SIMULATING LIGHT-WATER-REACTOR START-UP TRANSIENTS USING A POINT-KINETICS MODEL WITH A PRECALCULATED REACTIVITY-TABLE

by

S. A. PARRA

B.S. in Nuclear Engineering
University of Arizona
(1989)

Submitted to the Department of
Nuclear Engineering
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

in Nuclear Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
· February 1991

Signature of Author...
Santiago A. Parra
Department of Nuclear Engineering
December 1990

Certified by...
Allan F. Henry
Professor, Nuclear Engineering
Thesis Co–Supervisor

Certified by...
David D. Lanning
Professor, Nuclear Engineering
Thesis Co–Supervisor

SIMULATING LIGHT-WATER-REACTOR START-UP TRANSIENTS USING A POINT-KINETICS MODEL WITH A PRECALCULATED REACTIVITY-TABLE

by

S. A. PARRA

Submitted to the Department of Nuclear Engineering on
December 31, 1990 in partial fulfillment of the requirements
for the Degree of Master of Science in Nuclear Engineering

ABSTRACT

The objective of this research is to add a precalculated "reactivity-worth table" into the QUANDRY code so that during any transient involving control-rod motion, the correct reactivity can be inferred from this reactivity table and used in the point-kinetics calculations, thereby, reducing errors due to flux-shape changes.

Static criticality calculations corresponding to different control-rods position expected during the transient are performed. A table of reactivity versus control-rod positions is created and used during the transient. However, the thermal-hydraulic feedback effects are still calculated by the QUANDRY code. Therefore, the total reactivity at any time during the transient is, according to the point-kinetics model, the sum of the reactivity due to control-rod motion (read from the reactivity table) and the reactivity due to the thermal-hydraulic feedback effects computed by the code according to Perturbation Theory.

This new option in the point-kinetics model is then applied to a large Pressurized Water Reactor start-up transient involving an external source and thermal-hydraulic feedback effects. The results are compared with the full space- and time-dependent results for the same transient.

This modified point-kinetics model is shown to be more accurate in modeling a start-up transient than the unmodified perturbation theory point-kinetics model. The modified model, however, predicts either a lower or a higher final power depending on whether the initial or final flux-shape is employed in defining the point-kinetics parameters. Furthermore, it was found that the results depend on the weight functions used to define the point-kinetics parameters. The results show that using the final configuration of the reactor to define the weight functions give very inaccurate results no matter which flux-shape is used.

Thesis Co-Supervisor: Dr. Allan F. Henry
Title: Professor of Nuclear Engineering

Thesis Co-Supervisor: Dr. David D. Lanning
Title: Professor of Nuclear Engineering

ACKNOWLEDGMENTS

This research was undertaken from the advice of Dr. Allan F. Henry which is gratefully appreciated. Without his advice and patience, I might not have survived here for very long. Thanks to Dr. David D. Lanning who served as my thesis reader and Co-Supervisor. Also, special thanks is hereby given to Mr. Robert P. Jacqmin for his technical help and advice with the QUANDRY computer code and editing this thesis.

This research was sponsored by a fellowship from the Ford Foundation and the National Science Foundation.

TABLE OF CONTENTS

Page

Abstract 2

Acknowledgments. 3

Table of Contents. 4

List of Figures. 7

List of Tables 9

Dedication 10

Chapter 1. Introduction 11

1.1 Background. 11

1.2 Objective 13

1.3 Summary 14

Chapter 2. QUANDRY Theoretical Framework. 15

2.1 Introduction. 15

2.2 Static Case 16

2.2.1 Regular Eigenvalue-Problem. 17

2.2.2 Adjoint Eigenvalue-Problem. 19

2.2.3 External-Source Problem 20

2.3 Transient Case. 22

2.3.1 Time-Dependent Problem. 22

2.3.2 Point-Kinetics Parameters 26

2.3.3 Quasi-Static Option 27

2.4 Summary 30

Chapter 3. The New Reactivity Table Option in the QUANDRY Code 31

3.1 Introduction. 31

Page

3.2 The New Algorithm of the QUANDRY Transient Module. 32

3.3 Modification in the QUANDRY code. 37

3.3.1 Subroutine INPUTR 37

3.3.2 Subroutine RODTAB 37

3.4 Testing of the Reactivity Table Option. 37

3.4.1 Description of the Reactor-Core Model 38

3.4.2 Description of the Start-Up Transient 38

3.4.3 Computed Initial Point-Kinetics Parameters. 40

3.4.4 Test Results. 41

3.5 Summary 46

Chapter 4. Application: Point-kinetics versus Space-Kinetics 47

4.1 Introduction. 47

4.2 Generating the Reactivity Table 48

4.3 Initial Flux-Shape Results. 53

4.4 Final Flux-Shape Results. 62

4.5 Computing-Time Requirements 72

4.6 Summary 73

Chapter 5. Conclusion 74

5.1 Summary of the Modifications. 74

5.2 Summary of the Investigation. 74

5.3 Recommendations for Future Research 75

Page

References 77

Appendix 78

A. Subroutine INPUTR. 79

B. Subroutine RODTAB. 81

C. Description of the Salem-1 Model Reactor Core 83

I. Geometry. 83

II. Material Properties 85

III. Albedo-Type Boundary Conditions 86

IV. Delayed-Neutron Data. 86

V. Thermal-Hydraulic Data and Feedback Coefficients. 87

LIST OF FIGURES

Figure Page

1. The Original Algorithm of the QUANDRY Transient Module. 33

2. The Modified Algorithm of the QUANDRY Transient Module. 35

3. Reactivity versus Time using the Correct Reactivity from the Reference Calculation for the Reactivity-Table Option and Final-Adjoint Weighting 43

4. Normalized Amplitude versus Time using the Correct Reactivity from the Reference Calculation for the Reactivity-Table Option and Final-Adjoint Weighting 44

5. Normalized Power versus Time using the Correct Reactivity from the Reference Calculation for the Reactivity-Table Option and Final-Adjoint Weighting 45

6. Control-Rod Reactivity-Worth versus Control-Rod Position. 51

7. Reactivity versus Time using the Initial-Adjoint Weighting and the Initial Flux-Shape 55

8. Reactivity versus Time using the Final-Adjoint Weighting and the Initial Flux-Shape 56

9. Normalized Amplitude versus Time using the Initial-Adjoint Weighting and the Initial Flux-Shape. 58

10. Normalized Amplitude versus Time using the Final-Adjoint Weighting and the Initial Flux-Shape. 59

11. Normalized Power versus Time using the Initial-Adjoint Weighting and the Initial Flux-Shape. 60

Figure Page

12. Normalized Power versus Time using the Final-Adjoint Weighting and the Initial Flux-Shape. 61

13. Reactivity versus Time using the Initial-Adjoint Weighting and the Final Flux-Shape 66

14. Reactivity versus Time using the Final-Adjoint Weighting and the Final Flux-Shape. 67

15. Normalized Amplitude versus Time using the Initial-Adjoint Weighting and the Final Flux-Shape. 68

16. Normalized Amplitude versus Time using the Final-Adjoint Weighting and the Final Flux-Shape 69

17. Normalized Power versus Time using the Initial-Adjoint Weighting and the Final Flux-Shape. 70

18. Normalized Power versus Time using the Final-Adjoint Weighting and the Final Flux-Shape. 71

LIST OF TABLES

Tables Page

1. A Time Description of the Light Water Reactor Start-Up Transient 40

2. Reactivity Worth of the Control-Rods versus Control-Rod Position. 49

3. Reactivity Worth of the Control-Rods versus Time during the Start-Up Transient. 52

4. Figure Reference for the Results of the Point-Kinetics Calculations using the Initial Flux-Shape 54

5. Figure Reference for the Results of the Point-Kinetics Calculations using the Final Flux-Shape 65

6. Total CPU-Time Requirements. 72

DEDICATION

To Jose Pedro Parra,

my father who died in a car accident

on Saturday, July 1, 1989.

Chapter 1

INTRODUCTION

1.1 BACKGROUND

Accurate predictions of the spatial power distribution in large nuclear reactor cores under transient conditions are essential to the design, analysis, and safe operation of nuclear power plants. These predictions require detailed knowledge of the neutron density (or flux) as a function of space, energy, and time. This information can be obtained by solving the time-dependent, multi-dimensional, few-group, neutron diffusion equations. However, because of the complexity of analyzing Light-Water Reactors (LWRs) and because of the high computing-cost associated with the detailed numerical solution of these equations, the nuclear industry still makes extensive use of the simplest model of all, called the "point-kinetics model". This point-kinetics model consists of a set of seven first-order differential equations which depend only on time. Although formally exact in form, these equations can be solved in practice only when certain simplifying assumptions relative to the detailed neutron flux-shape and thermal-hydraulic conditions of the reactor are made. In addition, they give only a global picture of the reactor response to perturbations.

In the past ten years, "nodal methods" have emerged as an accurate and efficient technique for predicting criticality as well as the kinetics behavior of LWRs. Computer codes based on these coarse-mesh methods and on diffusion theory have been written and validated by comparisons with fine-mesh finite-difference codes. One such nodal code is the QUANDRY (Quadratic Analytic Nodal Diffusion Theory) code which has been developed at the Massachusetts Institute of Technology (MIT) [1]. The QUANDRY code solves the two-group, multi-dimensional, static and transient diffusion equations as derived from the Analytical Nodal Method. The availability of these nodal codes and of more efficient digital computers make it possible today to compute at a reasonable cost detailed space- and energy-dependent neutron distributions in a reactor following predetermined perturbations from equilibrium conditions. This detailed information can then be "averaged out" and compared with the results of the point-kinetics calculations.

One reason for the continuing success of the point-kinetics model is its simplicity. The model also has the attractive feature that all the kinetic behavior is determined by a single variable: "reactivity". The other integral quantities, namely the effective delayed-neutron fractions and the neutron-prompt lifetime do not vary significantly with time for most transients of interest. Therefore, for a given perturbation, if the time-dependence of the reactivity is known exactly, solving the point-kinetics equations will yield essentially the exact total power versus time response of the reactor. This would be an "exact" point-kinetics calculation. However, in practice, this is never the case since the reactivity is never known exactly and must be approximated somehow. Various procedures and methods are used to precompute values of the reactivity corresponding to different reactor conditions. One of the most common procedures used today is "Perturbation Theory". It consists of computing the reactivity by using a single, constant flux-shape. The resultant "approximate" point-kinetics calculation is valid only in a limited neighborhood around the initial equilibrium point.

An earlier study performed at MIT [2] has shown that the point-kinetics calculations can lead to completely erroneous predictions when applied to severe transients such as a reactor start-up. The main source of error was identified as the poor estimation of the reactivity away from equilibrium conditions where large flux-shape changes had occurred.

1.2 OBJECTIVE

The objective of the present investigation is to determine whether simple point-kinetics calculations based on improved reactivity estimates can reasonably approximate a Light-Water Reactor start-up transient. The essential idea consists of using as input to the point-kinetics model a precomputed table of 'reactivity versus control-rod position' to approximate the reactivity contribution of the moving control-rods. This procedure is aimed at reducing the large errors which would inevitably result if the reactivity were computed using a single, fixed flux-shape [2]. First, a table of 'reactivity versus control-rod position' is constructed from a series of steady-state, critical calculations using the static module of the QUANDRY code. Then, the transient module of the QUANDRY code is used to obtain: (i) reference solutions, (ii) "exact" point-kinetics results, and (iii) "approximate" point-kinetics results by the use of the precomputed reactivity table. In all cases, thermal-hydraulic feedback effects on cross-sections are computed by the QUANDRY code from three-dimensional temperature distribution calculations. In the "approximate" point-kinetics case, however, the net reactivity at any time during the transient is formed by adding the reactivity due to the thermal-hydraulic feedback and the reactivity inferred from the 'reactivity versus control-rod position' table.

1.3 SUMMARY

A simple procedure for approximating the time-dependence of the total power of a LWR based on the point-kinetics model and on a precomputed reactivity table has been outlined. This procedure is aimed at improving the results of more-approximate point-kinetics calculations without resorting to expensive computations. Comparisons will be made with the full space- and time-dependent reference results from the QUANDRY code.

In Chapter Two of this report, the main theoretical results which form the basis of the computer code QUANDRY are presented. In particular, the point-kinetics parameters are defined. Chapter Three describes how the table of 'reactivity versus control-rod position' is calculated and how the QUANDRY code has been modified to incorporate such a table. In Chapter Four, an application of this new code to a three-dimensional LWR model and to a start-up transient is presented. Finally, a summary of the investigation, conclusions about the modifications, and recommendations for future research are given in Chapter Five.

Chapter 2

QUANDRY THEORETICAL FRAMEWORK

2.1 INTRODUCTION

The QUANDRY code is a three-dimensional, two-group, Cartesian-geometry, nodal diffusion code which was developed at the Massachusetts Institute of Technology for the analysis of Light-Water Reactors [1]. The program is written in FORTRAN and contains two distinct modules: a "static module" for steady-state calculations and a "transient module" for time-dependent problems. The unknowns in the QUANDRY code are the node-averaged, group fluxes and the directional group net-leakages in the three-coordinate directions. In the case of steady-state, homogeneous problems, the eigenvalue (k_{eff}) is an additional unknown. The geometry and values for the homogeneous, nodal, group cross-sections, diffusion constants, discontinuity factors [3], and external source are assumed known. When feedback is considered, all thermal-hydraulic parameters and feedback coefficients must also be provided.

The group fluxes and directional net-leakages are found by simultaneously solving a "nodal balance equation" and three "coupling equations". The nodal balance equation is obtained without approximation by integrating the group diffusion equations over the volume of a given node. It relates the group fluxes in that particular node to all three directional, group net-leakages for that node. The coupling equations are only approximate. Each one of them relates the u-directed (u = x, y, z) group net-leakages for a given node to: (i) group fluxes for that node and its two nearest neighbors in the u-direction, and (ii) group net-leakages for the node and its four nearest neighbors in the two directions perpendicular to u. These coupling equations are derived from one-dimensional, two-group, analytical calculations in which the transverse-leakage term is approximated by a quadratic function [1].

For transient calculations, in addition to the node-averaged group fluxes and directional group net-leakages, the node-averaged precursor concentrations must also be calculated. Transients in the QUANDRY code are initiated by either

- simulating a control rod motion,
- changing the inlet coolant temperature,
- changing the coolant flow-rate,
- or varying the magnitude of the external source,

from an initial steady-state condition.

The overall transient is divided into time intervals, each being characterized by a fixed time-step size. Instantaneous values of the point-kinetics parameters can be computed at every time-step assuming that the weight functions appearing in the definition of the point-kinetics parameters have been specified. These point-kinetics parameters can be used in a quasi-static option in which shape calculations are performed infrequently.

2.2 STATIC PROBLEMS

In the static module of the QUANDRY code, a three-level iterative procedure is used to solve either a critical regular eigenvalue-problem, an adjoint eigenvalue-problem, or a non-homogeneous external-source problem. In all three cases, thermal-hydraulic feedback can be included.

The iterative procedure used in the QUANDRY code consists of [2]:

- outer or "fission source term" iterations,
  - at each outer iteration, inner iterations based on a modified block Gauss-Seidel Method,
    - at each inner iteration, flux iterations based on a block Successive Overrelaxation Method (Cyclic Chebyshev Semi-Iterative Method).

2.2.1 REGULAR EIGENVALUE-PROBLEM

Application of the Analytical Nodal Method to the three-dimensional, two-group, static diffusion equations leads to the following set of nodal equations [1]:

$$[\Sigma_t]\,[\overline{\Phi}] + h_y h_z[\overline{L}_x] + h_x h_z[\overline{L}_y] + h_x h_y[\overline{L}_z] = \frac{1}{\gamma}[M]\,[\overline{\Phi}]$$

$$-\,[F_x]\,[\overline{\Phi}] + [\overline{L}_x] - \frac{1}{h_y}[G_x]\,[\overline{L}_y] - \frac{1}{h_z}[G_x]\,[\overline{L}_z] = [0]$$

$$-\,[F_y]\,[\overline{\Phi}] - \frac{1}{h_x}[G_y]\,[\overline{L}_x] + [\overline{L}_y] - \frac{1}{h_z}[G_x]\,[\overline{L}_z] = [0]$$

$$-\,[F_z]\,[\overline{\Phi}] - \frac{1}{h_x}[G_z]\,[\overline{L}_x] - \frac{1}{h_y}[G_z]\,[\overline{L}_y] + [\overline{L}_z] = [0]$$

(1)

where the unknowns are: (i) the most positive value for the eigenvalue ($\gamma = k_{eff}$), (ii) the corresponding everywhere positive node-averaged fluxes, and (iii) the node-averaged, directional net-leakages. If the node-averaged flux and directional net-leakage vectors,

$$[\overline{\Phi}] = \begin{bmatrix} \overline{\Phi}_1^{i,j,k} \\ \overline{\Phi}_2^{i,j,k} \end{bmatrix} \tag{2}$$

and

$$[\overline{L}_u] = \begin{bmatrix} \overline{L}_{u1}^{i,j,k} \\ \overline{L}_{u2}^{i,j,k} \end{bmatrix} \; ; \; \mathrm{u=x,y,z} \tag{3}$$

are combined into a supervector,

$$[\Psi] = col([\overline{\Phi}],\ [\overline{L}_x],\ [\overline{L}_y],\ [\overline{L}_z]) \tag{4}$$

then Equation 1 can be rewritten in a condensed, supermatrix form as:

$$[\mathrm{H}_\gamma]\,[\Psi] = \frac{1}{\gamma}\,[\mathrm{P}]\,[\Psi] \tag{5}$$

or

$$[H_\gamma]\,[\Psi] = [0] \tag{6}$$

where

$$[H_\gamma] = \frac{1}{\gamma}\,[\mathrm{P}] - [\mathrm{H}_\gamma] \tag{7}$$

and with the following definitions:

$$[\mathrm{H}_\gamma] = \begin{bmatrix} [\Sigma_t] & h_y h_z [I] & h_x h_z [I] & h_x h_y [I] \\ -[F_x] & [I] & -\frac{1}{h_y}[G_x] & -\frac{1}{h_z}[G_x] \\ -[F_y] & -\frac{1}{h_x}[G_y] & [I] & -\frac{1}{h_z}[G_y] \\ -[F_z] & -\frac{1}{h_x}[G_z] & -\frac{1}{h_y}[G_z] & [I] \end{bmatrix} \tag{8}$$

$$[\mathrm{P}] = \begin{bmatrix} [M] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \end{bmatrix} \tag{9}$$

$$[M] = \{V^{i,j,k}[\chi]\,[\nu\Sigma_f^{i,j,k}]^T\}$$

where $V^{i,j,k}$ is the volume of node i,j,k;

$$[\chi] = \begin{bmatrix}1\\0\end{bmatrix}$$

$$[\nu\Sigma_f^{i,j,k}] = \begin{bmatrix}\nu\Sigma_{f1}^{i,j,k}\\ \nu\Sigma_{f2}^{i,j,k}\end{bmatrix}$$

$$[\Sigma_t] = \{V^{i,j,k}[\Sigma_t^{i,j,k} - \Sigma_s^{i,j,k}]\}$$

$$[\Sigma_t^{i,j,k}] = \begin{bmatrix}\Sigma_{t1}^{i,j,k} & 0\\ 0 & \Sigma_{t2}^{i,j,k}\end{bmatrix}$$

$$[\Sigma_s^{i,j,k}] = \begin{bmatrix}\Sigma_{11}^{i,j,k} & 0\\ \Sigma_{21}^{i,j,k} & \Sigma_{22}^{i,j,k}\end{bmatrix}$$

Because the matrices $[F_u]$ and $[G_u]$; u = x, y, z, depend slightly on the eigenvalue, the subscript γ is included in $[H_\gamma]$ and $[\mathcal{H}_\gamma]$. In addition, $[F_u]$ and $[G_u]$ are complicated functions of the cross-sections and diffusion constants; therefore, they must be updated whenever feedback effects are taken into consideration.

2.2.2 ADJOINT EIGENVALUE-PROBLEM

The static module of the original QUANDRY code has been modified to calculate the node-averaged, adjoint fluxes and adjoint, directional net-leakages for any reactor configuration. These adjoint fluxes and net-leakages can be used as weight functions in the definition of the point-kinetics parameters [4].

The adjoint form of the static, analytical, nodal diffusion equations is readily obtained from Equation 5:

$$[H_\gamma]^T[\Psi^*] = \frac{1}{\gamma}[P]^T[\Psi^*] \tag{10}$$

in which the most-positive eigenvalue γ is the same as in Equation 5. Equation 10 can be rewritten in a form similar to Equation 6:

$$[H_\gamma]^T[\Psi^*] = [0] \tag{11}$$

where

$$[H_\gamma]^T = \frac{1}{\gamma}[P]^T - [H_\gamma]^T \tag{12}$$

The unknown adjoint supervector $[\Psi^*]$ is a column vector made up of node-averaged, adjoint, group fluxes and adjoint, directional, group net-leakages:

$$[\Psi^*] = col([\Phi^*], [\bar{L}^*_x], [\bar{L}^*_y], [\bar{L}^*_z]) \tag{13}$$

The matrices $[H_\gamma]^T$ and $[P]^T$ are the transposes of $[H_\gamma]$ and $[P]$ as given by Equations 8 and 9, respectively.

2.2.3 EXTERNAL-SOURCE PROBLEM

Another modification recently added to the static and transient module of the QUANDRY code is the possibility of representing external sources anywhere within the reactor core. This external source option can be used to model local sources in a reactor for a start-up condition or the (γ,n) reactions for a shut-down condition [5].

When external sources are taken into consideration, the analytical nodal diffusion equations take on a form similar to Equation 1:

$$[\Sigma_t][\Phi] + h_y h_z[\bar{L}_x] + h_x h_z[\bar{L}_y] + h_x h_y[\bar{L}_z] = [M][\Phi] + [\bar{Q}_v]$$

$$- [F_x]\,[\bar{\Phi}] + [\bar{L}_x] - \frac{1}{h_y}[G_x]\,[\bar{L}_y] - \frac{1}{h_z}[G_x]\,[\bar{L}_z] - [\mathrm{H}_x]\,[\bar{Q}]$$

$$- [F_y]\,[\bar{\Phi}] - \frac{1}{h_x}[G_y]\,[\bar{L}_x] + [\bar{L}_y] - \frac{1}{h_z}[G_y]\,[\bar{L}_z] - [\mathrm{H}_y]\,[\bar{Q}]$$

$$- [F_z]\,[\bar{\Phi}] - \frac{1}{h_x}[G_z]\,[\bar{L}_x] - \frac{1}{h_y}[G_z]\,[\bar{L}_y] + [\bar{L}_z] - [\mathrm{H}_z]\,[\bar{Q}]$$

(14)

The external source terms can be grouped into a single supervector:

$$[Q] - col\,([\bar{Q}_v],\ [\mathrm{H}_x]\,[\bar{Q}],\ [\mathrm{H}_y]\,[\bar{Q}],\ [\mathrm{H}_z]\,[\bar{Q}]) \qquad (15)$$

where

$$[\bar{Q}_v] - V^{i,j,k}\begin{bmatrix}\bar{Q}^{i,j,k}\\ 0\end{bmatrix}$$

$$[\bar{Q}] - \begin{bmatrix}\bar{Q}^{i,j,k}\\ 0\end{bmatrix}$$

and where $\bar{Q}^{\,i,j,k}$ is the external source at node i,j,k. The $[\mathrm{H}_u]$; u = x, y, z, matrices are complicated functions of the cross-sections and diffusion constant similar to the $[F_u]$ and $[G_u]$ matrices. Using these definitions, Equation 14 can be rewritten as:

$$[\mathrm{H}_1]\,[\Psi] - [\mathrm{P}]\,[\Psi] + [Q] \qquad (16)$$

or equivalently,

$$[H_1]\,[\Psi] + [Q] - [0] \qquad (17)$$

with

$$[H_1] - [\mathrm{P}] - [\mathrm{H}_1] \qquad (18)$$

The matrix $[H_1]$ is the same as the matrix $[H_\gamma]$ in Equation 8 with γ set to 1.

2.3 TRANSIENT CASE

In the transient module of the QUANDRY code, a set of discretized, time-dependent, nodal diffusion equations is solved. The details of the temporal-discretization scheme is given in Reference [1]. At each time-step, a spatial (flux-shape) calculation is performed using the same iterative procedure as in the static case. Any transient calculation is preceded by a static calculation, the results of which determine the initial equilibrium conditions. As in the static case, thermal-hydraulic feedback effects can be included in the calculations.

2.3.1 TIME-DEPENDENT PROBLEM

The time-dependent analytical nodal diffusion equations embodied in the QUANDRY code are [1]:

$$\begin{aligned}
([M(t)] - [\Sigma_t(t)])[\overline{\Phi}(t)] &- h_y h_z[\overline{L}_x(t)] - h_x h_z[\overline{L}_y(t)] \\
&- h_x h_y[\overline{L}_z(t)] - \beta[M(t)][\overline{\Phi}(t)] + \sum_{d=1}^{D}\lambda_d[\overline{C}_d(t)] \\
&+ [\overline{Q}_v(t)] = [V]^{-1}\frac{d}{dt}[\overline{\Phi}(t)]
\end{aligned}$$

$$\begin{aligned}
[F_x(t)][\overline{\Phi}(t)] - [\overline{L}_x(t)] &+ \frac{1}{h_y}[G_x(t)][\overline{L}_y(t)] \\
&+ \frac{1}{h_z}[G_x(t)][\overline{L}_z(t)] + [H_x(t)][\overline{Q}(t)] = [0]
\end{aligned}$$

$$\begin{aligned}
[F_y(t)][\overline{\Phi}(t)] &+ \frac{1}{h_x}[G_y(t)][\overline{L}_x(t)] - [\overline{L}_y(t)] \\
&+ \frac{1}{h_z}[G_y(t)][\overline{L}_z(t)] + [H_y(t)][\overline{Q}(t)] = [0]
\end{aligned}$$

$$[F_z(t)]\,[\overline{\Phi}(t)] + \frac{1}{h_x}[G_z(t)]\,[\overline{L}_x(t)] + \frac{1}{h_y}[G_z(t)]\,[\overline{L}_y(t)]$$

$$- [\overline{L}_z(t)] + [\mathrm{H}_z(t)]\,[\overline{Q}(t)] = [0] \tag{19}$$

and

$$\beta_d[M_d(t)]\,[\overline{\Phi}(t)] - \lambda_d[\overline{C}_d(t)] = \frac{d}{dt}[\overline{C}_d(t)] \;; \quad \mathrm{d} = 1,2,3,\ldots,\mathrm{D} \tag{20}$$

where several quantities are simply the time-dependent generalizations of static quantities defined in the previous section. When no external source is present, the matrices $[\overline{Q}_V(t)]$ and $[\overline{Q}(t)]$ are zero. Moreover, the matrix $[M(t)]$ includes the eigenvalue γ which makes the source-free reactor initially critical.

By defining the time-dependent supervector:

$$[\Psi(t)] = col\,([\overline{\Phi}(t)],\ [\overline{L}_x(t)],\ [\overline{L}_y(t)],\ [\overline{L}_z(t)]) \tag{21}$$

Equations 19 and 20 can be rewritten in a supermatrix form as:

$$[H(t)]\,[\Psi(t)] - \beta\,[\mathrm{P}(t)]\,[\Psi(t)] + \sum_{d=1}^{D}\lambda_d[C_d(t)]$$

$$+ [Q(t)] = [\mathrm{V}]^{-1}\frac{d}{dt}[\Psi(t)] \tag{22}$$

and

$$\beta_d[M_d(t)]\,[\overline{\Phi}(t)] - \lambda_d[\overline{C}_d(t)] = \frac{d}{dt}[\overline{C}_d(t)] \;; \quad \mathrm{d} = 1,2,3,\ldots,\mathrm{D} \tag{23}$$

with the following definitions:

$$[C_d(t)] = col\,([\overline{C}_d(t)],\ [0],\ [0],\ [0]) \tag{24}$$

$$[\overline{C}_d(t)] = V^{i,j,k}\begin{bmatrix}\overline{C}_d^{i,j,k}(t)\\ 0\end{bmatrix}\ ;\ \mathrm{d} = 1,2,3,\ldots,\mathrm{D}$$

$$[\mathrm{V}]^{-1} = diag(V^{i,j,k}[v]^{-1})$$

$$[v] = \begin{bmatrix} v_1^{i,j,k} & 0\\ 0 & v_2^{i,j,k}\end{bmatrix}$$

where $v_g^{i,j,k}$ is the group-g neutron velocity for the composition inside node i,j,k.

$$\beta = \sum_{d=1}^{D}\beta_d$$

$$[H(t)] = [\mathrm{P}(t)] - [\mathrm{H}(t)] \qquad (25)$$

$$[\mathrm{H}(t)] = \begin{bmatrix} [\Sigma_t(t)] & h_y h_z[I] & h_x h_y[I] & h_x h_y[I]\\ -[F_x(t)] & [I] & -\frac{1}{h_y}[G_x(t)] & -\frac{1}{h_z}[G_x(t)]\\ -[F_y(t)] & -\frac{1}{h_x}[G_y(t)] & [I] & -\frac{1}{h_z}[G_y(t)]\\ -[F_z(t)] & -\frac{1}{h_x}[G_z(t)] & -\frac{1}{h_y}[G_z(t)] & [I]\end{bmatrix}$$

$$[\Sigma_t(t)] = \{V^{i,j,k}[\Sigma_t^{i,j,k}(t) - \Sigma_s^{i,j,k}(t)]\}$$

$$[\mathrm{P}(t)] = \begin{bmatrix} [M(t)] & [0] & [0] & [0]\\ [0] & [0] & [0] & [0]\\ [0] & [0] & [0] & [0]\\ [0] & [0] & [0] & [0]\end{bmatrix} \qquad (26)$$

$$[M(t)] = \{V^{i,j,k}[\chi_p]\frac{1}{\gamma}[\nu\Sigma_f^{i,j,k}(t)]^T\}$$

$$[\nu\Sigma_f^{i,j,k}(t)] = \begin{bmatrix} \nu\Sigma_{f1}^{i,j,k}(t) \\ \nu\Sigma_{f2}^{i,j,k}(t) \end{bmatrix}$$

$$[M_d(t)] = \{V^{i,j,k}[\chi_d]\frac{1}{\gamma}[\nu\Sigma_f^{i,j,k}(t)]^T\}$$

In the QUANDRY code, the prompt- and delayed-neutron spectra are such that:

$$[\chi_p] = [\chi_d] = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Therefore,

$$[M(t)] = [M_d(t)]$$

The eigenvalue (γ) is absent whenever external sources are present.

In the transient case, the matrices $[F_u(t)]$, $[G_u(t)]$; u = x, y, z, and $[H_\gamma(t)]$ have terms involving the "prompt and delayed frequencies" defined as:

$$\omega_{p_g}^{i,j,k}(t_{n+1}) = \frac{1}{\Delta t_n}\ln\left[\frac{\bar{\Phi}_g^{i,j,k}(t_{n+1})}{\bar{\Phi}_g^{i,j,k}(t_n)}\right] \tag{28}$$

and

$$\omega_d^{i,j,k}(t_{n+1}) = \frac{1}{\Delta t_n}\ln\left[\frac{\tilde{C}_d^{i,j,k}(t_{n+1})}{\tilde{C}_d^{i,j,k}(t_n)}\right] \ ; \ d = 1, 2, 3, \ldots, D \tag{29}$$

Because of these time-dependent frequencies, the $[F_u(t)]$ and $[G_u(t)]$ matrices must be updated during any transient even if no thermal-hydraulic feedback is considered.

Finally, the total neutronic power of the reactor is defined as:

$$P(t) = E\sum_{i,j,k} V^{i,j,k}[\Sigma_{f1}^{i,j,k}(t)\bar{\Phi}_1^{i,j,k}(t) + \Sigma_{f2}^{i,j,k}(t)\bar{\Phi}_2^{i,j,k}(t)] \tag{30}$$

where E is a conversion factor approximately equal to 3.2C4 10^{-11} J/fission.

2.3.2 POINT-KINETICS PARAMETERS

The point-kinetics equations can be derived exactly without any assumptions from the time-dependent nodal equations of the previous section, Equations 22 and 23 [4]. The results are:

$$\frac{d}{dt}n_{eff}(t) = \frac{\rho(t) - \beta}{\Lambda(t)}n_{eff}(t) + \sum_{d=1}^{D}\lambda_d C_{d_{eff}}(t) + q_{eff}(t) \tag{31}$$

and

$$\frac{d}{dt}C_{d_{eff}}(t) = \frac{\beta_d}{\Lambda(t)}n_{eff}(t) - \lambda_d C_{d_{eff}}(t)\ ; \quad d = 1,2,3,\ldots,D \tag{32}$$

with the following definitions for the point-kinetics parameters:

- amplitude function

$$n_{eff}(t) = \frac{1}{K}[\Psi^*]^T[V]^{-1}[\Psi(t)] = \frac{1}{K}[\Phi^*]^T[V]^{-1}[\Phi(t)] \tag{33}$$

- reactivity

$$\rho(t) = \frac{[\Psi^*]^T[H(t)]\,[\Psi(t)]}{[\Phi^*]^T[M(t)]\,[\Phi(t)]} \tag{34}$$

- prompt-neutron lifetime

$$\Lambda(t) = \frac{[\Phi^*]^T[V]^{-1}[\Phi(t)]}{[\Phi^*]^T[M(t)]\,[\Phi(t)]} \tag{35}$$

- effective delayed-neutron precursor concentrations

$$C_{d_{eff}} = \frac{1}{K}[\Psi^*]^T[C_d(t)] = \frac{1}{K}[\Phi^*]^T[\tilde{C}_d(t)]\ ; \quad d = 1,2,3,\ldots,D \tag{36}$$

– effective external source

$$q_{eff}(t) = \frac{1}{K}[\Psi^*]^T[Q(t)] \tag{37}$$

and where K is a normalization constant taken to be equal to:

$$K = \sum_{i,j,k} V^{i,j,k}[\bar{\Phi}_1^{*i,j,k} + \bar{\Phi}_2^{*i,j,k}]$$

Notice that the definition of the point-kinetics parameters depends on the weight functions making up the supervector $[\Psi^*]$. When weight functions have been defined, the QUANDRY code computes all the point-kinetics quantities from their defining equations (Equations 33 through 37) at every time-step. The weight functions are usually chosen as either unity, the regular fluxes and net-leakages themselves, or the adjoint quantities corresponding to a fixed particular configuration of the reactor.

However, it can be shown that the expression for reactivity in Equation 34 reduces to,

$$\rho = 1 - \frac{1}{k_{eff}} \tag{38}$$

where k_{eff} is the eigenvalue of the perturbed reactor, for any weight function chosen [6]. This is an "Adiabatic Approximation" result which is valid for any perturbation when Equation 25 is used to define the perturbated reactor condition with the eigenvalue factored out of the $[M(t)]$ matrix.

2.3.3 QUASI-STATIC OPTION

The basis of the "quasi-static approximation" is the formal separation of the flux into an amplitude function and a shape function. The amplitude function is a function of time only, while the shape function is normalized in such a way that most of the time-dependence of the flux is reflected in the amplitude function. In numerical calculations, such a splitting permits important computing time savings because (expensive) shape updates can be performed much more infrequently than (inexpensive) amplitude updates without introducing much error. This option is the most recent addition to the transient module of the QUANDRY code [2].

In addition to leading to significant computing-time reductions, this option can also makes it possible to perform "point-kinetics-type" calculations by suppressing all shape-updates. In such calculations, the initial or final shape function is used throughout the entire transient.

In an earlier study [4], it was shown that the use of fairly large time-steps resulted in erroneous values for the node-averaged fluxes and power densities. It was confirmed that the error was due largely to the amplitude part of the node-averaged fluxes which was poorly estimated. On the contrary, despite the large time-steps, the shape part of the fluxes displayed very little error for most transients of interest. To correct for these errors in the amplitude function, an amplitude correction scheme was incorporated into the QUANDRY code, which resulted in a new quasi-static option [2].

The time-dependent node-averaged fluxes and directional net-leakages can be expressed as the product of the amplitude function, as defined in Equation 33, and a shape function such that [6]:

$$[\Phi(t)] - n_{eff}(t)\,[S(t)] \tag{39}$$

Substituting Equation 33 into Equation 39 shows that the shape function must satisfy the following normalization condition:

$$K = [\overline{\phi^{*}}]^{T}[V]^{-1}[S(t)]$$

Therefore, even though the shape function, $[S(t)]$, is time-dependent, the integral quantity K is a constant, independent of time.

With the exception of very rapid transients, the variations in the shape part of the flux are limited and occur on a much slower time-scale than the variations in the amplitude part. The quasi-static approximation takes advantage of this situation by utilizing different time-steps for updating these two functions. Small time-steps are used to update the amplitude function since this function is very sensitive to time. On the other hand, large time-steps are used to update the shape function. In the case of a point-kinetics type calculation, the shape function is not updated at all and the initial or final shape function is used throughout the transient.

In practice, the QUANDRY code does not deal explicitly with the shape and amplitude functions, but instead (and equivalently) with the total flux, net-leakages, and amplitude function. At every amplitude time-step, all node-averaged fluxes and directional net-leakages are multiplied by an amplitude correction-factor defined as:

$$\frac{n_{eff}^{(PK)}}{n_{eff}^{(Q)}} \tag{40}$$

where

$n_{eff}^{(PK)}(t)$ - Amplitude function computed by solving the point-kinetics Equation 31.

$n_{eff}^{(Q)}(t)$ - Amplitude function computed by the QUANDRY code using Equation 33.

"Corrected" fluxes and directional net-leakages are defined as:

$$[\bar{\Phi}^{corr}(t)] = \frac{n_{eff}^{(PK)}(t)}{n_{eff}^{(Q)}(t)}[\bar{\Phi}(t)] \tag{41}$$

and

$$[\bar{L}_u^{corr}(t)] = \frac{n_{eff}^{(PK)}(t)}{n_{eff}^{(Q)}(t)}[\bar{L}_u(t)] \ ; \ u = x,y,z \tag{42}$$

Because the prompt frequencies (Equation 28) depend on the node-averaged group fluxes, they must also be corrected:

$$\omega_{p_g}^{i,j,k\,corr}(t_{n+1}) = \frac{1}{\Delta t_n}\ln\left[\frac{\bar{\Phi}_g^{i,j,k\,corr}(t_{n+1})}{\bar{\Phi}_g^{i,j,k}(t_n)}\right] \tag{43}$$

Equations 41, 42, and 43 make it possible to advance the transient calculation without recomputing the flux and net-leakage shapes at every time-step Δt_n. In the limit where no shape update at all are performed, the QUANDRY code operates in a "point-kinetics" mode.

2.4 SUMMARY

The main theoretical results of the QUANDRY code were presented in this chapter. The important features of the code are the computation of the point-kinetics parameters and the quasi-static option. These features are used in Chapter Four in combination with a "precalculated reactivity table" option presented in the next chapter.

Chapter 3

THE NEW REACTIVITY TABLE OPTION IN THE QUANDRY CODE

3.1 INTRODUCTION

In an earlier study [2], the quasi-static option of the QUANDRY code was used in a "point-kinetics" mode (no shape update) to assess the validity of point-kinetics in describing a start-up transient. This study showed, by comparisons with full space- and time-dependent calculations, that point-kinetics can lead to completely erroneous predictions for reactivity and total power. To improve these results (without resorting to expensive space-kinetics calculations), it was suggest [7] that the calculations be done by a hybrid "adiabatic" method which attempts to account for the large flux-shape changes that occur during such severe transients. In this hybrid method, the reactivity due to the moving control-rods is read from a precalculated table of 'control-rod-worth versus control-rod position', whereas the reactivity associated with thermal-hydraulic feedback effects is calculated by "Perturbation Theory", i.e. by using a fixed flux-shape corresponding to some reference control-rod position.

The idea behind this hybrid method is that if the reactivity changes caused by the control-rods can be well approximated, then the computed amplitude function should be very close to its true value. However, evaluation of this method required modifications in the QUANDRY code.

3.2 THE NEW ALGORITHM OF THE QUANDRY TRANSIENT MODULE

The algorithm of the QUANDRY transient module is shown in Figure 1.

This algorithm has been modified to include an option to read a precalculated 'reactivity versus time' table. The 'reactivity versus control-rod position' table is converted into a 'reactivity versus time' table by setting the rate of control-rods withdrawal from the reactor core constant. The only modification performed to the transient module was to add an interpolating routine (RODTAB) after step #5 in Figure 1. This subroutine returns at every time-step the reactivity due to the control-rods from the precalculated table. All cross-section changes reflecting moving control-rods are suppressed. Cross-sections now vary only because of thermal-hydraulic feedback effects. Figure 2 shows the new algorithm.

1. $\overline{\phi}_g(t_n)$, $\overline{L}_{g_u}(t_n)$, $\overline{C}_d(t_n)$, $\overline{T}_{fuel}(t_n)$, $\overline{T}_{coolant}(t_n)$, $\overline{\rho}_{coolant}(t_n)$, and $\rho(t_n)$, $\Lambda(t_n)$, $q_{eff}(t_n)$, $n_{eff}(t_n)$, $c_{d_{eff}}(t_n)$ are known at time t_n.
2. Take time-step Δt_n and advance to time t_{n+1}. Modify:

   a) cross-sections $\Sigma_g(t_n)$ because of input perturbation and feedback.

   b) external source $\overline{Q}(t_n)$, if time-dependent.
3. Update the matrices (if required).
4. Use $\overline{\phi}_g(t_n)$, $\overline{L}_{g_u}(t_n)$, $\Sigma_g(t_{n+1})$, $\overline{Q}(t_{n+1})$, and the matrices to compute the new fluxes $\overline{\phi}_g(t_{n+1})$ and net-leakages $\overline{L}_{g_u}(t_{n+1})$.
5. Use $\overline{\phi}_g(t_{n+1})$, $\overline{L}_{g_u}(t_{n+1})$, $\Sigma_g(t_{n+1})$, $\overline{Q}(t_{n+1})$, and the matrices to calculate the point-kinetics parameters $\rho(t_{n+1})$, $\Lambda(t_{n+1})$, $q_{eff}(t_{n+1})$, and $n_{eff}^{(Q)}(t_{n+1})$ from their definitions.
6. Use $\rho(t_n)$, $\Lambda(t_n)$, $q_{eff}(t_n)$, $n_{eff}(t_n)$, $c_{d_{eff}}(t_n)$, and $\rho(t_{n+1})$, $\Lambda(t_{n+1})$, $q_{eff}(t_{n+1})$ to compute $n_{eff}^{(PK)}(t_{n+1})$ by solving the point-kinetics equations.
7. Compute the amplitude correction factor,

$$\frac{n_{eff}^{(PK)}(t_{n+1})}{n_{eff}^{(Q)}(t_{n+1})}$$

and multiply all fluxes $\overline{\phi}_g(t_{n+1})$ and net-leakages $\overline{L}_{g_u}(t_{n+1})$ by this factor.

Figure 1. The Original Algorithm of the QUANDRY Transient Module [1].
(continued on the next page)

8. Set $n_{eff}(t_{n+1}) = n_{eff}^{(PK)}(t_{n+1})$.

9. Compute $\overline{C}_d(t_{n+1})$ from $\overline{C}_d(t_n)$, $\overline{\phi}_g(t_n)$, and the corrected fluxes $\overline{\phi}_g(t_{n+1})$.

10. Calculate the $c_{d_{eff}}(t_{n+1})$'s from their definitions.

11. Compute $\overline{T}_{fuel}(t_{n+1})$ and $\overline{T}_{coolant}(t_{n+1})$ from $\overline{\phi}_g(t_{n+1})$, and infer $\overline{\rho}_{coolant}(t_{n+1})$.

12. Compute $\Delta\Sigma_g(t_{n+1})$ to account for temperature and density effects, and return to step #1.

Figure 1. The Original Algorithm of the QUANDRY Transient Module [1].

1. $\overline{\phi}_g(t_n)$, $\overline{L}_{g_u}(t_n)$, $\overline{C}_d(t_n)$, $\overline{T}_{fuel}(t_n)$, $\overline{T}_{coolant}(t_n)$, $\overline{\rho}_{coolant}(t_n)$, and $\rho(t_n)$, $\Lambda(t_n)$, $q_{eff}(t_n)$, $n_{eff}(t_n)$, $c_{d_{eff}}(t_n)$ are known at time t_n.

2. Take time-step Δt_n and advance to time t_{n+1}. Modify:

   a) cross-sections $\Sigma_g(t_n)$ because of feedback.

   b) external source $\overline{Q}(t_n)$, if time-dependent.

3. Update the matrices (if required).

4. Use $\overline{\phi}_g(t_n)$, $\overline{L}_{g_u}(t_n)$, $\Sigma_g(t_{n+1})$, $\overline{Q}(t_{n+1})$, and the matrices to compute the new fluxes $\overline{\phi}_g(t_{n+1})$ and net-leakages $\overline{L}_{g_u}(t_{n+1})$.

5. Use $\overline{\phi}_g(t_{n+1})$, $\overline{L}_{g_u}(t_{n+1})$, $\Sigma_g(t_{n+1})$, $\overline{Q}(_{n+1})$, and the matrices to calculate the point-kinetics parameters $\rho_{FB}(t_{n+1})$, $\Lambda(t_{n+1})$, $q_{eff}(t_{n+1})$, and $n_{eff}^{(Q)}(t_{n+1})$ from their definitions. $\rho_{FB}(t_{n+1})$ is due to thermal-hydraulic feedback only, hence, the subscript 'FB'.

6. Use the precalculated reactivity table to find $\rho_{CR}(t_{n+1})$ by interpolation. $\rho_{CR}(t_{n+1})$ is the reactivity due to the moving control-rods only, hence, the subscript 'CR'. Compute $\rho(t_{n+1}) = \rho_{FB}(t_{n+1}) + \rho_{CR}(t_{n+1})$.

7. Use $\rho(t_n)$, $\Lambda(t_n)$, $q_{eff}(t_n)$, $n_{eff}(t_n)$, $c_{d_{eff}}(t_n)$, and $\rho(t_{n+1})$, $\Lambda(t_{n+1})$, $q_{eff}(t_{n+1})$ to compute $n_{eff}^{(PK)}(t_{n+1})$ by solving the point-kinetics equations.

Figure 2. The Modified Algorithm of the QUANDRY Transient Module. (continued on the next page).

8. Compute the amplitude correction factor,

$$\frac{n_{eff}^{(PK)}(t_{n+1})}{n_{eff}^{(Q)}(t_{n+1})}$$

and multiply all fluxes $\bar{\phi}_g(t_{n+1})$ and net-leakages $\bar{L}_{g_u}(t_{n+1})$ by this factor.

9. Set $n_{eff}(t_{n+1}) = n_{eff}^{(PK)}(t_{n+1})$.

10. Compute $\bar{C}_d(t_{n+1})$ from $\bar{C}_d(t_n)$, $\bar{\phi}_g(t_n)$, and the corrected fluxes $\bar{\phi}_g(t_{n+1})$.

11. Calculate the $c_{d_{eff}}(t_{n+1})$'s from their definitions.

12. Compute $\bar{T}_{fuel}(t_{n+1})$ and $\bar{T}_{coolant}(t_{n+1})$ from $\bar{\phi}_g(t_{n+1})$, and infer $\bar{\rho}_{coolant}(t_{n+1})$.

13. Compute $\Delta\Sigma_g(t_{n+1})$ to account for temperature and density effects, and return to step #1.

Figure 2. The Modified Algorithm of the QUANDRY Transient Module.

3.3 MODIFICATIONS IN THE QUANDRY CODE

The new algorithm described in the previous section was implemented in the QUANDRY code in the form of two subroutines: INPUTR and RODTAB.

3.3.1 SUBROUTINE INPUTR

Subroutine INPUTR contains the precalculated 'reactivity versus time' table in the form of a one-dimensional array. An example of such a routine is given in Appendix A.

This subroutine is called only once at the first time-step ($t_{n=0}$) in the transient module.

3.3.2 SUBROUTINE RODTAB

Subroutine RODTAB calculates the reactivity by interpolation in the array returned by subroutine INPUTR. This interpolation program is based on Neville's Algorithm [8]. Neville's Algorithm uses a Lagrange's polynomial interpolation-type scheme; however, it has the advantage of computing an error estimate of the interpolation. A listing of the subroutine RODTAB is given in Appendix B.

This subroutine is called immediately after subroutine PKPAR which computes values for all point-kinetics quantities. However, subroutine RODTAB is called only at specific time-steps which correspond to actual motion of the control-rods during a transient.

3.4 TESTING OF THE REACTIVITY TABLE OPTION

The new reactivity table option in the QUANDRY code has been tested by applying it to a simulated start-up transient in a three-dimensional LWR model. This transient displays external-source effects, rapid power rise, flux-shape changes, and thermal-hydraulic feedback effects. This transient is described below and will be used again in the applications of Chapter Four.

3.4.1 DESCRIPTION OF THE REACTOR-CORE MODEL

The Light Water Reactor used is a representation of the Salem-1 Pressurized Water Reactor core. The core is a three-dimensional model of the Salem-1 reactor core with an approximate height of 3.60 meters and 3.20 meters in diameter. This core is reflected on all sides, axially and radially. The quarter-core, nominal, thermal power is $P = 834.5\ 10^6$ W, the coolant inlet-temperature is $\overline{T}_{\text{inlet coolant}} = 555$ K = 282 °C, and the mass flow-rate is $3.868\ 10^6$ g s^{-1}. Because of symmetry in the core, only one quadrant of the core is modeled. This quarter core is subdivided into 18 horizontal planes for a total of 1458 nodes of equal size: 21.6 cm x 21.6 cm x 20.0 cm. A more complete description of the core model is given in Appendix C.

3.4.2 DESCRIPTION OF THE START-UP TRANSIENT

The initial configuration of the reactor core corresponds to a sub-critical core containing soluble poison with all control-rods fully inserted in the "hot standby" condition (where the temperature and pressure correspond to full power operating condition of the reactor). An external start-up source is introduced in the bottom plane, central axis, (node #1 of the model). The magnitude of this external source is $\overline{Q} = 5.502236\ 10^8$ neutrons cm^{-3} s^{-1}, which gives an initial equilibrium power of $P_0 = 83.45$ W or 10^{-7} P_{nominal}.

The transient consists in simulating a fast power rise to full power, critical conditions from t = 0 s to t = 400 s, see Table 1. The first part of the transient is the simulation of a fast (non-realistic) boron dilution by reduction of the group-two absorption cross-sections in all nodes (except those representing control assemblies because of technical restrictions within the QUANDRY code). This occurs during the first 120 s of the transient.

The second part of the transient represents control-rods withdrawal at a constant rate of 2 cm s^{-1} from t = 120 s to t = 200 s and is simulated by decreasing the group-two absorption cross-sections in compositions #16 through 23 successively (see Appendix C). When the new reactivity table option is used, the QUANDRY code skips this second part of the transient. Instead, the group-two absorption cross-sections corresponding to the rodded nodes remain unaltered and the reactivity representing control-rods withdrawal is added to the system using the 'reactivity versus control-rod position' table.

In the third part of the transient, control-rod motion stops temporarily, from t = 200 s to t = 300 s. When the reactor power reaches a high enough value, thermal-hydraulic feedback become significant and limit further power increase.

Finally, in the fourth step, control-rods withdrawal resumes at a constant rate of 2 cm s^{-1} until all control rods have been fully withdrawn from the reactor core, t = 300 s to t = 400 s. If the reactivity table option is used, this last part is also skipped. Fission cross-sections are divided beforehand by a factor of 1.058371 so that the final configuration of the reactor at t = 400 s corresponds to a reactor that is exactly critical at the specified total power level of 834.5 10^6 W.

Table 1. A Time Description of the Light-Water Reactor Start-Up Transient.

TIME SCALE (s)	TYPE OF TRANSIENT
0	Beginning of Transient Power = 83.45 W Reactivity* = $-1.04756\ 10^{-1}$ "Hot" standby condition Boron present, all control-rods fully inserted
0 TO 120	Boron Dilution In all nodes except the rodded nodes #16 through 33
120 TO 200	Control Rods Withdrawal Rate: 2 cm s^{-1} In nodes #16 through 23
200 TO 300	No Action Thermal-hydraulic feedback becomes significant
300 TO 400	Control Rods Withdrawal Rate: 2 cm s^{-1} in nodes #24 through 33

400	End of Transient
	Power = 834.5 10^6 W
	Reactivity = 0
	No Boron, all control-rods fully withdrawn

* Depends on the weight function used to define it.

3.4.3 COMPUTED INITIAL POINT-KINETICS PARAMETERS

The weight functions used to compute the point-kinetics parameters are adjoint fluxes and net-leakages corresponding to the (artificially-critical) configuration of the core at t = 120 s (all control-rods fully inserted and no boron present). The reactor is made exactly critical by dividing all fission cross-sections by the eigenvalue, k_{eff}, obtained from a source-free static calculation.

The initial values of the point-kinetics parameters computed by using these weight functions are:

ρ_o	$= -1.04756\ 10^{-1}$
Λ_o	$= 2.61910\ 10^{-5}$ s
n_{eff_o}	$= 4.28201\ 10^{0}$ cm^{-3}
q_{eff_o}	$= 1.71267\ 10^{4}$ cm^{-3} s^{-1}
$c_{d_{eff_o}}$	$= 3.44261\ 10^{3}$, d=1
	$7.73087\ 10^{3}$, d=2
	$1.88091\ 10^{3}$, d=3
	$1.50571\ 10^{3}$, d=4
	$1.05194\ 10^{2}$, d=5
	$7.72985\ 10^{0}$, d=6

with

$$\beta = 0.0065$$

for this reactor.

3.4.4 TEST RESULTS

The numerical parameters selected for the tests are:

$\theta_f = 1.0$

$\theta_p = 0.5$

Error reduction in fluxes: $\epsilon_\phi = 0.001$

Three inner iterations per time-iteration

Matrix-updating at every time-iteration

Diagonal-symmetry option selected [1]

and for the point-kinetics calculations:

$$\theta_n = 1.0$$

$$\theta_c = 0.5$$

$$\theta_q = 1.0 \ [4]$$

A simple test is performed to validate the new reactivity table option in the QUANDRY code. The test consists in using for the reactivity table the "exact" values from the reference space- and time-dependent calculation. In addition to validating the new option in the code, this test will give a precise idea of how closely the power trajectory can be predicted if one is able to precompute a "perfect" reactivity table.

Figure 3 shows the reactivity as a function of time from the reference QUANDRY calculation and from the interpolating routine. Since the reference reactivity itself is used in the table, the curves are identical (to within small interpolation-errors), as expected.

Figure 4 shows the normalized amplitude function as computed from this reactivity table using Equations 31 and 32. Also shown is the reference curve for comparisons. Since the prompt-neutron lifetime, Λ, in Equations 31 and 32 does not vary significantly during the transient, it is not surprising that the point-kinetics result agree very closely with the reference result.

Figure 5 compares the point-kinetics and reference normalized power as a function of time (Equation 30). A sizeable discrepancy appears between the two curves. This difference can be explained by flux-shape changes which are not accounted for in the point-kinetics case (constant shape). No matter how accurately the reactivity is known, the reactor

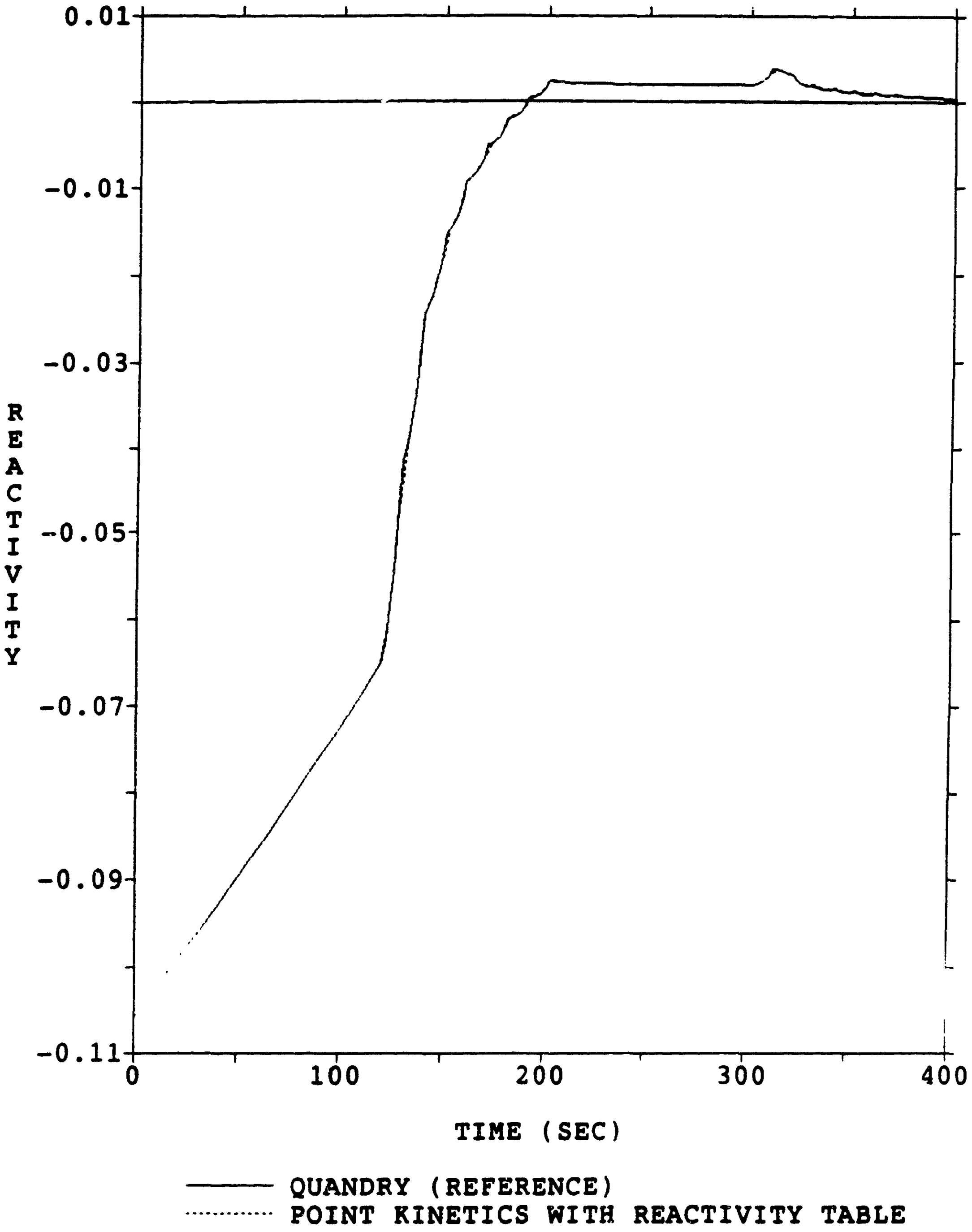


Figure 3. Reactivity versus Time using the Correct Reactivity from the Reference Calculation for the Reactivity-Table Option and Final-Adjoint Weighting.

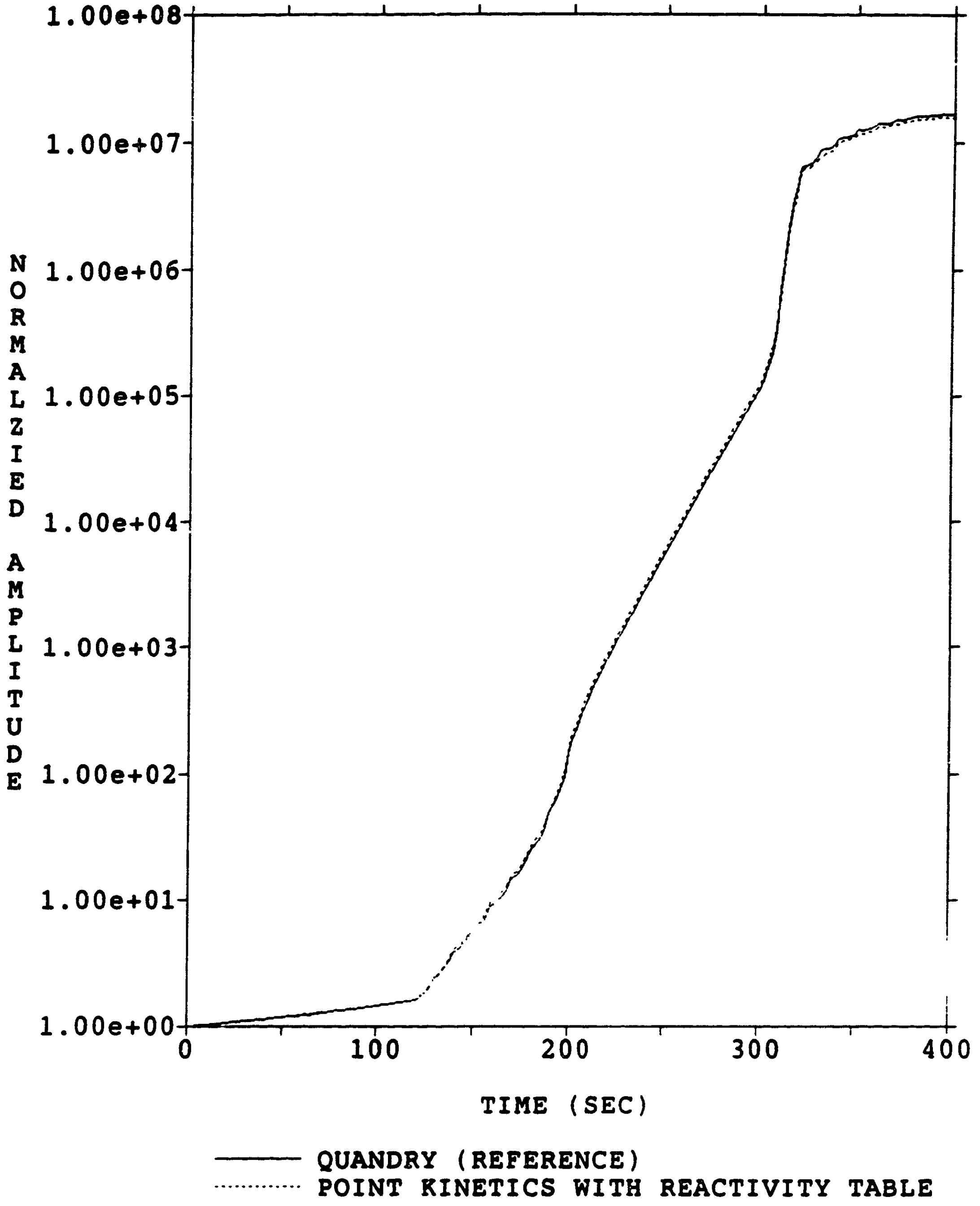


Figure 4. Normalized Amplitude versus Time using the Correct Reactivity from the Reference Calculation for the Reactivity-Table Option and Final-Adjoint Weighting.

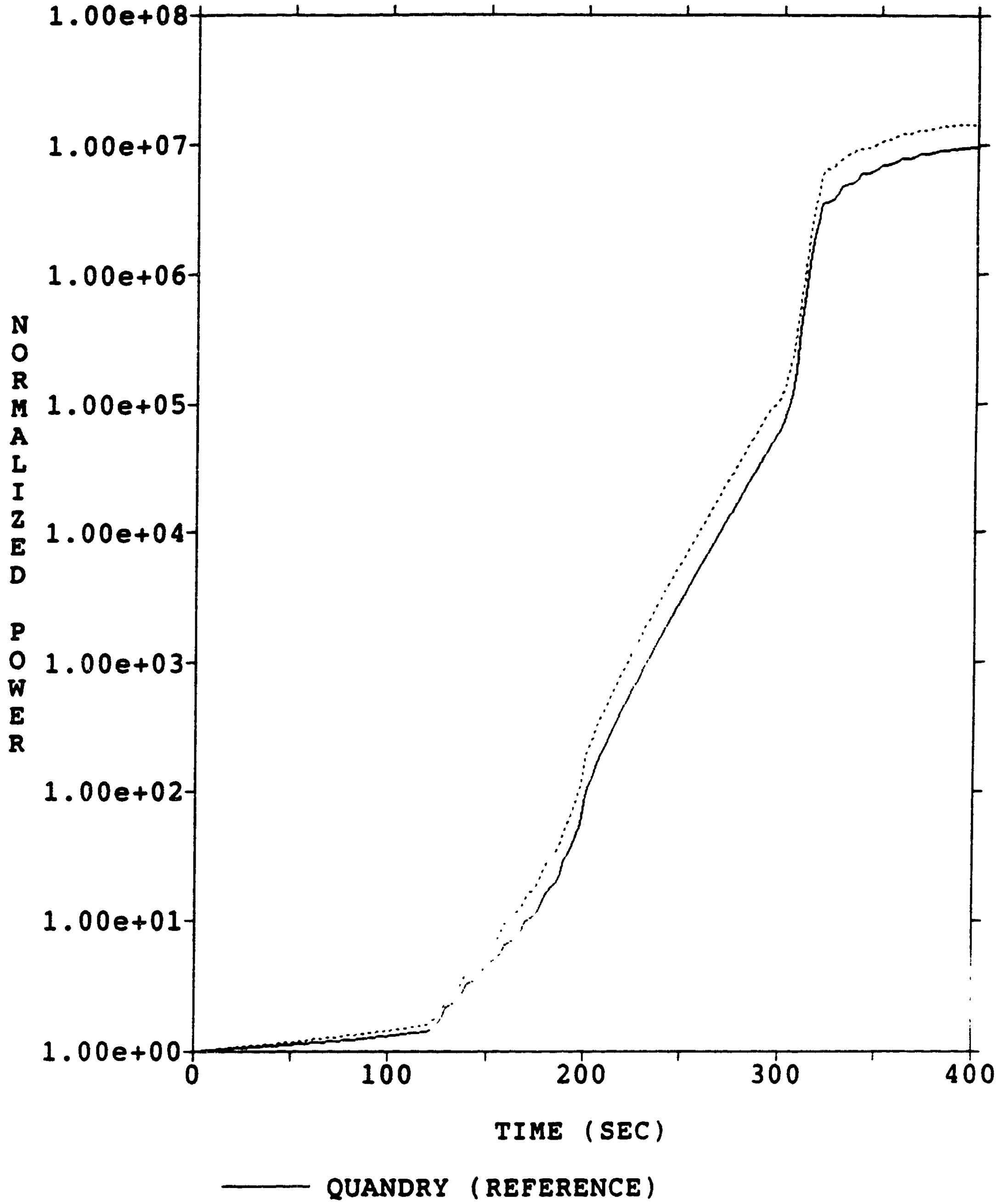


Figure 5. Normalized Power versus Time using the Correct Reactivity from the Reference Calculation for the Reactivity-Table Option and Final-Adjoint Weighting.

power (as computed by Equation 30) will generally be in error if a constant flux-shape is used throughout a transient involving flux-shape variations.

3.5 SUMMARY

A new reactivity table option has been incorporated into the transient module of the QUANDRY code. It is to be used whenever QUANDRY is run in a "point-kinetics" mode. The table is normally constructed from static calculations corresponding to reactor conditions expected at certain times during the transient under study. However, to validate this new option, the "exact" reactivity profile from a reference calculation was used for this table. Consistent results were obtained.

In Chapter Four, applications of this new option to a Light-Water Reactor are presented. The results are compared with a reference space- and time-dependent solution.

Chapter 4

APPLICATION: POINT-KINETICS VERSUS SPACE-KINETICS

4.1 INTRODUCTION

Today, the nuclear power industry still relies extensively on the point-kinetics calculations to predict a Light-Water Reactor transient response. Therefore, it is of interest to compare the point-kinetics results with the results of more accurate space- and time-dependent calculations to see whether large discrepancies exist between them.

An earlier study demonstrated that in the case of a LWR start-up transient, the point-kinetics calculations compare rather poorly with the space- and time-dependent calculations [2]. In particular, the reactivity as computed from Equation 34 was seriously in error. The main cause of error was found to be the inability of the point-kinetics to deal with the severe flux-shape changes that occurred during a start-up transient.

On the other hand, if the correct reactivity versus time profile were known, the results of the point-kinetics calculations could be improved significantly as demonstrated by the test of the previous chapter. This motivates the idea of using a precomputed reactivity table rather than calculating the reactivity using Equation 34. This procedure is often used in practice.

4.2 GENERATING THE REACTIVITY TABLE

In this research, the precalculated reactivity table is produced from the "Adiabatic Approximation Theory" [6]. However, no thermal-hydraulic feedback effects are included so that the reactivity table accounts only for control-rod effects. The reactor model used is the same as in Chapter Three Section Four, the Salem-1 Pressurized Water Reactor. The procedure employed to generate the reactivity table is given below.

First, an eigenvalue-problem is solved to determine the value of γ in Equation 1 that will make the reactor exactly critical at the end of the transient (no boron, control rods fully withdrawn, full power). This eigenvalue for the Salem-1 model reactor was computed to be

$$\gamma = k_{eff} = 1.058371$$

and is used to divide all the homogeneous, node-averaged fission cross-sections.

Next, a number of static eigenvalue-calculations are performed with the control-rods in different positions within the reactor core. The reactor conditions correspond to full power, "hot standby" (operating temperature and pressure) conditions. Thermal-hydraulic feedback effects are excluded from these calculations. From the computed eigenvalues, a table of 'reactivity versus control-rod position' is constructed using Equation 38 (see Table 2). These calculations were performed at full power and found to be (to five significant figures) power independent. Finally, the final position corresponding to all control rods out (360 cm) gives a reactivity of $1.37264\ 10^{-2}$. When thermal-hydraulic feedback effects are considered, they add $-1.37264\ 10^{-2}$ of feedback reactivity to

Table 2. Reactivity Worth of the Control-Rods versus Control-Rod Position.

CONTROL-ROD POSITION (cm)	REACTIVITY
0 - ALL RODS INSERTED	$- 1.86878\ 10^{-2}$
20	$- 1.86276\ 10^{-2}$
40	$- 1.82104\ 10^{-2}$
60	$- 1.62666\ 10^{-2}$
80	$- 1.12207\ 10^{-2}$
100	$- 5.52142\ 10^{-3}$
120	$- 9.12547\ 10^{-4}$
140	$2.57379\ 10^{-3}$
160	$5.20343\ 10^{-3}$
180	$7.21282\ 10^{-3}$
200	$8.77339\ 10^{-3}$
220	$1.00054\ 10^{-2}$
240	$1.09917\ 10^{-2}$
260	$1.17899\ 10^{-2}$
280	$1.24387\ 10^{-2}$
300	$1.29603\ 10^{-2}$
320	$1.33600\ 10^{-2}$
340	$1.36210\ 10^{-2}$
360 - ALL RODS WITHDRAWN	$1.37264\ 10^{-2}$

give a net reactivity equal to zero at the end of the transient.

A plot of reactivity versus control-rod position is shown in Figure 6, displaying the familiar S-shape curve.

Furthermore, from the knowledge of the control-rod positions and rate of removal from the reactor core (see Section 3.4.2), the 'reactivity versus position' table can be converted into a 'reactivity versus time' table. For this particular start-up transient, the control-rods are withdrawn at a constant rate of 2 cm s^{-1}. Table 3 shows the corresponding converted 'reactivity versus time' table. This table is inserted into subroutine INPUTR in the transient module of the QUANDRY code.

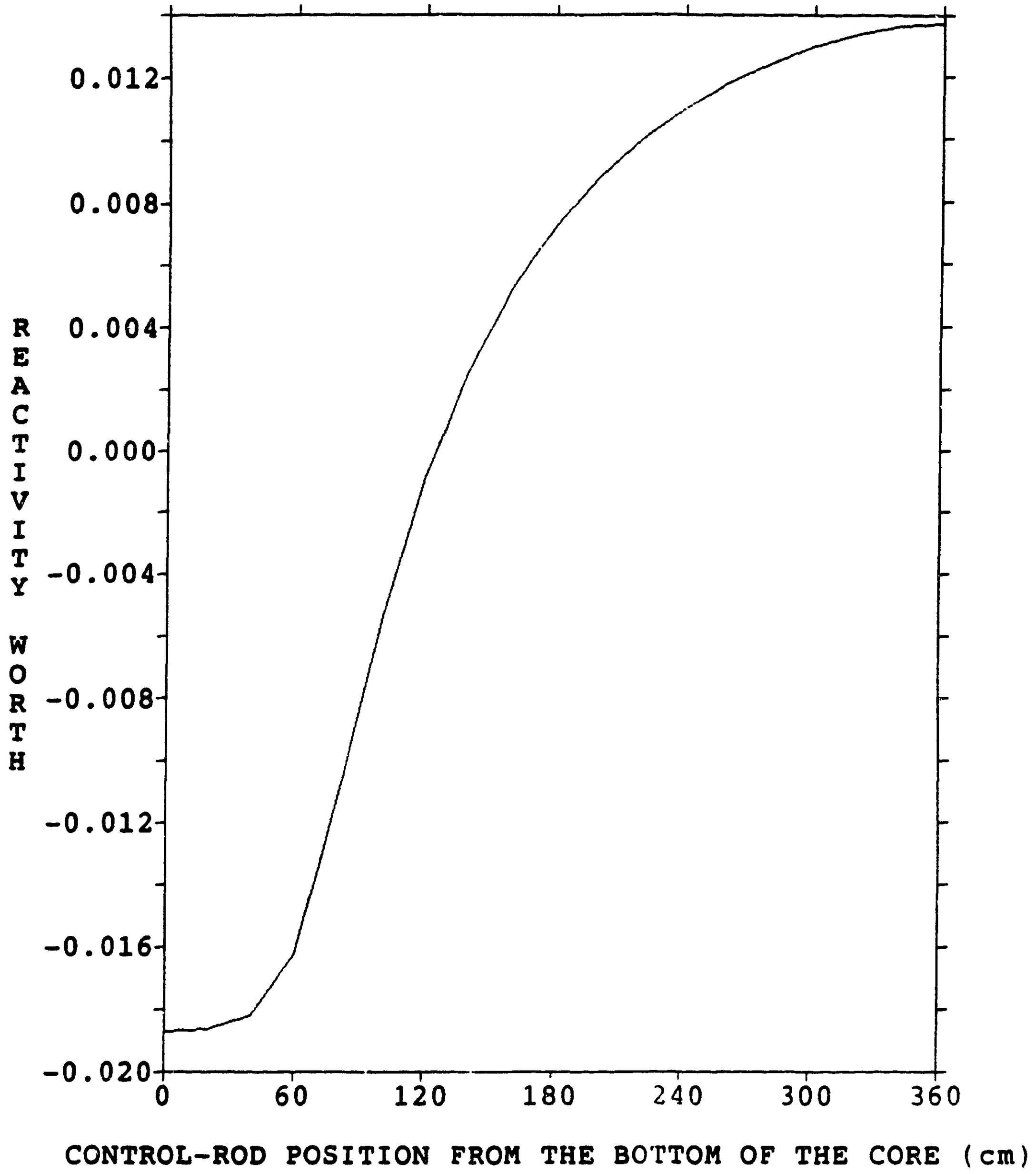


Figure 6. Control-Rod Reactivity-Worth versus Control-Rod Position.

Table 3. Reactivity Worth of the Control-Rods versus Time during the Start-Up Transient.

TIME (s)	REACTIVITY
120	– 1.86878 10^{-2}
130	– 1.86276 10^{-2}
140	– 1.82104 10^{-2}
150	– 1.62666 10^{-2}
160	– 1.12207 10^{-2}
170	– 5.52142 10^{-3}
180	– 9.12547 10^{-4}
190	2.57379 10^{-3}
200 and 300	5.20343 10^{-3}
310	7.21282 10^{-3}
320	8.77339 10^{-3}
330	1.00054 10^{-2}
340	1.09917 10^{-2}
350	1.17899 10^{-2}
360	1.24387 10^{-2}
370	1.29603 10^{-2}
380	1.33600 10^{-2}
390	1.36210 10^{-2}
400	1.37264 10^{-2}

4.3 INITIAL FLUX-SHAPE RESULTS

Two different sets of weight functions are employed in the computation of the point-kinetics parameters. The first set are the adjoint fluxes and net-leakages corresponding to the initial reactor configuration (at t = 0 s, with boron and all control-rods fully inserted). The second set are made up of adjoint quantities corresponding to be the final reactor configuration (at t = 400 s, no boron present and all control rods fully withdrawn). In the point-kinetics runs, either the initial flux-shape (t = 0 s) or the final flux-shape (t = 400 s) are used.

When the point-kinetics models are ran with the initial flux-shape option, they will first do a QUANDRY (t = 0 s condition) calculation to find the initial flux-shape and use this initial flux-shape throughout a transient [2]. Therefore, the reference (QUANDRY) and the point-kinetics models initial reactivity value, when using the initial flux-shape, are the same. Furthermore since the point-kinetics models use the initial flux-shape from QUANDRY, the reactivity values for the initial and final adjoint weighting will not be equal. This is due to the dependency of the initial QUANDRY flux-shape on the weight functions.

For each set of adjoint weight functions, the QUANDRY (full space- and time-dependent reference) results are plotted together in the same figure with two different types of point-kinetics models. The first point-kinetics model utilizes an amplitude correction scheme with the reactivity computed by Equation 34 throughout the transient. The second type of point-kinetics model utilizes the same amplitude correction scheme but now uses a precalculated reactivity table to infer the reactivity due to control-rod motion. The initial shape of the flux is used in the point-kinetics calculations. For easy reference to the graphs, the figure numbers for each graph are summarized in Table 4.

Table 4. Figure Reference for the Results of the Point-Kinetics Calculations using the Initial Flux-Shape.

	Initial Adjoint Weighting	Final Adjoint Weighting
Reactivity vs Time	Figure 7.	Figure 8.
Normalized Amplitude vs Time	Figure 9.	Figure 10.
Normalized Power vs Time	Figure 11.	Figure 12.

Notice that all of the reference plots have a cusp-like behavior. This non-physical cusping phenomenon is due to technical restrictions within the QUANDRY code. QUANDRY simulates control-rod motion by *homogeneously* varying the nodal cross-sections when control-rods are withdrawn or inserted in the nodes. Spatial variations of the cross-sections within a node can not be simulated.

Figures 7 and 8 reveal that the reactivity results for the point-kinetics calculations are different from the reference results. The amount of the difference depends on which adjoint weighting functions are used. This adjoint weighting dependency can be observed immediately in

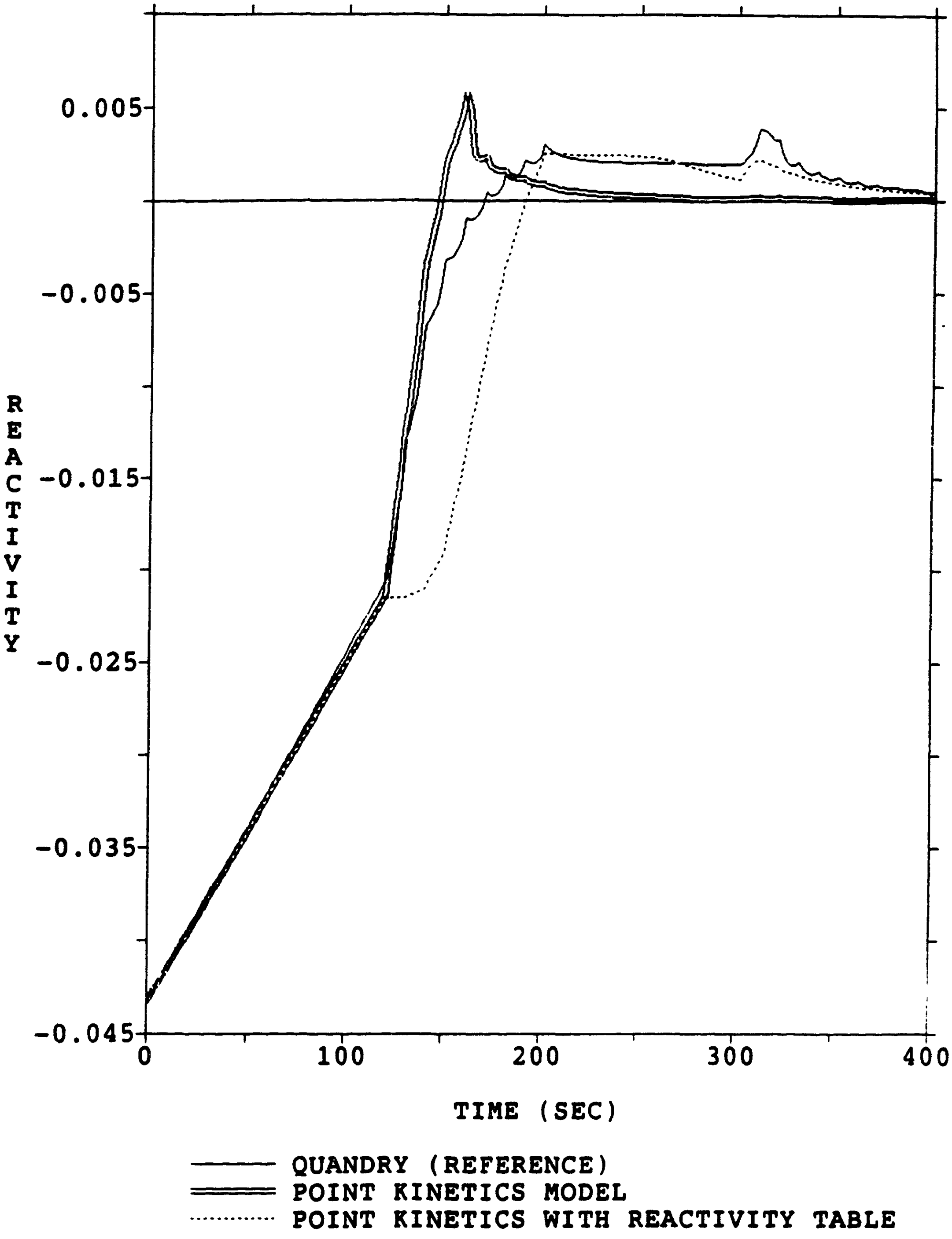


Figure 7. Reactivity versus Time using the Initial-Adjoint Weighting and the Initial Flux-Shape.

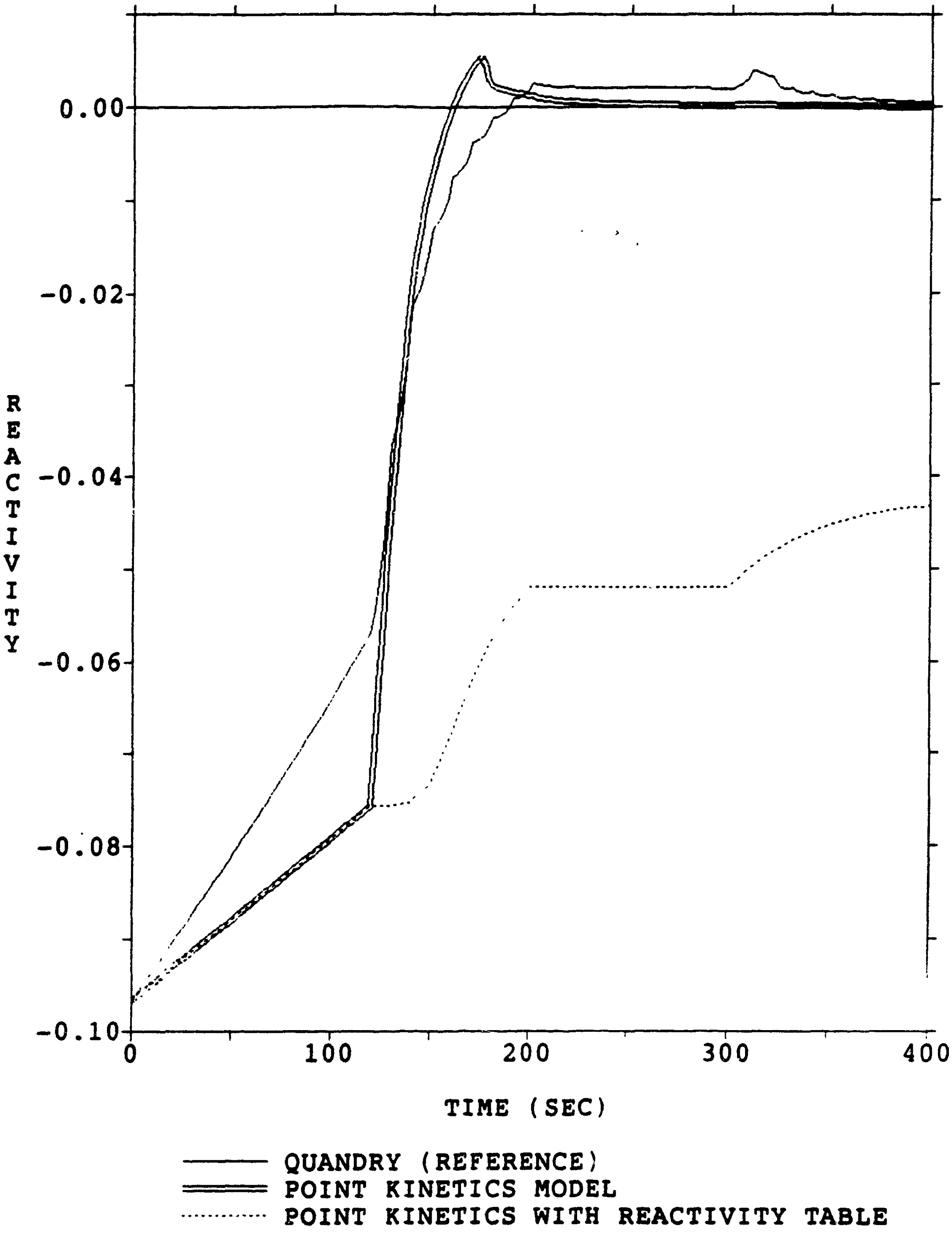


Figure 8. Reac:ivity versus Time using the Final-Adjoint Weighting and the Initial Flux-Shape.

the first part of the transient corresponding to the boron dilution. When the initial adjoint is used, the perturbation expression for reactivity is accurate and both point-kinetics models are close to the reference (Figure 7). With the final adjoint weighting (Figure 8), this is no longer the case.

Most important, Figure 7 shows that the point-kinetics model predicts criticality much earlier than the reference results while the point-kinetics model utilizing the reactivity table predicts criticality much later. On the other hand, the point-kinetics model utilizing the reactivity table simulates the control-rod withdrawal at t = 300 s much better than the regular point-kinetics model.

On the contrary, Figure 8 shows that the point-kinetics model utilizing the reactivity table does not even go critical when the final configuration of the reactor core is used to generate the adjoint weight functions needed to compute the reactivity. This is because the final weight functions give an initial reactivity at time t = 120 s (no boron, but all control-rods fully inserted) of $-7.5686\ 10^{-1}$. However, the total reactivity worth of the control-rods was found to be $3.24142\ 10^{-2}$. Therefore, the control-rods do not have enough reactivity worth to bring the reactor to a critical condition.

Furthermore, the normalized amplitude and power versus time plots are also in error. Since the reactivity is wrong, there is no reason to expect that the amplitude or the power will be correct. However, when the initial adjoint weighting functions are employed, the point-kinetics model utilizing the reactivity table simulates the amplitude and power much better than the regular point-kinetics (See Figures 9 and 11). On

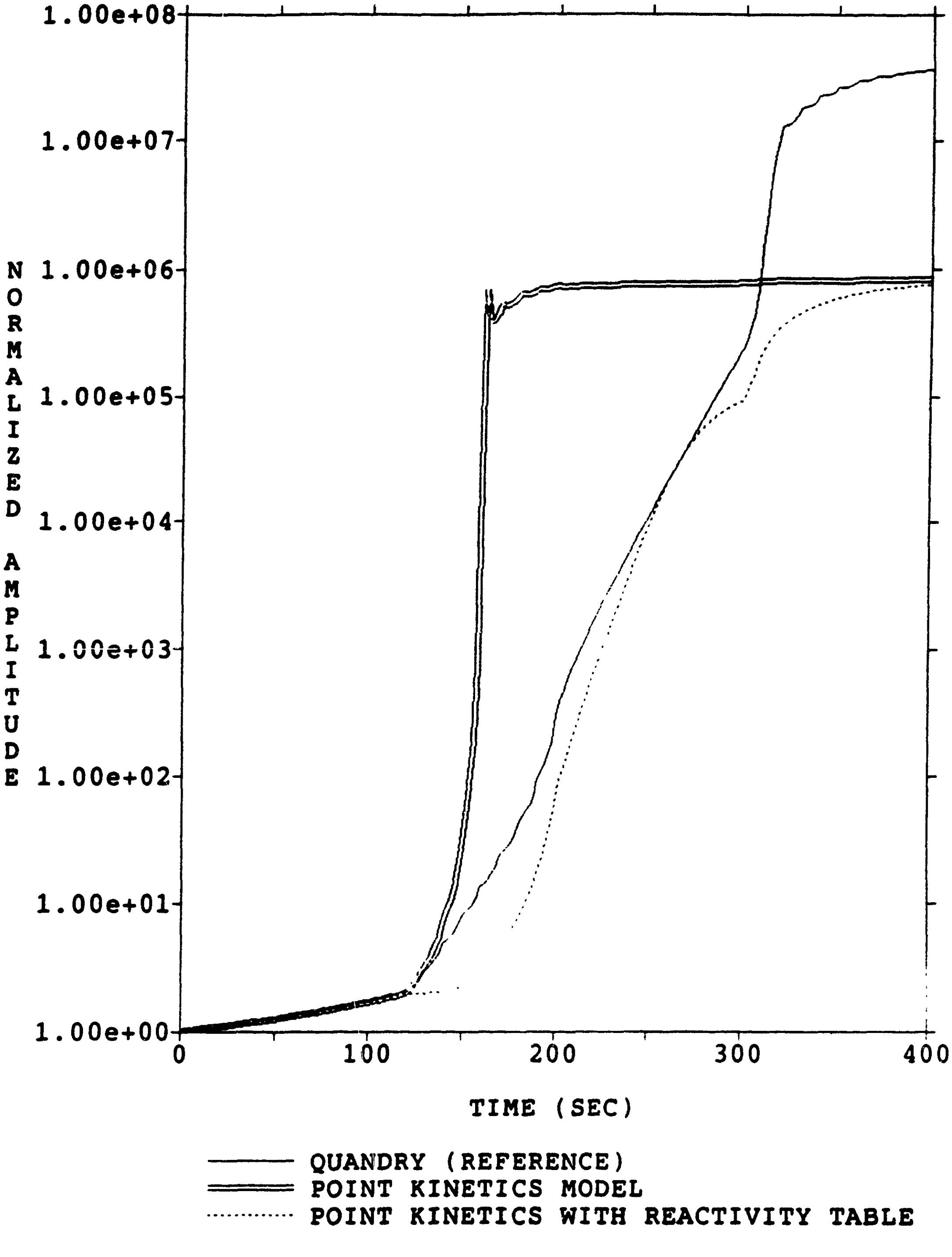


Figure 9. Normalized Amplitude versus Time using the Initial Adjoint Weighting and the Initial Flux-Shape.

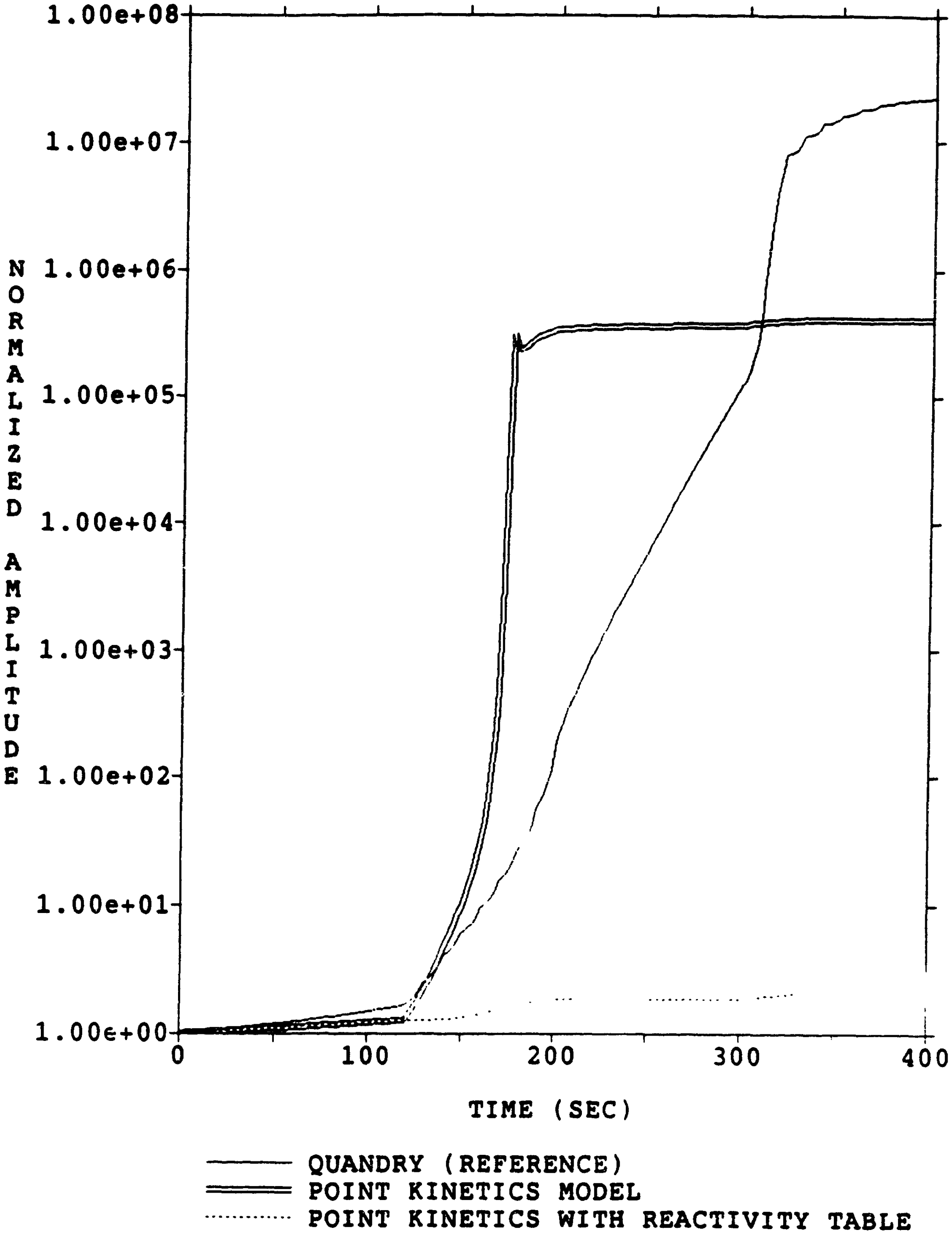


Figure 10. Normalized Amplitude versus Time using the Final-Adjoint Weighting and the Initial Flux-Shape.

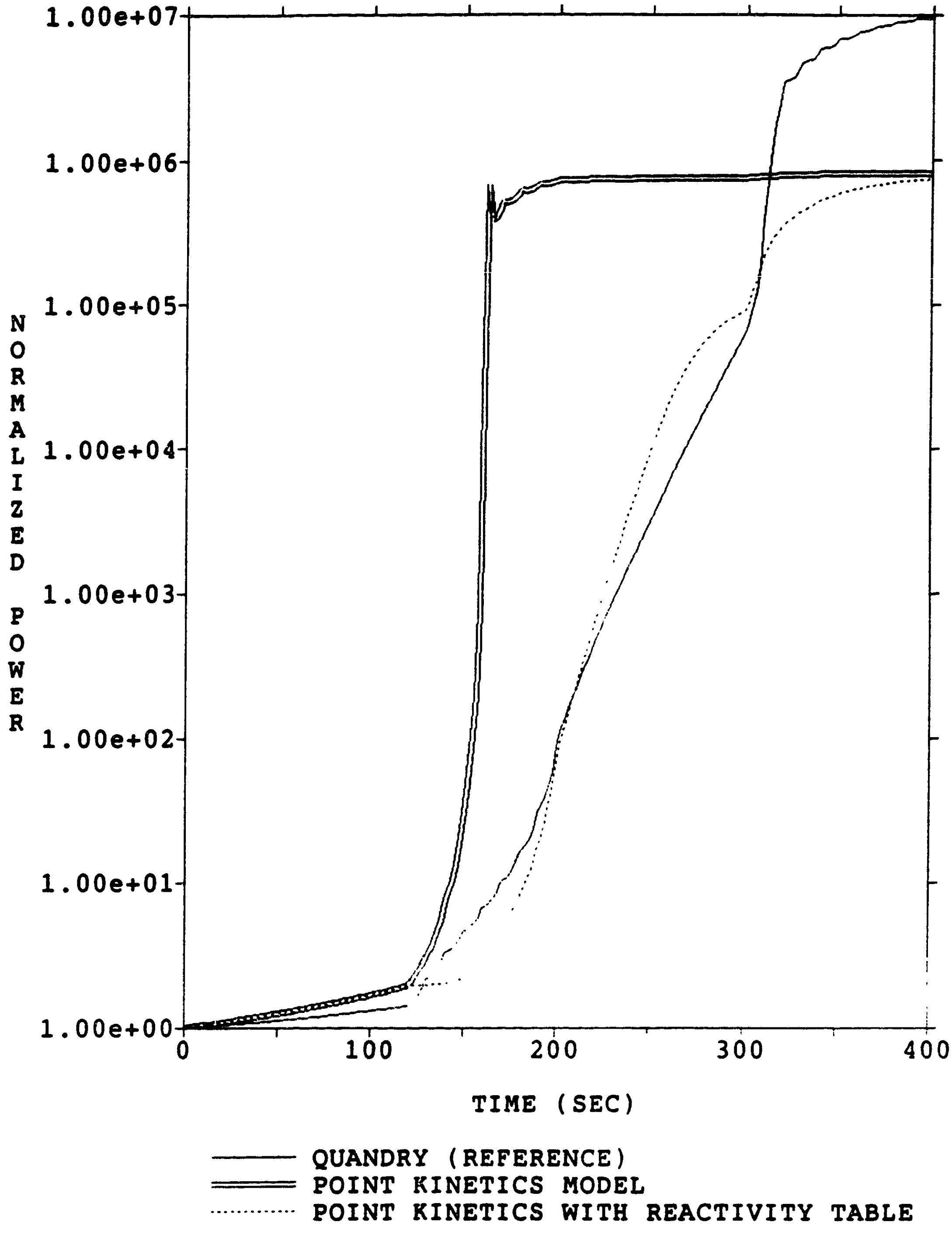


Figure 11. Normalized Power versus Time using the Initial-Adjoint Weighting and the Initial Flux-Shape.

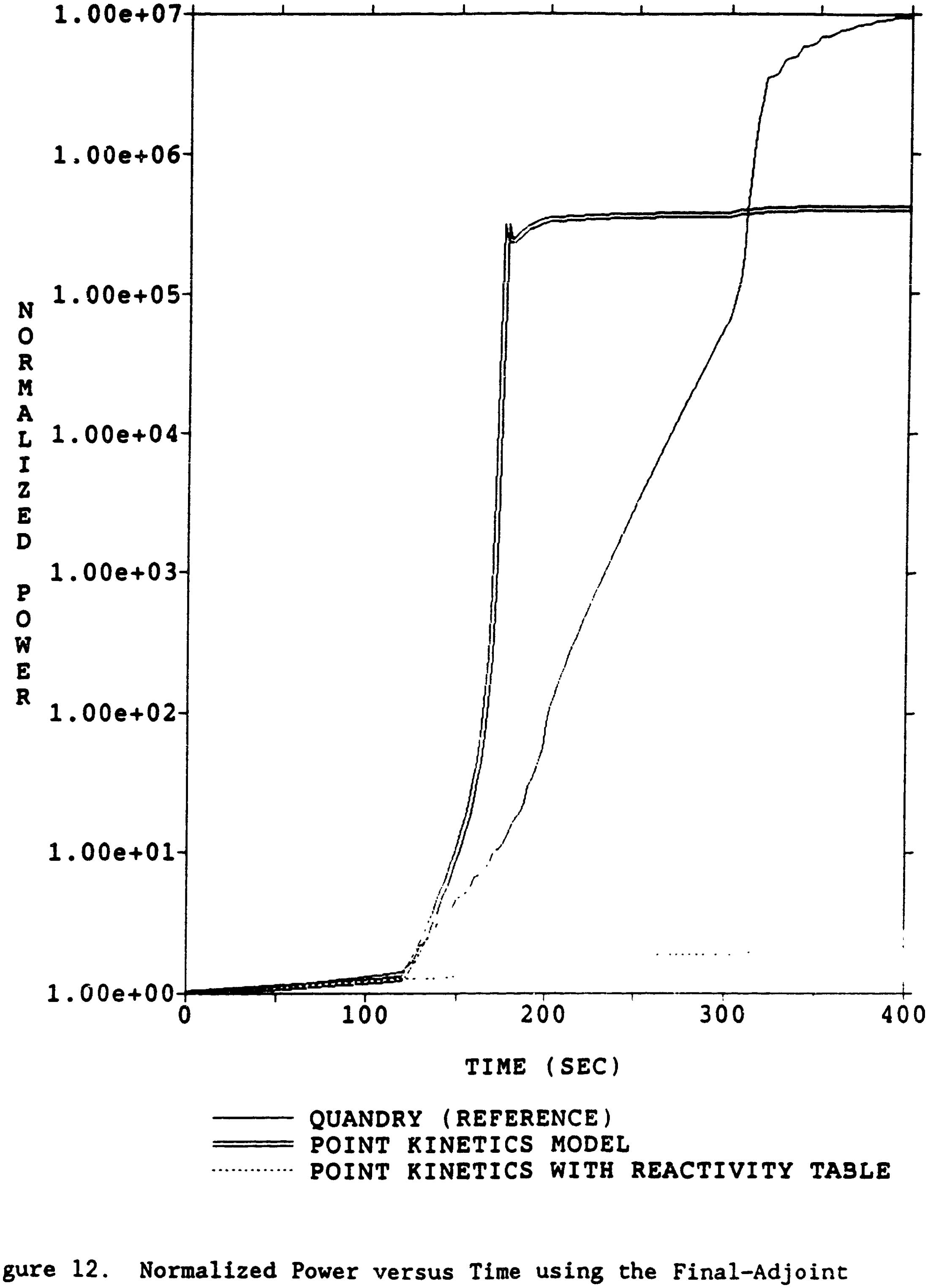


Figure 12. Normalized Power versus Time using the Final-Adjoint Weighting and the Initial Flux-Shape.

the contrary, when the final adjoint weight functions are used, the amplitude and power are wrong throughout the transient (See Figures 10 and 12).

Notice that the amplitude and the power plots (Figure 9 with 11, and Figure 10 with 12) are similar to each other. This similarity is due to the power being proportional to the amplitude function according to the following relationship,

$$P(t) \propto [\Sigma_f^{i,j,k}(t)]\,[\Phi(t)] = [\Sigma_f^{i,j,k}(t)]\,[S]\,n_{eff}(t)$$

Since the shape function is constant for the point-kinetics model and since the variation is small, they both behave the same.

4.4 FINAL FLUX-SHAPE RESULTS

In order to achieve the same results as the reference case at the end of the transient, the shape of the fluxes and net-leakages corresponding to the final critical configuration of the reactor are now used in the point-kinetics calculations. By using this final shape, the point-kinetics calculations will be forced toward the correct final conditions of the transient. As before, the initial and final adjoint weight functions are used for comparison.

Since the QUANDRY code does not deal with flux-shapes explicitly, the final node-averaged fluxes and directional net-leakages are recorded in a binary file by running an eigenvalue-problem corresponding to the static case of the final critical configuration of the reactor core. These fluxes and net-leakages must be rescaled because the amplitude function corresponds to the final critical condition (full power, end of transient) of the reactor. Therefore, it must be rescaled to confirm with the initial critical condition (initial power, beginning of transient) of the reactor.

The amplitude function within these final fluxes and net-leakages are rescaled in two parts at the beginning of the transient. First, these fluxes and net-leakages are multiplying by a factor of 0.993891 10^{-7}. This lowers the magnitude of the amplitude function from the final critical condition to the initial critical condition of the reactor to give an initial power level of $P_0 = 83.45$ W or 10^{-7} $P_{nominal}$. This scaling factor is not exactly equal to 10^{-7} because of thermal-hydraulic feedback effects on the fission cross-sections caused by going from cold to hot reactor conditions and vice versa. Secondly, they are multiplied by a factor of,

$$\frac{n_{eff_o}^{(PK)}}{n_{eff_o}^{(Q)}}$$

to attain the initial condition. This scaling factor will start the point-kinetics models using the correct amplitude function corresponding to the point-kinetics calculations instead of the amplitude function computed by QUANDRY. These initial amplitude functions are found by,

$$n_{eff_o}^{(PK)} = \frac{\Lambda_o}{-\rho_o} q_{eff_o}$$

$n_{eff_o}^{(Q)}$ = Calculated from QUANDRY using Equation 33

Since the point-kinetics parameters depend on the adjoint weight functions, the initial point-kinetics amplitude function, thus the scaling factors, will be different for each weight function employed. As a result, the scaling factors are:

- Initial Adjoint Weighting

$$\text{Scaling Factor} = 0.2632907 * 0.9938916\ 10^{-7}$$

- Final Adjoint Weighting

$$\text{Scaling Factor} = 0.4661142 * 0.9938916\ 10^{-7}$$

Since the final flux-shape is being used and the initial amplitude function is corrected, the initial power level will not be equal to 83.45 W $= 10^{-7}\ P_{nominal}$. Thus, the corresponding values for the initial equilibrium thermal power of the reactor are:

- Initial Adjoint Weighting

$$P_0 = 0.2632907 * 83.45 = 21.97\ \text{W}$$

- Final Adjoint Weighting

$$P_0 = 0.4661142 * 83.45 = 38.90\ \text{W}$$

The figure numbers for each graph corresponding to the final flux-shapes are summarized in Table 5.

Even with the final shape being employed, the results are still inaccurate compared to the reference solutions. Nevertheless, these results revel some interesting information.

For example, the point-kinetics model utilizing the reactivity table simulates the reference reactivity much better than the regular point-kinetics model when the initial adjoint weight function is used, (See Figure 13). However, Figure 14 demonstrates that using the final adjoint weight function is not acceptable for the same reason given

Table 5. Figure Reference for the Results of the Point-Kinetics Calculations using the Final Flux-Shape.

	Initial Adjoint Weighting	Final Adjoint Weighting
Reactivity vs Time	Figure 13.	Figure 14.
Normalized Amplitude vs Time	Figure 15.	Figure 16.
Normalized Power vs Time	Figure 17.	Figure 18.

before: the reactivity worth of the control-rods provided by the table is not sufficient to make the reactor go critical.

Figure 13 also displays the error associated with using the final flux-shape at the beginning of the transient. The reactivity for the point-kinetics models and the reference during the boron dilution transient are not the same because the final flux-shape does not correspond to the actual flux-shape of the reactor initially.

With initial adjoint weighting, the point-kinetics model utilizing the reactivity table gives the better representation of the total power and amplitude of the reactor, although there are still significant errors. This can be seen in Figures 15 and 17.

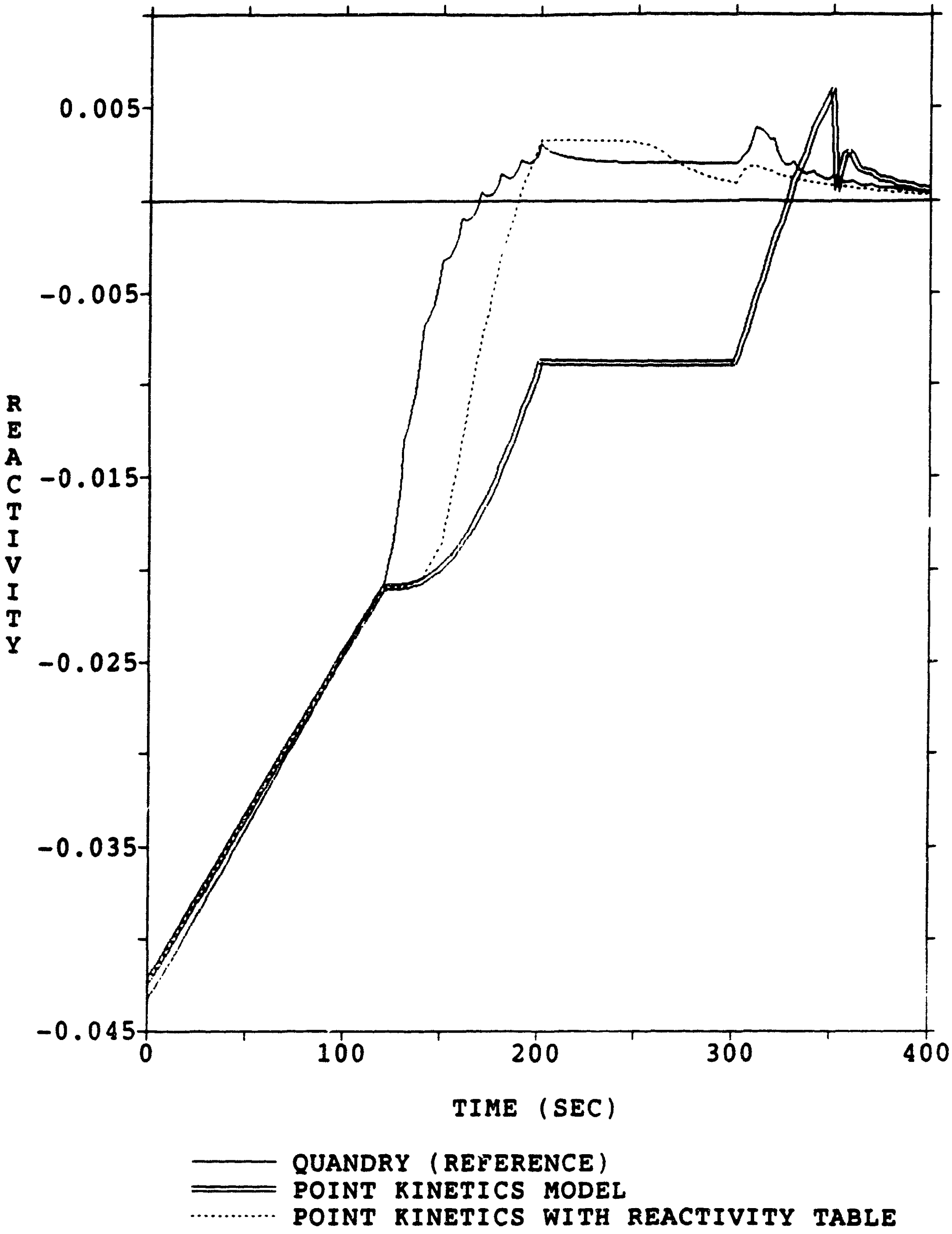


Figure 13. Reactivity versus Time using the Initial-Adjoint Weighting and the Final Flux-Shape.

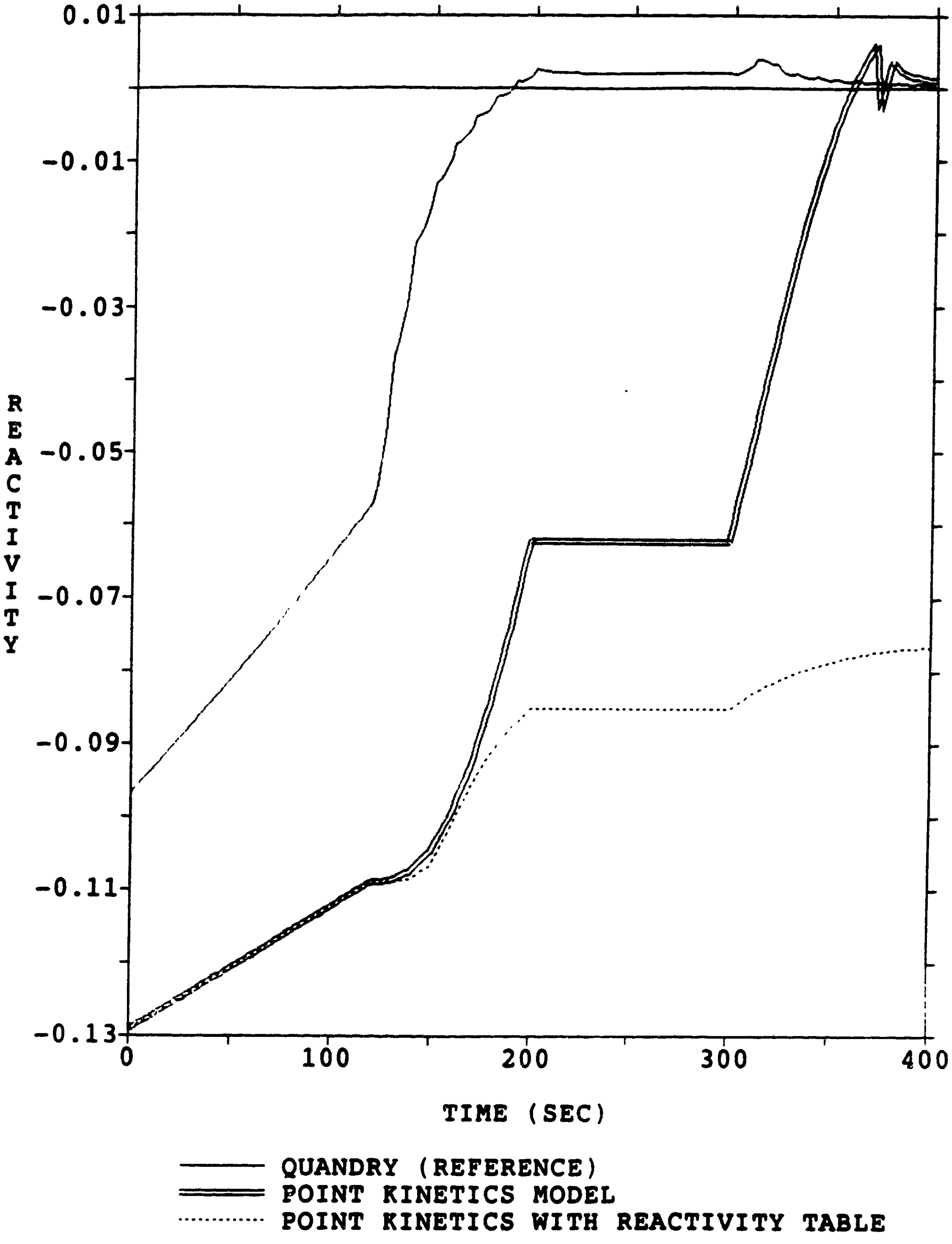


Figure 14. Reactivity versus Time using the Final-Adjoint Weighting and the Final Flux-Shape.

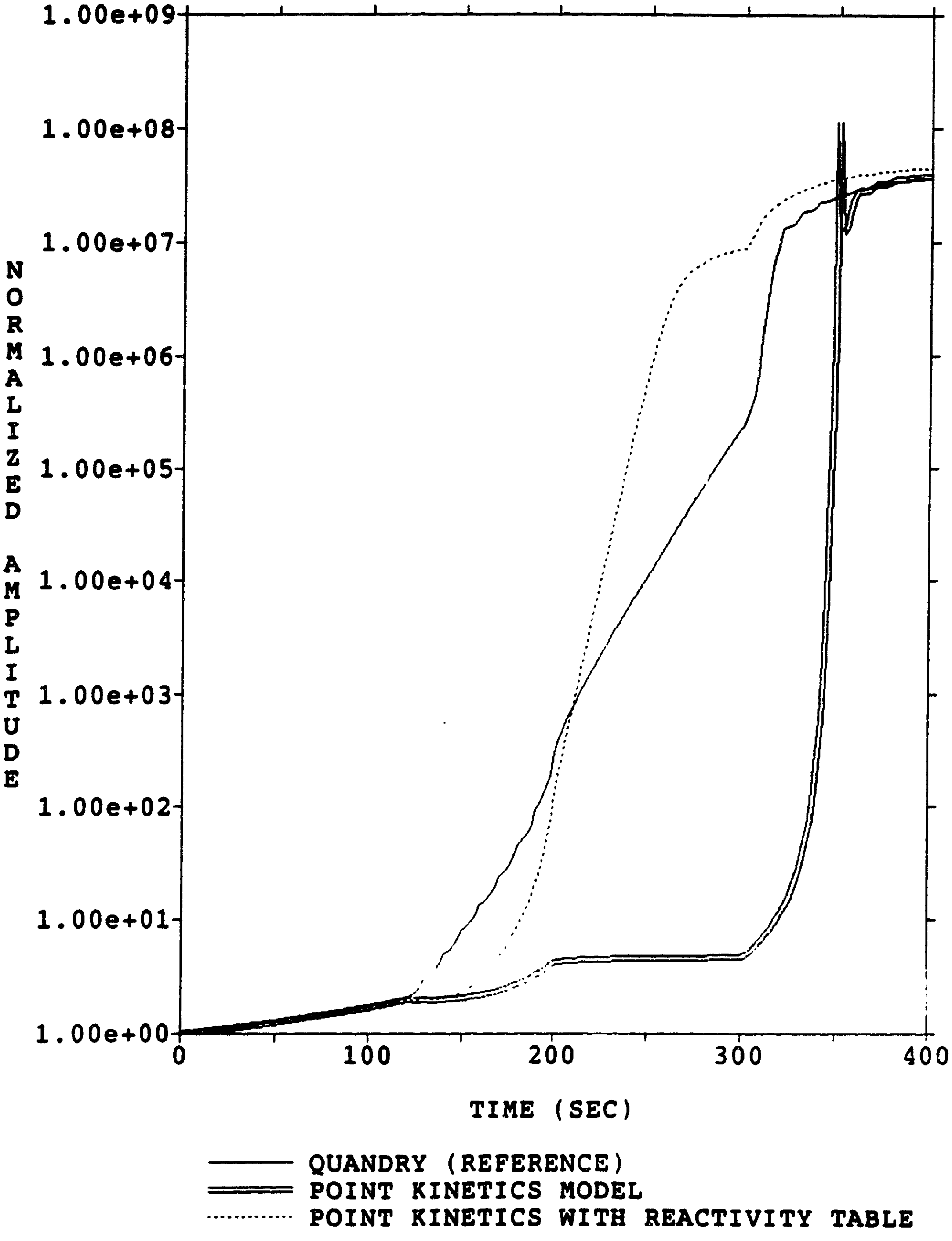


Figure 15. Normalized Amplitude versus Time using the Initial A.'joint weighting and the Final Flux-Shape.

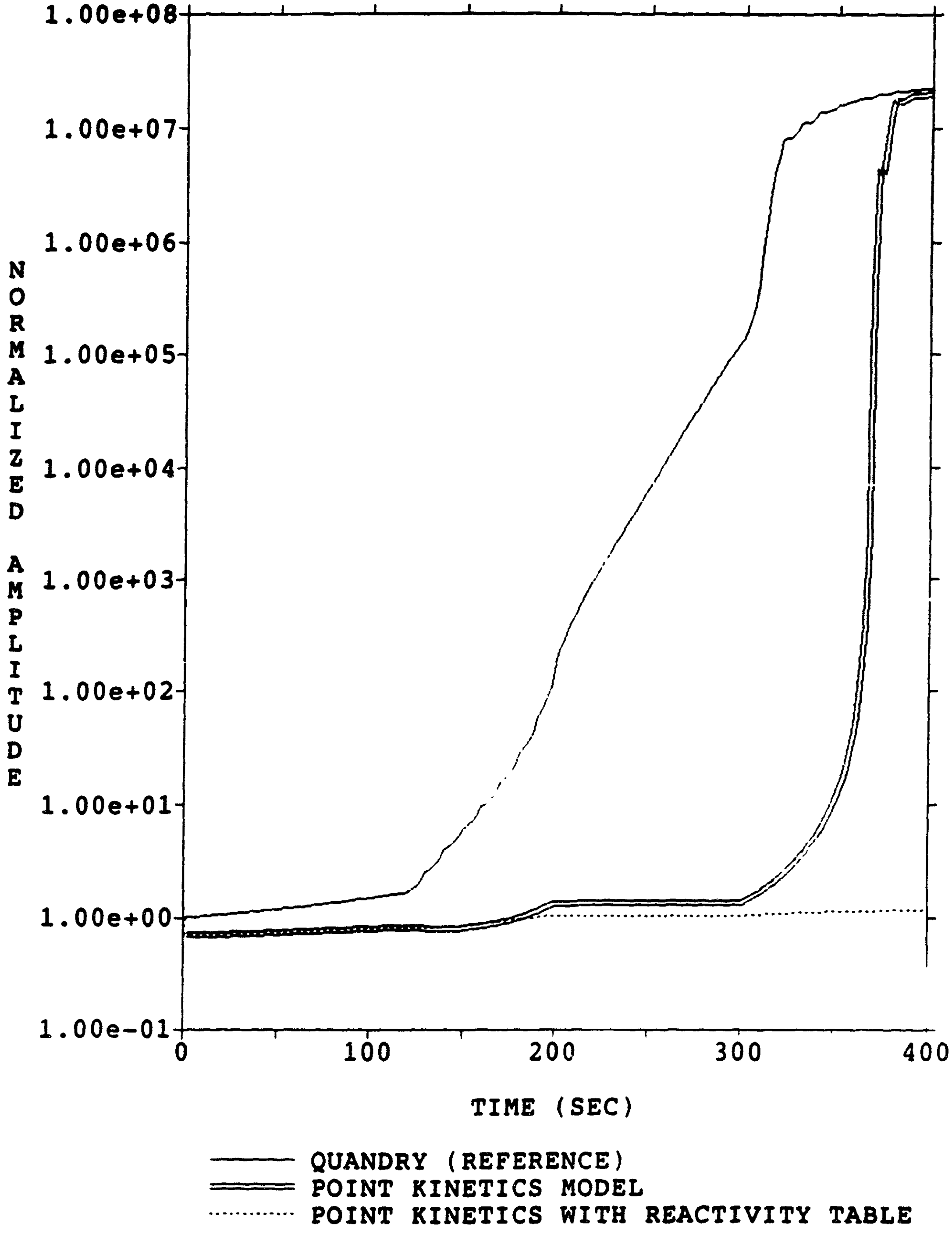


Figure 16. Normalized Amplitude versus Time using the Final-Adjoint Weighting and the Final Flux-Shape.

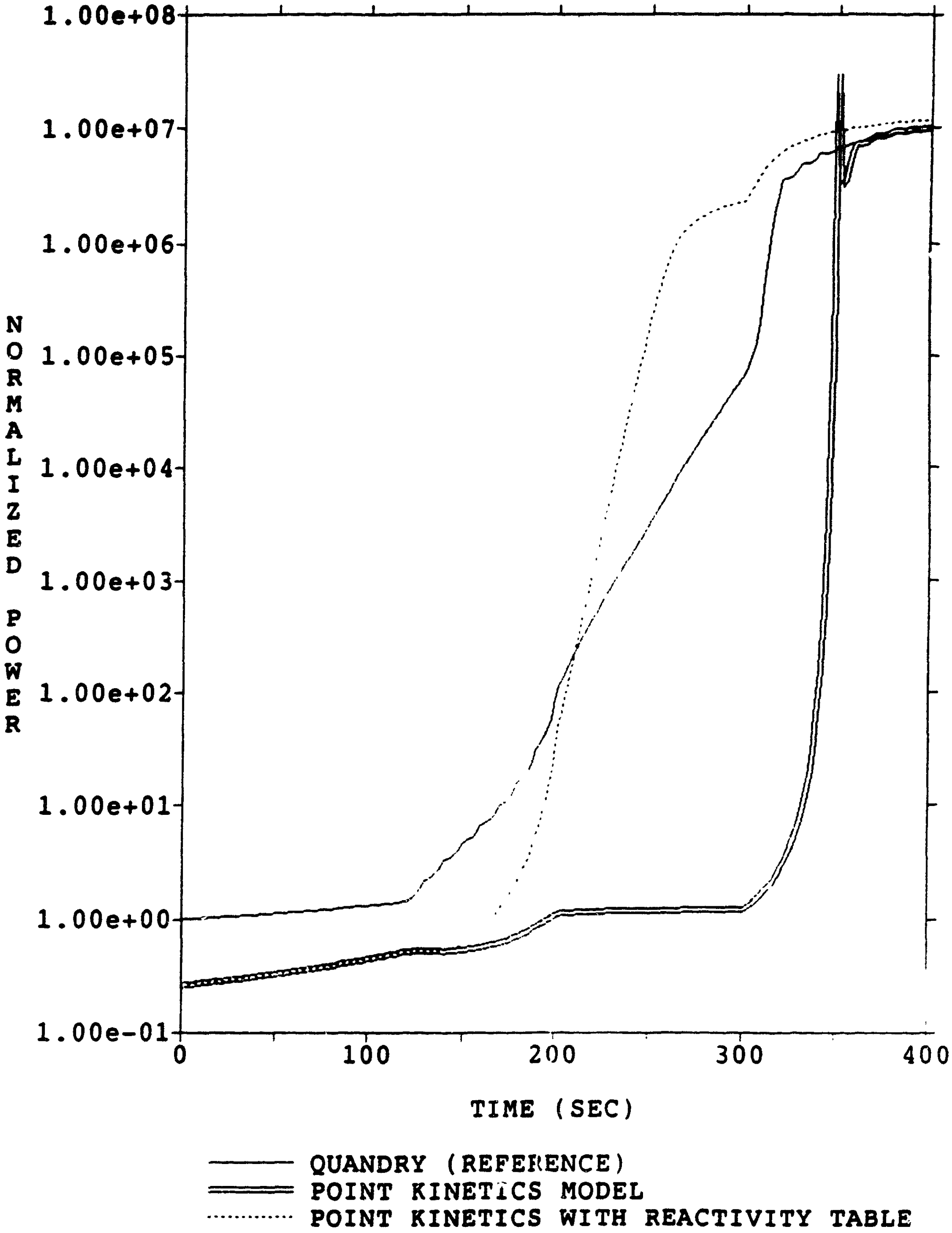


Figure 17. Normalized Power versus Time using the Initial-Adjoint Weighting and the Final Flux-Shape.

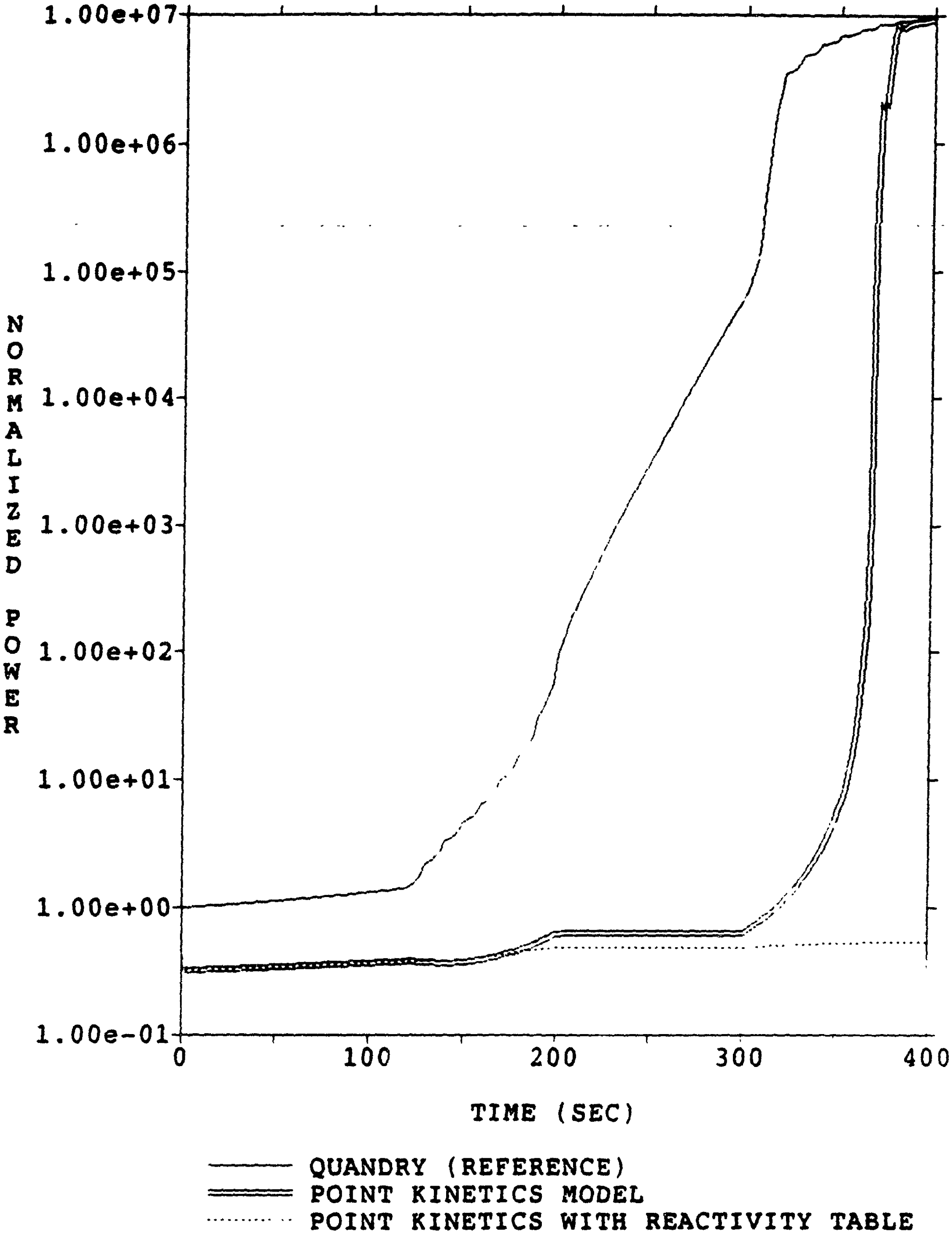


Figure 18. Normalized Power versus Time using the Final-Adjoint Weighting and the Final Flux-Shape.

4.5 COMPUTING-TIME REQUIREMENTS

It was found in an earlier study [2], that the reference space- and time-dependent calculations could represent a start-up transient very closely by using large time-steps without introducing any major errors. Large time-steps ($\Delta t = 1$ sec) compared to the dynamics of the transient. Therefore, all QUANDRY (reference) results were obtained using these large time-steps. For the point-kinetics models, since the goal is to achieve accurate results in the least amount of time, large time-steps where also used ($\Delta t = 1$ sec).

All calculations are performed on a DIGITAL/MicroVAX III computer. The total computing-time required for each type of calculation are given in Table 6.

Table 6. Total CPU-Time Requirements.

	CPU-TIME
QUANDRY (reference)	1 hr 56 min 21 sec
Point-Kinetics Model	1 hr 08 min 19 sec
Point-Kinetics with a Reactivity Table Option	1 hr 07 min 20 sec

Both point-kinetics models required less computing-time than the QUANDRY (reference) calculation when using the same time-steps. The reason being that reference calculation performed full flux-shape update calculations at every time-step while the point-kinetics models used a constant flux-shape throughout the transient. Unfortunately, the point-kinetics models took over an hour to run because they both utilized

QUANDRY's thermal-hydraulic feedback subroutines.

Furthermore, the point-kinetics model using a reactivity table required less computing-time than the regular point-kinetics model. This small savings in computing-time is expected because the reactivity is read from a table instead of being computed from Equation 34 during a control-rod transient.

4.6 SUMMARY

The results obtain from these calculations show that the point-kinetics results are in error compared to the reference results, even when a precalculated reactivity table is used. Various attempts to improve the point-kinetics results by using different adjoint weighting and flux-shapes were unsuccessful. This leads to one conclusion, the use of a single flux-shape throughout a start-up transient is not sufficient to describe the transient.

The recommendations for future research and the main summaries of this research are given in Chapter Five.

Chapter 5
CONCLUSION

5.1 SUMMARY OF THE MODIFICATIONS

A new reactivity table option has been created for the QUANDRY code. This option requires that a table of 'reactivity versus control-rod position' be produced either by static calculations or by some other method before running a transient calculation. This table makes it possible to use the QUANDRY code in a point-kinetics mode where, at any time-step, the net reactivity is formed by combining the tabulated value with a feedback value computed from a fixed flux-shape. This net reactivity is then used in the point-kinetics equations to infer a new value for the amplitude function. As a result, a combination of this amplitude function with the fixed shape function give the updated fluxes and net-leakages.

This new option was incorporated into the QUANDRY transient module with minimum modifications. These simple modifications were made to preserve the computing efficiency of the program as much as possible.

5.2 SUMMARY OF THE INVESTIGATIONS

The results of an application of this new option to a start-up transient are significantly better than if the net reactivity had been calculated directly by the perturbation formula during the transient. However, the results still compared poorly with the reference (space- and time-dependent) calculations which take into account external source effects, flux-shape variations, and thermal-hydraulic feedback effects. Therefore, further modifications of the point-kinetics model are needed to predict accurately a LWR start-up transient.

The results of the point-kinetics model also showed that using the final configuration of the reactor core as the adjoint weight functions gave very inaccurate results. Therefore, the use of the final adjoint weight functions to define the point-kinetics parameters during a start-up transient should be avoided.

For transients consisting of a small change in the flux-shape, the point-kinetics results appear to be correct. On the contrary, as the flux-shape starts to change considerably, the point-kinetics results are no longer valid. However, it is very difficult to determine exactly were the point-kinetics model fails with respect to flux-shape changes. Therefore, further research in this area is needed to determine exactly were the point-kinetics utilizing a constant flux-shape are valid.

5.3 RECOMMENDATIONS FOR FUTURE RESEARCH

One of the assumptions used in generating the precalculated reactivity table was that the reactor is initially in a "hot standby" condition. As a result, the reactivity table was calculated using the operating temperature and pressure of the reactor. However, in most start-up transients the reactor is in a "cold condition" corresponding to ambient temperature and pressure. Therefore, the temperature dependence on the reactivity table should be investigated.

Another area of research that should be investigated is the idea of precalculating the total reactivity of the reactor at certain times during the transient. This total reactivity would include contributions due to boron dilution, control-rods motion, and thermal-hydraulic feedback effects instead of using a hybrid "adiabatic" method employed in this research. The precalculated total reactivity table would now consist of a two-dimensional table were the point-kinetics model would interpolate the reactivity table to find the correct reactivity corresponding to a given time and power level.

This method would minimize any errors due to flux-shape variations in the net reactivity of the reactor. Furthermore, since the reactivity would be the correct reactivity corresponding to a particular time and (steady-state) power level during a transient, the point-kinetics results should be more accurate.

Finally, the results clearly demonstrated that the start-up transient can not be correctly simulated by the use of a single constant flux-shape throughout the transient. Therefore, research should be performed to add an option in the point-kinetics model where the shape function is recalculated whenever there is a large power change in the reactor. For example, a simple method would be to start using the initial shape function at the beginning of the transient and switch over to the final shape function somewhere during the transient. This change in the shape function can be triggered by the rise in the power level.

REFERENCES

[1] Smith K. S., "An Analytic Nodal Method for Solving the Two-Group, Multidimensional, Static and Transient Neutron Diffusion Equations," Nuclear Engineer's Thesis, Department of Nuclear Engineering, MIT, Cambridge, MA (March 1979).

[2] Jacqmin R. P., "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," Department of Nuclear Engineering, MIT, Cambridge, MA (November 1989).

[3] Smith K. S., "Spatial Homogenization Methods for Light Water Reactor Analysis," Ph.D. Thesis, Department of Nuclear Engineering, MIT, Cambridge, MA (June 1980).

[4] Jacqmin R. P., "Calculation of the Point-Kinetics Parameters from the QUANDRY Solution of the Two-Group, Time-Dependent, Nodal Diffusion Equations," Department of Nuclear Engineering, MIT, Cambridge, MA (July 1989).

[5] Jacqmin R. P., "Implementation of a Fixed-Source Option in the QUANDRY Nodal Code," Department of Nuclear Engineering, MIT, Cambridge, MA (January 1989).

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

[7] Taiwo T. A. and Henry A. F., "Perturbation Theory Based on a Nodal Model," **Nuclear Science and Engineering**, 92, 34 (1986).

[8] Flannery B. P., Press W. H., Teukolsky S. A., and Vetterling W. T., **Numerical Recipes: The Art of Scientific Computing**, Cambridge University Press (1989).

APPENDIX

APPENDIX A.

SUBROUTINE INPUTR

This subroutine contains the input variables used in the subroutine RODTAB. The input variables are reactivity as a function of time in the form of a one-dimensional array. This was accomplished by assuming that the control rods are being withdrawn at a constant rate of 2 cm/sec from the reactor core.

Definition of Variables:

XA = Independent tabulated time variable
YA = Dependent tabulated reactivity variable such
  that YA(j) = Function(XA(j))

```
Subroutine INPUTR(XA,YA)

Real XA, YA

Dimension XA(20), YA(20)

C     Time = 120 seconds
            XA(1) = 120
            YA(1) = -1.86878 E-02
C     Time = 130 seconds
            XA(2) = 130
            YA(2) = -1.86276 E-02
C     Time = 140 seconds
            XA(3) = 140
            YA(3) = -1.82104 E-02
C     Time = 150 seconds
            XA(4) = 150
            YA(4) = -1.62666 E-02
C     Time = 160 seconds
            XA(5) = 160
            YA(5) = -1.12207 E-02
C     Time = 170 seconds
            XA(6) =  170
            YA(6) = -5.52142 E-03
C     Time = 180 seconds
            XA(7) = 180
            YA(7) = -9.12547 E-04
C     Time = 190 seconds
            XA(8) = 190
            YA(8) =  2.57379 E-03
```

```
C      Time - 200 seconds
              XA(9) - 200
              YA(9) -  5.20343 E-03
C      Time - 300 seconds
              XA(10) - 300
              YA(10) -  5.20343 E-03
C      Time - 310 seconds
              XA(11) - 310
              YA(11) -  7.21282 E-03
C      Time - 320 seconds
              XA(12) - 320
              YA(12) -  8.77339 E-03
C      Time - 330 seconds
              XA(13) - 330
              YA(13) -  1.00054 E-02
C      Time - 340 seconds
              XA(14) - 340
              YA(14) -  1.09917 E-02
C      Time - 350 seconds
              XA(15) - 350
              YA(15) -  1.17899 E-02
C      Time - 360 seconds
              XA(16) - 360
              YA(16) -  1.24387 E-02
C      Time - 370 seconds
              XA(17) - 370
              YA(17) -  1.29603 E-02
C      Time - 380 seconds
              XA(18) - 380
              YA(18) -  1.33600 E-02
C      Time - 390 seconds
              XA(19) - 390
              YA(19) -  1.36210 E-02
C      Time - 400 seconds
              XA(20) - 400
              YA(20) -  1.37264 E-02

Return

End
```

APPENDIX B.

SUBROUTINE RODTAB

Given arrays XA and YA, each of length N, and given a value X, this subroutine returns a value Y. If P(x) is a polynomial of degree N - 1 such that $P(XA_i) = YA_i$, where i=1,...,N, then the returned value is Y = P(X). This polynomial interpolation is based on Neville's Algorithm [8].

Definition of Variables:

```
XA - Independent tabulated time variable
YA - Dependent tabulated reactivity variable such
       that YA(j) - Function(XA(j))
X  - Given time value
Y  - Reactivity value found from array time
       interpolation
DY - Error estimate based on interpolation
```

```
Subroutine RODTAB(XA,YA,N,X,Y)

Real XA, YA, X, Y, DY, DIF, DIFT, HO, HP, W, DEN, C, D

Integer NS, N, M, I

Parameter (NMAX - 20)

Dimension XA(N), YA(N), C(NMAX), D(NMAX)

C     N - Array length

      NS - 1
      DIF - ABS(X - XA(1))

C     Find the index NS of the closest table value

      DO 10 I - 1, N
            DIFT - ABS(X - XA(I))
            IF (DIFT .LT. DIF) THEN
                  NS - I
                  DIF - DIFT
            END IF

C     Initialize the table of C's and D's, the C's and D's
C     are the corrections which make the Neville's Algorithm
C     one order higher
```

```
            C(I) = YA(I)
            D(I) = YA(I)

10     CONTINUE


C      Initial approximation for Y

       Y = YA(NS)
       NS = NS - 1

C      Find the correct value for Y

       DO 30 M = 1, N - 1

             DO 20 I = 1, N - M
                   HO = XA(I) - X
8                  HP = XA(I + M) -X
                   W = C(I + 1) - D(I)
                   DEN = HO - HP

C      Update the C's and D's

                   DEN = W / DEN
                   D(I) = HP * DEN
                   C(I) = HO * DEN
20           CONTINUE

C      Decide which correction factor, C or D, to add to
C      the value of Y

             IF (2 * NS .LT. N - M) THEN
                   DY = C(NS + 1)
             ELSE
                   DY = D(NS)
             NS = NS - 1
             END IF

             Y = Y + DY

30     CONTINUE

       RETURN

       END
```

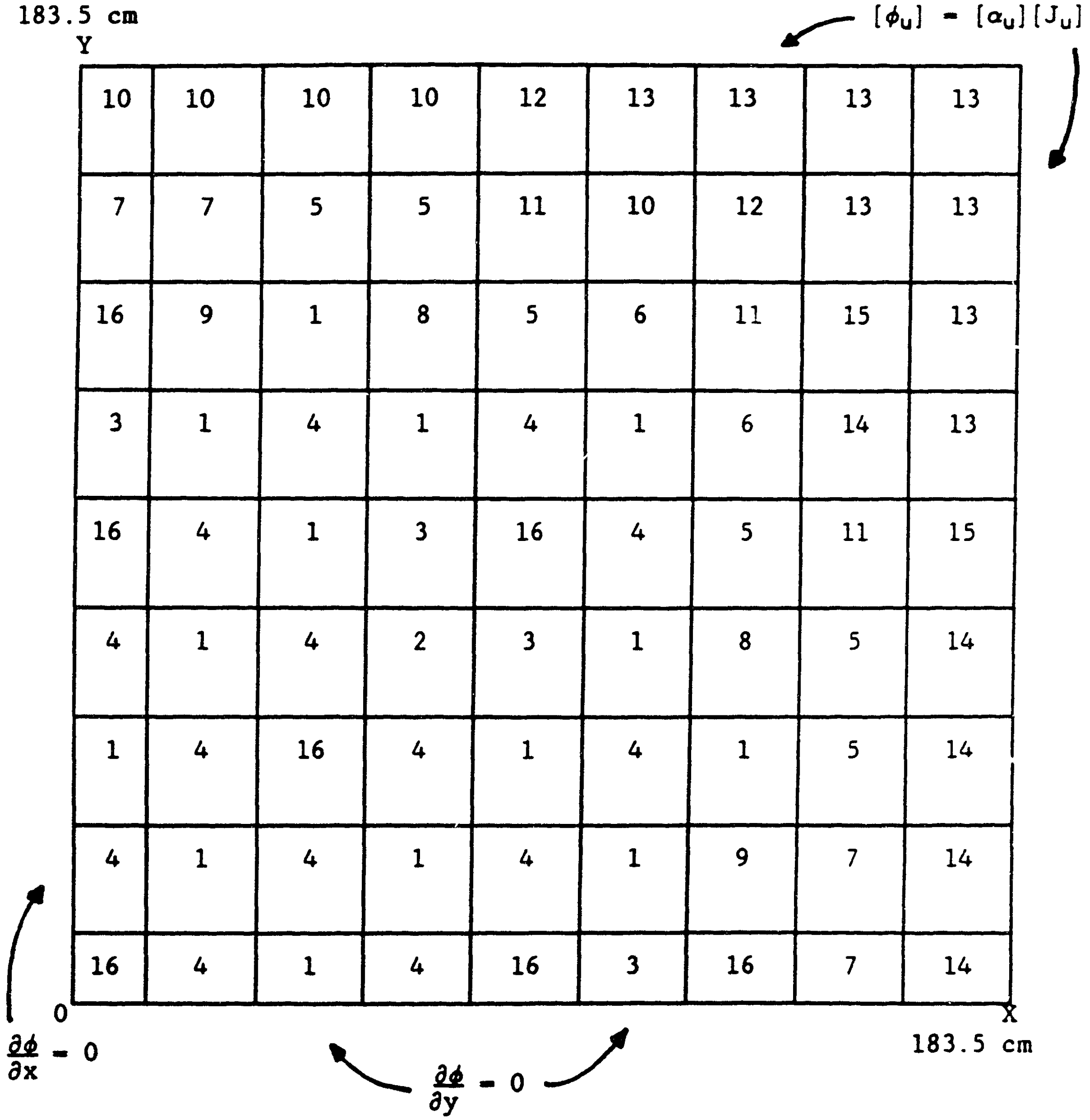
APPENDIX C.

DESCRIPTION OF THE SALEM-1 MODEL REACTOR CORE

I. GEOMETRY

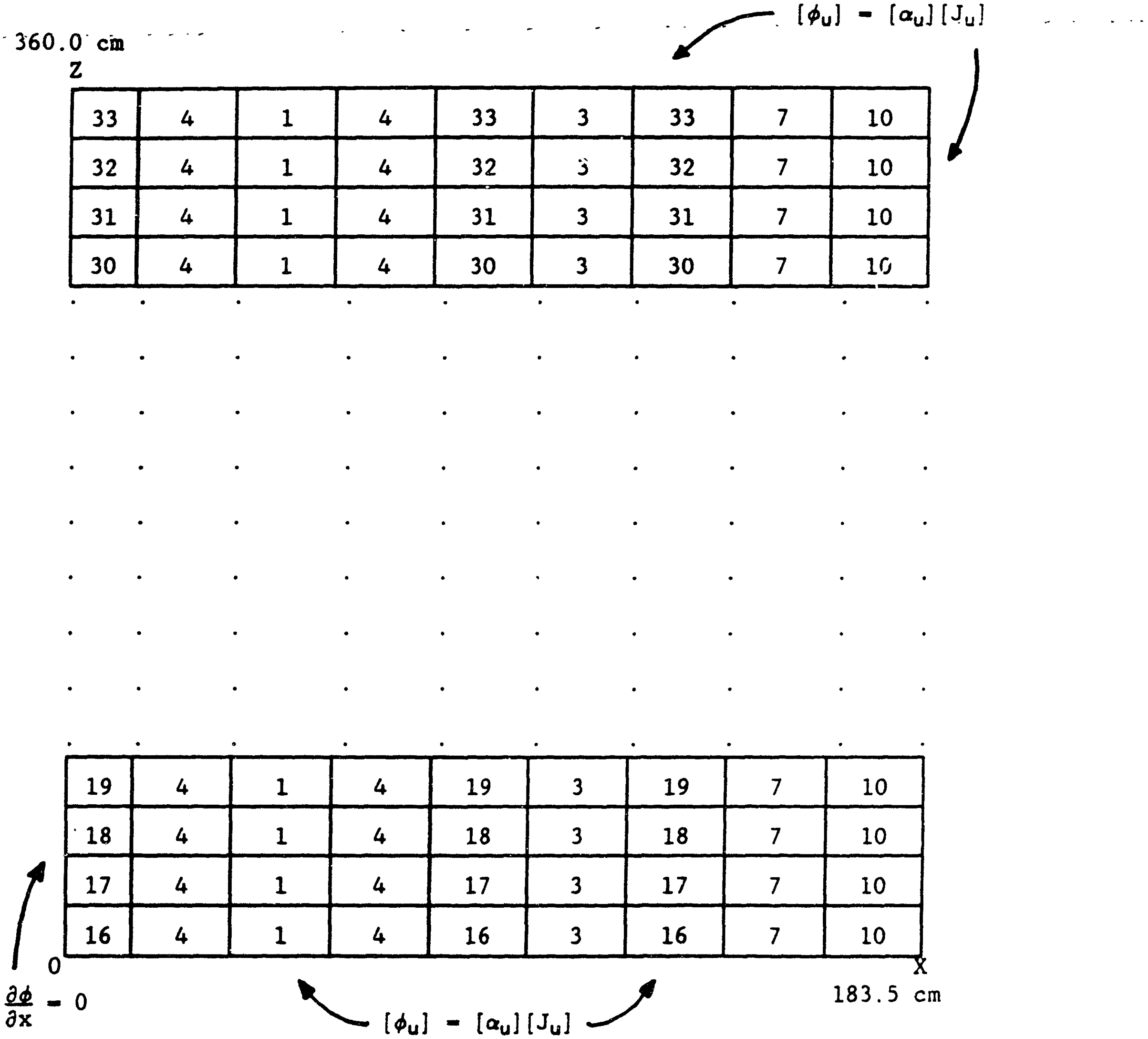
This is a three-dimensional representation of the first quadrant of the core. There is quarter-core symmetry. The quarter-core model contains 1458 nodes of equal sizes: 21.6 cm x 21.6 cm x 20.0 cm, with 18 nodes in the vertical direction.

1. X-Y CROSS SECTION IN THE BOTTOM (Z=0) PLANE (1/4 CORE)


183.5 cm
Y
$[\phi_u] = [\alpha_u][J_u]$
0
X
183.5 cm
$\frac{\partial\phi}{\partial x} = 0$
$\frac{\partial\phi}{\partial y} = 0$

10	10	10	10	12	13	13	13	13
7	7	5	5	11	10	12	13	13
16	9	1	8	5	6	11	15	13
3	1	4	1	4	1	6	14	13
16	4	1	3	16	4	5	11	15
4	1	4	2	3	1	8	5	14
1	4	16	4	1	4	1	5	14
4	1	4	1	4	1	9	7	14
16	4	1	4	16	3	16	7	14

2. X-Z CROSS-SECTION IN THE BOTTOM (Y=0) PLANE.


$[\phi_u] = [\alpha_u][J_u]$
360.0 cm
Z
0
X
183.5 cm
$\frac{\partial\phi}{\partial x} = 0$
$[\phi_u] = [\alpha_u][J_u]$

33	4	1	4	33	3	33	7	10
32	4	1	4	32	3	32	7	10
31	4	1	4	31	3	31	7	10
30	4	1	4	30	3	30	7	10

19	4	1	4	19	3	19	7	10
18	4	1	4	18	3	18	7	10
17	4	1	4	17	3	17	7	10
16	4	1	4	16	3	16	7	10

II. MATERIAL PROPERTIES

Comp. #	Group g	D (cm)	Σ_a (cm^{-1})	$\nu\Sigma_f$ (cm^{-1})	$\Sigma_{g'g}$ (cm^{-1})
1	1	1.3648	0.008887	0.005550	0.017245
	2	0.4826	0.130772	0.185823	0.0
2	1	1.3603	0.009661	0.006267	0.015942
	2	0.4776	0.169403	0.229195	0.0
3	1	1.3596	0.009957	0.006267	0.015398
	2	0.4798	0.181915	0.230258	0.0
4	1	1.3592	0.010104	0.006269	0.015128
	2	0.4810	0.188426	0.230923	0.0
5	1	1.3594	0.009509	0.006890	0.016386
	2	0.4673	0.169073	0.264760	0.0
6	1	1.35898	0.009692	0.006890	0.016049
	2	0.46853	0.176288	0.265397	0.0
7	1	1.35890	0.009730	0.006890	0.015981
	2	0.46875	0.177654	0.265512	0.0
8	1	1.3576	0.010252	0.006892	0.015022
	2	0.4728	0.200287	0.267778	0.0
9	1	1.3572	0.010399	0.006894	0.014752
	2	0.4740	0.206951	0.268552	0.0
10, 14	1	1.4957	0.002683	0.0	0.022923
	2	0.3637	0.051595	0.0	0.0
11	1	1.3933	0.003541	0.0	0.017943
	2	0.3659	0.068149	0.0	0.0
12, 15	1	1.6701	0.001220	0.0	0.031408
	2	0.3621	0.039330	0.0	0.0
13	1	1.7446	0.005960	0.0	0.035032
	2	0.3614	0.034208	0.0	0.0

Comp. #	Group g	D (cm)	Σ_a (cm^{-1})	$\nu\Sigma_f$ (cm^{-1})	$\Sigma_{g'g}$ (cm^{-1})
16 to 33 (rodded case)	1	1.321964	0.013482	0.005567	0.015178
	2	0.486196	0.211003	0.194976	0.0
16 to 33 (unrodded case)	1	1.321964	0.013482	0.005567	0.015178
	2	0.486196	0.101003	0.194976	0.0

$\nu = 2.5$

$\chi_{p_1} = 1.0$

$\chi_{p_2} = 0.0$

$f_{1_{u-}} = f_{2_{u+}} = 1.0 \qquad u = x,\ y,\ z$

$v_1 = 1.25\ 10^7\ \text{cm s}^{-1}$

$v_2 = 2.50\ 10^5\ \text{cm s}^{-1}$ for all compositions

III. ALBEDO-TYPE BOUNDARY CONDITIONS

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

Boundary	α_{11}	α_{12}	α_{21}	α_{22}
Axial	4.011	0.0	2.805	8.993
Radial	2.0	0.0	0.0	2.0

IV. DELAYED-NEUTRON DATA

$\chi_{d_1} = 1.0$

$\chi_{d_2} = 0.0$

$\beta = 0.00650$

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

V. THERMAL-HYDRAULIC DATA AND FEEDBACK COEFFICIENTS

1. THERMAL-HYDRAULIC DATA

Specific heat of the fuel	- 2.460 10^{6} erg g^{-1} K^{-1}
Specific heat of the coolant	- 5.430 10^{7} erg g^{-1} K^{-1}
Density of the fuel	- 1.030 10^{1} g cm^{-3}
Initial mass flow rate	- 3.868 10^{6} g s^{-1}
Film coefficient at the initial flow rate	- 3.293 10^{7} erg cm^{-2} s^{-1} K^{-1}
Conductivity/conduction length of the fuel clad	- 2.200 10^{6} erg cm^{-2} s^{-1} K^{-1}
Surface area of clad/ volume of coolant	- 3.097 10^{0} cm^{-1}
Volume fraction of the coolant	- 5.420 10^{-1}
Fuel reference temperature	- 5.330 10^{2} K
Coolant density temperature	- 7.975 10^{-1} g cm^{-3}
Coolant reference temperature	- 5.330 10^{2} K
Inlet temperature of the coolant	- 5.550 10^{2} K
Fraction of fission energy released in the coolant	- 2.600 10^{-2}
Coolant pressure	- 1.551 10^{7} Pa
Partial of density times enthalpy with respect to the coolant temperature	- 1.600 10^{7} erg cm^{-3} K^{-1}
Energy conversion factor	- 3.204 10^{-11} J/fission

2. FEEDBACK COEFFICIENTS

A. For compositions # 1 through 9 and 16 through 33

- fuel temperature

$$\frac{\partial}{\partial T^{f}}\frac{1}{D_1} = -6.6\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^{f}}\frac{1}{D_2} = -2.6\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^{f}}\Sigma_{c_1} = 3.3\ 10^{-7}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^{f}}\Sigma_{c_2} = -3.8\ 10^{-7}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^{f}}\Sigma_{21} = -8.5\ 10^{-8}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^{f}}\Sigma_{f_2} = -1.0\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^{f}}\nu\Sigma_{f_2} = -2.5\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

- coolant temperature

$$\frac{\partial}{\partial T^{c}}\frac{1}{D_1} = -8.0\ 10^{-5}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^{c}}\frac{1}{D_2} = -1.3\ 10^{-3}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\Sigma_{c_1} = 3.0\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\Sigma_{c_2} = -8.2\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\Sigma_{21} = -1.5\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\Sigma_{f_2} = -8.3\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\nu\Sigma_{f_2} = -2.075\ 10^{-5}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

B. For node compositions # 10 through 15

– fuel temperature

$$\frac{\partial}{\partial T^f}\frac{1}{D_1} = -6.6\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^f}\frac{1}{D_2} = -2.6\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^f}\Sigma_{c_1} = 3.3\ 10^{-7}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^f}\Sigma_{c_2} = -3.8\ 10^{-7}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^f}\Sigma_{21} = -8.5\ 10^{-8}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

- coolant temperature

$$\frac{\partial}{\partial T^c}\frac{1}{D_1} = -8.0\ 10^{-5}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\frac{1}{D_2} = -1.3\ 10^{-3}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\Sigma_{c_1} = 3.0\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\Sigma_{c_2} = -8.2\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$

$$\frac{\partial}{\partial T^c}\Sigma_{21} = -1.5\ 10^{-6}\ \mathrm{cm}^{-1}\mathrm{K}^{-1}$$