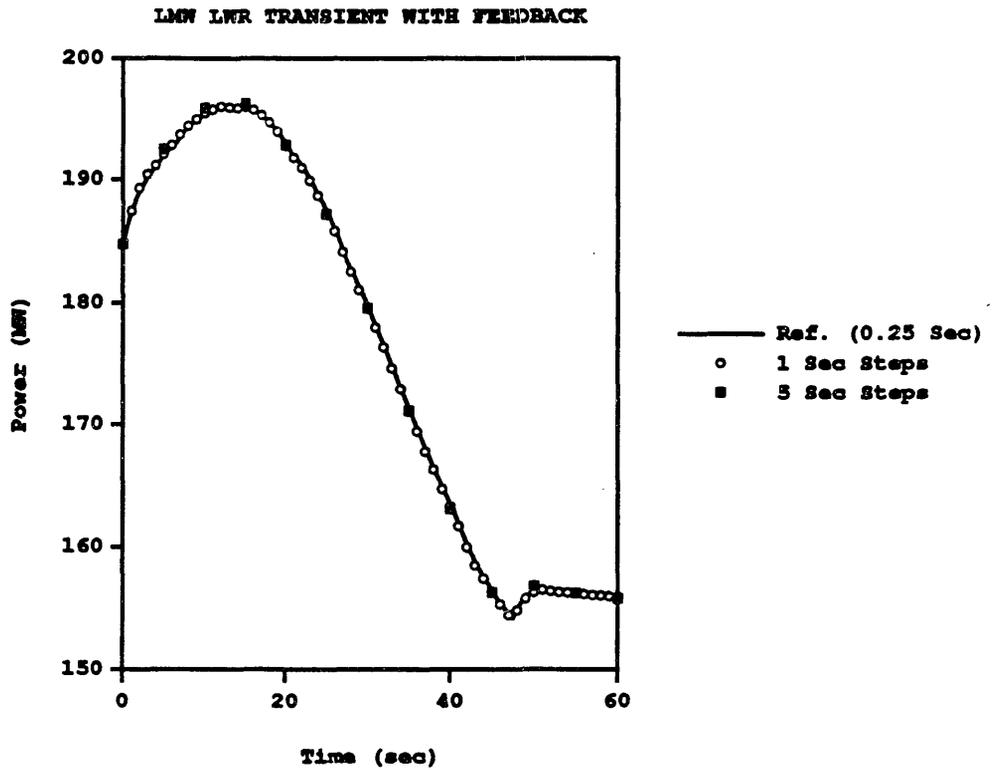


Figure 5-9: Total power vs. time for the 3-D LMW problem with feedback.

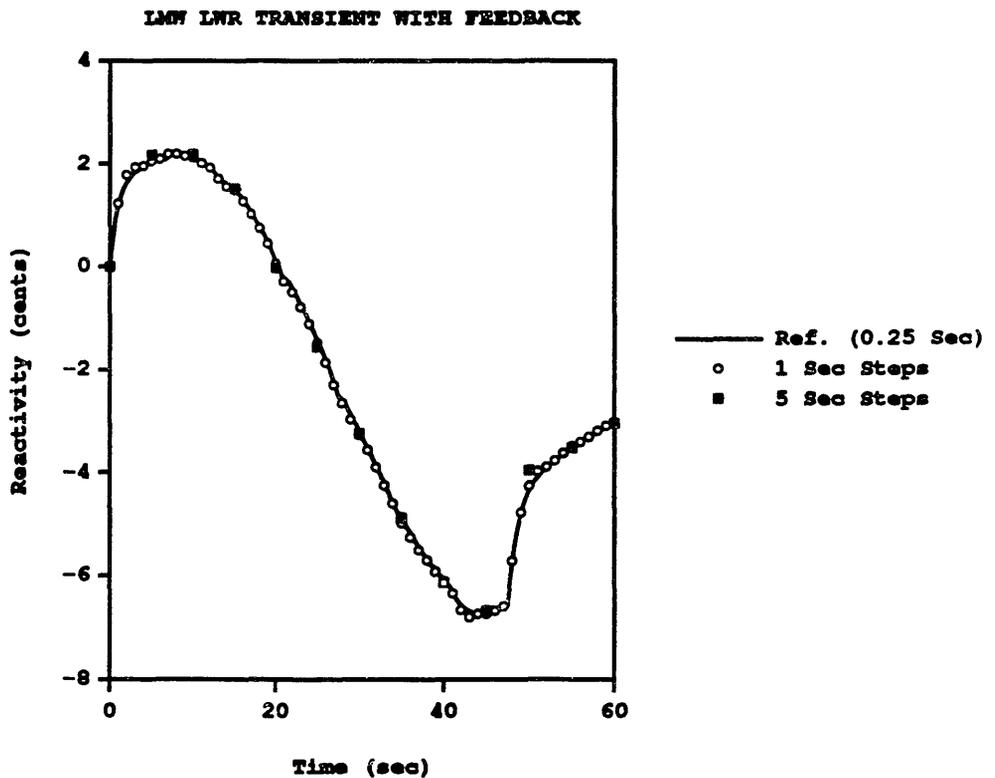


Figure 5-10: Reactivity vs. time for the 3-D LMW problem with feedback.

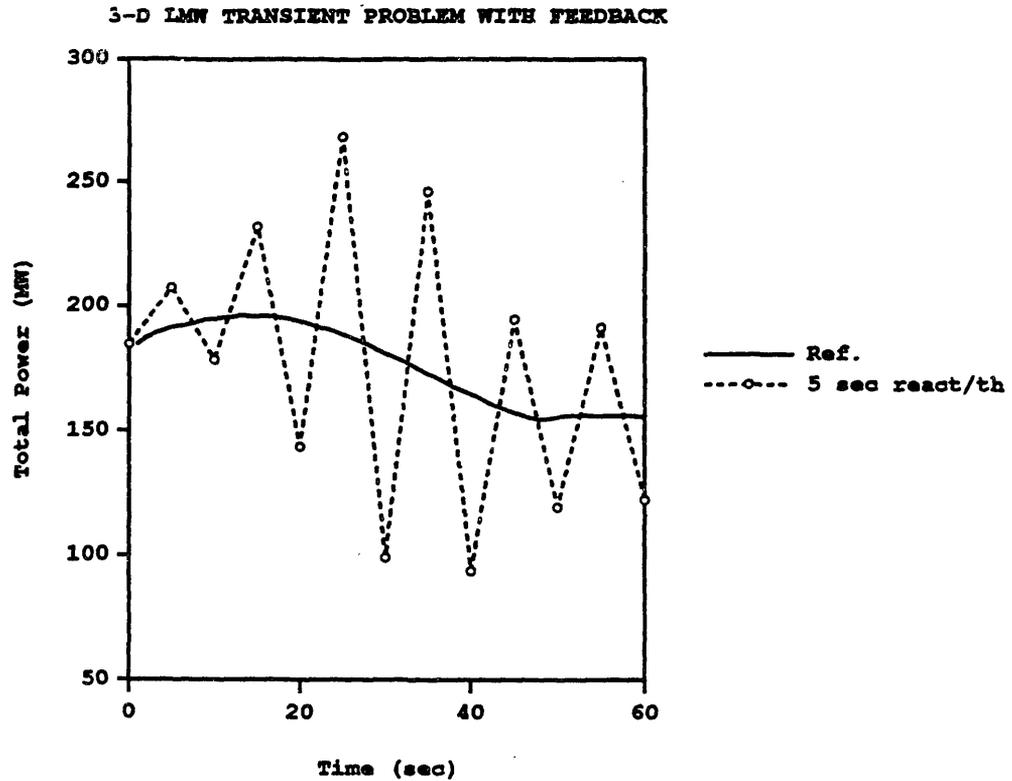


Figure 5-11: The 3-D LMW transient with feedback using 5 second shape and reactivity/thermal-hydraulic steps.

non-feedback problem, excellent results have been obtained with all time steps. An analysis of the nodal and assembly power densities shows that the largest errors are isolated to nodes which contain moving rods while the error in the remaining nodes is quite small.

Note that the reactivity/thermal-hydraulic calculations between the shape updates are essential in obtaining accurate results. A calculation employing 5 second shape and reactivity/thermal-hydraulic steps generates severe over-shoots in power, as shown in Figure 5-11. This is caused by the tandem sequence in which the neutronic and thermal-hydraulic calculations are performed. The reactivity is computed using cross sections from the previous thermal-hydraulics calculation. Thus, when we are in a power increase, the temperatures are too low, giving a reactivity which is too high. The power then increases rapidly resulting in a large increase in the temperatures which, in turn, gives a low value for the reactivity. The result is an

Table 5.18: Total power vs. time for the 3-D LMW problem with feedback, CONQUEST and QUANDRY solutions without cusping correction.

Time (sec)	QUANDRY [S-2]	CONQUEST
0.0	184.8	184.8
5.0	191.7	191.7
10.0	194.3	194.1
20.0	193.4	193.2
30.0	179.0	179.1
40.0	163.2	163.4
50.0	156.3	156.3
60.0	156.0	155.9

oscillation about the reference power.

A comparison of the calculations, using 1/4 second time steps, with and without the cusping correction are presented in Figures 5-12 and 5-13. These figures show that the cusping effects are much more significant than those in the non-feedback calculation. The cusping correction shows only slight distortions when the rod leaves one node and enters another. A comparison of the 1/4 second time step calculation without cusping correction between CONQUEST and QUANDRY is given in Table 5.18. The results are nearly identical, demonstrating that quartic polynomial method has accuracy which is comparable to that of the Analytic Nodal Method.

5.5 The LRA BWR Transient Problems

The LRA 2-D [A-2] and 3-D [A-3] benchmark problems represent a BWR with two neutron energy groups, two delayed precursor families and Doppler feedback with an adiabatic heatup model. The highly simplified BWR has a two-zone core consisting of 312 fuel assemblies ($15 \times 15 \times 300$ cm). The core is surrounded radially and axially by a 30 cm water reflector. Several of the control blades, represented as smeared absorbers in four adjacent assemblies, are withdrawn resulting in large local flux perturbations.

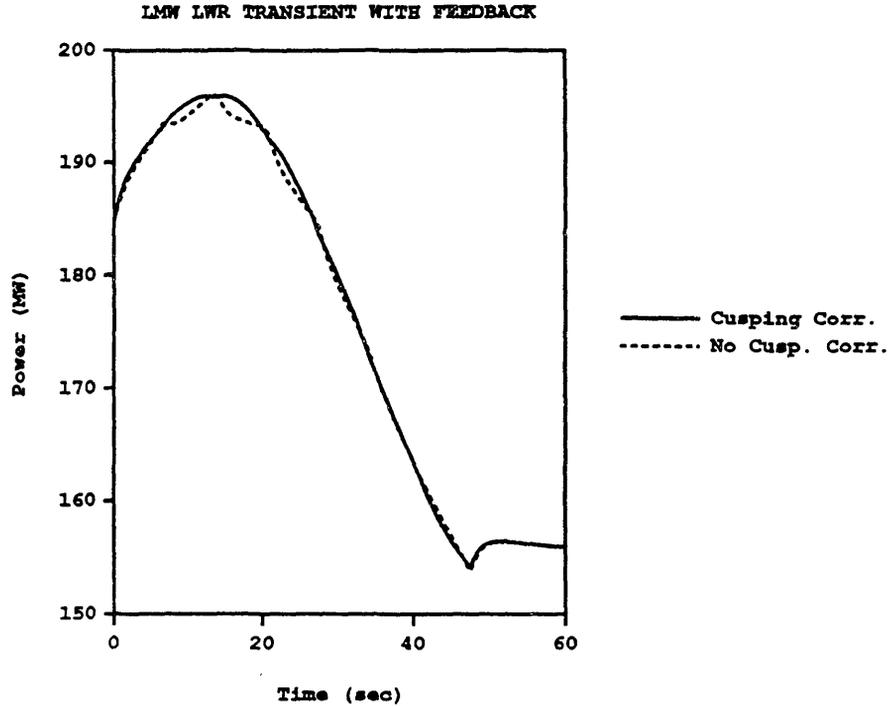


Figure 5-12: Power density vs. time for the 3-D LMW problem with feedback demonstrating the cusping correction.

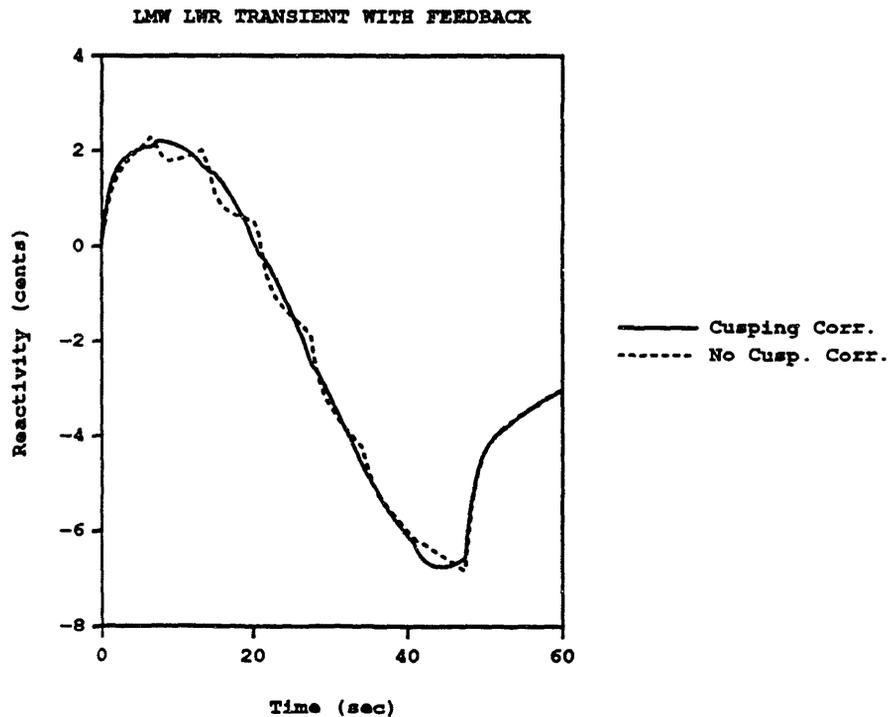


Figure 5-13: Reactivity vs. time for the 3-D LMW problem with feedback, demonstrating the cusping correction.

Table 5.19: A summary of the 2-D LRA problem static results for 15×15 cm nodes for several nodal codes.

	CONQUEST	QUAGMIRE [Z-1]	QUANDRY [S-2]
Outer Iterations	22	24	41
Eigenvalue ^a	0.996329	0.996329	0.99641
$\epsilon_{\max}(\text{node}, \%)$	+1.36	+1.41	-0.19
$\bar{\epsilon}(\text{node}, \%)$	0.40	0.42	0.07
CPU Time ^b (sec)	4.3	8.4 ^c	4.5 ^d

^aReference: 0.99636

^bDEC VS3100 M38

^cActual execution time: 6.5 sec. SGI 4D/210 (D.P.), 1/4 core

^dActual execution time: 2.7 sec. IBM 370/168

The transient is initiated from low power by the removal of an asymmetrically placed control blade at a speed of 150 cm/s resulting in a super-prompt critical configuration. The transient calculations are performed using quarter-core symmetry so that four control blades are actually being removed from the entire core. This transient is extremely difficult since the reactor power spans approximately 10 orders of magnitude during the transient with large spatial changes during the transient. The complete problem specifications are given in Appendix B.

5.5.1 The 2-D LRA Problem

The 2-D LRA problem was solved using assembly-sized meshes, 15×15 cm, and eighth-core symmetry. A summary of the static results are presented in Table 5.19 along with QUAGMIRE [Z-1] and QUANDRY [S-2] results. QUAGMIRE is a quartic polynomial code developed by Zerkle [Z-1] which is based upon polynomial equations which are similar to those of CONQUEST. The reference solution is a 16 node per assembly calculation by Shoher [A-3].

As expected, the CONQUEST and QUAGMIRE results are nearly identical with a maximum nodal error of about 1.4 percent which is quite reasonable for such a severe problem. The errors in the assembly power densities from QUANDRY, however, are

significantly smaller than those of CONQUEST and QUAGMIRE. The CONQUEST and QUANDRY execution times are comparable and are smaller than that of QUAGMIRE. The difference in execution times between CONQUEST and QUAGMIRE is most likely the result of the different expansion coefficient solution methods, as discussed in Chapter 2. A comparison of the normalized power distributions for several different node spacings are given in Appendix C

The transient problem was solved using the quasi-static method and fully-implicit differencing schemes. The following three different temporal meshes were used:

<u>92 Step</u>		<u>329 Step</u>		<u>1000 Step</u>	
<u>Interval</u>	<u>Δt_n</u>	<u>Interval</u>	<u>Δt_n</u>	<u>Interval</u>	<u>Δt_n</u>
$0 \leq t \leq 0.8$	100 ms	$0 \leq t \leq 0.8$	25 ms	$0 \leq t \leq 1.0$	10 ms
$0.8 \leq t \leq 1.0$	50 ms	$0.8 \leq t \leq 1.0$	10 ms	$1.0 \leq t \leq 1.3$	1 ms
$1.0 \leq t \leq 1.4$	10 ms	$1.0 \leq t \leq 1.3$	3 ms	$1.3 \leq t \leq 1.5$.5 ms
$1.4 \leq t \leq 1.5$	5 ms	$1.3 \leq t \leq 1.45$	1.5 ms	$1.5 \leq t \leq 2.0$	5 ms
$1.5 \leq t \leq 2.0$	50 ms	$1.45 \leq t \leq 2.0$	15 ms	$2.0 \leq t \leq 3.0$	10 ms
$2.0 \leq t \leq 3.0$	100 ms	$2.0 \leq t \leq 3.0$	15 ms		

The 329 and 1000 step temporal meshes were chosen so that the calculations would match published QUANDRY solutions [S-2]. The 92 step mesh was chosen to test the quasi-static solution procedure. The results of three fully-implicit calculations are presented in Table 5.20. A comparison of the 1000 step results indicate that although there are large spatial changes in the flux distribution, several time steps may be performed between discontinuity factor updates without causing significant errors. The 329 step solution is relatively close to the 1000 step solution. The normalized power densities and fuel temperatures for several times of interest are given in Appendix C. In addition, the plots of the mean power density and fuel temperature are given in Figures 5-14 and 5-15.

The quasi-static method was used to solve the 2-D LRA problem with 92 and 329 time steps. The results are shown in Table 5.21 along with a 92 step fully-implicit calculation. The quasi-static method does lead to increased accuracy, but also to increased execution time. The reason that the 92 step calculation does not lead to a reduction in execution time is that as the time step size increases, more iterative

work is required to compute the new shape function. Thus, the quasi-static method does not provide a significant benefit for severe, super-prompt critical transients in which small time steps are required for the calculation of the shape function. The fully-implicit procedure is recommended for such transients. A comparison of the CONQUEST results with other nodal methods are presented in Table 5.22.

5.5.2 The 3-D LRA Problem

The 3-D LRA transient problem is more severe than the 2-D problem because of larger radial flux tilts and axial shifts. The static calculations for three different mesh sizes are given in Table 5.23 where the smallest node size is used as the reference (a comparison of the power distributions are given in Appendix C). A comparison of the static calculation results of several nodal codes is given in Table 5.24.

This transient problem has proven to be very difficult because of stability problems. There are two causes of the instabilities in this calculation. The first problem occurs at the external boundaries where the fluxes and currents are very small because of the large reflector. The difficulty occurs because the discontinuity factors which are to be computed at the surface involve the ratio of the surface current to the surface fluxes, which are both small. Round-off effects cause these discontinuity factors to become absurd or result in an attempt to divide by a zero node-averaged flux. This difficulty has been overcome by simply setting the surface discontinuity factors to unity.

The other difficulty is that the corrected finite-difference equations for the reflector nodes may not be diagonally dominant because of the values of the discontinuity factor ratios. Simply setting the discontinuity factor ratios in the reflector region to unity resulted in rather large errors in the core power distribution because the control blade which has been removed from the core is near the reflector. Alternate procedures for ensuring the diagonal dominance of the corrected finite-difference equations are rather difficult to obtain.

By careful selection of the frequency of the discontinuity factor updating and a relatively tight convergence criterion (10^{-4}), a transient solution was obtained using

Table 5.20: A summary of the CONQUEST results for 2-D LRA transient using fully-implicit time differencing.

Number of time steps	329	329	1000	1000	Ref. ^a
Time steps per D.F. update	4	1	10; $t < 1.4$ 4; $t > 1.4$	1	—
Time to first peak (s)	1.429	1.429	1.438	1.438	1.436
Power at first peak (W/cm ³)	5623	5598	5505	5490	5411
Power at second peak (W/cm ³)	807	804	798	791	784
Power at $t = 3.0$ s	99.5	99.3	99.1	98.6	96.2
Average fuel temperature at $t = 3.0$ s	1113	1113	1107	1104	1087
Peak fuel temperature at $t = 3.0$ s	3042	3042	3023	3014	2948
CPU Time ^b (sec)	163.	324.	379.	811.	7030.

^aShober's fine temporal and spatial mesh calculation [A-3]

^bDEC VS3100 M38

Table 5.21: A summary of the CONQUEST results for 2-D LRA transient using the quasi-static method.

Number of time steps	92 ^a	92	329	Ref. ^b
Time steps per D.F. update	1	1	4	—
Time to first peak (s)	1.370	1.430	1.438	1.436
Power at first peak (W/cm ³)	5439	5589	5515	5411
Power at second peak (W/cm ³)	743	802	804	784
Power at $t = 3.0$ s	97.5	99.2	98.9	96.2
Average fuel temperature at $t = 3.0$ s	1154	1121	1108	1087
Peak fuel temperature at $t = 3.0$ s	3142	3074	3034	2948
CPU Time ^c (sec)	152.	186.	209.	7030.

^aFully-implicit calculation, for comparison

^bShober's fine temporal and spatial mesh calculation [A-3]

^cDEC VS3100 M38

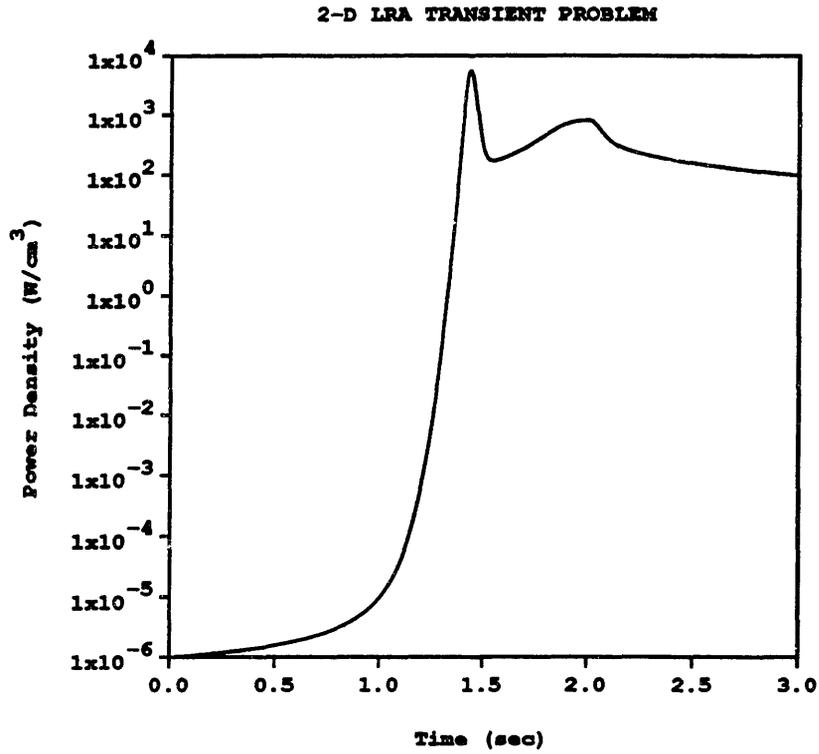


Figure 5-14: Power vs. time for the 2-D LRA transient problem.

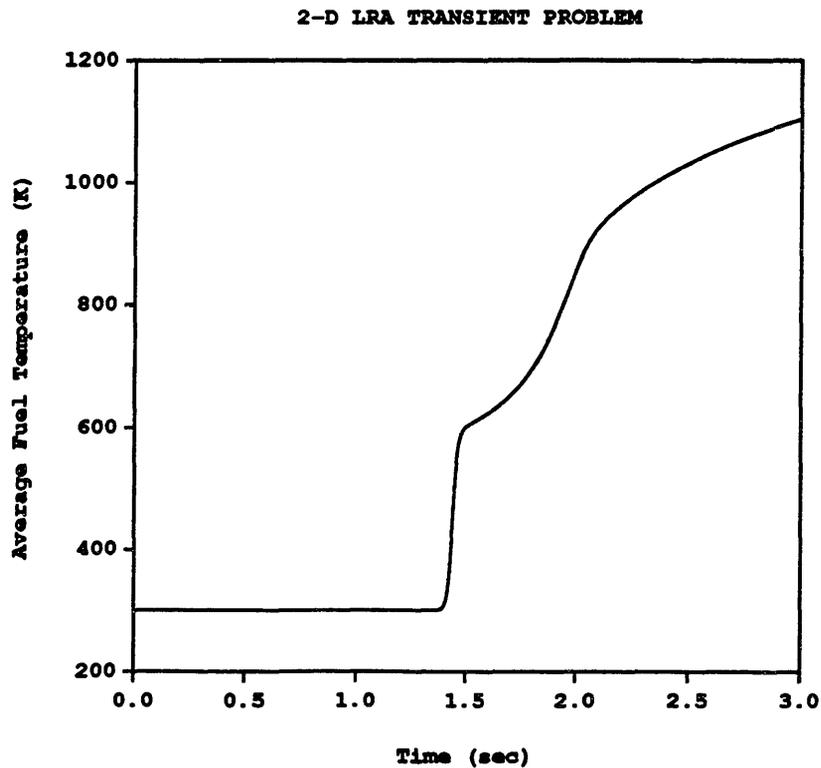


Figure 5-15: Fuel temperature vs. time for the 2-D LRA transient problem.

Table 5.22: A comparison of nodal solutions to the 2-D LRA transient problem.

	2DTD [S-11]	CUBBOX [A-2]	IQSBOX [A-2]	QUANDRY [S-2]	CONQUEST	Ref. ^a
Initial eigenvalue	0.99655	0.99633	0.99631	0.99641	0.99633	0.99636
Time to first peak (s)	1.426	1.421	1.445	1.435	1.437	1.436
Power at first peak (W/cm ³)	5552	5734	5451	5473	5505	5411
Power at second peak (W/cm ³)	815	~ 830	~ 800	797	798	784
Power at t = 3.0 s	97.0	~ 60	~ 100	97.5	99.1	96.2
Average fuel temperature at t = 3.0 s	1127	1070	1127	1108	1107	1087
Peak fuel temperature at t = 3.0 s	3112	2925	2989	3029	3023	2948
Number of time steps	1000	1200	522	1000	1000	2600
CPU Time (sec)	210. IBM 370/168	180. IBM 360/91	255. CYBER 175	188. IBM 370/168	379 DEC VS3100	1661 IBM 370/195
Computer						

^aShober's fine temporal and spatial mesh calculation [A-3]

Table 5.23: A summary of the CONQUEST 3-D LRA problem static results.

	15×15 $\times 25(15)$ cm	7.5×7.5 $\times 12.5(7.5)$ cm	5×5 $\times 12.5(7.5)$ cm
Nodes (1/8 Core)	1056	8096	17952
D.F. Updates	7	8	9
Outer Iterations	21	34	41
Eigenvalue	0.996361	0.996391	0.996368
$\epsilon_{\max}(\text{node},\%)$	-1.37	+0.20	ref.
$\bar{\epsilon}(\text{node},\%)$	0.40	0.04	ref.
$\epsilon_{\max}(\text{assembly},\%)$	+1.17	+0.19	ref.
$\bar{\epsilon}(\text{assembly},\%)$	0.22	0.03	ref.
CPU Time ^a (sec)	70.	762.	1988.

^aDEC VS3100 M38

Table 5.24: A comparison the coarse mesh 3-D LRA problem static results.

	CONQUEST	QUAGMIRE [Z-1]	QUANDRY [S-2]
Outer Iterations	21	22	35
Eigenvalue	0.996361	0.996360	0.99644
$\epsilon_{\max}(\text{node},\%)$	-1.37	+1.62	-0.38
$\bar{\epsilon}(\text{node},\%)$	0.40	0.45	0.08
CPU Time ^a (sec)	70.	202. ^b	30. ^c

^aDEC VS3100 M38

^bActual execution time: 154.7 sec. SGI 4D/210 (D.P.), 1/4 core

^cActual execution time: 18.7 sec. IBM 370/168

the following 410 time steps

<u>Time Interval</u>	<u>Δt_n</u>
$0 \leq t \leq 0.5$	25 ms
$0.5 \leq t \leq 0.6$	10 ms
$0.6 \leq t \leq 0.7$	2.5 ms
$0.7 \leq t \leq 0.8$	1.25 ms
$0.8 \leq t \leq 0.95$	1.0 ms
$0.95 \leq t \leq 1.0$	2.5 ms
$1.0 \leq t \leq 2.0$	20 ms
$2.0 \leq t \leq 3.0$	25 ms

Plots of the power and fuel temperature verses time are presented in Figures 5-16 and 5-17 and assembly-averaged power densities, planar power densities, and temperatures at several times of interest are given in Appendix C.

A comparison of the transient results of several nodal codes is presented in Table 5.25. Note that the QUANDRY calculation employed a very coarse mesh with 30×30 cm nodes. A static calculation performed with CONQUEST using 30 cm nodes resulted in large errors indicating that assembly-sized nodes may be the limit for which the quartic polynomial approximation gives accurate results. Hence, the quartic polynomial methods may not be well suited for such applications as super nodal methods [G-1] which uses four assemblies per node.

5.6 The PWR Operational Transient

The previous benchmark problems which have been analyzed, while good tests, represent relatively simple reactor models. In order to determine the accuracy of the polynomial method, and the efficiency of the quasi-static method a more realistic application is desired. The PWR operational transient discussed in this section was introduced by Jacqmin [J-2] for analysis of a nodal synthesis method. The reactor is representative of a Westinghouse 1000 MW_e pressurized water reactor. The core contains 193 fuel assemblies with dimensions of $21.591 \times 21.591 \times 360$ cm. The radial reflector is explicitly modelled and the axial reflector is represented by infinite-reflector albedo boundary conditions. The thermal-hydraulic feedback is performed using the WIGL model discussed in Chapter 4. The complete description of the reactor model is given in Appendix B.

Table 5.25: A comparison of nodal solutions to the 3-D LRA transient problem.

	CUBBOX [A-3]	IQSBOX [A-3]	QUANDRY [S-2]	CONQUEST
Number of Nodes	1936	1452	490	1452
Initial eigenvalue	0.99626	0.99624	0.99652	0.996361
Time to first peak (s)	0.906	0.894	0.906	0.905
Power at first peak (W/cm ³)	6268	5798	5739	5390
Time to second peak (s)	~ 1.5	~ 1.5	1.44	1.44
Power at second peak (W/cm ³)	~ 370	~ 400	412	431
Power at t = 3.0 s	~ 40	~ 40	70	39.9
Average fuel temperature at t = 3.0 s	~ 1000	~ 950	1033	953
Number of time steps	800	706	410	410
CPU Time (sec)	5800	2172	580	3889
Computer	AMDAHL V/6	CYBER 176	IBM 370/168	DEC VS3100

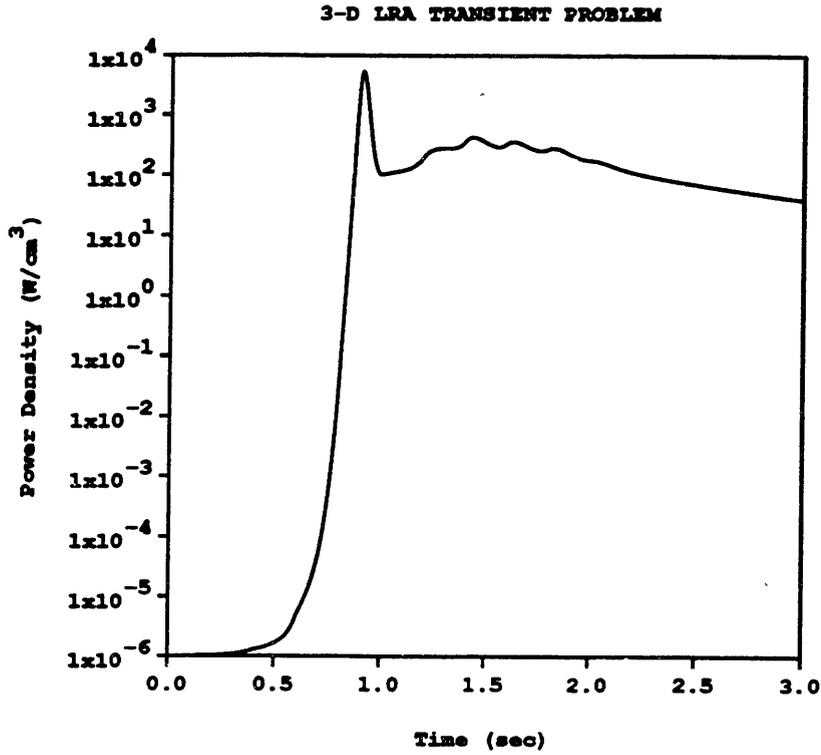


Figure 5-16: Power vs. time for the 3-D LRA transient problem.

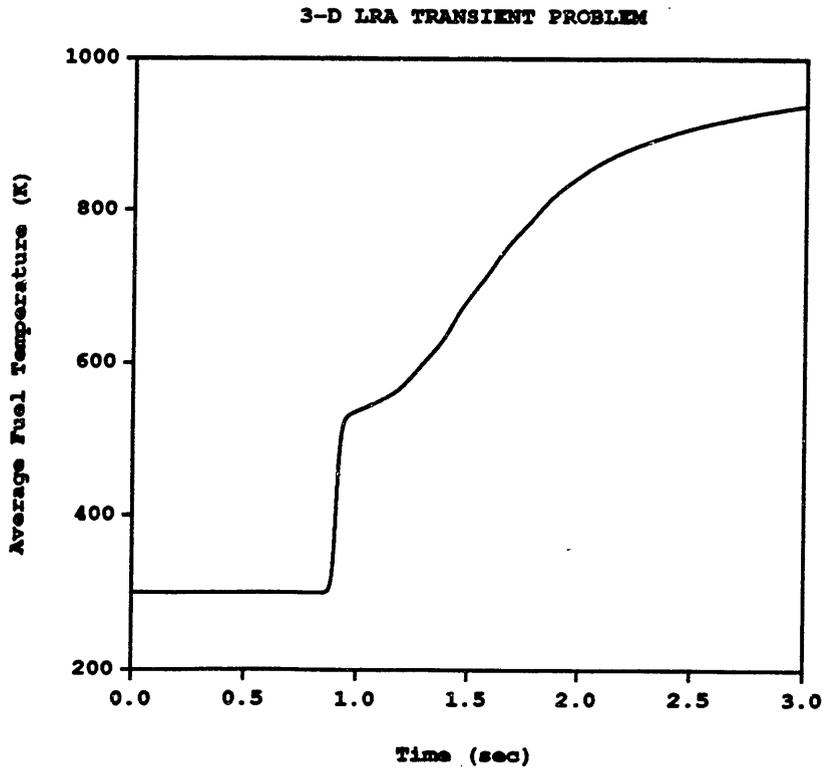


Figure 5-17: Fuel temperature vs. time for the 3-D LRA transient problem.

The reactor is initially critical at 20 % nominal power. Control rod banks C and D are partially inserted into the core. Static calculations were performed to determine the initial eigenvalue and power distribution using eighth-core symmetry. CONQUEST calculations were performed with the following node spacings:

Coarse: $21.5910 \times 21.591 \times 20$ cm
Fine: $10.7955 \times 10.7955 \times 10$ cm
Very Fine: $5.39775 \times 5.39775 \times 10$ cm

A summary of the static results are presented in Table 5.26 along with a coarse mesh calculation performed with QUANDRY. From the progression of calculations, the maximum error in the power of the finest mesh, which is considered the reference solution, is expected to less than 0.1 %. The largest errors in the coarse mesh calculation occur at nodes neighboring the reflector and are slightly larger than the errors which occurred in the previous benchmark problems. The average error, 0.69 %, indicates that the errors in most nodes, however, are quite small. The lower error in the QUANDRY calculation, -1.4 %, can be attributed to a more accurate solution in the large radial reflector. The CONQUEST execution time is roughly half that of QUANDRY which indicates the efficiency of the non-linear iteration scheme.

The transient is initiated by the removal of control rod banks C & D at a constant speed of 2 cm/s. As shown in Figure 5-18, rod bank C reaches the top of the core at $t = 60$ seconds while rod bank D continues its motion. All rod motion ceases at $t = 120$ seconds leaving rod bank D partially inserted. The transient is followed until $t = 180$ seconds when the reactor has nearly reached a new steady-state condition. A reference calculation was performed with CONQUEST using 1/4 second shape-update steps for $0 \leq t \leq 120$ seconds and 1/2 second shape updates for $120 \leq t \leq 180$ seconds and required approximately 177 minutes of CPU time. Additional calculations using doubled times steps were performed with both CONQUEST and QUANDRY. In the CONQUEST calculation the discontinuity factors were updated every time step and required 64 minutes of CPU time. The QUANDRY calculation employed its quasi-static option [J-3] and matrix updates were performed every other time step. The

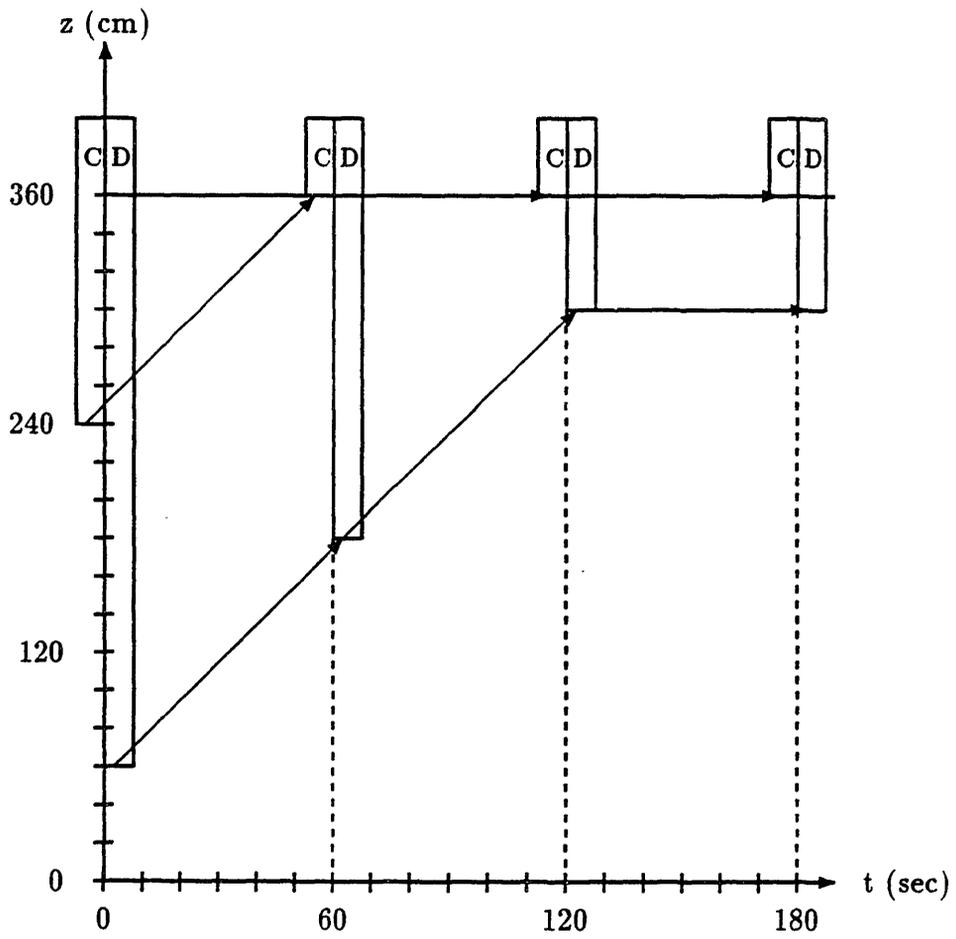


Figure 5-18: Control rod motions for the PWR operational transient.

Table 5.26: A summary of the results of the static calculation for the PWR Operational Transient.

Computer code	QUANDRY	CONQUEST	CONQUEST	CONQUEST
Spatial Mesh	Coarse	Coarse	Fine	Very Fine
Nodes (1/8 Core)	810	810	6156	23976
D.F. Updates	—	9	11	7
Outer Iterations	28	27	47	55
Eigenvalue	1.04551	1.04548	1.04523	1.04514
$\epsilon_{\max}(\text{node}, \%)$	-1.4	-3.1	-0.4	ref.
$\bar{\epsilon}(\text{node}, \%)$	0.5	0.7	0.1	ref.
$\epsilon_{\max}(\text{assembly}, \%)$	-1.2	-2.1	-0.3	ref.
$\bar{\epsilon}(\text{assembly}, \%)$	0.4	0.6	0.1	ref.
CPU Time ^a (sec)	181.	86.	796.	2940.

^aDEC VS3100 M38

QUANDRY calculation required 95 minutes of CPU time. Plots of total power and reactivity versus time for these three calculations are presented in Figures 5-19 and 5-20. The power versus time curves for all calculations lie virtually on top of one another indicating that the solutions are temporally converged. The reactivity plots are also very close with the differences being caused by cusping effects.

In order to show the efficiency and accuracy of the quasi-static method, a CONQUEST calculation was performed with shape-update steps of 5 seconds and reactivity/thermal hydraulic steps of 1 second. The power and reactivity versus time curves are presented in Figures 5-21 and 5-22. Examination of the reactivity versus time shows close agreement over most of the transient. Perturbations occur when shape updates are performed when a control rod is at a nodal interface. This is possibly related to the cusping correction or is a consequence of the large temporal mesh. The effect of these perturbations on the power versus time curve, however, are quite small. A fully implicit calculation with equivalent accuracy requires time steps of less than 1/2 second. Hence, the quasi-static option allows a ten-fold increase in shape-update time step.

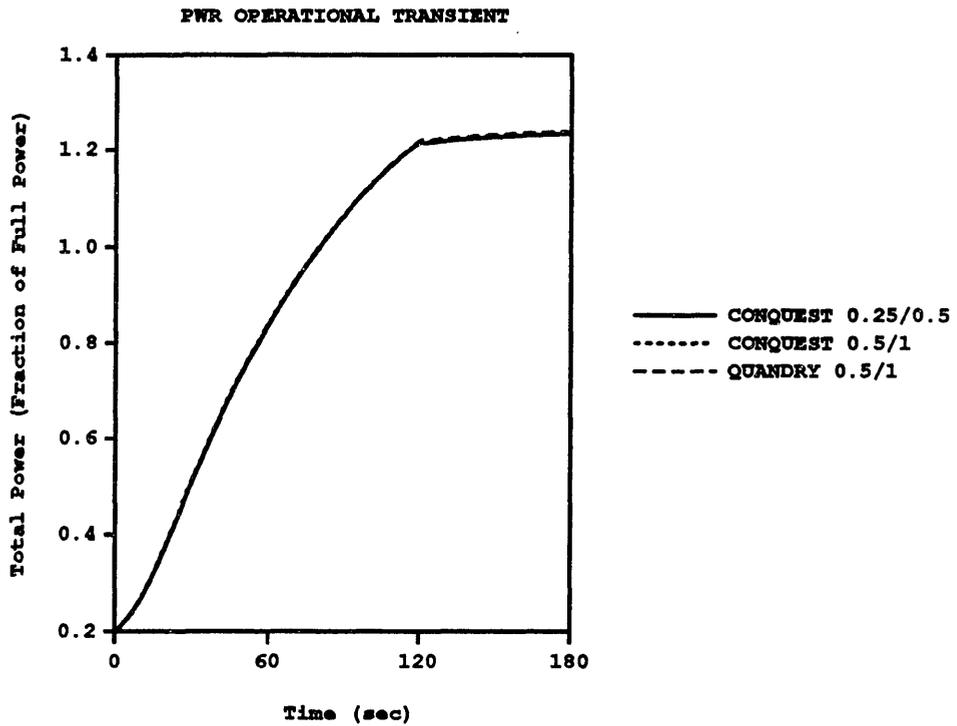


Figure 5-19: Power vs. time for the PWR operational transient demonstrating the temporal convergence of the solution.

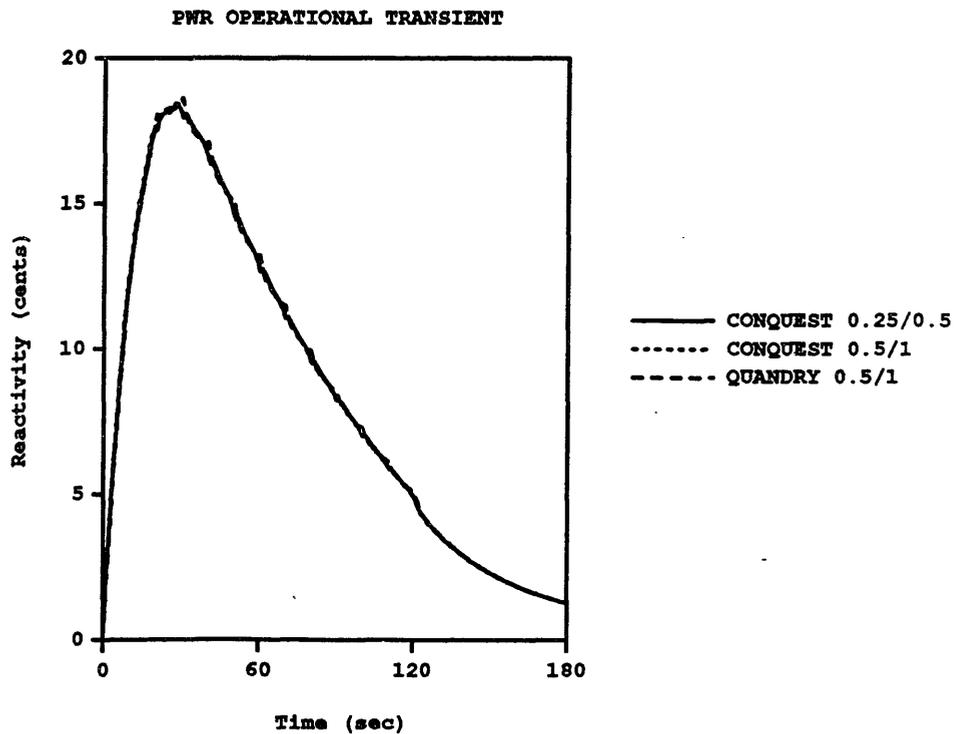


Figure 5-20: Reactivity vs. time for the PWR operational transient demonstrating the temporal convergence of the solution.

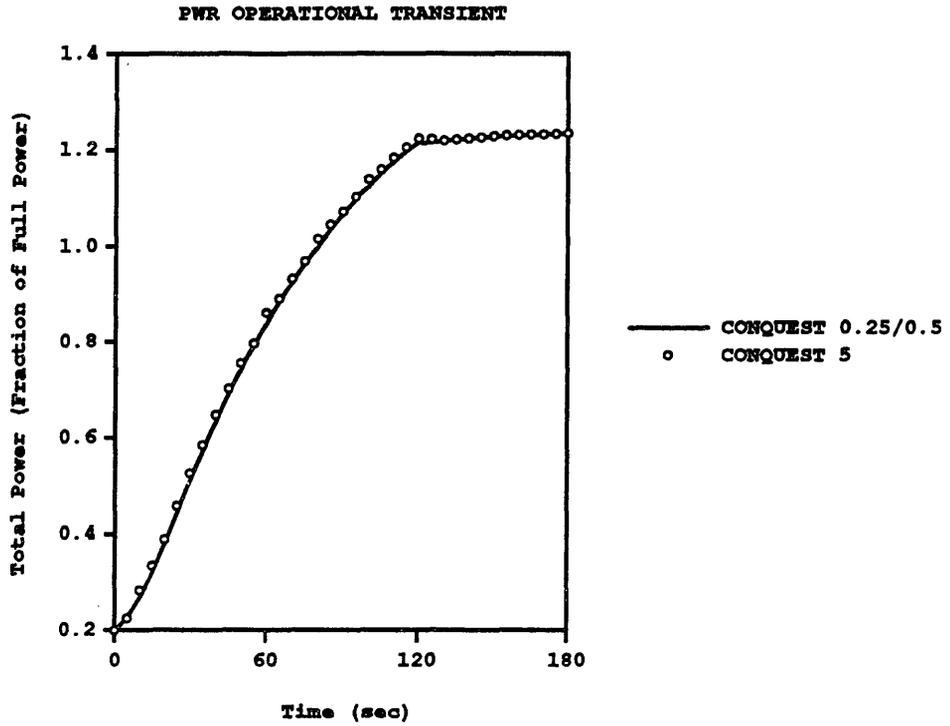


Figure 5-21: Power vs. time for the PWR operational transient, large time-step quasi-static solution.

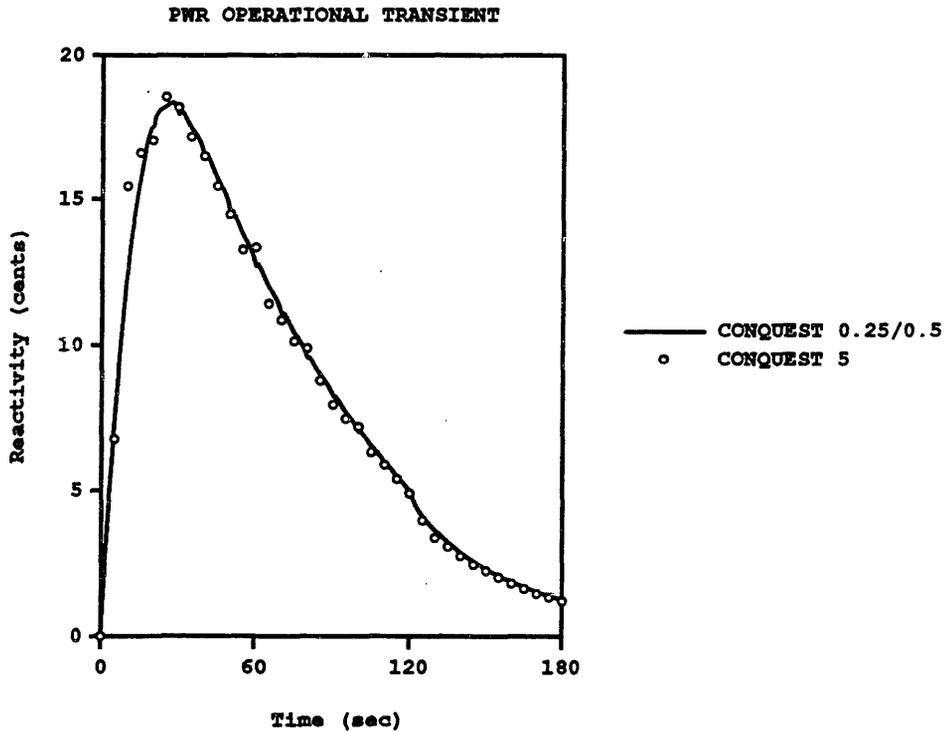


Figure 5-22: Reactivity vs. time for the PWR operational transient, large time-step quasi-static solution.

5.7 The PWR Coolant Inlet-Temperature Transient

Jacqmin [J-2] also introduced a transient problem which is driven by changing thermal-hydraulic conditions of the reactor. The reactor model is the same as previously used for the PWR operational transient. The reactor is initially in a steady-state, critical condition at nominal power of 3338 MW_{th}. All control rod banks are fully withdrawn except rod-bank D which is partially inserted (as in Figure 5-18 for $t \geq 120$ s). A transient is initiated thermally by a two-second exponential decrease in the coolant inlet temperature, from 555 K to 535 K, followed by an exponential increase to 555 K. The exact form of the perturbation is

$$T_{\text{inlet}}(t) = T_{\text{inlet}}(0) \exp(-t/\tau_1) + T_{\text{inlet}}(0) (1 - \exp(-t/\tau_2)), \quad (5.1)$$

where $\tau_1 = 2.0$ sec. and $\tau_2 = 2.206$ sec.

This transient is a good test of the neutron/thermal-hydraulic coupling since it is driven by the changing thermal-hydraulic conditions of the reactor. The initial critical condition of the reactor was determined for the coarse mesh nodalization using CONQUEST and QUANDRY. A summary of the calculations is presented in Table 5.27. The CONQUEST and QUANDRY nodal powers are compared with the QUANDRY calculation being the reference even though it is not spatially converged. The errors in the power distributions are expected to be similar to those of the initial static conditions of the PWR operational transient. A comparison of the QUANDRY and CONQUEST power distributions are presented in Appendix C

A reference calculation for the transient was performed with CONQUEST 1/16 second shape-update time steps and required 52.1 minutes of computation time. Discontinuity factor updates were performed every time step to ensure an accurate solution. Additional calculations were performed with doubled time steps (1/8 second) with CONQUEST and QUANDRY. The CONQUEST calculation had discontinuity factor updates performed every time step and required 22.6 minutes. In the QUANDRY calculation, matrix updates were performed every other time step and the calculation required 27.2 minutes. All calculations used a convergence criterion

Table 5.27: A summary of the CONQUEST and QUANDRY static calculations for the PWR coolant inlet-temperature problem.

	CONQUEST	QUANDRY
D.F. Updates	9	—
Outer Iterations	25	28
Eigenvalue	1.048269	1.048223
$\epsilon_{\max}(\text{node},\%)$	-2.2	ref.
$\bar{\epsilon}(\text{node},\%)$	0.64	ref.
$\epsilon_{\max}(\text{assembly},\%)$	-1.6	ref.
$\bar{\epsilon}(\text{assembly},\%)$	0.43	ref.
CPU Time ^a (sec)	85.	194.

^aDEC VS3100 M38

of 10^{-3} . Plots of total power and reactivity versus time for these three calculations are presented in Figures 5-23 and 5-24. The fact that these curves lie virtually on top of one another indicate that the 1/16 second time step is temporally converged.

This transient problem was also solved using 1 second shape-update time steps with reactivity/thermal-hydraulic steps of 1/8 and 1/2 seconds (requiring 9.6 and 7.3 minutes, respectively). The power and reactivity plots are shown with the reference calculation in Figures 5-25 and 5-26. Since this transient does not involve control rod motions the flux shape changes very slowly. Thus, the calculation with 1/8 second reactivity/thermal-hydraulic steps closely matches the reference. The 1/2 second reactivity/thermal-hydraulic step calculation, however, has significant error in power and reactivity. This occurs because the transient is driven by the changing thermal hydraulic conditions of the core. The 1/2 second update of the thermal-hydraulic conditions is not sufficient for this relatively quick transient.

The quasi-static method is well suited for thermal transients such as this coolant inlet-transient since the shape function is slowly varying. Since the effects of the changing thermal-hydraulic conditions on reactivity, and therefore the amplitude function, can be determined without computing the shape function, we obtain a substantial reduction in computing time over the fully-implicit procedure. In fact, for

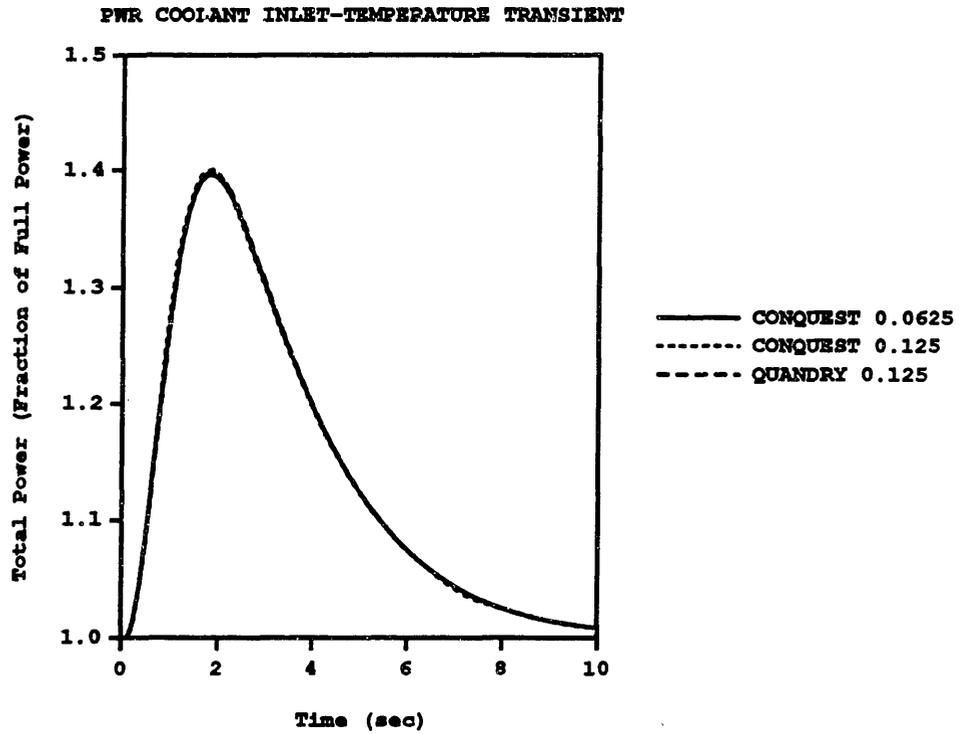


Figure 5-23: Power vs. time for the PWR coolant inlet-temperature transient demonstrating the temporal convergence.

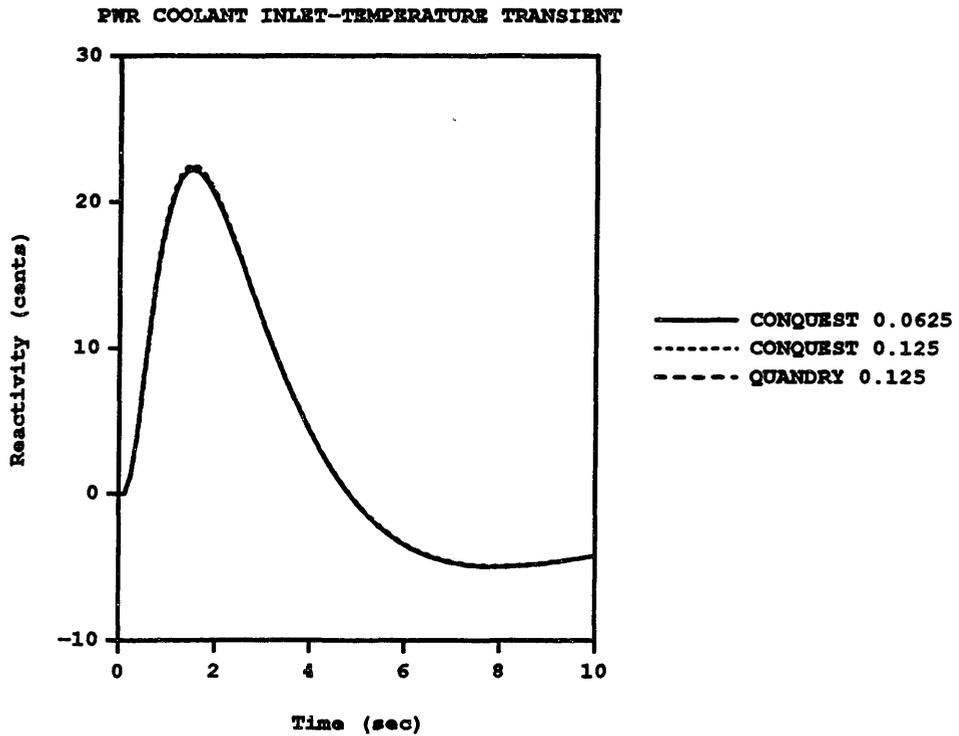


Figure 5-24: Reactivity vs. time for the PWR coolant inlet-temperature transient demonstrating the temporal convergence.

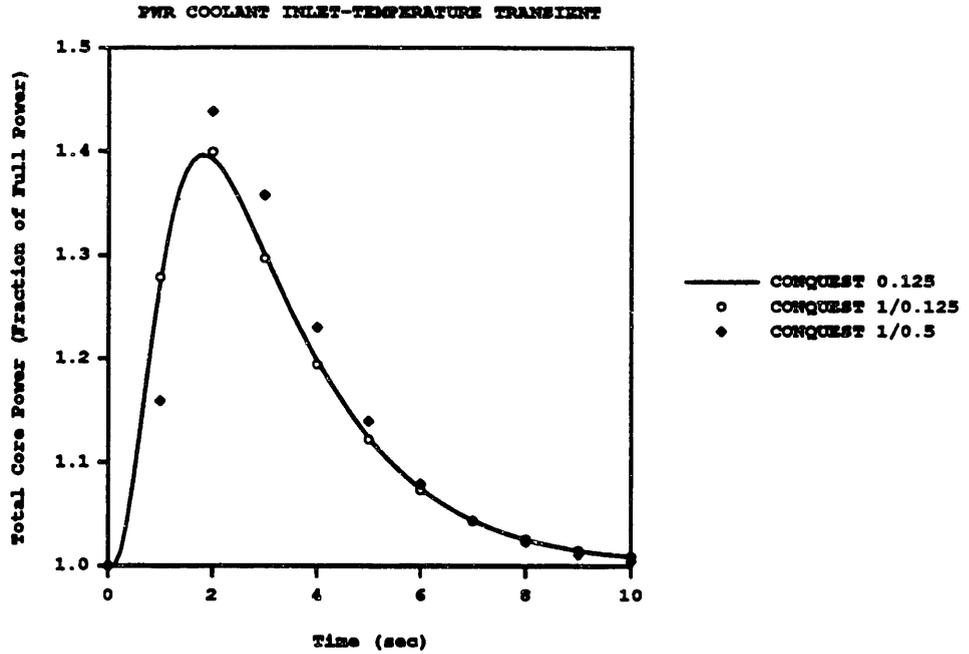


Figure 5-25: Power vs. time for the PWR coolant inlet-temperature transient, large time-step quasi-static solution.

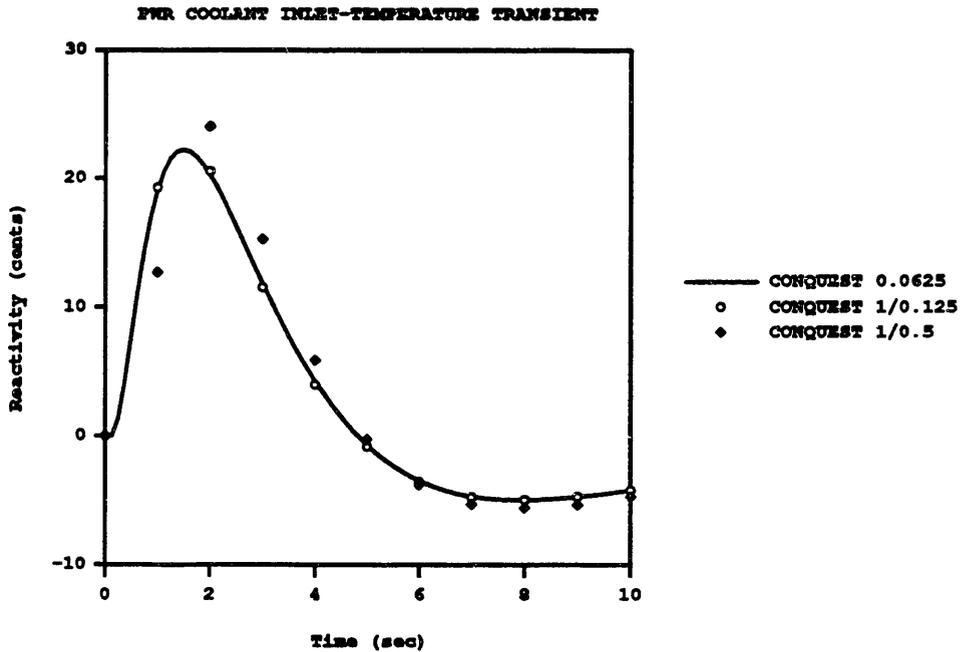


Figure 5-26: Reactivity vs. time for the PWR coolant inlet-temperature transient, large time-step quasi-static solution.

this transient, a fully-implicit method required thermal-hydraulic time steps of about 1/8 second to provided accurate results even though the flux shape changes slowly.

5.8 Summary

In this chapter the quasi-static polynomial nodal method was applied to several benchmark problems and the results compared to reference solutions. In addition, two additional PWR problems were analyzed to determine the accuracy of the method for more realistic problems. The results demonstrate that the polynomial approximation yields results which are as accurate as established nodal methods. In general the execution times for fully-implicit calculations were slightly shorter than those of the Analytic Nodal Method. When the quasi-static method is used with large time-steps a significant reduction in computation time may be obtained. The LRA problem results demonstrated that the quasi-static method does not provide much benefit for super-prompt critical transients because of the large changes in the flux shape.

The analysis of the TWIGL benchmark problem demonstrated that the discontinuity factors need not be updated every time step if the flux shape does not change significantly throughout the transient. Also, the use of the adjoint as a weight function for quasi-static calculations was shown to be less important than its use for point kinetics calculations because the shape function is updated periodically. The 3-D LMW benchmark showed the effectiveness of the quasi-static method for problems with and without thermal-hydraulic feedback. In addition, the simple control-rod cusping correction scheme was shown to be very effective and the nature of the thermal-hydraulic/neutronics coupling was studied.

In addition to the problems presented in this chapter, several additional problems were analyzed. This includes homogenous 4 and 7 group problems for which the analytic solutions could be obtained. Also, Cabral's thermal-hydraulic model was validated by the used of test problems given in reference [A-1]. All results closely matched the reference solutions.

Chapter 6

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

6.1 Overview of the Investigation

The objective of this thesis was the development of a quasi-static nodal method for the efficient solution of time-dependent, multidimensional, few-group neutron diffusion equations. The method places no restriction on the number or structure of energy groups. The nodal method also models the presence of time-varying neutron sources and includes two different thermal-hydraulic models.

In Chapter 2, a rigorous set of static equations having the finite-difference form was derived. Since the finite-difference approximation is accurate only for small mesh spacings, discontinuity factors were introduced leading to a set of equations which are formally exact. The discontinuity factors are computed during the calculation using non-linear iteration such that the corrected finite-difference scheme matches the solution of a polynomial nodal method, which provides accurate results for large mesh spacings. The polynomial method is of variable order allowing quadratic through quartic approximations. The expansion coefficients of the polynomials are obtained from continuity and weighted residual conditions.

In Chapter 3, the transient nodal equations were derived and the details of the quasi-static method were presented. First, the transient finite-difference equations which are corrected with discontinuity factors were obtained. The polynomial nodal method was then applied and, by the introduction of prompt and delayed frequencies, a form identical to the static equations was obtained. Next, the quasi-static approximation was applied to the nodal equations by factoring the flux into shape

and amplitude functions. The results of the derivation were precise mathematical equations for the point kinetics parameters in terms of the operators of the nodal method.

The numerical properties and solution methods for the static and transient equations were presented in Chapter 4. First, the properties of the static equations were discussed and methods of solving eigenvalue, source, criticality and adjoint problems were examined. Considerations in the selection of the theta required by the time integration of the spatially-dependent transient equations were presented. The solution methods for the transient equations were developed to be similar to the solution methods used for the static equations so that the non-linear iteration scheme may be easily applied. Finally, an adaptive procedure for the solution of the point kinetics equations was presented.

Applications of the quasi-static polynomial method, embodied in the CONQUEST code, were demonstrated in Chapter 5. The static and transient results for three benchmark problems and two additional PWR problems were presented. The accuracy of the polynomial nodal method was found to be consistent with established nodal methods having comparable execution times. The application of the quasi-static method, however, resulted in substantially reduced execution times for most problems. The analysis of the LRA transient problem showed that the quasi-static method does not offer any advantage for prompt super-critical transients in which the flux shape exhibits large, rapid changes. Stability problems were encountered in the 3-D LRA transient problem.

6.2 Conclusions

The application of the polynomial nodal method has lead to the following general conclusions:

1. The quartic polynomial nodal method provides accurate results. The errors in normalized power distributions are comparable to those of other nodal methods.

2. The non-linear iteration scheme reduces the storage requirements and leads to efficient solutions. No difficulties were encountered in generalizing the non-linear method to transient calculations. Stability problems, however, were encountered in problems with large reflectors.
3. The application of the quasi-static scheme substantially reduces the computation times of many problems when compared to the conventional space-time neutronics with fully-implicit procedures.
4. In problems involving thermal-hydraulic feedback the quasi-static method allows the effects of changing thermal-hydraulic conditions to be incorporated into the reactivity without having to update the shape function.
5. The quasi-static method does not lead to a significant reduction in computation time for severe transients in which the shape function changes rapidly. In the calculation of severe transients an increase in the shape update time step results in a substantial increase in the iterative solution time. The fully-implicit method is recommended for such transients.

6.3 Recommendations for Future Research

Several issues encountered through the course of this investigation remain unresolved and are therefore recommended as possible research areas.

6.3.1 Diagonal Dominance Required by the Iterative Methods

The iterative method used for the inner iterations of the static and transient nodal methods is the cyclic Chebyshev semi-iterative method (CCSI). The convergence of the CCSI method can be guaranteed if the coefficient matrix is diagonally dominant. Because of the introduction of discontinuity factors into the finite-difference equations, however, the diagonal dominance cannot be guaranteed.

In an attempt to force the corrected finite-difference equations to be diagonally dominant, Aragonés and Ahnert [A-4] have proposed the use of modified diffu-

sion coefficients. For demonstration purposes, consider the one-dimensional corrected finite-difference equation

$$\begin{aligned}
& \frac{1}{h_x^i} \left[\frac{h_x^i}{2D_g^i} \frac{f_{gx-}^i}{f_{gx+}^{i-1}} + \frac{h_x^{i-1}}{2D_g^{i-1}} \right]^{-1} \left(\frac{f_{gx-}^i}{f_{gx+}^{i-1}} \bar{\phi}_g^i - \bar{\phi}_g^{i-1} \right) \\
& + \frac{1}{h_x^i} \left[\frac{h_x^i}{2D_g^i} \frac{f_{gx+}^i}{f_{gx-}^{i+1}} + \frac{h_x^{i+1}}{2D_g^{i+1}} \right]^{-1} \left(\frac{f_{gx+}^i}{f_{gx-}^{i+1}} \bar{\phi}_g^i - \bar{\phi}_g^{i+1} \right) \\
& + \sum_{g'=1}^G \left[\frac{1}{\lambda} \chi_{g\nu} \Sigma_{fg'}^i + \Sigma_{gg'}^i \right] \bar{\phi}_{g'}^i + q_g^i.
\end{aligned} \tag{6.1}$$

With appropriate definitions this equation can be written in the following simple form

$$- L_g^i \phi_g^{i-1} + \left(\frac{f_{gx-}^i}{f_{x+}^{i-1}} L_g^i + \frac{f_{gx+}^i}{f_{gx-}^{i+1}} R_g^i + \Sigma_g^i \right) \phi_g^i - R_g^i \phi_g^{i+1} = S_g^i. \tag{6.2}$$

The diagonal dominance of this equation requires

$$\frac{f_{gx-}^i}{f_{x+}^{i-1}} L_g^i + \frac{f_{gx+}^i}{f_{gx-}^{i+1}} R_g^i + \Sigma_g^i > L_g^i + R_g^i. \tag{6.3}$$

According to the method presented by Aragonés and Ahnert, Eq.(6.2) is diagonally dominant provided that L_g^i , R_g^i , and the discontinuity factor ratios are nonnegative and bounded. Examination of the diagonal dominance condition, Eq.(6.3), however, shows that if the discontinuity factor ratios are small and positive, diagonal dominance may not be obtained. Alternate conditions for the diffusion coefficients, however, may possibly be obtained by closer examination of the diagonal dominance condition, Eq.(6.3).

6.3.2 Application to Multi-Group Analyses

The polynomial nodal method embodied in CONQUEST can handle any number or structure of energy groups. The benchmarking of the method, however, involved only two-group problems and homogenous 4 and 7 groups problems. The analysis of problems involving more than two energy groups would prove interesting.

6.3.3 Study of the Thermal-Hydraulic/Neutron Coupling

In the application of the quasi-static method to problems involving thermal-hydraulic feedback it was found that the thermal-hydraulic conditions of the reactor must be updated more frequently than the shape function to reflect changes in the reactivity. For complicated thermal-hydraulic models this may require substantial computation time. Methods of decoupling the thermal-hydraulic equations, perhaps by the use of "feedback coefficients", should be investigated.

6.3.4 Study of the Time Dependence of the Discontinuity Factors

In many of the transient calculations it was found that the discontinuity factors only need to be updated when significant changes in the spatial flux shape occurs. The use of the pre-computed discontinuity factors which are held constant, or obtained periodically using a table look-up procedure, may increase the computational efficiency of the transient calculation by eliminating the necessity of updating.

REFERENCES

- [A-1] B. N. Aviles "Digital Control Strategies for Spatially-Dependent Reactor Cores with Thermal-Hydraulic Feedback," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (February 1990).
- [A-2] "Argonne Code Center: Benchmark Problem Book," ANL-7416, Supplement 2, Argonne National Laboratory (June 1977).
- [A-3] "National Energy Software Center: Benchmark Problem Book," ANL-7416, Supplement 3, Argonne National Laboratory (December 1985).
- [A-4] J. M. Aragonés and C. Ahnert, "A Linear Discontinuous Finite Difference Formulation for Synthetic Coarse-Mesh Few-Group Diffusion Calculations," *Nucl. Sci. Eng.*, **94**, 309 (1986).
- [B-1] M. E. Byers, "A Transient Nodal Method for Reactor Models in R-Z Geometry," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (September 1992).
- [B-2] F. Bennewitz, H. Finnemann, M. R. Wagner, "Higher Order Corrections in Nodal Reactor Calculations," *Trans. Am. Nucl. Soc.*, **22**, 250 (1975).
- [C-1] E. L. L. Cabral, "Three Dimensional Real-Time Thermal-Hydraulic Model and Non-Linear Controller for Large PWR Cores," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (December 1988).
- [D-1] T. F. DeLorey, Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (in progress).
- [D-2] G. Dahlquist, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ (1974).
- [D-3] J. J. Dongarra, "Performance of Various Computers Using Standard Linear Equations Software," CS-89-85, University of Tennessee Computer Science Department (June 1992).
- [F-1] H. Finnemann, F. Bennewitz, and M. R. Wagner, "Interface Current Techniques for Multidimensional Reactor Calculations," *Atomkernenergie*, **30**, 123 (1977).

- [G-1] R. G. Gamino, "The Development and Application of Supernodal Methods to PWR Analysis," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (May 1986).
- [H-1] C. L. Hoxie, A. Weitzberg, and W. Herwig, "Accurate Solution of the Transient Three-Dimensional CMFD Equations," *Trans. Am. Nucl. Soc.*, **47**, 411 (1984).
- [H-2] K. F. Hansen and D. R. Ferguson, "Improved Temporal Truncation Error in Space-Dependent Kinetics Calculations," *Trans. Am. Nucl. Soc.*, **15**, 790 (1972).
- [H-3] A. F. Henry and N. J. Curlee, "Verification of A Method for Treating Neutron Space-Time Problems," *Nucl. Sci. Eng.*, **4**, 727 (1958).
- [H-4] A. F. Henry, *Nuclear Reactor Analysis*, MIT Press, Cambridge, MA (1975).
- [H-5] A. F. Henry, "Derivation of Nodal Equations Having the Finite-Difference Form," Course Notes, 22.213, Massachusetts Institute of Technology, (Fall 1989).
- [H-6] A. F. Henry and K. S. Smith, "Modern Nodal Methods for Analyzing Light Water Reactors," Course Notes, Special Summer Course 22.80s, Department of Nuclear Engineering, Massachusetts Institute of Technology (June 1989).
- [H-6] L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic Press, New York (1981).
- [H-7] L. A. Hageman and J.B. Yasinsky, "Comparison of Alternating-Direction Time-Differencing Methods and Other Implicit Methods for the Solution of the Neutron Group Diffusion Equations," *Nucl. Sci. Eng.*, **38**, 8 (1969).
- [J-1] H. Joo, "Resolution of the Control Rod Cusping Problem for Nodal Methods," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (February 1984).
- [J-2] R. P. Jacqmin, "A Semi-Experimental Nodal Synthesis Method for the On-Line Reconstruction of Three-Dimensional Neutron Flux Shapes and Reactivity," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (September 1991).
- [J-3] R. P. Jacqmin, "A New Quasi-Static Option in the Quandry Code and Its Application to the Comparison of Point Kinetics with Space Kinetics for Severe Transients Involving Thermal-Hydraulic Feedback," Massachusetts Institute of Technology (1989).
- [K-1] P. W. Kao, "Application of Supernodal Methods to Transient Analysis," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (August 1988).

- [L-1] R. D. Lawrence, "A Nodal Green's Function Method for Multidimensional Neutron Calculations," *Nucl. Sci. Eng.*, **76**, 218 (1980).
- [L-2] S. Langenbuch, W. Maurer, and W. Werner, "Coarse-Mesh Flux-Expansion Method for the Analysis of Space-Time Effects in Large Light Water Reactor Cores," *Nucl. Sci. Eng.*, **63**, 437 (1977).
- [L-3] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, New York (1973).
- [N-1] S. Nakamura, *Computational Methods in Engineering and Science with Applications to Fluid Dynamics and Nuclear Systems*, John Wiley & Sons, New York (1977).
- [O-1] K. O. Ott, D. A. Meneley, "Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics," *Nucl. Sci. Eng.* **36**, 402 (1969).
- [P-1] D. K. Parsons, "The Replacement of Reflectors and Baffles in Nodal Calculations by Albedo Boundary Conditions," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (February 1984).
- [P-2] W. H. Press et al. *Numerical Recipes*, Cambridge University Press, New York (1989).
- [S-1] K. S. Smith, "An Spatial Homogenization Methods for Light Water Reactor Analysis," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (June 1980).
- [S-2] K. S. Smith, "An Analytic Nodal Method for Solving the Two-Group, Multidimensional, Static and Transient Neutron Diffusion Equations," SM and NE Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (March 1979).
- [S-3] K. S. Smith, D.M Ver Planck, and M. Edenius, "An Advanced Nodal Method for LWR Analyses," *Trans. Am. Nucl. Soc.*, **50**, 532 (1985).
- [S-4] K. S. Smith, "Nodal Method Storage Reduction by Nonlinear Iteration," *Trans. Am. Nucl. Soc.*, **44**, 265 (1983).
- [S-5] Y. A. M. Shatilla, "A Simple Quadratic Nodal Model for Hexagonal Geometry," D.Sc. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (September 1992).
- [S-7] T. M. Sutton, "NODEX: A High Order NEM-Based Multigroup Nodal Code," in *Advances in Nuclear Engineering Computation and Radiation Shielding* (Proc. of Topical Meeting, Santa Fe, NM, April 9-13, 1989) Vol. 1, American Nuclear Society, LaGrange Park, IL (1989).
- [S-8] G. Strang *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA (1986).

- [S-9] T. M. Sutton "Wielandt Iteration as Applied to the Nodal Expansion Method," *Nucl. Sci. Eng*, **98**, 169 (1988).
- [S-10] W. M. Stacey *Space-Time Nuclear Reactor Kinetics*, Academic Press, New York (1969).
- [S-11] R. A. Shober, "A Nodal Method for Solving Transient Few-Group Neutron Diffusion Equations," ANL-78-51, Argonne National Laboratory (June 1978).
- [T-1] T. A. Taiwo "The Quasi-Static Nodal Model for Reactor Core Kinetics," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (October 1984).
- [V-1] A. V. Vota, N. J. Curlee, Jr., and A. F. Henry, "WIGL3-A Program for the Steady-State and Transient Solution of the One-Dimensional, Two-Group, Space-Time Diffusion Equations Accounting for Temperature, Xenon, and Control Feedback," WAPD-TM-788 (February 1969).
- [V-2] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J. (1962).
- [W-2] E. L. Wachspress *Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics*, Prentice-Hall, Englewood Cliffs, N.J. (1966).
- [Z-1] M. L. Zerkle, "Development of a Polynomial Nodal Method with Flux and Current Discontinuity Factors," Ph.D. Thesis, Department of Nuclear Engineering, Massachusetts Institute of Technology (June 1992).

Appendix A

THE QUADRATIC TRANSVERSE LEAKAGE MOMENTS AND COEFFICIENTS

In this appendix the transverse-leakage coefficient and moments required by the weighted residual equations will be derived. The transverse-leakage moments for node (l,m,n) in the u -direction are defined by

$$S_{gup}^{lmn} \equiv \int_{u_l}^{u_{l+1}} w_p(u) S_{gu}^{lmn}(u) du, \quad (\text{A.1})$$

where $w_p(u)$ is the weight function. For moments weighting we use the first and second order expansion functions given by

$$w_1(u) = \left(\frac{u - u_l}{h_u^l} \right) - \frac{1}{2}, \quad (\text{A.2a})$$

$$w_2(u) = 3 \left(\frac{u - u_l}{h_u^l} \right)^2 - 3 \left(\frac{u - u_l}{h_u^l} \right) + \frac{1}{2}. \quad (\text{A.2b})$$

The transverse-leakage moments are determined by assuming that its shape in the u -direction can be represented by a quadratic polynomial. The coefficients of the polynomial are obtained by requiring the quadratic approximation to preserve the transverse leakage in three adjacent nodes. Within the core interior, the quadratic transverse leakage which is fitted to the three adjacent nodes is used only for the central node. Nodes located on the reactor boundary, however, do not have nodes on both sides requiring that a biased quadratic fit be used. In addition, a flat transverse-leakage approximation may be used at the reactor surface.

A.1 The Quadratic Transverse-Leakage Approximation

For the quadratic transverse-leakage approximation we represent the transverse leakage by

$$S_{gu}^{mn}(u) = \bar{S}_{gu}^{lmn} + \left(\bar{S}_{gu}^{l-1,mn} - \bar{S}_{gu}^{lmn} \right) \rho_{u_l}^{l-1}(u) + \left(\bar{S}_{gu}^{l+1,mn} - \bar{S}_{gu}^{lmn} \right) \rho_{u_l}^{l+1}(u), \quad (\text{A.3})$$

where the u -direction node-averaged transverse-leakage for node (l, m, n) is given by

$$\bar{S}_{gu}^{lmn} \equiv \frac{1}{h_u^l} \int_{u_l}^{u_l+1} S_{gu}^{mn}(u) du,$$

and the quadratic polynomials are

$$\rho_{u_l}^{l-1}(u) = a_{u_l}^- + b_{u_l}^- \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^- \left(\frac{u - u_l}{h_u^l} \right)^2, \quad (\text{A.4a})$$

$$\rho_{u_l}^{l+1}(u) = a_{u_l}^+ + b_{u_l}^+ \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^+ \left(\frac{u - u_l}{h_u^l} \right)^2. \quad (\text{A.4b})$$

Substituting of the transverse-leakage approximation, Eq. (A.3), into Eq.(A.1) and performing the required integration gives the following equations for the transverse leakage moments

$$S_{gu1}^{lmn} = \frac{1}{12} \left[(b_{u_l}^- + c_{u_l}^-) \bar{S}_{gu}^{l-1,mn} - (b_{u_l}^- + b_{u_l}^+ + c_{u_l}^- + c_{u_l}^+) \bar{S}_{gu}^{lmn} + (b_{u_l}^+ + c_{u_l}^+) \bar{S}_{gu}^{l+1,mn} \right], \quad (\text{A.5a})$$

$$S_{gu2}^{lmn} = \frac{1}{60} \left[c_{u_l}^- \bar{S}_{gu}^{l-1,mn} - (c_{u_l}^- + c_{u_l}^+) \bar{S}_{gu}^{lmn} + c_{u_l}^+ \bar{S}_{gu}^{l+1,mn} \right]. \quad (\text{A.5b})$$

The quadratic polynomial coefficients are obtained by requiring the transverse-leakage approximation to preserve the average transverse leakages in each of three adjacent nodes. This results in the following constraints being placed on $\rho_{u_l}^{l-1}(u)$ and $\rho_{u_l}^{l+1}(u)$:

$$\frac{1}{h_u^{l-1}} \int_{u_{l-1}}^{u_l} \rho_{u_l}^{l-1}(u) du = 1,$$

$$\begin{aligned}
\frac{1}{h_u^l} \int_{u_i}^{u_{i+1}} \rho_{u_i}^{l-1}(u) du &= 0, \\
\frac{1}{h_u^{l+1}} \int_{u_{i+1}}^{u_{i+2}} \rho_{u_i}^{l-1}(u) du &= 0, \\
\frac{1}{h_u^{l-1}} \int_{u_{i-1}}^{u_i} \rho_{u_i}^{l+1}(u) du &= 0, \\
\frac{1}{h_u^l} \int_{u_i}^{u_{i+1}} \rho_{u_i}^{l+1}(u) du &= 0, \\
\frac{1}{h_u^{l+1}} \int_{u_{i+1}}^{u_{i+2}} \rho_{u_i}^{l+1}(u) du &= 1.
\end{aligned}$$

Integrating Eq. (A.3) over the three adjacent nodes and applying the above constraints gives the quadratic coefficients

$$\begin{aligned}
a_{u_i}^- &= \frac{h(h+h_p)}{(h_m+h+h_p)(h_m+h)}, \\
b_{u_i}^- &= -\frac{2h(2h+h_p)}{(h_m+h+h_p)(h_m+h)}, \\
c_{u_i}^- &= \frac{3h^2}{(h_m+h+h_p)(h_m+h)}, \\
a_{u_i}^+ &= -\frac{h_m h}{(h_m+h+h_p)(h+h_p)}, \\
b_{u_i}^+ &= \frac{2h(h_m-h)}{(h_m+h+h_p)(h+h_p)}, \\
c_{u_i}^+ &= \frac{3h^2}{(h_m+h+h_p)(h+h_p)},
\end{aligned}$$

where

$$\begin{aligned}
h_m &= h_u^{l-1}, \\
h &= h_u^l, \\
h_p &= h_u^{l+1}.
\end{aligned}$$

The transverse-leakage expansion coefficients are functions only of the reactor geometry. For equal-sized nodes, $h_m = h = h_p$, the coefficients are constants given

by

$$\begin{aligned} a_{u_l}^- &= \frac{1}{3}, & a_{u_l}^+ &= -\frac{1}{6}, \\ b_{u_l}^- &= -1, & b_{u_l}^+ &= 0, \\ c_{u_l}^- &= \frac{1}{2}, & c_{u_l}^+ &= \frac{1}{2}. \end{aligned}$$

A.2 LHS-Biased Quadratic Transverse-Leakage Approximation

For nodes adjacent to the external reactor surface, Zerkle [Z-1] introduced a biased quadratic transverse-leakage approximation. The transverse leakage in the u -direction in node (l,m,n) , which is on the LHS reactor surface is given by

$$S_{gu}^{mn}(u) = \bar{S}_{gu}^{lmn} + \left(\bar{S}_{gu}^{l+1,mn} - \bar{S}_{gu}^{lmn} \right) \rho_{u_l}^{l+1}(u) + \left(\bar{S}_{gu}^{l+2,mn} - \bar{S}_{gu}^{lmn} \right) \rho_{u_l}^{l+2}(u), \quad (\text{A.6})$$

where

$$\rho_{u_l}^{l+1}(u) = a_{u_l}^+ + b_{u_l}^+ \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^+ \left(\frac{u - u_l}{h_u^l} \right)^2, \quad (\text{A.7a})$$

$$\rho_{u_l}^{l+2}(u) = a_{u_l}^{++} + b_{u_l}^{++} \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^{++} \left(\frac{u - u_l}{h_u^l} \right)^2. \quad (\text{A.7b})$$

Substituting the transverse-leakage approximation, Eq. (A.6), into Eq.(A.1) and performing the required integration gives the following equations for the transverse leakage moments

$$\begin{aligned} S_{gu1}^{lmn} &= \frac{1}{12} \left[(b_{u_l}^+ + c_{u_l}^+) \bar{S}_{gu}^{l+1,mn} - (b_{u_l}^+ + b_{u_l}^{++} + c_{u_l}^+ + c_{u_l}^{++}) \bar{S}_{gu}^{lmn} \right. \\ &\quad \left. + (b_{u_l}^{++} + c_{u_l}^{++}) \bar{S}_{gu}^{l+2,mn} \right], \end{aligned} \quad (\text{A.8a})$$

$$S_{gu2}^{lmn} = \frac{1}{60} \left[c_{u_l}^+ \bar{S}_{gu}^{l+1,mn} - (c_{u_l}^+ + c_{u_l}^{++}) \bar{S}_{gu}^{lmn} + c_{u_l}^{++} \bar{S}_{gu}^{l+2,mn} \right]. \quad (\text{A.8b})$$

The quadratic polynomial coefficients are obtained by requiring the transverse-leakage approximation, Eq. (A.3), to preserve the average transverse leakages in each of three nodes (l,m,n) , $(l+1,m,n)$, and $(l+2,m,n)$. This results in the following

constraints being placed on $\rho_{u_i}^{l+1}(u)$ and $\rho_{u_i}^{l+2}(u)$:

$$\frac{1}{h_u^l} \int_{u_i}^{u_{i+1}} \rho_{u_i}^{l+1}(u) du = 0,$$

$$\frac{1}{h_u^{l+1}} \int_{u_{i+1}}^{u_{i+2}} \rho_{u_i}^{l+1}(u) du = 1,$$

$$\frac{1}{h_u^{l+2}} \int_{u_{i+2}}^{u_{i+3}} \rho_{u_i}^{l+1}(u) du = 0,$$

$$\frac{1}{h_u^l} \int_{u_i}^{u_{i+1}} \rho_{u_i}^{l+2}(u) du = 0,$$

$$\frac{1}{h_u^{l+1}} \int_{u_{i+1}}^{u_{i+2}} \rho_{u_i}^{l+2}(u) du = 0,$$

$$\frac{1}{h_u^{l+2}} \int_{u_{i+2}}^{u_{i+3}} \rho_{u_i}^{l+2}(u) du = 1.$$

Integrating Eq. (A.6) over the three adjacent nodes and applying the above constraints gives the following quadratic coefficients

$$a_{u_i}^+ = -\frac{h \left[h(h + 4h_p + 2h_{pp}) + 3h_p(h_p + h_{pp}) + h_{pp}^2 \right]}{(h + h_p + h_{pp})(h + h_p)(h_p + h_{pp})},$$

$$b_{u_i}^+ = \frac{2h \left[h(2h + 6h_p + 3h_{pp}) + 3h_p(h_p + h_{pp}) + h_{pp}^2 \right]}{(h + h_p + h_{pp})(h + h_p)(h_p + h_{pp})},$$

$$c_{u_i}^+ = -\frac{3h^2(h + 2h_p + h_{pp})}{(h + h_p + h_{pp})(h + h_p)(h_p + h_{pp})},$$

$$a_{u_i}^{++} = \frac{h(h + h_p)}{(h + h_p + h_{pp})(h_p + h_{pp})},$$

$$b_{u_i}^{++} = -\frac{2h(2h + h_p)}{(h + h_p + h_{pp})(h_p + h_{pp})},$$

$$c_{u_i}^{++} = \frac{3h^2}{(h + h_p + h_{pp})(h_p + h_{pp})},$$

where

$$h = h_u^l,$$

$$h_p = h_u^{l+1},$$

$$h_{pp} = h_u^{l+2}.$$

The transverse-leakage expansion coefficients are functions of only the reactor geometry. For equal-sized nodes, $h = h_p = h_{pp}$, the coefficients are constants given by

$$\begin{aligned} a_{u_l}^+ &= -\frac{7}{6}, & a_{u_l}^{++} &= \frac{1}{3}, \\ b_{u_l}^+ &= 3, & b_{u_l}^{++} &= -1, \\ c_{u_l}^+ &= -1, & c_{u_l}^{++} &= \frac{1}{2}. \end{aligned}$$

A.3 RHS-Biased Quadratic Transverse-Leakage Approximation

Similarly, the transverse leakage in the u -direction in node (l, m, n) which is on the RHS reactor surface is given by

$$S_{gu}^{mn}(u) = \bar{S}_{gu}^{lmn} + (\bar{S}_{gu}^{l-1, mn} - \bar{S}_{gu}^{lmn}) \rho_{u_l}^{l-1}(u) + (\bar{S}_{gu}^{l-2, mn} - \bar{S}_{gu}^{lmn}) \rho_{u_l}^{l-2}(u), \quad (\text{A.9})$$

where

$$\rho_{u_l}^{l-1}(u) = a_{u_l}^- + b_{u_l}^- \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^- \left(\frac{u - u_l}{h_u^l} \right)^2, \quad (\text{A.10a})$$

$$\rho_{u_l}^{l-2}(u) = a_{u_l}^{--} + b_{u_l}^{--} \left(\frac{u - u_l}{h_u^l} \right) + c_{u_l}^{--} \left(\frac{u - u_l}{h_u^l} \right)^2. \quad (\text{A.10b})$$

Substituting the transverse-leakage approximation, Eq. (A.9), into Eq.(A.1) and performing the required integration gives the following equations for the transverse leakage moments

$$\begin{aligned} S_{gu1}^{lmn} &= \frac{1}{12} \left[(b_{u_l}^- + c_{u_l}^-) \bar{S}_{gu}^{l-1, mn} - (b_{u_l}^- + b_{u_l}^{--} + c_{u_l}^- + c_{u_l}^{--}) \bar{S}_{gu}^{lmn} \right. \\ &\quad \left. + (b_{u_l}^{--} + c_{u_l}^{--}) \bar{S}_{gu}^{l-2, mn} \right], \end{aligned} \quad (\text{A.11a})$$

$$S_{gu2}^{lmn} = \frac{1}{60} \left[c_{u_l}^- \bar{S}_{gu}^{l-1, mn} - (c_{u_l}^- + c_{u_l}^{--}) \bar{S}_{gu}^{lmn} + c_{u_l}^{--} \bar{S}_{gu}^{l-2, mn} \right]. \quad (\text{A.11b})$$

The quadratic polynomial coefficients are obtained by requiring the transverse-leakage approximation, Eq. (A.3), to preserve the average transverse leakages in each of three nodes (l, m, n) , $(l - 1, m, n)$, and $(l - 2, m, n)$. This results in the following

constraints being placed on $\rho_{u_i}^{l-1}(u)$ and $\rho_{u_i}^{l-2}(u)$

$$\frac{1}{h_u^l} \int_{u_i}^{u_{i+1}} \rho_{u_i}^{l-1}(u) du = 0,$$

$$\frac{1}{h_u^{l-1}} \int_{u_{i-1}}^{u_i} \rho_{u_i}^{l-1}(u) du = 1,$$

$$\frac{1}{h_u^{l-2}} \int_{u_{i-2}}^{u_{i-1}} \rho_{u_i}^{l-1}(u) du = 0,$$

$$\frac{1}{h_u^l} \int_{u_i}^{u_{i+1}} \rho_{u_i}^{l-2}(u) du = 0,$$

$$\frac{1}{h_u^{l-1}} \int_{u_{i-1}}^{u_i} \rho_{u_i}^{l-2}(u) du = 0,$$

$$\frac{1}{h_u^{l-2}} \int_{u_{i-2}}^{u_{i+1}} \rho_{u_i}^{l-2}(u) du = 1.$$

Integrating Eqs. (A.10a) and (A.10b) over the three adjacent nodes and applying the above constraints gives the following quadratic coefficients

$$a_{u_i}^- = \frac{h [h_{mm}(h_{mm} + 3h_m + h) + h_m(3h_m + 2h)]}{(h_{mm} + h_m + h)(h_m + h)(h_{mm} + h_m)},$$

$$b_{u_i}^- = -\frac{2h [h_{mm}(h_{mm} + 3h_m) + 3h_m^2 - h^2]}{(h_{mm} + h_m + h)(h_m + h)(h_{mm} + h_m)},$$

$$c_{u_i}^- = -\frac{3h^2(h_{mm} + 2h_m + h)}{(h_{mm} + h_m + h)(h_m + h)(h_{mm} + h_m)},$$

$$a_{u_i}^{--} = -\frac{hh_{mm}}{(h_{mm} + h_m + h)(h_{mm} + h_m)},$$

$$b_{u_i}^{--} = \frac{2h(h_m - h)}{(h_{mm} + h_m + h)(h_{mm} + h_m)},$$

$$c_{u_i}^{--} = \frac{3h^2}{(h_{mm} + h_m + h)(h_{mm} + h_m)},$$

where

$$h_{mm} = h_u^{l-2},$$

$$h_m = h_u^{l-1},$$

$$h = h_u^l.$$

The transverse-leakage expansion coefficients are functions of only the reactor geometry. For equal-sized nodes, $h_{mm} = h_m = h$, the coefficients are constants given by

$$\begin{aligned} a_{u_i}^- &= \frac{5}{6}, & a_{u_i}^{--} &= -\frac{1}{6}, \\ b_{u_i}^- &= -1, & b_{u_i}^{--} &= 0, \\ c_{u_i}^- &= -1, & c_{u_i}^{--} &= \frac{1}{2}. \end{aligned}$$

A.4 The Flat Transverse-Leakage Approximation

In the flat transverse-leakage approximation, the transverse-leakage in the u -direction is assumed to be constant and equal to the node-averaged u -direction transverse leakage

$$S_{gu}^{lmn} = \bar{S}_{gu}^{lmn}. \quad (\text{A.12})$$

Substituting of the flat transverse-leakage approximation into Eq.(A.1) and performing the required integration reveals that for this approximation the transverse leakage moments are zero

$$S_{gup}^{lmn} = 0, \quad p = 1, 2. \quad (\text{A.13})$$

Appendix B

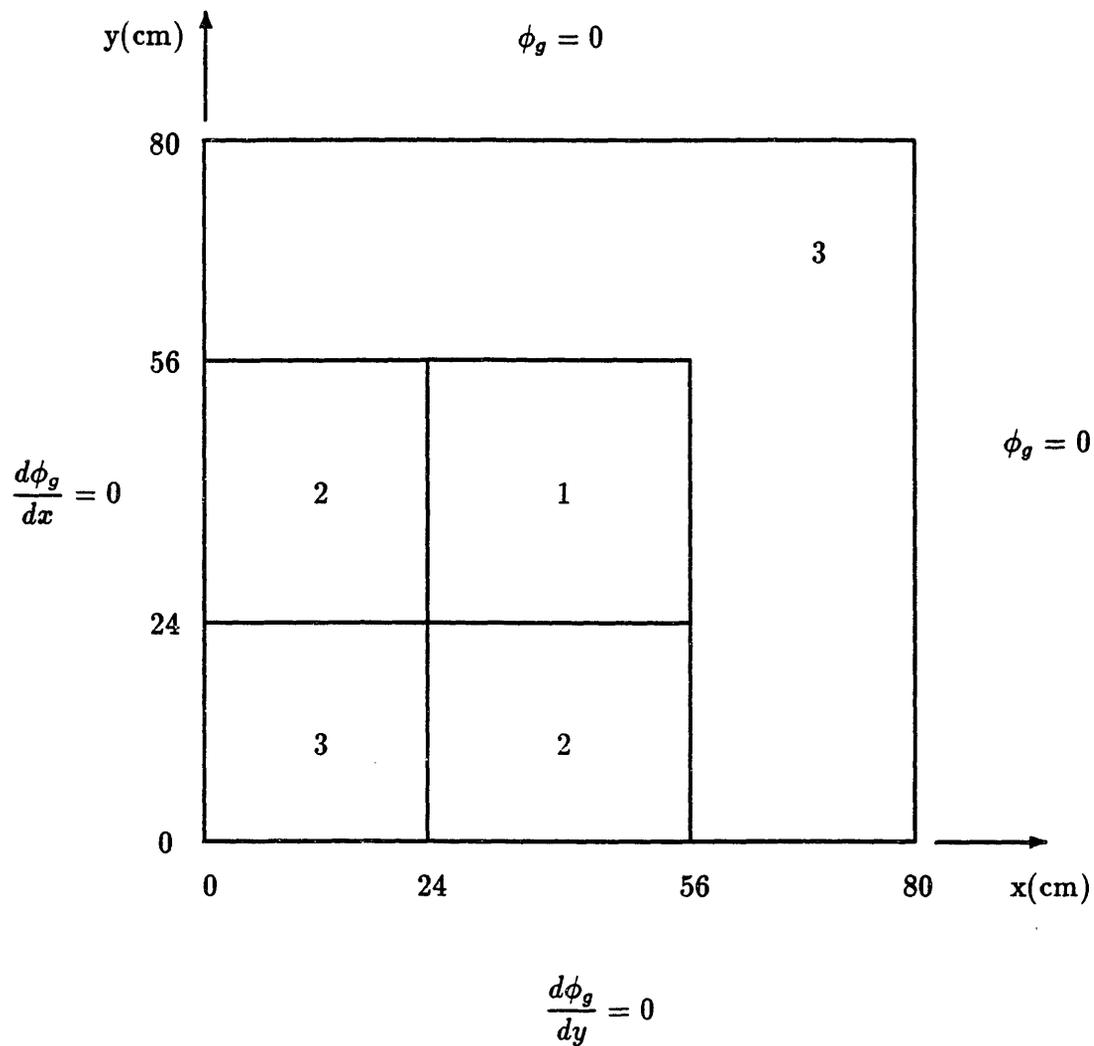
PROBLEM SPECIFICATIONS

- B.1 The TWIGL 2-D Seed-Blanket Reactor Kinetics Problem
- B.2 The LMW LWR Transient Problem
- B.3 The LRA BWR Kinetics Benchmark Problem
- B.4 The PWR Transient Problems

B.1 The TWIGL 2-D Seed-Blanket Reactor Kinetics Problem

Geometry:

Quadrant of TWIGL Reactor



Material Properties:

Composition	Group, g	D_g (cm)	Σ_{ag} (cm ⁻¹)	$\nu\Sigma_{fg}$ (cm ⁻¹)	Σ_{21} (cm ⁻¹)
1	1	1.4	0.01	0.007	0.01
	2	0.4	0.15	0.2	
2	1	1.4	0.01	0.007	0.01
	2	0.4	0.15	0.2	
3	1	1.3	0.008	0.003	0.01
	2	0.5	0.05	0.06	

$$\chi_1 = 1.0$$

$$\chi_2 = 0.0$$

$$\nu = 2.43$$

$$v_1 = 1 \times 10^7 \text{ cm/s}$$

$$v_2 = 2 \times 10^5 \text{ cm/s}$$

Delayed Neutron Data:

Family, d	β_d	λ_d (s ⁻¹)
1	0.0075	0.08

Perturbations:

In composition 1,

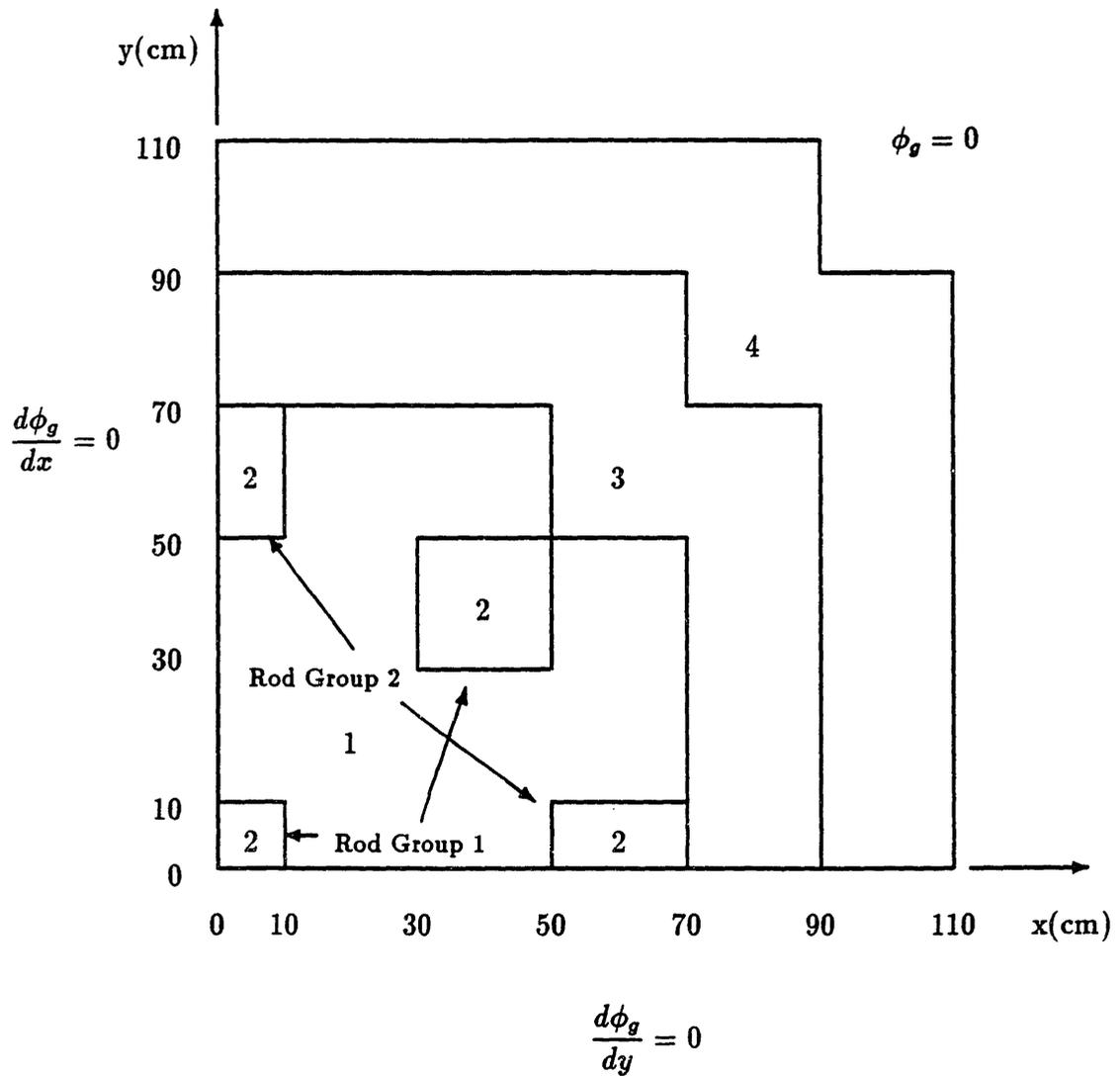
$$\text{Step: } \Delta\Sigma_{a2} = -0.0035; \quad t = 0$$

$$\text{Ramp: } \Sigma_{a2}(t) = \begin{cases} \Sigma_{a2}(0) \{1 - 0.11667t\} & t \leq 0.2 \\ \Sigma_{a2}(0) \{0.97666\} & t > 0.2 \end{cases}$$

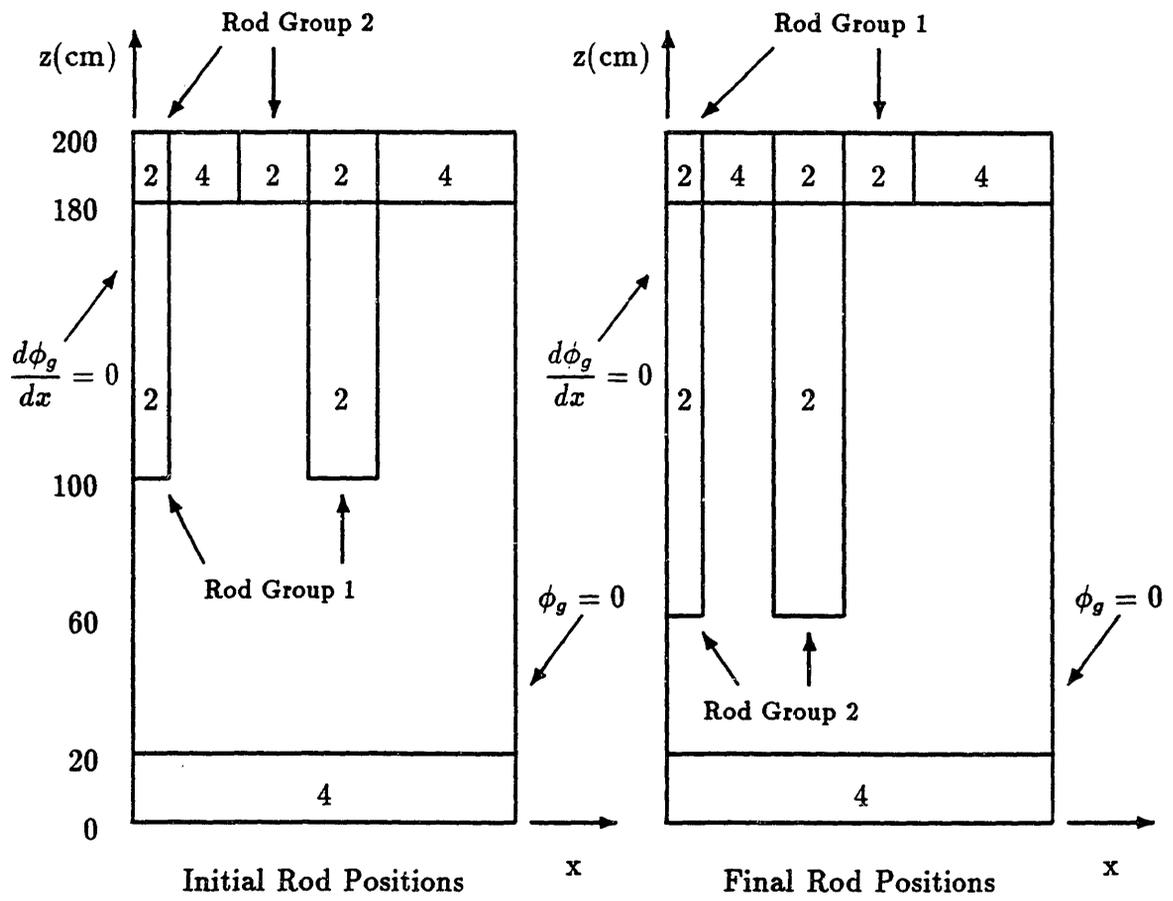
B.2 The LMW LWR Transient Problem

Geometry:

Quadrant of Reactor Horizontal Section



Reactor Vertical Section



Material Properties:

Composition	Group, g	D_g (cm)	Σ_{ag} (cm^{-1})	$\nu\Sigma_{fg}$ (cm^{-1})	Σ_{21} (cm^{-1})
1	1	1.423913	0.01040206	0.006477691	0.0175555
	2	0.356306	0.08766217	0.1127328	
2	1	1.423913	0.01095206	0.006477691	0.0175555
	2	0.356306	0.08766217	0.1127328	
3	1	1.425611	0.01099263	0.007503284	0.01717768
	2	0.350574	0.09925634	0.1378004	
3	1	1.634227	0.002660573	0.0	0.02759693
	2	0.264002	0.04936351	0.0	

$$\chi_1 = 1.0$$

$$\chi_2 = 0.0$$

$$\nu = 2.5$$

$$v_1 = 1.25 \times 10^7 \text{ cm/s}$$

$$v_2 = 2.5 \times 10^5 \text{ cm/s}$$

Delayed Neutron Data:

Family, d	β_d	λ_d (s^{-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

$$\chi_{d1} = 1.0$$

$$\chi_{d2} = 0.0, \quad d = 1, 2, \dots, 6$$

Energy Conversion Factor:

$$3.204 \times 10^{-11} \text{ J/fission}$$

Perturbation:

Rod Group 1 removed at 3.0 cm/s, $0 \leq t \leq 26.666$ s

Rod Group 2 inserted at 3.0 cm/s, $7.5 \leq t \leq 47.5$ s

WIGL Thermal-Hydraulic Parameters:

$$C_f = 2.46 \times 10^6 \text{ ergs/g/K}$$

$$C_c = 5.43 \times 10^7 \text{ ergs/g/K}$$

$$\rho_f = 10.3 \text{ g/cm}^3$$

$$W_0 = 2.2 \times 10^6 \text{ g/s}$$

$$h_0 = 2.71 \times 10^7 \text{ ergs/cm}^2/\text{s/K}$$

$$A_h = 2.59 \text{ 1/cm}$$

$$U = 2.2 \times 10^6 \text{ ergs/cm}^2/\text{s/K}$$

$$V_c/(V_c + V_f) = 0.559$$

$$r = 0.0$$

$$\frac{\partial \rho_c H}{\partial \bar{T}_c} = 1.60 \times 10^7 \text{ ergs/cm}^3/\text{K}$$

$$\text{Pressure} = 1.53 \times 10^7 \text{ Pa}$$

$$\text{Coolant Inlet Temp.} = 533 \text{ K}$$

$$\text{Initial Power} = 184.8 \text{ MW}_{\text{th}} \text{ (quarter-core)}$$

Macroscopic Cross Section Derivatives:

Parameter, Σ	$\frac{\partial \Sigma}{\partial \rho_c}$	$\frac{\partial \Sigma}{\partial \bar{T}_c}$	$\frac{\partial \Sigma}{\partial \bar{T}_f}$
D_1^{-1}	+0.41	-8.0×10^{-5}	-6.6×10^{-6}
D_2^{-1}	+2.7	-1.3×10^{-3}	-2.6×10^{-6}
Σ_{c1}	$+2.83 \times 10^{-3}$	$+3.0 \times 10^{-6}$	$+3.3 \times 10^{-7}$
Σ_{c2}	$+1.4 \times 10^{-2}$	-8.2×10^{-6}	-3.7×10^{-7}
$\nu \Sigma_{f1}$	+0.0	+0.0	+0.0
$\nu \Sigma_{f2}^\dagger$	$+4.132 \times 10^{-2}$	-2.017×10^{-5}	-2.43×10^{-6}
Σ_{f1}	+0.0	+0.0	+0.0
Σ_{f2}^\dagger	$+1.7 \times 10^{-2}$	-8.3×10^{-6}	-1.0×10^{-6}
Σ_{21}	$+2.4 \times 10^{-2}$	-1.5×10^{-6}	$+8.5 \times 10^{-8}$

\dagger Zero for reflector materials (composition # 3)

$$\rho_0 = 0.7961 \text{ g/cm}^3$$

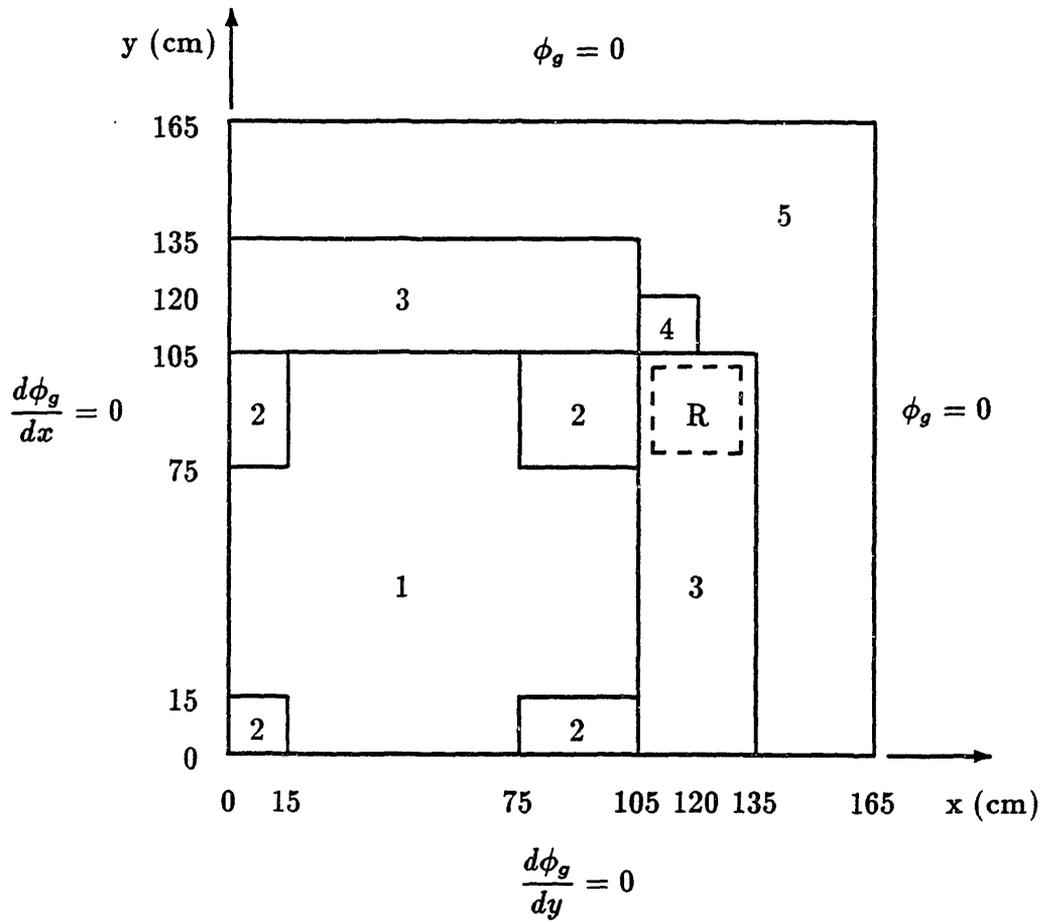
$$\bar{T}_{f0} = 533 \text{ K}$$

$$\bar{T}_{c0} = 533 \text{ K}$$

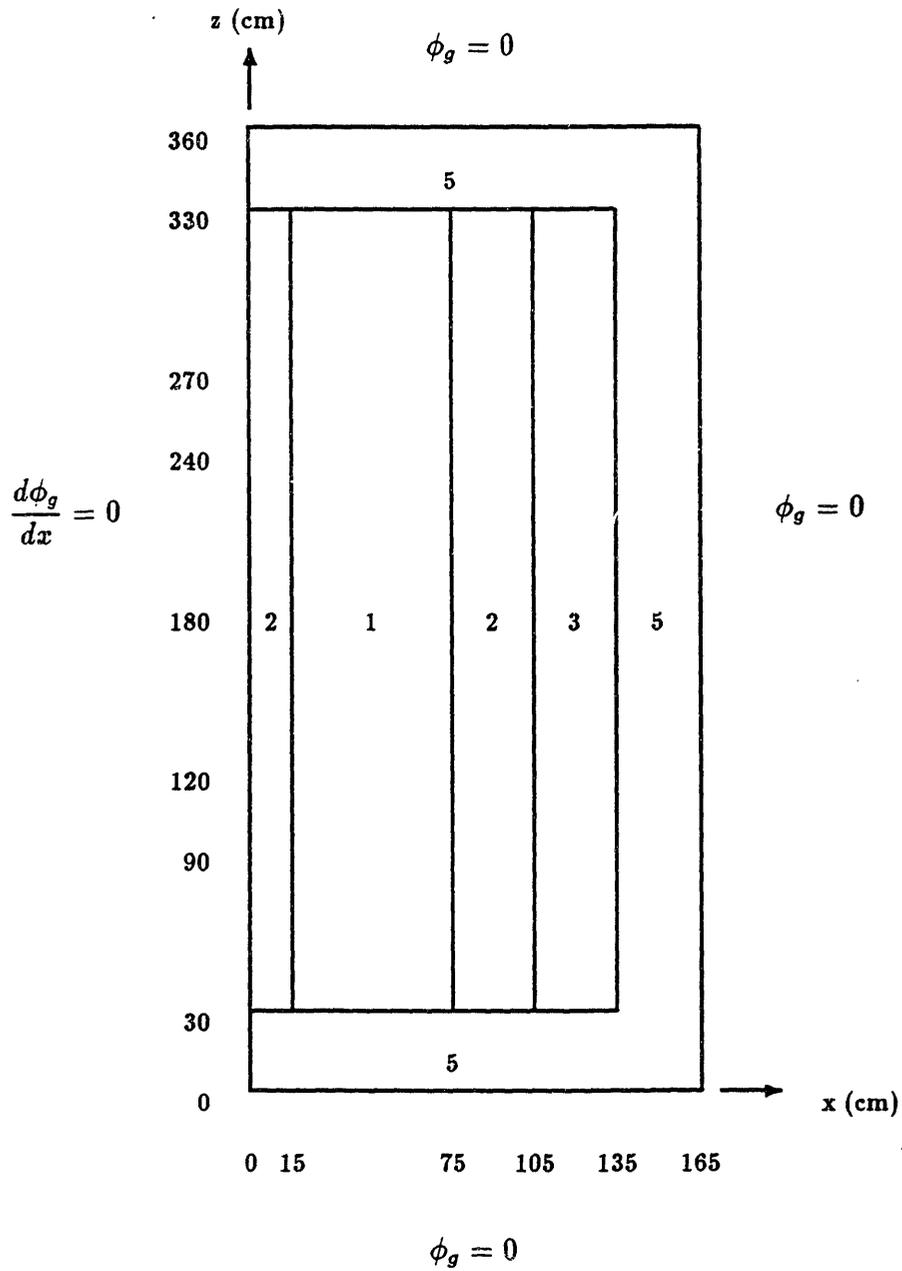
B.3 The LRA BWR Kinetics Benchmark Problem

Geometry:

Quadrant of Reactor Horizontal Section



Reactor Vertical Section, $y = 0$



Material Properties:

Composition	Group, g	D_g (cm)	Σ_{ag} (cm ⁻¹)	$\nu\Sigma_{fg}$ (cm ⁻¹)	Σ_{21} (cm ⁻¹)
1	1	1.255	0.008252	0.004602	0.02533
	2	0.211	0.1003	0.1091	
2	1	1.268	0.007181	0.004609	0.02767
	2	0.1902	0.07047	0.08675	
3	1	1.259	0.008002	0.004663	0.02617
	2	0.2091	0.08344	0.1021	
4	1	1.259	0.008002	0.004663	0.02617
	2	0.2091	0.073324	0.1021	
5	1	1.257	0.0006034	0.0	0.04754
	2	0.1592	0.01911	0.0	

Axial buckling of 10^{-4} for all compositions in 2-D problem.

$$\chi_1 = 1.0$$

$$\chi_2 = 0.0$$

$$\nu = 2.43$$

$$v_1 = 3.0 \times 10^7 \text{ cm/s}$$

$$v_2 = 3.0 \times 10^5 \text{ cm/s}$$

Delayed Neutron Data:

Family, d	β_d	λ_d (s ⁻¹)
1	0.0054	0.00654
2	0.001087	1.35

$$\chi_{d1} = 1.0$$

$$\chi_{d2} = 0.0, \quad d = 1, 2$$

Adiabatic Feedback Data:

$$\alpha \sum_{g=1}^G \Sigma_{fg}(\mathbf{r}, t) \phi_g(\mathbf{r}, t) = \frac{\partial}{\partial t} T(\mathbf{r}, t)$$

$$\Sigma_{a1}(\mathbf{r}, t) = \Sigma_{a1}(\mathbf{r}, 0) \left\{ 1 + \gamma \left(\sqrt{T(\mathbf{r}, t)} - \sqrt{T_0} \right) \right\}$$

where

$$\alpha = 3.83 \times 10^{-11} \text{ Kcm}^3$$

$$\gamma = 2.034 \times 10^{-3} \text{ K}^{-1/2}$$

$$T_0 = 300 \text{ K}$$

Energy Conversion Factor:

$$\text{Power} = \epsilon \int_{V_{\text{core}}} \sum_{g=1}^G \Sigma_{fg}(\mathbf{r}, t) \phi_g(\mathbf{r}, t) d\mathbf{r}$$

$$\epsilon = 3.204 \times 10^{-11} \text{ J/fission}$$

Transient Initial Conditions:

Mean power density at $t = 0$: 10^{-6} W/cm^3

Fuel temperature at $t = 0$: 300 K

Perturbation:

3-D: Control rod (R) removed downward at speed of 150 cm/s

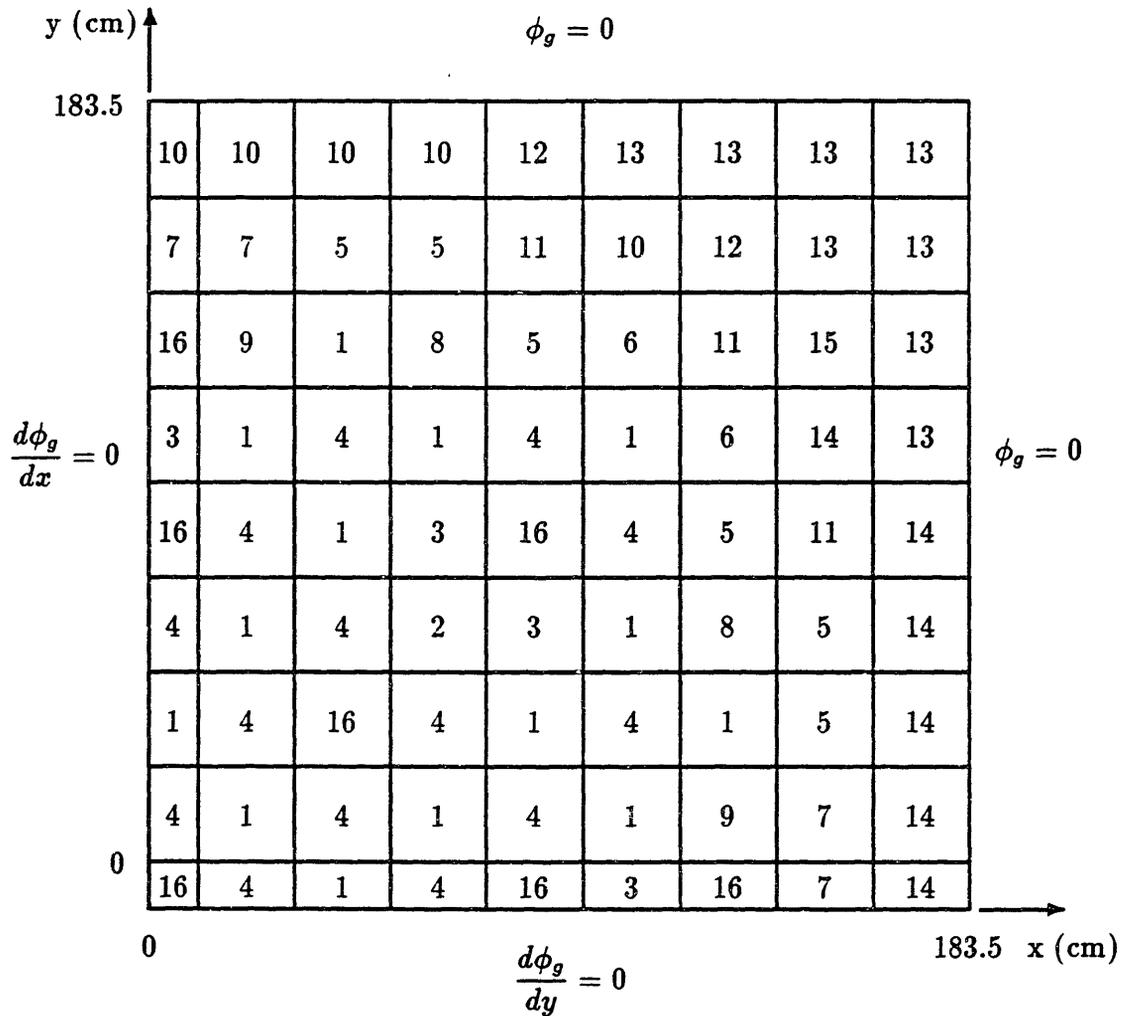
2-D: Control Rod composition (R) is given by

$$\Sigma_{a2}(t)/\Sigma_{a2}(0) = \begin{cases} 1 - 0.0606184 \times t & t < 2.0s \\ 0.8787631 & t > 2.0s \end{cases}$$

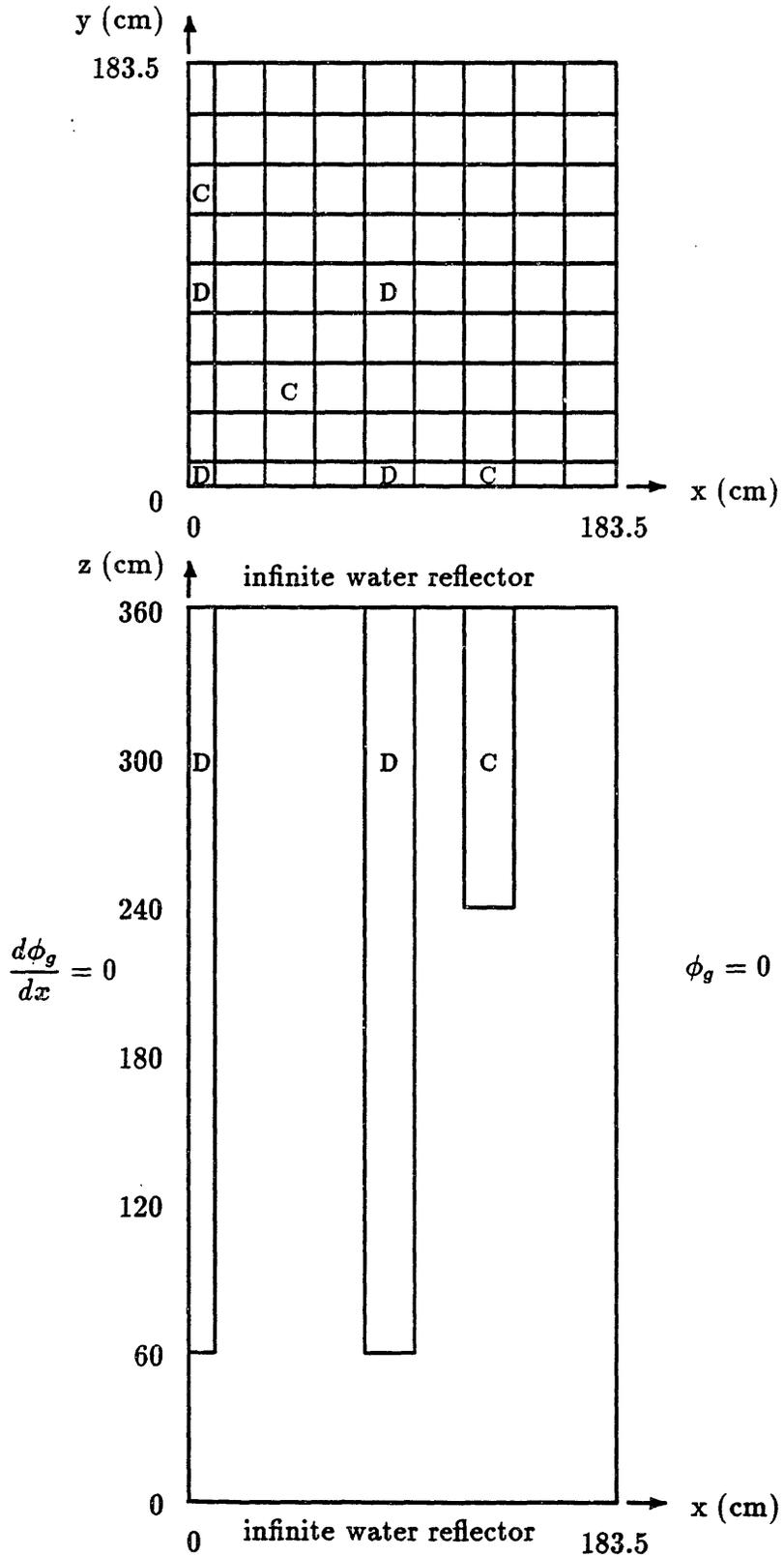
B.4 The PWR Transient Problems

Geometry

Quadrant of the Reactor, Assembly dimensions 21.591 cm × 21.591 cm. In Unrodded Planes, Composition #16 is Replaced by Composition #1.



Control Rod Bank D & C Locations and Reactor Vertical Section



Material Properties:

Composition	Group, <i>g</i>	D_g (cm)	Σ_{ag} (cm ⁻¹)	$\nu\Sigma_{fg}$ (cm ⁻¹)	Σ_{21} (cm ⁻¹)
1	1	1.3648	0.008887	0.005550	0.017245
	2	0.4826	0.130772	0.185823	
2	1	1.3603	0.009661	0.006267	0.015942
	2	0.4776	0.169403	0.229195	
3	1	1.3596	0.009957	0.006267	0.015398
	2	0.4798	0.181915	0.230258	
4	1	1.3592	0.010104	0.006269	0.015128
	2	0.4810	0.188426	0.230923	
5	1	1.3594	0.009509	0.006890	0.016386
	2	0.4673	0.169073	0.264760	
6	1	1.35898	0.0096925	0.0068905	0.0160495
	2	0.46853	0.1762888	0.2653975	
7	1	1.35890	0.009730	0.006890	0.015981
	2	0.46875	0.177654	0.265512	
8	1	1.3576	0.010252	0.006892	0.015022
	2	0.4728	0.200287	0.267778	
9	1	1.3572	0.010399	0.006894	0.014752
	2	0.4740	0.206951	0.268552	
10,14	1	1.4957	0.002683	0.0	0.022923
	2	0.3637	0.051595	0.0	
11	1	1.3933	0.003541	0.0	0.017943
	2	0.3659	0.068149	0.0	
12,15	1	1.6701	0.001220	0.0	0.031408
	2	0.3621	0.039330	0.0	
13	1	1.7446	0.000596	0.0	0.035032
	2	0.3614	0.034208	0.0	
16	1	1.321964	0.013482	0.055670	0.015178
	2	0.486198	0.161003	0.194976	

$$\chi_1 = 1.0$$

$$\chi_2 = 0.0$$

$$\nu = 2.5$$

$$v_1 = 1.25 \times 10^7 \text{ cm/s}$$

$$v_2 = 2.5 \times 10^5 \text{ cm/s}$$

Axial Albedo Boundary Conditions

Γ_{11}	Γ_{12}	Γ_{21}	Γ_{22}
4.011	0.0	2.805	8.993

Delayed Neutron Data:

Family, d	β_d	λ_d (s^{-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

$$\chi_{d1} = 1.0$$

$$\chi_{d2} = 0.0, \quad d = 1, 2, \dots, 6$$

Macroscopic Cross Section Derivatives:

Parameter, Σ	$\frac{\partial \Sigma}{\partial \rho_c}$	$\frac{\partial \Sigma}{\partial \bar{T}_c}$	$\frac{\partial \Sigma}{\partial \bar{T}_f}$
D_1^{-1}	+0.41	-8.0×10^{-5}	-6.6×10^{-6}
D_2^{-1}	+2.7	-1.3×10^{-3}	-2.6×10^{-6}
Σ_{c1}	$+2.83 \times 10^{-3}$	$+3.0 \times 10^{-6}$	$+3.3 \times 10^{-7}$
Σ_{c2}	$+1.4 \times 10^{-2}$	-8.2×10^{-6}	-3.7×10^{-7}
$\nu \Sigma_{f1}$	+0.0	+0.0	+0.0
$\nu \Sigma_{f2}^\dagger$	$+4.132 \times 10^{-2}$	-2.017×10^{-5}	-2.43×10^{-6}
Σ_{f1}	+0.0	+0.0	+0.0
Σ_{f2}^\dagger	$+1.7 \times 10^{-2}$	-8.3×10^{-6}	-1.0×10^{-6}
Σ_{21}	$+2.4 \times 10^{-2}$	-1.5×10^{-6}	$+8.5 \times 10^{-8}$

† Zero for reflector materials (compositions # 10 — 15)

$$\rho_0 = 0.7961 \text{ g/cm}^3$$

$$\bar{T}_{f0} = 533 \text{ K}$$

$$\bar{T}_{c0} = 533 \text{ K}$$

WIGL Thermal-Hydraulic Parameters:

$$C_f = 2.46 \times 10^6 \text{ ergs/g/K}$$

$$C_c = 5.43 \times 10^7 \text{ ergs/g/K}$$

$$\rho_f = 10.3 \text{ g/cm}^3$$

$$W_0 = 2.2 \times 10^6 \text{ g/s}$$

$$h_0 = 2.71 \times 10^7 \text{ ergs/cm}^2\text{/s/K}$$

$$A_h = 2.59 \text{ 1/cm}$$

$$U = 2.2 \times 10^6 \text{ ergs/cm}^2\text{/s/K}$$

$$V_c/(V_c + V_f) = 0.559$$

$$r = 0.0$$

$$\frac{\partial \rho_c H}{\partial \bar{T}_c} = 1.60 \times 10^7 \text{ ergs/cm}^3\text{/K}$$

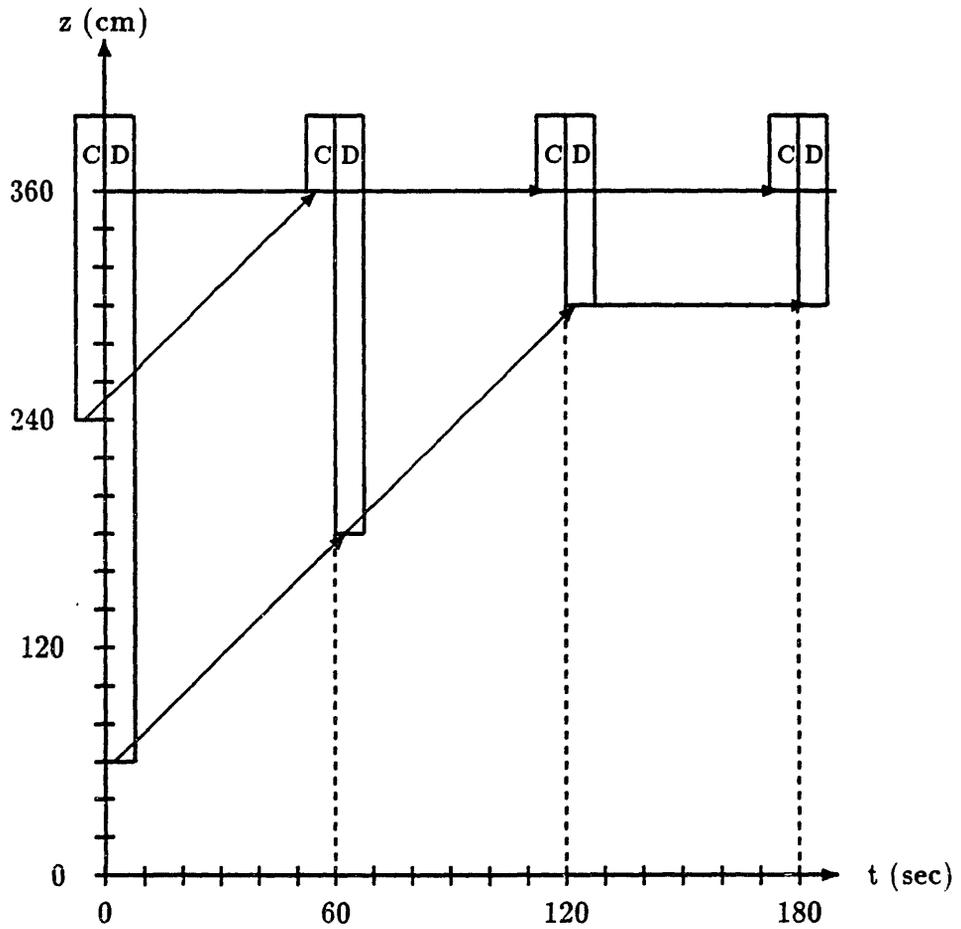
$$\text{Pressure} = 1.53 \times 10^7 \text{ Pa}$$

$$\text{Coolant Inlet Temp.} = 533 \text{ K}$$

$$\text{Initial Power} = 184.8 \text{ MW}_{\text{th}} \text{ (quarter-core)}$$

TRANSIENT #1: Rod Motion Transient Perturbation

Rod banks C & D are removed at a velocity of 2 cm/s



TRANSIENT #2: Coolant Inlet-Temperature Perturbation

The coolant inlet temperature is varied according to

$$T_{\text{inlet}}(t) = T_{\text{inlet}}(0) \exp(-t/\tau_1) + T_{\text{inlet}}(0) (1 - \exp(-t/\tau_2))$$

where

$$\tau_1 = 2.0 \text{ s}$$

$$\tau_2 = 2.206 \text{ s}$$

Control rod positions are the same as for transient #1 for $t \geq 120 \text{ s}$.

Appendix C

SELECTED RESULTS OF PROBLEM ANALYSES

- Figure C-1: 3-D LMW problem without feedback, comparison of initial static solutions.
- Figure C-2: 3-D LMW problem with feedback, comparison of initial static solutions.
- Figure C-3: 2-D LRA problem without feedback, comparison of initial static solutions.
- Figure C-4: 3-D LMW problem transient problem, normalized power distributions and fuel temperatures.
- Figure C-5: 3-D LRA problem without feedback, comparison of initial static solutions.
- Figure C-6: 3-D LMW problem transient problem, normalized assembly power distributions and fuel temperatures.
- Figure C-7: PWR operational transient, comparison of initial static solutions.
- Figure C-8: PWR coolant inlet-temperature transient, comparison of initial static solutions.

3-D LMW Problem Without Feedback

Static Solution

Assembly Averaged Errors

				0.8592 +0.04% +0.32%	0.4341 -0.02% -0.68%
			1.1227 +0.02% +0.18%	0.9800 +0.03% +0.26%	0.6272 -0.01% -0.09%
		1.5897 +0.00% +0.06%	1.3961 +0.00% +0.07%	1.0833 +0.01% +0.09%	0.7082 -0.05% -0.48%
⊕	1.5544 +0.00% +0.10%	1.6547 +0.00% +0.06%	1.4402 +0.00% +0.04%	0.9802 +0.00% +0.05%	0.7267 -0.05% -0.50%

⊕ ⊕
 ⊕

Maximum and Average Errors

Node Maximum Error	Node Average Error	Assembly Maximum Error	Assembly Average Error
—	—	—	—
+0.16 %	+0.02 %	-0.05 %	+0.01 %
-1.20 %	+0.29 %	-0.68 %	+0.15 %

Figure C-1: 3-D LMW problem without feedback, comparison of initial static solutions.

3-D LMW Problem With Feedback Static Solution

Assembly Averaged Errors

				0.9087 +0.06% +0.41%	0.4745 -0.01% -0.61%
Reference $5 \times 5 \times 10$ cm		1.1128	1.0081	0.6709	
$10 \times 10 \times 10$ cm, Errors		+0.02%	+0.04%	-0.01%	
$20 \times 20 \times 20$ cm, Errors		+0.18%	+0.31%	-0.05%	
		1.4933 -0.00% +0.03%	1.3505 +0.00% +0.06%	1.0933 +0.02% +0.10%	0.7461 -0.06% -0.49%
⊕ —	1.4283 -0.00% +0.06%	1.5383 -0.01% +0.03%	1.3815 -0.00% +0.03%	0.9842 +0.00% +0.04%	0.7619 -0.07% -0.54%
⊕	⊕	⊕	⊕	⊕	⊕

Maximum and Average Errors

Node Maximum Error	Node Average Error	Assembly Maximum Error	Assembly Average Error
—	—	—	—
+0.18 %	+0.03 %	-0.07 %	+0.01 %
-1.24 %	+0.33 %	-0.61 %	+0.15 %

Figure C-2: 3-D LMW problem with feedback, comparison of initial static solutions.

Figure C-4: 2-D LRA transient problem, normalized power distributions and fuel temperatures.

NORMALIZED POWER DISTRIBUTION AT TIME = 0.00000e+00										
TOTAL CORE POWER = 1.75500e-02										
AVERAGE POWER DENSITY = 1.00000e-06										
Y =	9	8	7	6	5	4	3	2	1	
Y = 9	9.1607e-01	8.6004e-01	8.1956e-01	8.4608e-01	9.2637e-01	9.6560e-01	8.3880e-01	0.0000e+00	0.0000e+00	0.0000e+00
Y = 8	1.4772e+00	1.2790e+00	1.1688e+00	1.2179e+00	1.4221e+00	1.6779e+00	1.6187e+00	1.3247e+00	0.0000e+00	0.0000e+00
Y = 7	1.6697e+00	1.1488e+00	9.6524e-01	1.0209e+00	1.3377e+00	2.0626e+00	2.1719e+00	1.6187e+00	8.3880e-01	8.3880e-01
Y = 6	1.3945e+00	9.4014e-01	7.8306e-01	8.4363e-01	1.1521e+00	1.8635e+00	2.0626e+00	1.6779e+00	9.6560e-01	9.6560e-01
Y = 5	7.9104e-01	6.7352e-01	6.2010e-01	6.8022e-01	8.6789e-01	1.1521e+00	1.3377e+00	1.4221e+00	9.2637e-01	9.2637e-01
Y = 4	5.1326e-01	4.9243e-01	4.9427e-01	5.5471e-01	6.8022e-01	8.4363e-01	1.0209e+00	1.2179e+00	8.4608e-01	8.4608e-01
Y = 3	4.1511e-01	4.0886e-01	4.2619e-01	4.9427e-01	6.2010e-01	7.8306e-01	9.6524e-01	1.1688e+00	8.1956e-01	8.1956e-01
Y = 2	4.4293e-01	4.0256e-01	4.0886e-01	4.9243e-01	6.7352e-01	9.4014e-01	1.1488e+00	1.2790e+00	8.6004e-01	8.6004e-01
Y = 1	6.2046e-01	4.4293e-01	4.1511e-01	5.1326e-01	7.9104e-01	1.3945e+00	1.6697e+00	1.4772e+00	9.1607e-01	9.1607e-01
I =	1	2	3	4	5	6	7	8	9	
NODE AVERAGED FUEL TEMPERATURES AT TIME = 0.00000e+00										
REACTOR AVERAGED FUEL TEMPERATURE = 3.00000e+02										
Y = 9	3.0000e+02									
Y = 8	3.0000e+02									
Y = 7	3.0000e+02									
Y = 6	3.0000e+02									
Y = 5	3.0000e+02									
Y = 4	3.0000e+02									
Y = 3	3.0000e+02									
Y = 2	3.0000e+02									
Y = 1	3.0000e+02									
I =	1	2	3	4	5	6	7	8	9	

Figure C-4: 2-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 4.00000e-01										
TOTAL CORE POWER = 2.42727e-02										
AVERAGE POWER DENSITY = 1.38306e-06										
Y =	9	8	7	6	5	4	3	2	1	X =
Y = 9	8.3928e-01	7.9623e-01	7.7575e-01	8.2562e-01	9.3206e-01	9.9772e-01	9.8816e-01	0.0000e+00	0.0000e+00	0.0000e+00
Y = 8	1.3522e+00	1.1830e+00	1.1067e+00	1.1913e+00	1.4382e+00	1.7470e+00	1.7375e+00	1.4756e+00	0.0000e+00	0.0000e+00
Y = 7	1.5268e+00	1.0610e+00	9.1439e-01	1.0020e+00	1.3613e+00	2.1625e+00	2.3603e+00	1.8874e+00	1.0167e+00	1.0167e+00
Y = 6	1.2747e+00	8.6600e-01	7.4182e-01	8.2856e-01	1.1733e+00	1.9543e+00	2.2405e+00	1.9486e+00	1.1547e+00	1.1547e+00
Y = 5	7.2330e-01	6.2215e-01	5.8645e-01	6.6459e-01	8.7554e-01	1.1944e+00	1.4240e+00	1.5523e+00	1.0247e+00	1.0247e+00
Y = 4	4.6894e-01	4.5426e-01	4.6503e-01	5.3548e-01	6.7276e-01	8.5127e-01	1.0468e+00	1.2617e+00	8.8145e-01	8.8145e-01
Y = 3	3.7788e-01	3.7540e-01	3.9755e-01	4.6971e-01	5.9845e-01	7.6451e-01	9.5086e-01	1.1581e+00	8.1481e-01	8.1481e-01
Y = 2	4.0097e-01	3.6715e-01	3.7797e-01	4.6171e-01	6.3763e-01	8.9493e-01	1.0987e+00	1.2289e+00	8.2880e-01	8.2880e-01
Y = 1	5.5981e-01	4.0225e-01	3.8183e-01	4.7810e-01	7.4266e-01	1.3141e+00	1.5781e+00	1.4005e+00	8.7037e-01	8.7037e-01
X = 1 2 3 4 5 6 7 8 9										
NODE AVERAGED FUEL TEMPERATURES AT TIME = 4.00000e-01										
REACTOR AVERAGED FUEL TEMPERATURE = 3.00000e+02										
Y = 9	3.0000e+02									
Y = 8	3.0000e+02									
Y = 7	3.0000e+02									
Y = 6	3.0000e+02									
Y = 5	3.0000e+02									
Y = 4	3.0000e+02									
Y = 3	3.0000e+02									
Y = 2	3.0000e+02									
Y = 1	3.0000e+02									
X = 1 2 3 4 5 6 7 8 9										

Figure C-4: 2-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 8.00000e-01										
TOTAL CORE POWER = 5.38370e-02										
AVERAGE POWER DENSITY = 3.06764e-06										
Y = 9	7.3009e-01	7.0539e-01	7.1343e-01	7.9637e-01	9.4010e-01	1.0433e+00	9.5831e-01	0.0000e+00	0.0000e+00	0.0000e+00
Y = 8	1.1744e+00	1.0464e+00	1.0183e+00	1.1535e+00	1.4609e+00	1.8449e+00	1.9063e+00	1.8902e+00	0.0000e+00	0.0000e+00
Y = 7	1.3235e+00	9.3637e-01	8.4205e-01	9.7481e-01	1.3945e+00	2.3037e+00	2.6271e+00	2.2691e+00	1.2744e+00	1.2744e+00
Y = 6	1.1042e+00	7.6541e-01	6.6314e-01	8.0702e-01	1.2033e+00	2.0827e+00	2.4923e+00	2.3323e+00	1.4271e+00	1.4271e+00
Y = 5	6.2696e-01	5.4906e-01	5.3859e-01	6.4219e-01	8.8628e-01	1.2541e+00	1.5464e+00	1.7369e+00	1.1652e+00	1.1652e+00
Y = 4	4.0591e-01	3.9997e-01	4.2352e-01	5.0811e-01	6.6218e-01	8.6207e-01	1.0835e+00	1.3240e+00	9.3190e-01	9.3190e-01
Y = 3	3.2496e-01	3.2773e-01	3.5683e-01	4.3470e-01	5.6766e-01	7.3813e-01	9.3043e-01	1.1429e+00	8.0814e-01	8.0814e-01
Y = 2	3.4137e-01	3.1680e-01	3.3413e-01	4.1809e-01	5.8661e-01	8.3069e-01	1.0278e+00	1.1578e+00	7.8450e-01	7.8450e-01
Y = 1	4.7358e-01	3.4434e-01	3.3451e-01	4.2811e-01	6.7390e-01	1.1998e+00	1.4475e+00	1.2913e+00	8.0548e-01	8.0548e-01
X = 1	2	3	4	5	6	7	8	9		
NODE AVERAGED FUEL TEMPERATURES AT TIME = 8.00000e-01										
REACTOR AVERAGED FUEL TEMPERATURE = 3.00000e+02										
Y = 9	3.0000e+02									
Y = 8	3.0000e+02									
Y = 7	3.0000e+02									
Y = 6	3.0000e+02									
Y = 5	3.0000e+02									
Y = 4	3.0000e+02									
Y = 3	3.0000e+02									
Y = 2	3.0000e+02									
Y = 1	3.0000e+02									
X = 1	2	3	4	5	6	7	8	9		

Figure C-4: 2-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 1.20000e+00										
TOTAL CORE POWER = 1.20896e+01										
AVERAGE POWER DENSITY = 6.88866e-04										
Y = 9	5.9313e-01	5.9057e-01	6.3243e-01	7.5471e-01	9.4318e-01	1.09335e+00	1.0414e+00	0.0000e+00	0.0000e+00	0.0000e+00
Y = 8	9.5202e-01	8.7435e-01	9.0408e-01	1.0995e+00	1.4797e+00	1.9576e+00	2.1144e+00	1.9652e+00	0.0000e+00	0.0000e+00
Y = 7	1.0696e+00	7.8025e-01	7.4940e-01	9.3627e-01	1.4283e+00	2.4703e+00	2.9627e+00	2.7792e+00	1.6291e+00	1.6291e+00
Y = 6	8.9183e-01	6.3740e-01	6.0859e-01	7.7695e-01	1.2352e+00	2.2357e+00	2.8112e+00	2.8461e+00	1.8008e+00	1.8008e+00
Y = 5	5.0774e-01	4.5830e-01	4.7821e-01	6.1231e-01	8.9635e-01	1.3257e+00	1.7009e+00	1.9774e+00	1.3505e+00	1.3505e+00
Y = 4	3.2854e-01	3.3300e-01	3.7165e-01	4.7284e-01	6.4691e-01	8.7334e-01	1.1279e+00	1.4013e+00	9.9478e-01	9.9478e-01
Y = 3	2.6036e-01	2.6929e-01	3.0632e-01	3.9040e-01	5.2765e-01	7.0284e-01	9.0187e-01	1.1201e+00	7.9657e-01	7.9657e-01
Y = 2	2.6892e-01	2.5535e-01	2.8002e-01	3.6319e-01	5.2117e-01	7.4701e-01	9.3444e-01	1.0631e+00	7.2452e-01	7.2452e-01
Y = 1	3.6864e-01	2.7375e-01	2.7619e-01	3.6527e-01	5.8573e-01	1.0509e+00	1.2770e+00	1.1476e+00	7.1935e-01	7.1935e-01
X = 1	2	3	4	5	6	7	8	9		
MODE AVERAGED FUEL TEMPERATURES AT TIME = 1.20000e+00										
REACTOR AVERAGED FUEL TEMPERATURE = 3.00000e+02										
Y = 9	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 8	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 7	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 6	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 5	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 4	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 3	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 2	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
Y = 1	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02	3.0000e+02
X = 1	2	3	4	5	6	7	8	9		

Figure C-4: 2-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 1.40000e+00

TOTAL CORE POWER = 1.04308e+07
 AVERAGE POWER DENSITY = 5.94348e+02

Y = 9	5.3697e-01	5.4213e-01	5.9543e-01	7.3068e-01	9.3483e-01	1.1042e+00	1.0889e+00	0.0000e+00	0.0000e+00
Y = 8	8.6153e-01	8.0267e-01	8.5265e-01	1.0684e+00	1.4740e+00	1.9894e+00	2.1941e+00	2.0858e+00	0.0000e+00
Y = 7	9.6670e-01	7.1598e-01	7.0845e-01	9.1379e-01	1.4310e+00	2.5231e+00	3.1013e+00	3.0328e+00	1.8145e+00
Y = 6	8.0841e-01	5.8534e-01	5.7628e-01	7.6001e-01	1.2399e+00	2.2871e+00	2.9463e+00	3.1042e+00	1.9965e+00
Y = 5	4.6060e-01	4.2190e-01	4.5289e-01	5.9720e-01	8.9558e-01	1.3499e+00	1.7655e+00	2.0881e+00	1.4388e+00
Y = 4	2.9857e-01	3.0667e-01	3.5034e-01	4.5689e-01	6.3783e-01	8.7476e-01	1.1441e+00	1.4327e+00	1.0210e+00
Y = 3	2.3593e-01	2.4683e-01	2.8615e-01	3.7160e-01	5.0938e-01	6.8547e-01	8.8659e-01	1.1063e+00	7.8852e-01
Y = 2	2.4193e-01	2.3213e-01	2.5877e-01	3.4045e-01	4.9258e-01	7.0912e-01	8.9106e-01	1.0180e+00	6.9537e-01
Y = 1	3.2964e-01	2.4728e-01	2.5348e-01	3.3939e-01	5.4743e-01	8.865e-01	1.1985e+00	1.0810e+00	6.7898e-01

X = 1 2 3 4 5 6 7 8 9

NODE AVERAGED FUEL TEMPERATURES AT TIME = 1.40000e+00

REACTOR AVERAGED FUEL TEMPERATURE = 3.07939e+02

Y = 9	3.0428e+02	3.0432e+02	3.0474e+02	3.0581e+02	3.0743e+02	3.0877e+02	3.0848e+02	3.0000e+02	3.0000e+02
Y = 8	3.0686e+02	3.0639e+02	3.0679e+02	3.0849e+02	3.1171e+02	3.1579e+02	3.1740e+02	3.1652e+02	3.0000e+02
Y = 7	3.0770e+02	3.0570e+02	3.0564e+02	3.0726e+02	3.1137e+02	3.2003e+02	3.2459e+02	3.2399e+02	3.1434e+02
Y = 6	3.0642e+02	3.0466e+02	3.0458e+02	3.0604e+02	3.0985e+02	3.1815e+02	3.2336e+02	3.2456e+02	3.1578e+02
Y = 5	3.0367e+02	3.0336e+02	3.0380e+02	3.0475e+02	3.0711e+02	3.1072e+02	3.1400e+02	3.1654e+02	3.1139e+02
Y = 4	3.0238e+02	3.0244e+02	3.0279e+02	3.0363e+02	3.0675e+02	3.0695e+02	3.0908e+02	3.1137e+02	3.0810e+02
Y = 3	3.0188e+02	3.0196e+02	3.0228e+02	3.0295e+02	3.0405e+02	3.0545e+02	3.0704e+02	3.0879e+02	3.0626e+02
Y = 2	3.0193e+02	3.0185e+02	3.0206e+02	3.0271e+02	3.0392e+02	3.0564e+02	3.0709e+02	3.0809e+02	3.0553e+02
Y = 1	3.0262e+02	3.0197e+02	3.0202e+02	3.0270e+02	3.0436e+02	3.0783e+02	3.0954e+02	3.0860e+02	3.0540e+02

X = 1 2 3 4 5 6 7 8 9

Figure C-4: 2-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 3.00000e+00

TOTAL CORE POWER = 1.73176e+06
 AVERAGE POWER DENSITY = 9.86756e+01

Y = 9	5.0041e-01	5.0427e-01	5.5203e-01	6.7682e-01	8.7232e-01	1.0511e+00	1.0548e+00	0.0000e+00	0.0000e+00
Y = 8	7.9908e-01	7.4271e-01	7.8605e-01	9.8367e-01	1.3694e+00	1.8946e+00	2.1939e+00	2.2421e+00	0.0000e+00
Y = 7	8.9931e-01	6.6297e-01	6.5311e-01	8.4201e-01	1.3341e+00	2.4244e+00	3.1672e+00	3.5009e+00	2.2518e+00
Y = 6	7.5569e-01	5.4635e-01	5.3561e-01	7.0581e-01	1.1645e+00	2.2115e+00	3.0238e+00	3.5799e+00	2.4418e+00
Y = 5	4.3683e-01	3.9948e-01	4.2674e-01	5.6121e-01	8.4679e-01	1.3041e+00	1.7794e+00	2.2192e+00	1.5890e+00
Y = 4	2.9017e-01	2.9687e-01	3.3639e-01	4.3526e-01	6.0725e-01	8.4164e-01	1.1250e+00	1.4444e+00	1.0530e+00
Y = 3	2.3661e-01	2.4531e-01	2.8008e-01	3.5825e-01	4.8685e-01	6.5444e-01	8.5288e-01	1.0780e+00	7.7942e-01
Y = 2	2.4984e-01	2.3622e-01	2.5712e-01	3.3033e-01	4.6991e-01	6.7075e-01	8.4208e-01	9.6824e-01	6.6842e-01
Y = 1	3.4607e-01	2.5508e-01	2.5376e-01	3.2983e-01	5.2086e-01	9.2746e-01	1.1254e+00	1.0163e+00	6.4374e-01

X = 1 2 3 4 5 6 7 8 9

MODE AVERAGED FUEL TEMPERATURES AT TIME = 3.00000e+00

REACTOR AVERAGED FUEL TEMPERATURE = 1.10428e+03

Y = 9	7.1627e+02	7.1955e+02	7.5948e+02	8.6308e+02	1.0230e+03	1.1636e+03	1.1540e+03	3.0000e+02	3.0000e+02
Y = 8	9.6574e+02	9.1899e+02	9.5529e+02	1.1198e+03	1.4361e+03	1.8551e+03	2.0641e+03	2.0514e+03	3.0000e+02
Y = 7	1.0484e+03	8.5197e+02	8.4400e+02	1.0008e+03	1.4046e+03	2.2824e+03	2.8237e+03	2.9529e+03	1.9604e+03
Y = 6	9.2658e+02	7.5304e+02	7.4428e+02	8.8524e+02	1.2609e+03	2.1033e+03	2.7042e+03	3.0134e+03	2.1100e+03
Y = 5	6.5976e+02	6.2890e+02	6.5164e+02	7.6286e+02	9.9660e+02	1.3632e+03	1.7252e+03	2.0395e+03	1.5273e+03
Y = 4	5.3611e+02	5.4186e+02	5.7488e+02	6.5688e+02	7.9813e+02	9.8753e+02	1.2109e+03	1.4582e+03	1.1372e+03
Y = 3	4.8982e+02	4.9752e+02	5.2698e+02	5.9232e+02	6.9888e+02	8.3684e+02	9.9776e+02	1.1775e+03	9.3143e+02
Y = 2	4.9802e+02	4.8832e+02	5.0707e+02	5.6897e+02	6.8580e+02	8.5323e+02	9.9515e+02	1.0975e+03	8.4650e+02
Y = 1	5.7272e+02	5.0228e+02	5.0378e+02	5.6852e+02	7.2855e+02	1.0672e+03	1.2330e+03	1.1419e+03	8.3171e+02

X = 1 2 3 4 5 6 7 8 9

3-D LRA Problem, Static Solution Comparison

Assembly Averaged Errors

									1.3270 +0.19% -0.10%						
								2.1594 +0.10% +0.61%	1.6202 +0.08% -0.04%	0.8456 +0.08% -0.74%					
							1.8510 +0.09% +0.68%	2.0497 +0.09% +0.64%	1.6785 +0.06% +0.01%	0.9711 +0.03% -0.50%					
						0.8643 +0.03% +0.40%	1.1520 +0.02% +0.02%	1.3390 +0.01% -0.08%	1.4211 +0.04% +0.11%	0.9320 -0.02% -0.56%					
							0.5527 +0.00% +0.32%	0.6784 +0.01% +0.23%	0.8436 +0.01% -0.00%	1.0225 -0.00% -0.14%	1.2212 +0.01% -0.25%	0.8527 -0.06% -0.74%			
							0.4243 -0.02% +0.35%	0.4924 -0.01% +0.32%	0.6184 -0.01% +0.22%	0.7830 -0.01% -0.02%	0.9671 -0.03% -0.19%	1.1729 -0.03% -0.34%	0.8267 -0.09% -0.84%		
							0.3999 -0.03% +0.57%	0.4071 -0.03% +0.35%	0.4907 -0.03% +0.27%	0.6707 -0.02% +0.37%	0.9402 -0.05% -0.04%	1.1511 -0.07% -0.23%	1.2812 -0.04% -0.17%	0.8671 -0.11% -0.80%	
							0.6125 +0.01% +1.17%	0.4407 -0.05% +0.39%	0.4135 -0.03% +0.30%	0.5123 -0.03% +0.11%	0.7906 -0.06% -0.00%	1.3852 +0.00% +0.62%	1.6607 -0.01% +0.51%	1.4812 -0.06% -0.27%	0.9242 -0.11% -0.86%

Reference 5 x 5 x 12.5(7.5) cm Powers
7.5 x 7.5 x 12.5(7.5) cm Errors
15 x 15 x 25(15) cm Errors

Maximum and Average Errors

Node Maximum Error	Node Average Error	Assembly Maximum Error	Assembly Average Error
+0.20 %	+0.04 %	+0.19 %	+0.03 %
-1.37 %	+0.40 %	+1.17 %	+0.22 %

Figure C-5: 3-D LRA problem, comparison of initial static solutions.

Figure C-6: 3-D LRA transient problem, normalized power distributions and fuel temperatures.

NORMALIZED POWER DISTRIBUTION AT TIME = 0.00000e+00

TOTAL CORE POWER = 5.26500e+00
 AVERAGE POWER DENSITY = 1.00000e-06

Y = 9	9.1625e-01	8.6023e-01	8.1983e-01	8.4656e-01	9.2717e-01	9.6672e-01	8.3994e-01	0.0000e+00	0.0000e+00
Y = 8	1.4774e+00	1.2791e+00	1.1689e+00	1.2183e+00	1.4232e+00	1.6797e+00	1.6209e+00	1.3268e+00	0.0000e+00
Y = 7	1.6893e+00	1.1482e+00	9.6491e-01	1.0209e+00	1.3383e+00	2.0643e+00	2.1743e+00	1.6209e+00	8.3994e-01
Y = 6	1.3937e+00	9.3940e-01	7.8240e-01	8.4319e-01	1.1522e+00	1.8645e+00	2.0643e+00	1.6797e+00	9.6672e-01
Y = 5	7.9015e-01	6.7265e-01	6.1930e-01	6.7956e-01	8.6751e-01	1.1522e+00	1.3383e+00	1.4232e+00	9.2717e-01
Y = 4	5.1233e-01	4.9152e-01	4.9340e-01	5.5392e-01	6.7956e-01	8.4319e-01	1.0209e+00	1.2183e+00	8.4656e-01
Y = 3	4.1414e-01	4.0791e-01	4.2528e-01	4.9340e-01	6.1930e-01	7.8240e-01	9.6491e-01	1.1689e+00	8.1983e-01
Y = 2	4.4181e-01	4.0154e-01	4.0791e-01	4.9152e-01	6.7265e-01	9.3940e-01	1.1482e+00	1.2791e+00	8.6023e-01
Y = 1	6.1892e-01	4.4181e-01	4.1414e-01	5.1233e-01	7.9015e-01	1.3937e+00	1.6893e+00	1.4774e+00	9.1625e-01

X = 1 2 3 4 5 6 7 8 9

MODE AVERAGED FUEL TEMPERATURES AT TIME = 0.00000e+00

REACTOR AVERAGED FUEL TEMPERATURE = 2.99999e+02

Y = 9	3.0000e+02								
Y = 8	3.0000e+02								
Y = 7	3.0000e+02								
Y = 6	3.0000e+02								
Y = 5	3.0000e+02								
Y = 4	3.0000e+02								
Y = 3	3.0000e+02								
Y = 2	3.0000e+02								
Y = 1	3.0000e+02								

X = 1 2 3 4 5 6 7 8 9

Figure C-6: 3-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 4.00000e-01

TOTAL CORE POWER = 6.78445e+00
 AVERAGE POWER DENSITY = 1.28859e-06

Y = 9	8.3577e-01	7.9203e-01	7.7022e-01	8.1836e-01	9.2382e-01	9.9082e-01	8.8614e-01	0.0000e+00	0.0000e+00
Y = 8	1.3466e+00	1.1769e+00	1.0988e+00	1.1813e+00	1.4268e+00	1.7387e+00	1.7410e+00	1.4978e+00	0.0000e+00
Y = 7	1.5201e+00	1.0553e+00	9.0764e-01	9.9345e-01	1.3516e+00	2.1559e+00	2.3752e+00	1.9577e+00	1.0800e+00
Y = 6	1.2686e+00	8.6299e-01	7.3604e-01	8.2133e-01	1.1651e+00	1.9489e+00	2.2554e+00	2.0184e+00	1.2185e+00
Y = 5	7.1949e-01	6.1818e-01	5.8166e-01	6.5842e-01	8.6842e-01	1.1889e+00	1.4271e+00	1.5706e+00	1.0437e+00
Y = 4	4.6618e-01	4.5112e-01	4.6114e-01	5.3040e-01	6.6672e-01	8.4540e-01	1.0434e+00	1.2628e+00	8.8427e-01
Y = 3	3.7563e-01	3.7276e-01	3.9426e-01	4.6534e-01	5.9300e-01	7.5831e-01	9.4480e-01	1.1529e+00	8.1198e-01
Y = 2	3.9886e-01	3.6484e-01	3.7509e-01	4.5775e-01	6.3217e-01	8.8779e-01	1.0908e+00	1.2212e+00	8.2395e-01
Y = 1	5.5712e-01	3.9991e-01	3.7905e-01	4.7419e-01	7.3666e-01	1.3040e+00	1.5687e+00	1.3911e+00	8.6479e-01

X = 1 2 3 4 5 6 7 8 9

NODE AVERAGED FUEL TEMPERATURES AT TIME = 4.00000e-01

REACTOR AVERAGED FUEL TEMPERATURE = 2.99939e+02

Y = 9	3.0000e+02								
Y = 8	3.0000e+02								
Y = 7	3.0000e+02								
Y = 6	3.0000e+02								
Y = 5	3.0000e+02								
Y = 4	3.0000e+02								
Y = 3	3.0000e+02								
Y = 2	3.0000e+02								
Y = 1	3.0000e+02								

X = 1 2 3 4 5 6 7 8 9

Figure C-6: 3-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 8.00000e-01

TOTAL CORE POWER = 2.67376e+05
 AVERAGE POWER DENSITY = 5.07836e-02

Y = 9	5.0479e-01	5.1323e-01	5.7119e-01	7.1194e-01	9.2404e-01	1.1055e+00	1.0935e+00	0.0000e+00	0.0000e+00
Y = 8	8.0962e-01	7.5971e-01	8.1856e-01	1.0430e+00	1.4618e+00	2.0015e+00	2.2437e+00	2.1738e+00	0.0000e+00
Y = 7	9.0745e-01	6.7703e-01	6.8037e-01	8.9377e-01	1.4239e+00	2.5479e+00	3.1924e+00	3.2321e+00	1.9791e+00
Y = 6	7.5652e-01	5.5316e-01	5.5333e-01	7.4358e-01	1.2345e+00	2.3106e+00	3.0336e+00	3.3033e+00	2.1645e+00
Y = 5	4.3212e-01	3.9862e-01	4.3399e-01	5.8251e-01	8.8762e-01	1.3572e+00	1.8024e+00	2.1633e+00	1.5036e+00
Y = 4	2.7979e-01	2.8928e-01	3.3457e-01	4.4261e-01	6.2806e-01	8.6860e-01	1.1479e+00	1.4481e+00	1.0361e+00
Y = 3	2.2042e-01	2.3193e-01	2.7180e-01	3.5654e-01	4.9327e-01	6.6895e-01	8.7105e-01	1.0919e+00	7.8007e-01
Y = 2	2.2509e-01	2.1705e-01	2.4403e-01	3.2373e-01	4.7117e-01	6.8102e-01	8.5919e-01	9.8517e-01	6.7419e-01
Y = 1	3.0585e-01	2.3047e-01	2.3813e-01	3.2117e-01	5.2051e-01	9.3758e-01	1.1452e+00	1.0357e+00	6.5144e-01

X = 1 2 3 4 5 6 7 8 9

MODE AVERAGED FUEL TEMPERATURES AT TIME = 8.00000e-01

REACTOR AVERAGED FUEL TEMPERATURE = 2.99999e+03

Y = 9	3.0000e+02								
Y = 8	3.0000e+02								
Y = 7	3.0000e+02								
Y = 6	3.0000e+02								
Y = 5	3.0000e+02								
Y = 4	3.0000e+02								
Y = 3	3.0000e+02								
Y = 2	3.0000e+02								
Y = 1	3.0000e+02								

X = 1 2 3 4 5 6 7 8 9

Figure C-6: 3-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 1.40000e+00									
TOTAL CORE POWER = 2.06243e+09									
AVERAGE POWER DENSITY = 3.91725e+02									
Y =	9	8	7	6	5	4	3	2	1
Y = 9	4.9345e-01	5.0090e-01	5.5557e-01	6.9008e-01	8.9587e-01	1.0783e+00	1.0737e+00	0.0000e+00	0.0000e+00
Y = 8	7.8867e-01	7.3853e-01	7.9263e-01	1.0060e+00	1.4104e+00	1.9469e+00	2.2256e+00	2.2288e+00	0.0000e+00
Y = 7	8.8670e-01	6.5812e-01	6.5830e-01	8.6147e-01	1.3745e+00	2.4874e+00	3.1932e+00	3.4087e+00	2.1530e+00
Y = 6	7.4147e-01	5.4011e-01	5.3779e-01	7.1982e-01	1.1984e+00	2.2637e+00	3.0427e+00	3.4824e+00	2.3418e+00
Y = 5	4.2607e-01	3.9241e-01	4.2545e-01	5.6832e-01	8.6512e-01	1.3313e+00	1.7976e+00	2.2091e+00	1.5652e+00
Y = 4	2.7970e-01	2.8847e-01	3.3180e-01	4.3617e-01	6.1487e-01	8.5492e-01	1.1395e+00	1.4551e+00	1.0545e+00
Y = 3	2.2450e-01	2.3499e-01	2.7274e-01	3.5476e-01	4.8772e-01	6.6005e-01	8.6179e-01	1.0876e+00	7.8390e-01
Y = 2	2.3350e-01	2.2320e-01	2.4754e-01	3.2411e-01	4.6723e-01	6.7221e-01	8.4724e-01	9.7488e-01	6.7172e-01
Y = 1	3.2089e-01	2.3909e-01	2.4282e-01	3.2228e-01	5.1656e-01	9.2667e-01	1.1291e+00	1.0214e+00	6.4604e-01

X =	1	2	3	4	5	6	7	8	9
Y = 9	4.5639e+02	4.5686e+02	4.7036e+02	5.0686e+02	5.6336e+02	6.1181e+02	6.0511e+02	3.0000e+02	3.0000e+02
Y = 8	5.5037e+02	5.3161e+02	5.4306e+02	6.0115e+02	7.1360e+02	8.6060e+02	9.2736e+02	9.1240e+02	3.0000e+02
Y = 7	5.8130e+02	5.0648e+02	5.0152e+02	5.5695e+02	7.0110e+02	8.1025e+02	1.1922e+03	1.2121e+03	8.6435e+02
Y = 6	5.3503e+02	4.6907e+02	4.6413e+02	5.1398e+02	6.4813e+02	9.4696e+02	1.1487e+03	1.2326e+03	9.1637e+02
Y = 5	4.3436e+02	4.2319e+02	4.2947e+02	4.6891e+02	5.5223e+02	6.8174e+02	8.0573e+02	9.0970e+02	7.2694e+02
Y = 4	3.8761e+02	3.8937e+02	4.0092e+02	4.3020e+02	4.8075e+02	5.4805e+02	6.2854e+02	7.1287e+02	5.9707e+02
Y = 3	3.7001e+02	3.7288e+02	3.8324e+02	4.0691e+02	4.4559e+02	4.9553e+02	5.5342e+02	6.1787e+02	5.2798e+02
Y = 2	3.7278e+02	3.6916e+02	3.7600e+02	3.9877e+02	4.4190e+02	5.0372e+02	5.5585e+02	5.9294e+02	5.0095e+02
Y = 1	4.0014e+02	3.7427e+02	3.7487e+02	3.9890e+02	4.5837e+02	5.8415e+02	6.4571e+02	6.1166e+02	4.9638e+02

NODE AVERAGED FUEL TEMPERATURES AT TIME = 1.40000e+00									
REACTOR AVERAGED FUEL TEMPERATURE = 6.46904e+02									
Y =	9	8	7	6	5	4	3	2	1
Y = 9	4.5639e+02	4.5686e+02	4.7036e+02	5.0686e+02	5.6336e+02	6.1181e+02	6.0511e+02	3.0000e+02	3.0000e+02
Y = 8	5.5037e+02	5.3161e+02	5.4306e+02	6.0115e+02	7.1360e+02	8.6060e+02	9.2736e+02	9.1240e+02	3.0000e+02
Y = 7	5.8130e+02	5.0648e+02	5.0152e+02	5.5695e+02	7.0110e+02	8.1025e+02	1.1922e+03	1.2121e+03	8.6435e+02
Y = 6	5.3503e+02	4.6907e+02	4.6413e+02	5.1398e+02	6.4813e+02	9.4696e+02	1.1487e+03	1.2326e+03	9.1637e+02
Y = 5	4.3436e+02	4.2319e+02	4.2947e+02	4.6891e+02	5.5223e+02	6.8174e+02	8.0573e+02	9.0970e+02	7.2694e+02
Y = 4	3.8761e+02	3.8937e+02	4.0092e+02	4.3020e+02	4.8075e+02	5.4805e+02	6.2854e+02	7.1287e+02	5.9707e+02
Y = 3	3.7001e+02	3.7288e+02	3.8324e+02	4.0691e+02	4.4559e+02	4.9553e+02	5.5342e+02	6.1787e+02	5.2798e+02
Y = 2	3.7278e+02	3.6916e+02	3.7600e+02	3.9877e+02	4.4190e+02	5.0372e+02	5.5585e+02	5.9294e+02	5.0095e+02
Y = 1	4.0014e+02	3.7427e+02	3.7487e+02	3.9890e+02	4.5837e+02	5.8415e+02	6.4571e+02	6.1166e+02	4.9638e+02

Figure C-6: 3-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 2.00000e+00

TOTAL CORE POWER = 9.39129e+08
 AVERAGE POWER DENSITY = 1.78372e+02

Y = 9	4.6057e-01	4.7058e-01	5.2823e-01	6.6564e-01	8.7659e-01	1.0713e+00	1.0848e+00	0.0000e+00	0.0000e+00
Y = 8	7.3505e-01	6.9273e-01	7.5302e-01	9.7057e-01	1.3825e+00	1.9415e+00	2.2697e+00	2.3325e+00	0.0000e+00
Y = 7	8.2545e-01	6.1698e-01	6.2570e-01	8.3253e-01	1.3519e+00	2.4922e+00	3.2894e+00	3.6575e+00	2.3546e+00
Y = 6	6.9195e-01	5.0706e-01	5.1204e-01	6.9697e-01	1.1789e+00	2.2713e+00	3.1344e+00	3.7354e+00	2.5496e+00
Y = 5	3.9876e-01	3.6958e-01	4.0576e-01	5.5014e-01	8.5020e-01	1.3298e+00	1.8318e+00	2.2952e+00	1.6440e+00
Y = 4	2.6303e-01	2.7269e-01	3.1661e-01	4.2079e-01	5.9984e-01	8.4384e-01	1.1386e+00	1.4681e+00	1.0709e+00
Y = 3	2.1205e-01	2.2266e-01	2.5985e-01	3.4006e-01	4.7027e-01	6.4036e-01	8.4174e-01	1.0686e+00	7.7364e-01
Y = 2	2.2113e-01	2.1161e-01	2.3519e-01	3.0847e-01	4.4522e-01	6.4120e-01	8.1051e-01	9.3647e-01	6.4772e-01
Y = 1	3.0422e-01	2.2666e-01	3.2023e-01	3.0539e-01	4.8899e-01	8.7712e-01	1.0698e+00	9.6977e-01	6.1526e-01

X = 1 2 3 4 5 6 7 8 9

NODE AVERAGED FUEL TEMPERATURES AT TIME = 2.00000e+00

REACTOR AVERAGED FUEL TEMPERATURE = 8.57875e+02

Y = 9	5.4055e+02	5.4257e+02	5.6597e+02	6.2641e+02	7.1955e+02	8.0111e+02	7.9501e+02	3.0000e+02	3.0000e+02
Y = 8	6.8477e+02	6.5785e+02	6.7937e+02	7.7539e+02	9.5981e+02	1.2028e+03	1.3223e+03	1.3123e+03	3.0000e+02
Y = 7	7.3225e+02	6.1894e+02	6.1473e+02	7.0824e+02	9.4115e+02	1.4505e+03	1.7809e+03	1.8294e+03	1.2575e+03
Y = 6	6.6148e+02	5.6142e+02	5.5669e+02	6.3683e+02	8.5725e+02	1.3458e+03	1.6908e+03	1.8633e+03	1.3432e+03
Y = 5	5.0712e+02	4.8940e+02	5.0277e+02	5.6748e+02	7.0329e+02	9.1584e+02	1.1245e+03	1.3047e+03	1.0081e+03
Y = 4	4.3549e+02	4.3888e+02	4.5811e+02	5.0576e+02	5.8778e+02	6.9751e+02	8.2693e+02	9.6983e+02	7.8385e+02
Y = 3	4.0853e+02	4.1307e+02	4.3023e+02	4.6818e+02	5.3003e+02	6.1011e+02	7.0345e+02	8.0771e+02	6.6510e+02
Y = 2	4.1291e+02	4.0754e+02	4.1859e+02	4.5456e+02	5.2236e+02	6.1950e+02	7.0195e+02	7.6136e+02	6.1723e+02
Y = 1	4.5532e+02	4.1537e+02	4.1661e+02	4.5426e+02	5.4705e+02	7.4321e+02	8.3958e+02	7.8718e+02	6.0758e+02

X = 1 2 3 4 5 6 7 8 9

Figure C-6: 3-D LRA transient problem, normalized power distributions and fuel temperatures, continued.

NORMALIZED POWER DISTRIBUTION AT TIME = 3.00000e+00

TOTAL CORE POWER = 2.09843e+08
 AVERAGE POWER DENSITY = 3.98562e+01

Y = 9	4.8220e-01	4.8879e-01	5.4103e-01	6.7164e-01	8.7439e-01	1.0608e+00	1.0693e+00	0.0000e+00	0.0000e+00
Y = 8	7.6984e-01	7.1974e-01	7.7088e-01	9.7754e-01	1.3755e+00	1.9170e+00	2.2307e+00	2.2859e+00	0.0000e+00
Y = 7	8.6540e-01	6.4166e-01	6.4011e-01	8.3718e-01	1.3421e+00	2.4564e+00	3.2247e+00	3.5754e+00	2.3003e+00
Y = 6	7.2620e-01	5.2795e-01	5.2423e-01	7.0108e-01	1.1706e+00	2.2393e+00	3.0765e+00	3.6534e+00	2.4928e+00
Y = 5	4.1905e-01	3.8534e-01	4.1653e-01	5.5541e-01	8.4764e-01	1.3154e+00	1.8037e+00	2.2550e+00	1.6150e+00
Y = 4	2.7738e-01	2.8536e-01	3.2677e-01	4.2796e-01	6.0308e-01	8.4189e-01	1.1307e+00	1.4550e+00	1.0612e+00
Y = 3	2.2506e-01	2.3458e-01	2.7034e-01	3.4933e-01	4.7846e-01	6.4724e-01	8.4715e-01	1.0733e+00	7.7661e-01
Y = 2	2.3645e-01	2.2464e-01	2.4666e-01	3.1980e-01	4.5800e-01	6.5680e-01	8.2711e-01	9.5337e-01	6.5878e-01
Y = 1	3.2688e-01	2.4179e-01	2.4261e-01	3.1817e-01	5.0574e-01	9.0385e-01	1.0996e+00	9.9478e-01	6.3059e-01

X = 1 2 3 4 5 6 7 8 9

NODE AVERAGED FUEL TEMPERATURES AT TIME = 3.00000e+00

REACTOR AVERAGED FUEL TEMPERATURE = 9.53904e+02

Y = 9	5.7836e+02	5.8103e+02	6.0881e+02	6.7996e+02	7.8963e+02	8.8641e+02	8.8116e+02	3.0000e+02	3.0000e+02
Y = 8	7.4512e+02	7.1448e+02	7.4041e+02	8.5338e+02	1.0700e+03	1.3571e+03	1.5023e+03	1.4970e+03	3.0000e+02
Y = 7	8.0006e+02	6.6940e+02	6.6545e+02	7.7308e+02	1.0489e+03	1.6484e+03	2.0212e+03	2.1183e+03	1.4434e+03
Y = 6	7.1836e+02	8.0292e+02	5.9821e+02	6.9480e+02	9.5125e+02	1.5282e+03	1.9391e+03	2.1585e+03	1.5448e+03
Y = 5	5.3992e+02	5.1966e+02	5.3571e+02	6.1175e+02	7.7123e+02	1.0216e+03	1.2699e+03	1.4866e+03	1.1384e+03
Y = 4	4.5716e+02	4.3140e+02	4.8390e+02	5.3975e+02	6.3593e+02	7.6506e+02	9.1770e+02	1.0887e+03	8.6911e+02
Y = 3	4.2607e+02	4.3140e+02	4.5148e+02	4.9580e+02	5.6803e+02	6.6166e+02	7.7105e+02	8.9343e+02	7.2714e+02
Y = 2	4.3127e+02	4.2503e+02	4.3791e+02	4.7974e+02	5.5855e+02	6.7149e+02	7.6754e+02	8.3705e+02	6.6955e+02
Y = 1	4.8064e+02	4.3416e+02	4.3557e+02	4.7926e+02	5.8691e+02	8.1458e+02	9.2651e+02	8.6589e+02	6.5749e+02

X = 1 2 3 4 5 6 7 8 9

PWR Operational Transient Initial Static Solution

Assembly Averaged Errors

				0.7981 +0.20% +0.92% +0.73%	0.6230 -0.13% -1.78% -0.60%			
				0.6420 +0.07% +0.31% -0.01%	0.8005 -0.04% +0.07% +0.42%	1.0061 +0.15% +0.22% +0.93%		
CONQUEST Very Fine Mesh, Power	0.8905	0.8767	1.1182	1.1807	0.9598			
CONQUEST Fine Mesh, Error	+0.04%	-0.08%	+0.19%	-0.03%	+0.20%			
CONQUEST Coarse Mesh, Error	+0.31%	-0.15%	+1.16%	-0.07%	+0.39%			
QUANDRY Coarse Mesh, Error	-0.24%	-0.43%	+0.44%	+0.56%	+0.92%			
	0.8841 +0.16% +1.28% -0.21%	0.8464 -0.12% -0.38% -0.96%	1.0653 +0.15% +1.15% +0.08%	1.1106 -0.13% -0.61% -0.39%	1.4066 +0.13% +0.86% +0.64%	1.3392 -0.08% -0.43% +0.82%		
	0.8336 +0.17% +1.01% -0.71%	0.8022 -0.12% -0.25% -1.16%	0.9192 +0.15% +1.04% -0.39%	0.8763 -0.15% -0.45% -0.71%	1.2223 +0.10% +0.66% +0.06%	1.3418 -0.24% -1.07% -0.07%	1.2172 -0.25% -1.62% +0.03%	
€	0.5311 +0.07% +0.39% -0.90%	0.6922 -0.11% +0.14% -1.01%	0.8809 +0.18% +0.92% -0.87%	0.7653 -0.12% +0.02% -0.73%	0.6885 +0.01% +0.13% -0.29%	1.0748 -0.17% -0.36% -0.19%	1.3370 +0.01% -0.37% -0.23%	1.2131 -0.31% -2.10% -0.34%
€								
€								

Maximum and Average Errors

Node Maximum Error	Node Average Error	Assembly Maximum Error	Assembly Average Error
—	—	—	—
-0.44 %	+0.14 %	-0.31 %	+0.11 %
-3.05 %	+0.69 %	-2.10 %	+0.58 %
-1.40 %	+0.53 %	-1.16 %	+0.44 %

Figure C-6: PWR operational transient, comparison of initial static solutions.

PWR Coolant Inlet-Temperature Transient Initial Static Solution

Assembly Averaged Errors

					0.8211 0.8230 +0.24%	0.5942 0.5876 -1.12%		
				1.0028 1.0101 +0.73%	0.8626 0.8595 -0.36%	0.9362 0.9297 -0.70%		
QUANDRY Normalized Power		0.9726	0.9722	1.0874	1.0319	0.8015		
CONQUEST Normalized Power		0.9754	0.9730	1.0947	1.0258	0.7975		
Difference		+0.29%	+0.08%	+0.67%	-0.59%	-0.50%		
		1.0175 1.0285 +1.09%	0.9079 0.9102 +0.26%	1.0847 1.0932 +0.78%	1.0234 1.0210 -0.24%	1.1963 1.1991 +0.23%	1.0992 1.0865 -1.16%	
		1.0085 1.0202 +1.16%	0.9143 0.9181 +0.42%	1.0468 1.0578 +1.05%	0.9874 0.9880 +0.06%	1.1868 1.1936 +0.58%	1.1715 1.1609 -0.90%	1.0102 0.9948 -1.53%
⊕	0.9773 0.9890 +1.20%	0.9019 0.9068 +0.55%	1.0214 1.0333 +1.16%	0.9438 0.9467 +0.31%	1.0993 1.1076 +0.76%	1.1291 1.1269 -0.19%	1.2314 1.2311 -0.03%	1.0274 1.0112 -1.58%
⊕								⊕

Maximum and Average Errors

Node Maximum Error	Node Average Error	Assembly Maximum Error	Assembly Average Error
—	—	—	—
-2.19 %	+0.64 %	-1.58 %	+0.43 %

Figure C-7: PWR coolant inlet-temperature transient, comparison of initial static solutions.