VARIATIONAL DERIVATION OF MODAL-NODAL
FINITE DIFFERENCE EQUATIONS IN
SPATIAL REACTOR PHYSICS

by

Patrick G. Bailey, Allan F. Henry

July 1972

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

AEC Research and Development Report
Contract AT(11-1)-3052
U.S. Atomic Energy Commission
VARIATIONAL DERIVATION OF MODAL-NODAL
FINITE DIFFERENCE EQUATIONS IN
SPATIAL REACTOR PHYSICS

by

Patrick G. Bailey, Allan F. Henry
July 1972

COO - 3052 - 5
MITNE - 138

AEC Research and Development Report
Contract AT(11-1) - 3052
U.S. Atomic Energy Commission
VARIATIONAL DERIVATION OF MODAL-NODAL
FINITE DIFFERENCE EQUATIONS IN
SPATIAL REACTOR PHYSICS

by

PATRICK G. BAILEY

Submitted to the Department of Nuclear Engineering on July 18, 1972, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

A class of consistent coarse mesh modal-nodal approximation methods is presented for the solution of the spatial neutron flux in multigroup diffusion theory. The methods are consistent in that they are systematically derived as an extension of the finite element method by utilizing general modal-nodal variational techniques. Detailed subassembly solutions, found by imposing zero current boundary conditions over the surface of each subassembly, are modified by piecewise continuous Hermite polynomials of the finite element method and used directly in trial function forms. Methods using both linear and cubic Hermite basis functions are presented and discussed.

The proposed methods differ substantially from the finite element methods in which homogeneous nuclear constants, homogenized by flux weighting with detailed subassembly solutions, are used. However, both schemes become equivalent when the subassemblies themselves are homogeneous.

One-dimensional, two-group numerical calculations using representative PWR nuclear material constants and 18-cm subassemblies were performed using entire subassemblies as coarse mesh regions. The results indicate that the proposed methods can yield comparable if not superior criticality measurements, comparable regional power levels, and extremely accurate subassembly fine flux structure with little increase of computational effort in comparison with existing coarse mesh methods.

Thesis Supervisors: Allan F. Henry
Kent F. Hansen

Titles: Professors of Nuclear Engineering
ACKNOWLEDGMENTS

I would like to express my deep appreciation and sincerest gratitude to Dr. Allan F. Henry whose advice and consultation have been of considerable help during the course of this work. Thanks must also be extended to Dr. Kent F. Hansen, who reviewed the final manuscript and provided many beneficial discussions and programming assistance.

It has also been a privilege to have known and worked with the late Dr. T. J. Thompson, whose untimely death was a great loss to the nation as well as to the department at M.I.T., and Dr. Elias P. Gyftopoulos. Their insistence on high standards of education and quality as well as clarity of expression will continue to remain with me always.

Fellowship support by the U.S. Atomic Energy Commission is also gratefully acknowledged. The work was performed under USAEC Contract AT (30-1) - 3903. All computations were performed on IBM 360/65 and 370/155 computers at the M.I.T. Information Processing Center.

Special appreciation is extended to my wife, Alice, whose patience and understanding have provided immeasurable support throughout the course of this work.

Finally, special thanks are expressed to Mrs. Mary Bosco, who has once again demonstrated her expert ability as a superb technical typist.
# TABLE OF CONTENTS

Abstract 2
Acknowledgments 3
List of Figures 7
List of Tables 10

Chapter 1. Introduction 11
  1.1 Preface 11
  1.2 The Time-Independent, Multigroup Diffusion Theory Equations 13
  1.3 Solution Methods 15
    1.3.1 Nodal Methods 16
    1.3.2 Modal Methods 19
    1.3.3 Modal-Nodal Methods 25

  2.1 Calculus of Variations Applied to Diffusion Theory 28
  2.2 Discontinuous Trial Functions 30
  2.3 The Finite Element Approximation Methods 34
    2.3.1 The Conventional Finite Difference Equations 35
    2.3.2 Multichannel Polynomial Synthesis 37
    2.3.3 The Linear Basis Function Approximation 46
    2.3.4 The Cubic Hermite Basis Function Approximation 47
Appendix A. Table of Symbols

Appendix B. Difference Equation Coefficients Resulting from Use of the Finite Element Approximation Methods
   B.1 Coefficients of the Conventional Finite Difference Equations
   B.2 Coefficients of the Linear Finite Element Method Equations
   B.3 Coefficients of the Cubic Hermite Finite Element Method Equations

Appendix C. Difference Equation Coefficients Resulting from Use of the Proposed Approximation Methods
   C.1 Coefficient-Integrands of the Proposed Approximation Method Equations Using Linear Basis Functions
   C.2 Coefficient-Integrands of the Proposed Approximation Method Equations Using Cubic Hermite Basis Functions

Appendix D. Description of the Computer Programs
   D.1 Description of Program REF2G
   D.2 Description of Program LINEAR
   D.3 Description of Program CUBIC
   D.4 Description of Program ANALYZE

Appendix E. Sample Input and Output Data Blocks for Programs REF2G, LINEAR, CUBIC, and ANALYZE
   (Included in only the first six copies)
   E.1 REF2G
   E.2 LINEAR
   E.3 CUBIC
   E.4 ANALYZE

Appendix F. Source Listings of the Programs
   (Included in only the first six copies)
   F.1 REF2G
   F.2 LINEAR
   F.3 CUBIC
   F.4 ANALYZE
<table>
<thead>
<tr>
<th>No.</th>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>1.1</td>
<td>Illustration of One-Dimensional Multichannel Synthesis</td>
<td>23</td>
</tr>
<tr>
<td>1.2</td>
<td>1.2</td>
<td>Illustration of One-Dimensional Overlapping Multichannel Synthesis</td>
<td>23</td>
</tr>
<tr>
<td>2.1</td>
<td>2.1</td>
<td>Conventional Nodal Finite Difference Approximation Trial Function Forms</td>
<td>36</td>
</tr>
<tr>
<td>2.2</td>
<td>2.2</td>
<td>Matrix Form of the Conventional Finite Difference Equations</td>
<td>38</td>
</tr>
<tr>
<td>2.3</td>
<td>2.3</td>
<td>Basis Functions of Equations 2.23 for N = 0, 1, 2, and 3</td>
<td>43</td>
</tr>
<tr>
<td>2.4</td>
<td>2.4</td>
<td>Cubic B Spline $\Omega^B_k(z)$</td>
<td>45</td>
</tr>
<tr>
<td>2.5</td>
<td>2.5</td>
<td>Cubic Hermite Basis Functions $\Omega^H_1(z)$ and $\Omega^H_2(z)$</td>
<td>45</td>
</tr>
<tr>
<td>2.6</td>
<td>2.6</td>
<td>Matrix Form of the Linear Finite Element Method Approximation</td>
<td>48</td>
</tr>
<tr>
<td>2.7</td>
<td>2.7</td>
<td>Matrix Form of the Cubic Hermite Finite Element Method Approximation</td>
<td>51</td>
</tr>
<tr>
<td>4.1</td>
<td>4.1</td>
<td>Solution of $\mathbf{A}<em>{\mathbf{F}} = \frac{1}{\lambda} \mathbf{B}</em>{\mathbf{F}}$ Using the Fission Source Power Iteration Method Without Fission Source Renormalization</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>4.2</td>
<td>Subassembly Notations and Detailed Solutions</td>
<td>79</td>
</tr>
<tr>
<td>5.1</td>
<td>5.1</td>
<td>Subassembly Configuration Geometries</td>
<td>87</td>
</tr>
<tr>
<td>5.2</td>
<td>5.2</td>
<td>Mesh Geometry in Half a Subassembly</td>
<td>88</td>
</tr>
<tr>
<td>5.3</td>
<td>5.3</td>
<td>Subassembly Detailed Flux Solutions for the One-Group Case</td>
<td>89</td>
</tr>
<tr>
<td>5.4</td>
<td>5.4</td>
<td>Subassembly Detailed Flux Solutions for the Two-Group Case</td>
<td>90</td>
</tr>
<tr>
<td>5.5</td>
<td>5.5</td>
<td>Subassembly Detailed Adjoint Flux Solutions for the Two-Group Case</td>
<td>91</td>
</tr>
<tr>
<td>5.6</td>
<td>5.6</td>
<td>Geometry of the Four Case Studies Composed of Types of Subassemblies</td>
<td>96</td>
</tr>
<tr>
<td>No.</td>
<td>Case 1: One-Group Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>Case 1: One-Group Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.9</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td>Case 1: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.11</td>
<td>101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.12</td>
<td>Case 1: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.13</td>
<td>102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.14</td>
<td>Case 1: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.15</td>
<td>103</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.16</td>
<td>Case 1: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.17</td>
<td>104</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.18</td>
<td>Case 1: Two-Group Fast Results Using Cubic Hermite Finite Element Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.19</td>
<td>105</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.20</td>
<td>Case 1: Two-Group Thermal Results Using Cubic Hermite Finite Element Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.21</td>
<td>106</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.22</td>
<td>Case 1: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 9-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.23</td>
<td>107</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.24</td>
<td>Case 1: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 9-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.25</td>
<td>108</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.26</td>
<td>Case 1: Two-Group Fast Results Using Cubic Hermite Finite Element Approximations and 9-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.27</td>
<td>109</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.28</td>
<td>Case 1: Two-Group Thermal Results Using Cubic Hermite Finite Element Approximations and 9-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.29</td>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.30</td>
<td>Case 2: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.31</td>
<td>111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.32</td>
<td>Case 2: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.33</td>
<td>112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.34</td>
<td>Case 2: Two-Group Fast Results Using Cubic Hermite Finite Element Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.35</td>
<td>113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.36</td>
<td>Case 2: Two-Group Thermal Results Using Cubic Hermite Finite Element Approximations and 18-cm Coarse Mesh Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.37</td>
<td>114</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.20 Case 2: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions 115
5.21 Case 2: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions 116
5.22 Case 2: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions 117
5.23 Case 3: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions 119
5.24 Case 3: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions 120
5.25 Case 3: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions 121
5.26 Case 3: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions 122
5.27 Case 4: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions 125
5.28 Case 4: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions 126
5.29 Case 4: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions 127
5.30 Case 4: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions 128
F.1 Structure of Program REF2G 185
F.2 Structure of Program LINEAR 243
F.3 Structure of Program CUBIC 288
F.4 Structure of Program ANALYZE 351
# LIST OF TABLES

<table>
<thead>
<tr>
<th>No.</th>
<th>Table Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Matrix Order $N$ of the Proposed Linear Basis Function Approximations as a Function of the Imposed Boundary Conditions</td>
</tr>
<tr>
<td>5.1</td>
<td>Representative Two-Group, 18-cm, PWR Subassembly Regional Nuclear Constants</td>
</tr>
<tr>
<td>5.2</td>
<td>Representative One-Group, 18-cm, PWR Subassembly Regional Nuclear Constants</td>
</tr>
<tr>
<td>5.3</td>
<td>Homogenized Subassembly One-Group Nuclear Constants</td>
</tr>
<tr>
<td>5.4</td>
<td>Homogenized Subassembly Two-Group Nuclear Constants</td>
</tr>
<tr>
<td>5.5</td>
<td>Test Results Using Three Consecutive Type A Subassemblies</td>
</tr>
<tr>
<td>5.6</td>
<td>Results of Case 1</td>
</tr>
<tr>
<td>5.7</td>
<td>Two-Group Results of Case 2</td>
</tr>
<tr>
<td>5.8</td>
<td>Results of Case 3</td>
</tr>
<tr>
<td>5.9</td>
<td>Results of Case 4</td>
</tr>
<tr>
<td>D.1</td>
<td>Sample Storage Requirements and Execution Times of the Programs for Two-Group Results</td>
</tr>
</tbody>
</table>
Chapter 1
INTRODUCTION

1.1 Preface

The large variety of approximation methods and techniques used in computational reactor analysis and simulation has caused the area of numerical reactor physics to become one of the most exciting areas in applied nuclear reactor physics today. The application of numerical analysis is most important in two phases of reactor design; feasibility studies and safety analysis. The primary consideration of the reactor physicist has been and must continue to be the safety of the reactor during and after any foreseeable nuclear accident. A realistic safety analysis can be obtained only if all the physical processes occurring within the reactor can be adequately described and related. Since all of these processes can be shown to be dependent upon the neutron density distribution throughout the reactor core, a detailed solution of the spatial neutron flux is vital.\(^1\)

The dynamic characteristics of a reactor strongly depend upon the spatial approximation and solution of the neutron flux. Approximation methods utilizing gross averaging of the flux near localized strong absorption and production regions, such as cruciform control rods or small water channels, can lead to inaccurate results. Large errors may result from the use of such methods in spatial kinetics problems such as depletion and xenon oscillation calculations. Much attention has therefore been focused upon approximation methods
which can obtain detailed spatial neutron flux distributions within large reactor cores.

The Boltzmann neutron transport equation\(^2\) is considered to be a sufficiently detailed description of the physical processes occurring within a nuclear reactor, and naturally is most difficult to solve. The P-1 and diffusion theory approximations\(^3\) greatly simplify the transport equation into more tractable equations which have been found to approximate adequately the flux distributions for most large-core reactors such as PWR, BWR, and LMFBR core geometries. The advent of high speed digital computers has enabled widespread use of diffusion theory because of its simple mathematical form and straightforward numerical solution techniques inherent with its use.

The treatment of the spatial approximation in diffusion theory is the primary concern of this report. There is in existence an increasingly abundant variety of such approximation methods currently in use. Fine mesh methods,\(^4\) for example, can yield very accurate results through the use of extremely large numbers of unknowns. However, such methods may well exceed the storage capacity of present day computers, as well as being exceedingly costly. Coarse mesh methods and particularly synthesis techniques,\(^5\) on the other hand, have recently become attractive as the number of unknowns can be drastically reduced, although the accuracy of many of these methods is in doubt.

The purpose of this report is twofold: first, to present the general development of variational approximation methods used to derive
difference approximations to the neutron diffusion equation; and second, to extend this development in order to develop systematically a class of consistent coarse mesh approximation methods which can approximate accurately the detailed spatial neutron flux and can also be easily incorporated into present day computer codes. As this report will deal only with the spatial approximation, the inclusion of time dependence will be set aside for future study.

1.2 The Time-Independent, Multigroup Diffusion Theory Equations

The energy discretized multigroup P-1 approximation to the Boltzmann neutron transport equation excluding time dependence can be written in standard group notation for each energy group as follows:

\[ j_g(r) + D_g(r) \nabla \phi_g(r) = 0 \]  

\[ \nabla \cdot j_g(r) + \sum_{g'} \Sigma_{gg'}(r) \phi_{g'}(r) = \frac{1}{\lambda} \chi_g \sum_{g'} \nu \Sigma_{fg'}(r) \phi_{g'}(r) \]  

where the group index \( g \) runs from the highest energy group, 1, to the lowest energy group, \( G \). The symbols and notation used throughout this report are summarized in Appendix A. Equations 1.1 are the standard \( P-1 \) equations which relate the vector neutron current \( j_g(r) \) for each energy group \( g \) with the scalar neutron flux \( \phi_g(r) \). The current may be eliminated via Fick's law, Eq. 1.1a, in order to obtain the multigroup diffusion equation:

\[ -\nabla \cdot D_g(r) \nabla \phi_g(r) + \sum_{g'} \Sigma_{gg'}(r) \phi_{g'}(r) = \frac{1}{\lambda} \chi_g \sum_{g'} \nu \Sigma_{fg'}(r) \phi_{g'}(r) \]  

(1.2)
Equation 1.2 can be written in operator matrix notation as

\[-\nabla \cdot \mathbf{D}(r) \nabla \phi(r) + [\mathbf{IM}(r) - \mathbf{T}(r)] \phi(r) = \frac{1}{\lambda} \mathbf{IB}(r) \phi(r) \]  

(1.3)

where \( \mathbf{D}, \mathbf{IM}, \mathbf{T}, \) and \( \mathbf{IB} \) are \( G \times G \) group matrices defined by

\[ \mathbf{ID}(r) = \text{Diag}[D_1(r) \ldots D_g(r) \ldots D_G(r)] \]  

(1.4a)

\[ \mathbf{IM}(r) = \text{Diag}[\Sigma_1(r) \ldots \Sigma_g(r) \ldots \Sigma_G(r)] \]  

(1.4b)

\[ \mathbf{T}(r) = \begin{bmatrix}
0 & -\Sigma_{12}(r) & \ldots & -\Sigma_{1G}(r) \\
-\Sigma_{21}(r) & 0 & \ldots & -\Sigma_{2G}(r) \\
\vdots & \vdots & \ddots & \vdots \\
-\Sigma_{G1}(r) & -\Sigma_{G2}(r) & \ldots & 0
\end{bmatrix} \]  

(1.4c)

\[ \mathbf{IB}(r) = \begin{bmatrix}
\chi_1 \\
\vdots \\
\vdots \\
\chi_G
\end{bmatrix} \begin{bmatrix}
\nu \Sigma_{f1}(r) & \ldots & \nu \Sigma_{fG}(r)
\end{bmatrix} \]  

(1.4d)

and \( \phi(r) \) is the group flux vector

\[ \phi(r) = \text{Col} [\phi_1(r) \ldots \phi_G(r)] \]  

(1.4e)

In problems where no upscattering is present, \( \Sigma_{gg'}(r) = 0 \) for \( g < g' \), and \( \mathbf{T} \) becomes \( G \times G \) lower triangular.

It is also convenient to define the group current vector \( \mathbf{j}(r) \)

\[ \mathbf{j}(r) = \text{Col} [j_1(r) \ldots j_G(r)] \]  

(1.4f)

and the \( G \times G \) group absorption, scattering and production matrix \( \mathbf{A}(r) \)

\[ \mathbf{A}(r) = \mathbf{IM}(r) - \mathbf{T}(r) - \frac{1}{\lambda} \mathbf{IB}(r) \]  

(1.4g)
Equations 1.1 and 1.2 may then be written simply as

\[ J(r) + D(r) \nabla \Phi(r) = 0 \] (1.5a)

\[ \nabla \cdot J(r) + \Lambda(r) \Phi(r) = 0 \] (1.5b)

and

\[ -\nabla \cdot D(r) \nabla \Phi(r) + \Lambda(r) \Phi(r) = 0 \] (1.6)

respectively. These forms of the group diffusion equations will be used throughout this report. The boundary conditions on \( \Phi(r) \) are of the homogeneous Neumann or Dirichlet type, while the normal component of the current \( J(r) \) is required to be continuous across all internal interfaces.

1.3 Solution Methods

All of the solution methods which can be employed in order to obtain approximate solutions to the time-dependent, multigroup diffusion equations may be conveniently classified as belonging in the area of either nodal analysis or modal analysis, or a combination of the two: modal-nodal analysis. The principal concept in each of these analyses is that the neutron flux, a continuous function of many variables, may be approximated as a set of unknown coefficients and/or functions of possibly fewer variables. The ultimate goals of such approximation methods are to produce easily solvable coupled equations which relate the unknowns to each approximation and yield results of acceptable accuracy at a low cost. Various commonly used methods and their drawbacks are discussed below.
1.3.1 Nodal Methods

Nodal methods involve the local approximation of an average flux at points called nodes, where each node represents a distinct region within the reactor in which the average flux is defined. An ordered set of nodes connected by a grid of mesh lines is then used to approximate the spatial flux behavior. The accuracy of such methods is generally governed by the internodal coupling or neutron current approximation inherent in each method.

A. Conventional Finite Difference Equations

The common finite difference equations used in diffusion theory can be derived using Taylor series expansion, variational techniques, or box integration methods about each spatial node. The second-order diffusion term at each node is replaced by three-point difference equations relating consecutive nodes in each spatial direction. The resulting band-structured matrix equations exhibit many advantageous mathematical properties and can be solved with the use of simple solution algorithms.

The attractiveness of these difference equations is further enhanced by the fact that, for properly posed problems (including proper boundary conditions), the approximation can be shown to converge to the solution of the differential equation as the mesh size approaches zero. Also, the accuracy of the approximation can be shown to be in general of order $\theta(h)$, thus error estimates for the approximation are available. It is for these reasons that these equations are frequently invoked as "exact" solutions to diffusion equation problems. The main disadvantage, however, is that, as the
number of nodes increases, the amount of labor and cost involved in order to obtain an accurate solution increases geometrically. A point of diminishing returns is then quickly reached where further accuracy is prohibitively expensive. Another disadvantage is that any known physical insight or \textit{a priori} detailed flux behavior cannot be used with this approximation.

A formal derivation of the conventional difference equations is given in section 2.3 of Chapter 2.

B. Gross Coupling Models

In gross coupling or coarse mesh nodal techniques an attempt is made to decrease drastically the number of nodes needed for solution without significantly decreasing solution accuracy. Many such methods have been proposed by postulating various forms of neutronic coupling or communication interaction between nodes.

1. Phenomenological Model$^{9,10,11}$

From a physical viewpoint, the reactor can be divided into several distinct regions, each represented by a node located somewhere in that region. Equations of balance relating state variables of interest (average neutron flux, regional power, etc.) can then be written for each region and between region nodes. Internodal coupling is governed by a set of coefficients, say $p_{ij}$, which may account for the number of neutrons born in region $i$ which appear in region $j$. A set of algebraic equations can then be written which describe the coupled core dynamics of the nodal interactions.
The principal drawback of such methods lies in the definition of the interaction parameters $p_{ij}$. Although the describing equations of the phenomenological model can be directly formulated from diffusion theory, the method of calculating the coefficients $p_{ij}$ remains unclear. However, the physical simplicity of this model has made it very appealing in coupled kinetics methods development. Much of the work in this field is based on deriving approximations which reduce to this simple conceptual model.

2. Effective D/L Coupling

These methods are very similar to finite difference approximations in that the structural forms of the resulting difference equations are identical. In order to compensate for the use of large internodal mesh spacing, the reactor constants, and the diffusion coefficients in particular, may be altered so that they correspond in an average sense to those obtained from a fine mesh calculation. In this way it is hoped that the gross internodal coupling will be sufficiently improved to compensate for the large mesh spacings.

It has been shown that such methods can indeed improve internodal coupling for large mesh regions; however, the results are generally not satisfactory since the coupling constants are dependent in an unpredictable way on changes in the properties of the nodes.

3. Fission Source Coupling

The assumption that the reactor flux can be separated into partial region fluxes due to nodal fission sources permits a consistent derivation of nodal coupled kinetics equations from multigroup
diffusion theory. Fission modes can be found from detailed flux solutions which are then used to account for internodal coupling. This method gives reasonably accurate results for fast and thermal reactor transients, although the number of nodes necessary to achieve an accurate solution must increase as the form of the spatial flux becomes more detailed.

4. Multichannel Coupling

By partitioning the reactor into regions called channels and allowing only adjacent channel-to-channel interactions, coupling coefficients $p_{ij}$ can be found which represent the net leakage of neutrons from channel $i$ into channel $j$ in terms of the corresponding averaged channel fluxes. The coupling coefficients can be calculated using diffusion theory or variational techniques which yield the diffusion equations as stationary conditions. This model is appealing in that it can be shown to reduce to the conventional difference equations when a regular grid of small channel regions is used.

The above examples of gross coupling models are generally unsatisfactory because they require the use of average fluxes defined within large regions of the reactor. More acceptable results are obtained by utilizing known or \textit{a priori} detailed spatial flux shapes in the regions in the approximation method.

1.3.2 Modal Methods

Modal methods imply an extensive rather than local approximation to the spatial neutron flux. In general the flux is represented by a combination of known functions defined over the regions of interest.
with unknown functions as mixing coefficients. Depending upon the approximation employed, relationships among these coefficients can be derived which are hopefully simpler to solve than the original equation.

A. **Helmholtz Modes**

The diffusion equation for a completely homogeneous reactor formally has an infinite solution set of eigenvalues and corresponding orthogonal eigenfunctions, called Helmholtz modes, which satisfy the homogeneous boundary conditions. For the general case of a heterogeneous reactor, the spatially dependent flux can be approximated as a linear combination of these modes. The major difficulty with this approach is that a large number of modes is required in order to approximate the solution flux, and thus the appeal for this simplistic modal approach is quickly lost.

B. **Lambda** and **Omega Modes**

Although included in the class of modal approximations, these methods require the use of known spatial solutions for time-dependent analysis. Lambda modes belong to the set of detailed flux solutions of the time-independent diffusion equations which correspond to different lambda eigenvalues.

A set of detailed flux solutions can also be found from the time-dependent diffusion equations by allowing the time-dependent flux to be separable and given in the form $e^{\omega t}$. The solutions of the resulting equations, called $\omega$ modes, correspond to different omega eigenvalues.
Both of these methods have successfully been used in the transient analysis of coupled nodal kinetics.

C. Synthesis Methods\textsuperscript{5,21}

The use of synthesis techniques for the derivation of modal approximations is the most exciting and fastest growing area of reactor analysis methods development. This can be attributed to the fact that all diffusion theory approximation schemes, both modal and nodal, and those including time dependence, can be ultimately derived from one single variational principle. Each approximation scheme is therefore dependent solely upon the form of the trial functions used to represent the flux, current, and weighting functions (or adjoint functions) in the synthesis procedure. The outstanding advantage of the synthesis method is that knowledge of \textit{a priori} detailed flux shapes or other physical insights can be incorporated directly into the approximation method.

1. Multichannel Synthesis\textsuperscript{22}

This method may be viewed as a modal extension of the multichannel gross coupling method. Assuming the flux to be separable in its variables \((x,y,z)\), the number of unknowns can be reduced by specifying detailed flux shapes in any dimension. A common example assumes that in each channel, \(k\), of the reactor the flux trial function, \(U_k(x,y,z)\), can be expressed as the product of a known transverse flux, \(\psi_k(x,y)\), with an unknown spatially dependent axial flux, \(\rho_k(z)\), as:

\[
U_k(x,y,z) = \rho_k(z)\psi_k(x,y)
\]  (1.7)
The specification of the flux in two dimensions reduces the problem to an approximation involving only one dimension. If, however, the flux is approximated by a full spatial solution times an unknown constant,

$$U_k(x, y, z) = F_k \psi_k(x, y, z)$$  \hspace{1cm} (1.8)

the method reduces to an approximation similar to the multichannel nodal method. Figure 1.1 illustrates the resulting flux shape characteristics of such a method for the one-dimensional case.

The major disadvantage of these multichannel synthesis methods lies in the fact that in general the flux is discontinuous at channel interfaces.\textsuperscript{23,24} Therefore the adjacent channel coupling currents, which then must be continuous across these interfaces,\textsuperscript{†} are defined in terms of averaged channel fluxes. Although these methods can produce detailed flux distributions in each channel, their accuracy appears to be not much better than nodal multichannel methods because of the averaged gross neutronic coupling requirements inherent in these methods.\textsuperscript{25}

2. Overlapping Multichannel Synthesis\textsuperscript{26,27}

The interchannel neutronic coupling can be improved by requiring that the flux trial functions be continuous across channel interfaces. This can be accomplished by modulating the known expansion functions, \(\psi_k\), by piecewise continuous normalized polynomial functions, \(p_k\), which are nonzero only within coupled channels

\textsuperscript{†}Variational techniques used with diffusion theory in general do not allow the flux and current to be simultaneously discontinuous. Further clarification is given in section 2.2 of Chapter 2.
Figure 1.1. Illustration of One-Dimensional Multichannel Synthesis

Figure 1.2. Illustration of One-Dimensional Overlapping Multichannel Synthesis
of interest, providing the expansion functions are continuous over all channels for which the corresponding polynomial functions are nonzero. Such polynomials are required to be normalized to unity at the coupling interface and zero along the external boundary of the channels in order to preserve flux trial function continuity.

In one dimension represented by the continuous variable \( z \) and \( K \) mesh regions bounded by the nodes \( z_k \) where \( k = 1 \) to \( K + 1 \), for example, the simple linear functions

\[
p_k(z) = \begin{cases} 
\frac{z - z_{k-1}}{z_k - z_{k-1}} & z_{k-1} \leq z \leq z_k \\
\frac{z_{k+1} - z}{z_{k+1} - z_k} & z_k \leq z \leq z_{k+1} \\
0 & \text{otherwise}
\end{cases}
\] (1.9)

satisfy these conditions. The flux can then be approximated as

\[
U(z) = \sum_{k=1}^{K} F_k p_k(z) \psi_k(z)
\] (1.10)

where the set of \( F_k \)'s are the unknowns of the method. The resulting flux shape characteristics of this approximation are illustrated in Figure 1.2.

Approximations based on this synthesis method are dependent upon the class of overlapping polynomial functions used as well as the form of the current trial functions employed. The form of the current is extremely important in that it specifies the coupling interaction between regions and in this sense governs the usefulness and accuracy
of the approximation. Work performed with this method to date has used current trial functions of a form similar to those of the flux trial function. Although the results of these investigations have been encouraging, such methods do not reduce to more simple known approximation methods. In addition, the band-structured matrix equations which arise from the use of such methods do not exhibit mathematical properties desired of such approximation schemes and may be difficult and costly to solve.

1.3.3 Modal-Nodal Methods

Approximation methods have also been developed in which the flux has a known extensive definition, or shape, and the unknowns are local flux values averaged in accordance with their corresponding extensive definition. Such modal-nodal methods retain all of the advantages of modal methods while generally reducing the number of unknowns and producing matrix equations which have desirable mathematical properties for numerical approximation and solution.

The finite element method is the best example of a modal-nodal approximation. Greater accuracy than that of conventional difference techniques can be obtained by allowing the flux in each region of interest to be represented as a polynomial which is continuous at region interfaces. The forms of the flux approximations and the resulting difference equations which arise from the use of the finite element method are described in detail in section 2.3 of Chapter 2.
The purpose of this report is to present an original and consistent class of modal-nodal coarse mesh approximation methods which retain given or known detailed flux structure within the regions of interest, while providing detailed neutronic coupling between adjacent regions. These methods are consistent in that they are derived from a general variational principle and are a systematic extension of the finite element method as applied to diffusion theory reactor analysis.

For purposes of simplicity, the methods will be developed for the case of one-dimensional, time-dependent, multigroup diffusion theory, although it is expected that these methods can be extended to the general spatially dependent kinetics problem with relative ease.

The remainder of this report is organized as follows. Chapter 2 summarizes the use of variational principles and synthesis techniques in time-independent diffusion theory. The difference equations of the finite element methods applied in one dimension are derived using modal-nodal trial function forms in order to illustrate the use of these techniques. The forms of the proposed approximation methods are given in Chapter 3. The resulting finite difference equations are presented and boundary conditions discussed for approximation methods involving both linear and cubic Hermite basis functions. The numerical properties of the resulting matrix equations, as well as their numerical solution scheme, and useful programming techniques are discussed in Chapter 4. Chapter 5 presents results of the proposed methods for four representative one-dimensional PWR configurations, and compares the results with those of coarse mesh finite element methods. Finally, Chapter 6 presents conclusions and recommendations as well as comments concerning the possibility of extending the proposed methods to multidimensional geometries.
Chapter 2

VARIATIONAL DERIVATION OF FINITE DIFFERENCE APPROXIMATIONS IN TIME-INDEPENDENT MULTIGROUP DIFFUSION THEORY

The application of variational calculus to the describing equations of physical systems is perhaps the most general and powerful method of obtaining approximate solutions in mathematical physics. Variational methods seek to combine known "trial functions" into approximate solutions through the use of a variational functional which characterizes the equations of the system.

Essentially, variational methods consist of first finding a characteristic functional whose first-order variation when set to zero yields the describing equations of the system as its Euler equations. A class of trial functions, given in terms of known functions and unknown coefficients (or functions), is then chosen to approximate the solutions of the describing equations. These trial functions are then substituted into the variational functional, and its first variation is set to zero. Allowing arbitrary variations in all of the trial function unknowns results in a set of relationships among the unknowns. These relationships when solved then yield the "best" obtainable approximate solution within the space of trial functions given.

Variational methods can be thought of as a class of weighted residual methods since "weighting functions" appear in the functional and in the equations that result from setting the first variation of the functional to zero. The weighting functions are determined by the
form of the functional itself; or equivalently, by the set of Euler equations selected to describe the system. In non-self adjoint problems, the adjoint equations are generally included in the set of Euler equations. The inclusion of corresponding "adjoint trial functions" in the functional results in adjoint weighting in the variation equations and allows greater approximation flexibility of the variational method.

2.1 Calculus of Variations Applied to Diffusion Theory

The time-independent multigroup diffusion equations as given by Eq. 1.3 can be written as

\[ H \phi = \frac{1}{\lambda} B \phi \]  

(2.1a)

where

\[ H = - \nabla \cdot D \nabla + M - T \]  

(2.1b)

Since the multigroup diffusion equations are not self-adjoint, it is convenient to introduce the adjoint diffusion equations

\[ H^* \phi^* = \frac{1}{\lambda} B^* \phi^* \]  

(2.2a)

where \( H^* \) and \( B^* \) are the adjoint operators corresponding to \( H \) and \( B \), respectively, and are defined as:3

\[ H^* = H^T = - \nabla \cdot D \nabla + M - T^T \]  

(2.2b)

\[ B^* = B^T \]  

(2.2c)

since \( D \) and \( M \) are diagonal. \( \phi^* \) is the group adjoint flux vector, or importance vector, which must obey the same boundary conditions as \( \phi \).28
The exact solutions \( \Phi(r) \) and \( \Phi^*(r) \) of the diffusion equations and the adjoint diffusion equations can be approximated by flux and adjoint flux trial functions denoted as \( U(r) \) and \( U^*(r) \) using a variational functional of the form

\[
\mathcal{F}_1 [U, U^*] = \frac{1}{\lambda} \frac{\int_u U^T H U \, dr}{\int_u U^T B U \, dr}
\]

where it is assumed that the group-theory flux trial function vectors \( U^* \) and \( U \) as well as the group current vectors \( D \gamma U^* \) and \( D \gamma U \) are everywhere continuous, and that \( U^* \) and \( U \) vanish outside the reactor region \( R \). Allowing arbitrary trial function variations, denoted by \( \delta U^* \) and \( \delta U \), making \( \mathcal{F}_1 \) stationary first with respect to \( U^* \) and then with respect to \( U \) results in the following equations:

\[
\int_R \delta U^* \left[ \frac{1}{\lambda} H U^* - I H \right] \, dr = 0 \quad (2.4a)
\]

\[
\int_R \left[ U^* \left( \frac{1}{\lambda} H U^* - I H \right) + I B \right] \delta U \, dr = 0 \quad (2.4b)
\]

The above equations, containing the desired Euler equations, are the equations upon which the approximation method is based.

A significant characteristic of this approximation form is the property of exact solution reproduction. Although general choices of the trial functions \( U \) and \( U^* \) result in approximate eigenvalues which may differ substantially from the exact solution eigenvalue, the exact solutions, when chosen within the given class of trial functions, are yielded as the result of the approximation along with the exact solution eigenvalue.
The nature of the above approximation depends solely upon the forms of the flux trial functions given. Each trial function can be defined in terms of unknown coefficients (or functions) and known functions. Independent variation of the unknown coefficients of the adjoint trial function in Eq. 2.4a will yield the "best" flux solution obtainable for that class of flux and adjoint flux trial functions given. The corresponding "best" adjoint flux solution can be found in an analogous manner using Eq. 2.4b. These techniques are illustrated in the next section.

Another functional incorporating the flux and adjoint flux diffusion equations can be defined as

\[ \mathcal{F}_2[U^*, U] = \int_R U^*^T [\rho H U - \frac{1}{\lambda} \rho B U] \, dr \] (2.5)

Although the forms of the above functionals differ, it can be shown that both produce the same variation equations, Eqs. 2.4, when made stationary. The form of \( \mathcal{F}_2 \) and its first variation are much less complex than the form and first variation of \( \mathcal{F}_1 \). For these reasons, functionals of the form of \( \mathcal{F}_2 \) will be used in this report.

2.2 Discontinuous Trial Functions

The addition of discontinuous flux trial functions into the class of allowable trial functions for use in diffusion theory variational methods greatly enhances and generalizes the versatility of such methods. However, special provisions must be made in the approximation method itself in order that such trial functions can be properly used. In order to account for the discontinuities in the flux (and in general also the current) trial functions, it is necessary to
include special terms specifying continuity conditions directly within
the approximation method. This can be accomplished through the use
of a variational functional whose Euler equations include the $P-1$
equations and continuity conditions for both flux and current. A
general functional of this type which allows discontinuous flux,
current, and adjoint trial functions can be derived from previous
work\textsuperscript{30,31} and is given as follows:

$$\mathcal{F}[\mathbf{U}^*, \mathbf{V}, \mathbf{U}, \mathbf{V}, \alpha, \beta] = \int_{\mathbb{R}} \left\{ \mathbf{U}^T \left[ \nabla \cdot \mathbf{V} + \mathbf{A} \mathbf{U} \right] + \mathbf{V}^T \left[ \nabla \mathbf{U} + \mathbf{D}^{-1} \mathbf{V} \right] \right\} d\mathbf{r}$$

$$+ \int_{\Gamma} \mathbf{\hat{n}} \cdot \left\{ [\mathbf{U}^T \alpha + \mathbf{U}^T (I-\alpha)] (\mathbf{V}^+ - \mathbf{V}^-) \right\} d\mathbf{s}$$

$$+ \left[ \mathbf{V}^T \beta + \mathbf{V}^T (I-\beta) \right] (\mathbf{U}^+ - \mathbf{U}^-) \right\} d\mathbf{s}$$  \hspace{1cm} (2.6)$$

where $\mathbf{U}^*$, $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{V}$ are the group flux and group current approxi-
mations to $\Phi^*$, $\Phi$, $\mathbf{J}^*$, and $\mathbf{J}$, respectively, and where the first
integral extends over the volume $\mathbb{R}$ of the reactor and the second
extends over all interior surfaces $\Gamma$ upon which discontinuities are
defined. $\mathbf{\hat{n}}$ is the unit vector perpendicular to interior surfaces, and
quantities evaluated on sides of surfaces toward which $\mathbf{\hat{n}}$ is pointing
are denoted with the subscript $\left( + \right)$. Quantities evaluated on sides of
surfaces from which $\mathbf{\hat{n}}$ is pointing are denoted with the subscript $\left( - \right)$. $\alpha$ and $\beta$ are in general $G \times G$ undefined variable matrices, and $I$ is in
general a $G \times G$ unit matrix, which allow a general treatment of the
discontinuities.

The restrictions generally imposed upon trial functions for use
in functionals of this type are the following:
1. The trial functions must be piecewise continuous.

2. The trial functions \( U \) and \( V^* \) as well as \( U^* \) and \( V \) are not allowed to be discontinuous at the same point.

3. The components of \( U^T V^* \) and \( U^* T V \) normal to the exterior surface of the reactor must vanish.

Due to restriction 2, the general quantities \( \alpha \) and \( \beta \) always cancel and are never used within these approximation methods.

The first variation of \( \mathcal{F} \) can be found in a straightforward manner, and can be simplified to the following form which indicates the desired \( P-1 \) and adjoint \( P-1 \) equations and the trial function continuity conditions as Euler equations:

\[
\delta \mathcal{F} = \int_R \left\{ \delta U^* \left[ \nabla \cdot (V + \Lambda U) \right] + \delta V^* \left[ \nabla U + D^{-1} V \right] 
+ \left[ -\nabla U^* + \nabla^* D^{-1} \right] \cdot \delta V + \left[ -\nabla \cdot (V^* + U^* \Lambda) \right] \delta U \right\} \, dr 
+ \int_\Gamma \hat{n} \cdot \left\{ \delta U^* \left( V_+ - V_- \right) + \delta V^* \left( U_+ - U_- \right) 
+ (U_+ - U_-)^* \delta V + (V_+ - V_-)^* \delta U \right\} \, ds \quad (2.7)
\]

In most applications, only approximations to the flux and current solutions are desired. In such instances variations in only the adjoint trial functions need be taken. Setting the first variation of \( \mathcal{F} \) equal to zero under these conditions and imposing the above trial function restrictions results in the following variation equation for flux and current approximations:
\[
\int_R \{ \delta U^* \text{T} [\nabla \cdot \underline{V} + \underline{A} U] + \delta \underline{V}^* \text{T} \cdot [\nabla U + \mathbb{D}^{-1} \underline{V}] \} \, dr \\
+ \int_\Gamma \hat{n} \cdot \{ \delta U^* \text{T} (\underline{V}_+ - \underline{V}_-) + \delta \underline{V}^* \text{T} (U_+ - U_-) \} \, ds = 0 
\] (2.8)

The above approximation can also be expressed independently of adjoint trial functions. If the adjoint trial functions are defined as

\[
\begin{align*}
U^* &= U \\
\underline{V}^* &= -\underline{V}
\end{align*}
\] (2.9a, b)

then Eq. 2.8 reduces to the Rayleigh-Ritz Galerkin method, a weighted residual method based upon flux weighting.

Regardless of the choice of weighting, the variation equations can be further simplified for those approximation methods which require the currents to obey explicitly Fick's laws:

\[
\begin{align*}
\underline{V} &= -\mathbb{D} \nabla U \\
\underline{V}^* &= +\mathbb{D} \nabla U^*
\end{align*}
\] (2.10a, b)

Under these conditions the variation equations for discontinuous flux and discontinuous current trial functions reduce to

\[
\int_R \{ \delta U^* \text{T} \underline{A} U - \delta \underline{V}^* \text{T} \cdot \mathbb{D}^{-1} \underline{V} \} \, dr \\
+ \int_\Gamma \hat{n} \cdot \{ (\delta U_+^* - \delta U_-^*) \underline{V}_+ + \delta \underline{V}_-^* \text{T} (U_+ - U_-) \} \, ds = 0 
\] (2.11)

If in addition the flux is required to be everywhere continuous, the variation equations reduce to the appealing forms

\[
\int_R \{ \delta U^* \text{T} \underline{A} U - \delta \underline{V}^* \text{T} \cdot \mathbb{D}^{-1} \underline{V} \} \, dr = 0 
\] (2.12a)
or equivalently

$$\int_{\mathbb{R}} \left\{ \delta U^T \Lambda U + (\nabla \delta U^T) \cdot D(\nabla U) \right\} \, dr = 0 \quad (2.12b)$$

Variation equations 2.11 and 2.12 are the approximation equations which are used with the finite element methods and the proposed approximation methods.

2.3 The Finite Element Approximation Methods$^{32,33}$

This section introduces the notation and techniques used in conjunction with the modal-nodal variational analysis of the finite element method approximations in one-dimensional multigroup diffusion theory. These fundamentals are presented in these simple approximations before applying them to the more general proposed approximation method in the next chapter.

The one-dimensional problem is defined by the continuous variable $z$ and divided into $K$ adjoining regions which are in general inhomogeneous. Each region $k$ is bounded by nodes $z_k$ and $z_{k+1}$ and has width $h_k = z_{k+1} - z_k$. It is convenient to define the dimensionless variable $x$ within each region $k$ as

$$x = \frac{z - z_k}{h_k} \quad (2.13a)$$

so that region $k$ can be described in terms of $z$ as

$$z_k \leq z \leq z_k + h_k = z_{k+1} \quad (2.13b)$$

or equivalently in terms of $x$ as

$$0 \leq x \leq 1 \quad (2.13c)$$

for each of the regions $k, k=1$ to $K$. This notation will be used throughout this report.
2.3.1 The Conventional Finite Difference Equations

The conventional nodal flux-averaged, three-point, finite difference equations of one-dimensional diffusion theory can be derived from Eq. 2.8 using discontinuous flux and current multigroup column vector trial functions of the following form:7,8†

\[
U(z) = F_k \begin{cases} 
    z_{k-1} + \frac{1}{2} h_{k-1} & \text{if } k = 1, \\
    z & \text{if } k \in \{2, \ldots, K\}, \\
    z_{K+1} & \text{if } k = K+1
\end{cases}
\]

\[
U^*(z) = F_k^* \begin{cases} 
    z_{k-1} + \frac{1}{2} h_{k-1} & \text{if } k = 1, \\
    0 & \text{otherwise}
\end{cases}
\]

\[
V(z) = G_k \begin{cases} 
    z_{k} & \text{if } k \in \{1, \ldots, K\}, \\
    0 & \text{otherwise}
\end{cases}
\]

\[
V^*(z) = G_k^* \begin{cases} 
    z_{k} & \text{if } k \in \{1, \ldots, K\}, \\
    0 & \text{otherwise}
\end{cases}
\]

The forms of these trial functions are illustrated in Figure 2.1.

Inserting these trial functions into variation equation 2.8 results in the equation

\[
\delta F^*_1 \left\{ \int_{0}^{\frac{1}{2}} \Lambda_1 F^*_1 h dx + G_1 - G_0 \right\} \\
+ \sum_{k=2}^{K} \delta F^*_k \left\{ \int_{\frac{1}{2}}^{1} \Lambda_{k-1} F^*_k h_{k-1} dx + \int_{0}^{\frac{1}{2}} \Lambda_k F^*_k h dx + G_k - G_{k-1} \right\} \\
+ \delta F^*_{K+1} \left\{ \int_{\frac{1}{2}}^{1} \Lambda_{K} F^*_{K+1} h dx + G_{K+1} - G_K \right\} \\
+ \sum_{k=1}^{K} \delta G^*_k \left\{ \int_{0}^{1} \Pi_{k}^{-1} G k h dx + F_{k+1} - F_k \right\} = 0 
\] (2.15)

†Shifting the domain of definition of the trial functions results in other approximation schemes with equivalently averaged nuclear constants.34
Figure 2.1. Conventional Nodal Finite Difference Approximation Trial Function Forms
Independent variation of all $F_k^*$ and $G_k^*$ then results in a system of 2K+1 equations and 2K+3 unknowns (including $G_0$ and $G_{K+1}$). The choice of boundary conditions supplies the missing equations. Zero flux boundary conditions can be imposed by setting $F_1 = F_{K+1} = 0$, which also requires $\delta F_1^* = \delta F_{K+1}^* = 0$ thereby eliminating $G_0$ and $G_{K+1}$, and results in a system of 2K-1 equations and 2K-1 unknowns. Symmetry boundary conditions can be imposed on the left by $G_0 = -G_1$ and on the right by $G_{K+1} = -G_K$, resulting in a system of 2K+1 equations and 2K+1 unknowns.

Elimination of all $G_k$, $k=1$ to $K$, results in the standard three-point difference equations

\[ b_1 F_1 + c_1 F_2 = 0 \]  
\[ a_k F_{k-1} + b_k F_k + c_k F_{k+1} = 0; \quad k = 2 \text{ to } K \]
\[ a_{K+1} F_K + b_{K+1} F_{K+1} = 0 \]

where Eqs. 2.16a and 2.16c are used for the cases of symmetry boundary conditions. The $G \times G$ matrix coefficients \{a_k, b_k, c_k\} are of the form $A - \frac{1}{\lambda} B$ and are defined assuming homogeneous regional nuclear constants in section 1 of Appendix B. The matrix form of Eqs. 2.16 for the use of zero flux boundary conditions on the left and symmetry on the right is illustrated in Figure 2.2.

2.3.2 Multichannel Polynomial Synthesis

The one-dimensional neutron flux $\phi_k(z)$ defined as nonzero only within each region $k$ for each region ($k=1$ to $K$) can be approximated within each region as a polynomial of order $N$ by the power series
Figure 2.2. Matrix Form of the Conventional Finite Difference Equations. Boundary conditions chosen are zero flux on the left and symmetry on the right.

\[ a_k F_{k-1} + b_k F_k + c_k F_{k+1} = 0; \ k = 2 \text{ to } K. \]

\[ a_{K+1} F_K + b_{K+1} F_{K+1} = 0. \]

where: \( b_k = \beta_k - \frac{1}{\lambda} \epsilon_k; \ k = 2 \text{ to } K+1. \)
where the distinction between \( z \) and \( x \) is understood since \( 0 \leq x \leq 1 \) within each region \( k \). Such approximations are not useful in diffusion theory because: (1) the resulting matrix equations relating the \( a_{k,i} \)'s contain full matrices similar to Hibern matrices which may be very difficult to solve; and (2) such matrices are almost always highly singular and may produce numerical instabilities in the solution method. These difficulties can be eliminated by employing polynomials in the trial functions in the following form:

\[
U_k^{(N)}(z) = \sum_{i=0}^{N} a_{k,i} x^i 
\]

(2.17)

where the \( p_i^{(N)}(x) \) are polynomials in \( x \) of degree \( N \). This form is convenient because for a particular selection of the \( p_i^{(N)}(x) \) the unknowns \( F_{k+\frac{i}{N}} \) can be defined as the approximate flux solution evaluated at points \( z_i + \frac{i}{N} \) within region \( k \). For high order approximations, \( i > 0 \), the flux can be made continuous by imposing the following restrictions on \( p_i^{(N)}(x) \):

\[
p_i^{(N)}(\frac{\ell}{N}) = \begin{cases} 1 & \ell = i \\ 0 & \ell \neq i \end{cases} \text{ for } \ell = 0 \text{ to } N
\]

(2.19)

The specific polynomial flux approximations of this form through degree \( N=3 \) are given below:

\[
U_k^{(0)}(x) = F_k
\]

(2.20a)

\[
U_k^{(1)}(x) = (1-x)F_k + x F_{k+1}
\]

(2.20b)
\[ U_k^{(2)}(x) = (1-3x+2x^2)F_k + (4x-4x^2)F_{k+\frac{1}{2}} + (-x+2x^2)F_{k+1} \]  \hspace{1cm} (2.20c)

\[ U_k^{(3)}(x) = \left( 1 - \frac{11}{2} x + 9x^2 - \frac{9}{2} x^3 \right) F_k + \left( 9x - \frac{45}{2} x^2 + \frac{27}{2} x^3 \right) F_{k+\frac{1}{3}} + \left( -\frac{9}{2} x + 18x^2 - \frac{27}{2} x^3 \right) F_{k+\frac{2}{3}} + \left( x - \frac{9}{2} x^2 + \frac{9}{2} x^3 \right) F_{k+1} \]  \hspace{1cm} (2.20d)

An immediate drawback of these approximations lies in the definitions of the corresponding current trial functions. Given a flux polynomial approximation of degree \( N \), polynomial approximations for the current can be of order zero through \( N \), and may even be of higher order than the flux approximation. Each set of chosen trial function pairs ultimately results in a characteristic complex band-structured matrix problem which may or may not have desirable numerical solution properties and is usually very difficult to solve.

Such problems can be eliminated by noting that the use of variational analysis attempts to force the current approximation to obey Fick's law. The obvious solution is direct use of Fick's law in the trial function forms

\[ V_k(z) = -D_k(z) \frac{d}{dz} U_k(z) \]  \hspace{1cm} (2.21)

which results in simple band-structured matrix equations relating only flux unknowns. The use of current polynomial approximations of order \( N-1 \) as given in Eqs. 2.20 with flux approximations of order \( N \), however, does not improve the situation.

The accuracy of these difference equations can be found first by eliminating all non-integer subscripted unknowns, then expanding the resulting three-point difference equations in a Taylor series about
node \( k \), and comparing results to the exact three-point difference solution known for the one-dimensional case.\(^7,35\) By comparison of terms containing equal powers of \( h_k \), it can be shown that the \( N=1 \) and \( N=2 \) polynomial approximations are accurate to order \( \theta(h^2) \) while the \( N=3 \) approximation is accurate to order \( \theta(h^3) \).

The approximation of a function by a polynomial of order \( N \) leads immediately to the concept of basis functions. The \( N+1 \) polynomial functions which multiply the \( N+1 \) unknowns in Eq. 2.18 form a basis for the approximation and can be called basis functions. The simplicity of basis functions becomes apparent in an error analysis of the approximation as follows. An approximate solution \( U^{(N)}(z) \) of order \( N \) to the exact one-dimensional solution \( \phi(z) \) can be expressed as

\[
U^{(N)}(z) = \sum_{k=1}^{K} \phi(z_k) \Omega_k^{(N)}(z) \tag{2.22a}
\]

where \( \Omega_k^{(N)}(z) \) is a basis function of order \( N \) centered about node \( z_k \). By Taylor series expansion about any node, it can be shown\(^36\) that if \( \Omega_k(z) \) satisfies:

\[
\sum_{k=1}^{K} z_k^\alpha \Omega_k^{(N)}(z) = \left( \frac{z}{h_k} \right)^\alpha \text{ for } |\alpha| \leq N \tag{2.22b}
\]

then \( U^{(N)}(z) \) is an approximation to \( \phi(z) \) accurate to order \( \theta(h_k^{N+1}) \).

Basis functions found using Eqs. 2.22 are unique for each \( N \) and generally extend over surrounding regions. The forms of the basis functions for \( N \leq 3 \) are summarized below and illustrated in Figure 2.3. Since the following basis functions are symmetric, only the right half, \( z \geq z_k \), is expressly given.
\[ N = 0: \quad \Omega_k^{(0)}(z) = \begin{cases} 1 & z_k \leq z < z_k + \frac{1}{2} h_k \\ 0 & \text{otherwise} \end{cases} \]  \hspace{1cm} (2.23a)

\[ N = 1: \quad \Omega_k^{(1)}(z) = \begin{cases} (1-x) & z_k \leq z \leq z_{k+1} \\ 0 & \text{otherwise} \end{cases} \]  \hspace{1cm} (2.23b)

\[ N = 2: \quad \Omega_k^{(2)}(z) = \begin{cases} \frac{3}{4} - x^2 & z_k \leq z \leq z_k + \frac{1}{2} h_k \\ \frac{1}{2} \left( \frac{3}{2} - x \right)^2 & z_k + \frac{1}{2} h_k \leq z \leq z_{k+1} \\ \frac{1}{2} \left( \frac{1}{2} - x \right)^2 & z_{k+1} \leq z \leq z_{k+1} + \frac{1}{2} h_{k+1} \\ 0 & \text{otherwise} \end{cases} \]  \hspace{1cm} (2.23c)

\[ N = 3: \quad \Omega_k^{(3)}(z) = \begin{cases} \frac{1}{36} \left( 30 - 54 x^2 + 28 x^3 \right) & z_k \leq z \leq z_{k+1} \\ \frac{1}{36} \left( 4 - 24 x + 30 x^2 - 11 x^3 \right) & z_{k+1} \leq z \leq z_{k+2} \\ \frac{1}{36} \left( -1 + 3 x - 3 x^2 + x^3 \right) & z_{k+2} \leq z \leq z_{k+3} \\ 0 & \text{otherwise} \end{cases} \]  \hspace{1cm} (2.23d)

where \( 0 \leq x \leq 1 \) within each region \( k \) in the above cases.

Use of these basis functions results in approximate solutions which are continuous for \( N \geq 1 \) and whose derivatives \( dU^{(N)}(z)/dz \) through \( d^{N-1}U^{(N)}(z)/dz^{N-1} \) are also continuous. In high order approximations in diffusion theory, it is advantageous to retain flux and current continuity and employ basis functions defined over two adjacent regions in order to produce three-point difference equations. This can be accomplished in the \( N=3 \) approximations with the cubic Hermite basis functions.
Figure 2.3. Basis Functions of Eqs. 2.23 for \( N = 0, 1, 2, \) and 3
The above cubic basis function $\Omega_k^{(3)}(z)$ can be constructed from a combination of either cubic B splines, $\Omega_k^B(z)$, or cubic Hermite polynomials, $\Omega_k^H_1(z)$ and $\Omega_k^H_2(z)$, as follows:

$$
\Omega_k^{(3)}(z) = -\frac{1}{6} \Omega_k^B(z) + \frac{4}{3} \Omega_k^B(z) - \frac{1}{6} \Omega_k^B(z+1)
$$

(2.24)

where:

$$
\Omega_k^B(z) = \frac{2}{3} \Omega_k^H_1(z) + \frac{1}{6} \Omega_k^H_1(z) - \frac{1}{2} \Omega_k^H_2(z)
$$

(2.25)

The forms of these cubic B and Hermite polynomials are given below and illustrated in Figures 2.5 and 2.6. Again, only the right half of the functions are expressly given as $\Omega_k^B$ and $\Omega_k^H_1$ are symmetric, while $\Omega_k^H_2$ is antisymmetric.

$$
\Omega_k^B(z) = \begin{cases} 
\frac{2}{3} - x^2 + \frac{1}{2} x^3 & z_k \leq z \leq z_{k+1} \\
\frac{1}{6} (1-x)^3 & z_{k+1} \leq z \leq z_{k+2} \\
0 & \text{otherwise}
\end{cases}
$$

(2.26)

$$
\Omega_k^H_1(z) = \begin{cases} 
1 - 3x^2 + 2x^3 & z_k \leq z \leq z_{k+1} \\
0 & \text{otherwise}
\end{cases}
$$

(2.27a)

$$
\Omega_k^H_2(z) = \begin{cases} 
x - 2x^2 + x^3 & z_k \leq z \leq z_{k+1} \\
0 & \text{otherwise}
\end{cases}
$$

(2.27b)

where again $0 \leq x \leq 1$ in each region $k$.

The fact that the cubic Hermite polynomials form a basis for the cubic basis functions and extend over only two adjacent regions makes them very attractive for use in diffusion theory approximation methods.
Figure 2.4. Cubic B Spline $\Omega^B_k(z)$ of Eq. 2.26

Figure 2.5. Cubic Hermite Basis Functions
$H^1_k(z)$ and $H^2_k(z)$ of Eqs. 2.27
2.3.3 The Linear Basis Function Approximation

The group flux trial functions defined as nonzero within each region \( k \) can be expressed in modal-nodal form in terms of linear basis functions as

\[
U_k(z) = (1-x)F_k + xF_{k+1} \quad ; \quad k = 1 \text{ to } K.
\]

where \( F_k \) is the approximate group flux column vector at node \( z_k \) and \( 0 \leq x \leq 1 \) with each region \( k \). Although the flux trial functions are continuous, the current trial functions defined within each region by Eqs. 2.10 are not, and are given by

\[
V_k(z) = \frac{1}{h_k} \mathbf{D}_k(x) \left[ F_k - F_{k+1} \right] \quad ; \quad k = 1 \text{ to } K.
\]

Insertion of these trial function forms into variation equation 2.12a results in the equation

\[
\sum_{k=1}^{K} h_k \int_0^1 \left\{ [(1-x)\delta F_k^* + x\delta F_{k+1}^*]^T \mathbf{D}_k(x)(1-x)F_k + xF_{k+1} \right\} dx = 0
\]

Allowing arbitrary variations in all \( F_k^* \) results in a system of \( K+1 \) equations and \( K+1 \) unknowns which can be written as:

\[
b_1 F_1 + c_1 F_2 = 0 \quad (2.30a)
\]

\[
a_k F_{k-1} + b_k F_k + c_k F_{k+1} = 0 \quad ; \quad k = 2, K. \quad (2.30b)
\]

\[
a_{K+1} F_K + b_{K+1} F_{K+1} = 0 \quad (2.30c)
\]
where the $G \times G$ matrix coefficients $\{a_k, b_k, c_k\}$ are of the form $A - \frac{1}{A} B$ and are defined assuming homogeneous regional nuclear constants in section 2 of Appendix B. Zero flux boundary conditions can be imposed by use of only Eq. 2.30b with $F_1 = F_{K+1} = 0$, while symmetry boundary conditions require the use of the other equations as well. The matrix form of these equations for the boundary conditions of zero flux on the left and symmetry on the right is given in Figure 2.6.

2.3.4 The Cubic Hermite Basis Function Approximation$^{38,39}$

The cubic Hermite polynomials can be incorporated into modal-nodal flux trial functions which allow continuous flux and continuous current by defining the flux trial functions within each region $k$ as

$$U_k(z) = (1-3x^2+2x^3)F_k + (3x^2-2x^3)F_{k+1}$$
$$+ (-x+2x^2-x^3) \frac{\theta}{h_k} D_k^{-1}(x)G_k + (x^2-x^3) \frac{\theta}{h_k} D_k^{-1}(x)G_{k+1} \quad (2.31a)$$

$$U_k^*(z) = (1-3x^2+2x^3)F_k^* + (3x^2-2x^3)F_{k+1}^*$$
$$+ (-x+2x^2-x^3) \frac{\theta}{h_k} D_k^{-1}(x)G_k^* + (x^2-x^3) \frac{\theta}{h_k} D_k^{-1}(x)G_{k+1}^* \quad (2.31b)$$

where $k = 1$ to $K$.

$F_k$ is again the approximate group flux solution vector at node $z_k$, and $G_k$ is proportional to the approximate group current solution vector at node $z_k$. Application of Fick's law defines the current trial functions for each $k$ as
Figure 2.6. Matrix Form of the Linear Finite Element Method Approximation. Eqs. 2.35 for the case of zero flux on the left and symmetry boundary conditions on the right.

where:

\[ a_k = \alpha_k - \frac{1}{\lambda} \delta_k \]
\[ b_k = \beta_k - \frac{1}{\lambda} \epsilon_k \]
\[ c_k = \gamma_k - \frac{1}{\lambda} \xi_k \]

k = 2 to K+1.
\[ V_k(z) = \frac{1}{h_k} D_k(x)(6x-6x^2) [F_{k+1} - F_k] \]
\[ + (1-4x+3x^2)G_k + (-2x+3x^2)G_{k+1} \quad (2.31c) \]

\[ V_k^*(z) = \frac{1}{h_k} D_k(x)(6x-6x^2) [F_k^* - F_k^*] \]
\[ + (-1+4x-3x^2)G_k^* + (2x-3x^2)G_{k+1}^* \quad (2.31d) \]

Continuity of flux and current are automatically guaranteed since

\[ U_k(0) = U_{k-1}(h_{k-1}) = F_k \]
\[ U_k^*(0) = U_{k-1}^*(h_{k-1}) = F_k^* \quad (2.32) \]
\[ V_k(0) = V_{k-1}(h_{k-1}) = \theta G_k \]
\[ V_k^*(0) = V_{k-1}^*(h_{k-1}) = -\theta G_k^* \]

The normalization constant \( \theta \) is introduced in order to produce stiffness matrices having small condition numbers and can be chosen such that \( \frac{\theta}{D_k(0)} = 1 \).

Insertion of these trial function forms into variation equation 2.12a results in a lengthy equation which can be written as follows:

\[ \dagger \text{Such a choice of } -G_k^* \text{ allows the matrix of coefficients to be positive definite. Cf., Chapter 4.} \]
\[
\delta F^*_1 \begin{bmatrix} b1_{1F1} & b2_{1G1} & c1_{1F2} & c2_{1G2} \end{bmatrix} \\
\delta G^*_1 \begin{bmatrix} b3_{1F1} & b4_{1G1} & c3_{1F2} & c4_{1G2} \end{bmatrix} \\
+ \sum_{k=2}^{K} \delta F^*_k \begin{bmatrix} a1_{kF_{k-1}} & a2_{kG_{k-1}} & b1_{kF_k} & b2_{kG_k} & c1_{kF_{k+1}} & c2_{kG_{k+1}} \end{bmatrix} \\
+ \sum_{k=2}^{K} \delta G^*_k \begin{bmatrix} a3_{kF_{k-1}} & a4_{kG_{k-1}} & b3_{kF_k} & b4_{kG_k} & c3_{kF_{k+1}} & c4_{kG_{k+1}} \end{bmatrix}
\]
\[
\delta F^*_{K+1} \begin{bmatrix} a1_{K+1F_{K}} & a2_{K+1G_K} & b1_{K+1F_{K+1}} & b2_{K+1G_{K+1}} \end{bmatrix} \\
\delta G^*_{K+1} \begin{bmatrix} a3_{K+1F_{K}} & a4_{K+1G_K} & b3_{K+1F_{K+1}} & b4_{K+1G_{K+1}} \end{bmatrix} = 0.
\]

(2.33)

where the $G \times G$ matrix coefficients $\{a1, \ldots, c4\}$ are of the form $A - \frac{1}{\lambda} B$ and are defined assuming homogeneous regional nuclear constants in section 3 of Appendix B.

The choice of either zero flux, $F_k = 0$ as well as $\delta F^*_k = 0$, or zero current, $G_k = 0$ as well as $\delta G^*_k = 0$, boundary conditions for $k = 1$ or $K + 1$ along with arbitrary variations of the remaining $F^*_k$ and $G^*_k$ results in a system of $2K$ equations and $2K$ unknowns. Figure 2.7 illustrates the matrix form of such a system for the case of zero flux on the left and zero current on the right boundary conditions.

The basis functions and approximation techniques presented in this section are applied to the proposed approximation methods in the next chapter. Also, various techniques for treating zero flux and symmetry boundary conditions are discussed. The matrix properties of the equations resulting from the above finite element approximations and their solution methods are discussed in Chapter 4.
Figure 2.7. Matrix Form of the Cubic Hermite Finite Element Method Approximation. Eqs. 2.39 for the case of zero flux on the left and symmetry boundary conditions on the right.

where:  
\[
\begin{align*}
    a_n = \alpha_n - \frac{1}{\lambda} \delta n_k \\
    b_n = \beta_n - \frac{1}{\lambda} \epsilon n_k \\
    c_n = \gamma_n - \frac{1}{\lambda} \xi n_k
\end{align*}
\]

n = 1 to 4; and

k = 1 to K+1.
Chapter 3

DEVELOPMENT OF A CONSISTENT COARSE MESH APPROXIMATION METHOD

3.1 Formulation

The finite element methods have been shown\textsuperscript{32,33} to approximate accurately flux solutions and criticality measurements of multigroup diffusion theory when applied to problems allowing homogeneous nuclear material within the mesh regions. Use of such homogeneous material, while simplifying the calculation of the matrix elements (since numerical integrations are not required), may result in limiting the region mesh sizes allowed unless some type of homogenization procedure is used. If the mesh spacing is chosen such that some or all mesh regions are heterogeneous, then direct application of the variational techniques given in Chapter 2 results in weight averaging the nuclear constants with products of the basis functions and their derivatives, as given by the approximation. Although such a procedure is a direct application of the finite element technique, the accuracy of such methods depends upon the placement of the mesh regions and may vary significantly as their placement is altered.

A more useful homogenization procedure which is commonly used in reactor diffusion theory analysis allows the nuclear material within each mesh region to be homogenized by flux weighting with an assumed flux shape determined \textit{a priori} within that region in order (hopefully) to preserve reaction rates.
In large reactors the core can be thought of as composed of a lattice of heterogeneous fuel subassemblies containing fuel, clad, coolant channels, and/or absorption control rods. Each subassembly can be divided into several distinct homogeneous regions whose few-group microcell macroscopic nuclear constants are found by multi-group energy-dependent calculations. Detailed subassembly solutions, $\psi_k(r)$, are then found for each subassembly $k$ by assuming that the current on the boundary of the subassemblies is zero. Flux weighting the nuclear material in each subassembly with the corresponding detailed subassembly solution for each subassembly region then results in regional homogeneous nuclear constants $\langle \Sigma_k \rangle$ which may better approximate the physics of the region.

$$\langle \Sigma_k \rangle = \frac{\int_k \psi_k(r) \Sigma_k(r) \, dr}{\int_k \psi_k(r) \, dr} \quad (3.1)$$

Proper use of detailed flux weighted constants can lead to accurate criticality measurements, but the detailed a priori fine flux structure within each region is lost since it appears only in cross-section homogenization and not in the approximation. Attempts to retain the fine flux structure have only recently been proposed in several multichannel synthesis approximations. Unfortunately, each of these approximations are approximations in themselves and do not reduce to desirable approximations if the detailed flux solutions are themselves constant, as would be the case in large homogeneous regions.
Just as the discontinuous multichannel synthesis approximation method can be shown to reduce to low order difference equations (of the type which could result using the finite element method with constant or flat basis functions) when constant trial functions are used, approximation methods are presented below which retain the given detailed flux structure and also reduce to the higher order finite element approximations. The use of linear or cubic Hermite basis functions in the approximation provides flux continuity and results in better approximation accuracy.

The approximations are presented and discussed for the case of one-dimensional, multigroup diffusion theory. Extension to higher dimensions remains a problem that will require some further study. The approximations which are the linear basis functions are considered in the next section, while the approximations using the cubic Hermite basis functions are considered in section 3.3.

3.2 The Proposed Linear Basis Function Approximations

The proposed approximation method utilizing linear basis functions and defined as nonzero within each mesh region \( k, k=1 \) to \( K \), is given by the following modal-nodal trial function forms:

\[
U_k(z) = \psi_k(x) \left[ \psi_k^{-1}(0)(1-x)F_k + \psi_k^{-1}(1)x F_{k+1} \right] \quad (3.2a)
\]

\[
U_k^*(z) = \psi_k^*(x) \left[ \psi_k^*^{-1}(0)(1-x)F_k^* + \psi_k^*^{-1}(1)x F_{k+1}^* \right] \quad (3.2b)
\]

\[
V_k(z) = \eta_k(x) \left[ \psi_k^{-1}(0)(1-x)F_k + \psi_k^{-1}(1)x F_{k+1} \right]
+ \frac{1}{h_k} D_k(x) \psi_k(x) \left[ \psi_k^{-1}(0)F_k - \psi_k^{-1}(1)F_{k+1} \right] \quad (3.2c)
\]
\[ V_k^*(z) = \eta_k^*(x) [ \psi_k^{-1}(0)(1-x)F_k^* + \psi_k^{-1}(1)xF_{k+1}^* ] \]

\[ + \frac{1}{h_k} D_k(x) \psi_k^*(x) [ \psi_k^{-1}(0)F_k^* - \psi_k^{-1}(1)F_{k+1}^* ] \]

(3.2d)

where:

\[ x = (z - z_k) / h_k \]

(3.2e)

and \( 0 \leq x \leq 1 \), as \( z_k \leq z \leq z_{k+1} \), for each region \( k = 1 \) to \( K \).

\( F_k \) is the unknown approximate group flux column vector at node \( z_k \), and \( \psi_k^*, \eta_k^* \), and \( \eta_k \) are \( G \times G \) diagonal matrices composed of the detailed group flux \( \psi_{g_k}(z) \) and group current \( \eta_{g_k}(z) \) solutions, and their adjoints, defined as nonzero only within region \( k \). Because of the variable transformation between \( z \) and \( x \), \( \psi_k(0) \) represents \( \psi_k(z_k) \), and \( \psi_k(1) \) represents \( \psi_k(z_{k+1}) \); neither of which, for the moment, is allowed to be zero for any region. The detailed current solutions are given from the detailed flux solutions by Fick's law as

\[ \eta_k(z) = -D_k(z) \frac{d\psi_k(z)}{dz} \]  

(3.3a)

\[ \eta_k^*(z) = +D_k(z) \frac{d\psi_k^*(z)}{dz} \]  

(3.3b)

As a result, the current trial functions are related to the flux trial functions by analogous expressions.

Continuity of the flux is imposed by the form of the trial functions since

\[ U_k(0) = U_{k-1}(h_{k-1}) = F_k \]  

(3.4a)

\[ U_k^*(0) = U_{k-1}^*(h_{k-1}) = F_k^* \]  

(3.4b)

The current trial functions, however, are discontinuous. It is evident
by comparison to Eqs. 2.28, that this approximation reduces to the linear basis function finite element method if the detailed flux solutions for each group are taken to be constant.

Insertion of these trial function forms in Eq. 2.12a results in the following variation equation:

\[
\sum_{k=1}^{K} h_k \int_0^1 \left\{ \psi_k^*(x) \psi_k^* \mathbf{T}^{-1} \right. \\
\left. (0)(1-x) \delta F_k^* \mathbf{A}_k(x) U_k(x) \right. \\
\left. + \psi_k^* \mathbf{T}^{-1} \right. \\
\left. (1)x \delta F_{k+1}^* \mathbf{A}_k(x) U_k(x) \right. \\
\left. + \left[ \eta_k^* \mathbf{T}^{-1} \right. \\
\left. (0)(1-x) - \frac{1}{h_k} D_k(x) \psi_k^* \mathbf{T}^{-1} \right. \\
\left. (0) \right] \delta F_k^* \mathbf{D}_k^{-1}(x) V_k(x) \right. \\
\left. + \left[ \eta_k^* \mathbf{T}^{-1} \right. \\
\left. (1)x + \frac{1}{h_k} D_k(x) \psi_k^* \mathbf{T}^{-1} \right. \\
\left. (1) \right] \delta F_{k+1}^* \mathbf{D}_k^{-1}(x) V_k(x) \right\} dx = 0
\]

(3.5)

This equation can be written in the form

\[
\delta F_1^* \left[ b_1 F_1 + c_1 F_2 \right] \\
+ \sum_{k=2}^{K} \delta F_k^* \left[ a_k F_{k-1} + b_k F_k + c_k F_{k+1} \right] \\
+ \delta F_{K+1}^* \left[ a_{K+1} F_K + b_{K+1} F_{K+1} \right] = 0
\]

(3.6)

where the $G \times G$ matrix coefficients \{a_k, b_k, c_k\} are integral quantities of the form $A - \frac{1}{\lambda} B$ and are defined in detail in section 1 of Appendix C.

External zero flux boundary conditions are easily imposed by setting $F_1 = F_{K+1} = 0$. This requires that $F_1^*$ and $F_{K+1}^*$ must then also be zero, which in turn requires the $\delta F_1^*$ and $\delta F_{K+1}^*$ coefficients in
Eqs. 3.5 and 3.6 to vanish. Allowing independent variations in the remaining $F_k^*$, $k = 2$ to $K$, results in a matrix problem of the form illustrated in Figure 2.6 which would contain $K-1$ equations and $K-1$ unknowns.

Zero current boundary equations are found using symmetry considerations. If, for example, a zero current or symmetry boundary condition is imposed on the right at $z_{K+1}$, then a "boundary condition equation" can be derived by assuming a pseudo-region $k = K + 1$ of width $h_K$ having mirror image properties of region $K$ about $z_{K+1}$ with corresponding symmetric flux and antisymmetric current properties of the detailed flux and current solutions. These properties in pseudo-region $K+1$ can be related to properties of region $K$ as a function of $x$ in each region as

\begin{align*}
\mathbb{D}_{K+1}(x) &= \mathbb{D}_K(1-x) \quad (3.7a) \\
\mathbf{A}_{K+1}(x) &= \mathbf{A}_K(1-x) \quad (3.7b)
\end{align*}

and

\begin{align*}
U_{K+1}(x) &= U_K(1-x) \quad (3.8a) \\
U^*_{K+1}(x) &= U^*_K(1-x) \quad (3.8b) \\
V_{K+1}(x) &= -V_K(1-x) \quad (3.8c) \\
V^*_{K+1}(x) &= -V^*_K(1-x) \quad (3.8d)
\end{align*}

The addition of pseudo-region $K+1$ to the summation in Eq. 3.5 results in the calculation of coefficients $a_{K+1}$ and $b_{K+1}$ in Eq. 3.6. Detailed definitions of the $G \times G$ zero current coefficient matrices $b_1$, $c_1$, $a_{K+1}$, and $b_{K+1}$, all of which vanish for the case of zero flux boundary conditions, are also given in Appendix C.1. If symmetry is imposed
on both sides of the problem, independent variations in $F^*_k$ for $k = 1$ to $K+1$ result in a matrix problem of $K+1$ equations and $K+1$ unknowns of similar form as illustrated in Figure 2.6.

Other boundary conditions may be imposed on the approximation, including albedo and reflector boundary conditions, which specify the flux to current ratio at the boundaries. Such conditions will always lead to a variation equation of the form of Eq. 3.6, where in general the matrix coefficients $a_2, b_2, b_K, c_K$ as well as the boundary coefficients $b_1, c_1, a_{K+1}, b_{K+1}$ will have modified definitions.

A serious drawback of the approximation given by Eqs. 3.2 is that it does not allow the use of detailed flux solutions containing explicit zero flux boundary conditions. For this reason the exact solution, $\psi_k(z) = \Phi_k(z)$ for all $k$, is excluded from the class of admissible trial function forms. However, such detailed solutions can be allowed by modifying the trial function forms in the boundary regions. If a detailed solution $\psi_1(z)$ is given in the first region with the zero flux condition $\psi_1(a_1) = 0$, for example, the trial functions of Eqs. 3.2 could be modified for region $k=1$ as

$$U_1(z) = \psi_1(x)\psi_1^{-1}(1)F_2$$  \hspace{1cm} (3.9a)
$$U^*_1(z) = \psi^*_1(x)\psi_1^{-1}(1)F_2$$  \hspace{1cm} (3.9b)
$$V_1(z) = \eta_1(x)\psi_1^{-1}(1)F_2$$  \hspace{1cm} (3.9c)
$$V^*_1(z) = \eta^*_1(x)\psi_1^{-1}(1)F_2^*$$  \hspace{1cm} (3.9d)

In this way, the imposed zero flux boundary condition is explicitly given by $\psi_1(z)$ rather than in the form of the trial function. Similar
trial functions can be given for an explicit zero flux boundary condition in the last region, \( k=K \).

The use of these special trial functions in the boundary regions alters the definitions of the matrix coefficients \( b_2 \) and \( b_K \) as given in Eq. 3.6. Detailed definitions of these coefficients when these special trial functions are used are also included in Appendix C.

Regardless of the types of boundary conditions imposed, Eq. 3.6 results in an \( N \times N \) matrix problem of the form

\[
\mathbf{A}\mathbf{F} = \frac{1}{\lambda} \mathbf{IB}\mathbf{F} \quad (3.10)
\]

where \( \mathbf{A} \) and \( \mathbf{I} \) are independent of \( \lambda \). The order \( N \) of the matrix equations is dependent upon the chosen boundary conditions, and is given for various choices in Table 3.1.

<table>
<thead>
<tr>
<th>Boundary Condition Type</th>
<th>Matrix Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>on Left</td>
<td>on Right</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
3.3 The Proposed Cubic Hermite Basis Function Approximation

The proposed modal-nodal approximation method utilizing the cubic Hermite polynomials

\[ p_1(x) = 1 - 3x^2 + 2x^3 \]
\[ p_2(x) = 3x^2 - 2x^3 \]
\[ p_3(x) = -x + 2x^2 - x^3 \]
\[ p_4(x) = x^2 - x^3 \]  

and their negative derivatives

\[ q_1(x) = 6x - 6x^2 \]
\[ q_2(x) = -6x + 6x^2 \]
\[ q_3(x) = 1 - 4x + 3x^2 \]
\[ q_4(x) = -2x + 3x^2 \]  

and defined as nonzero only within each mesh region \( k \), is given by the below regional trial function forms. \( F_k \) and \( G_k \) are again the unknown group column approximate flux and current solutions at \( z_k \) respectively, and the remaining symbols have been previously defined. As in the cubic Hermite finite element method described in section 2.3.4 of Chapter 2, \( \theta \) is an optional normalization parameter.

\[ U_k(z) = \psi_k(x)[\psi_k^{-1}(0)p_1(x)F_k + \psi_k^{-1}(1)p_2(x)F_{k+1} + h_k \theta D_k^{-1}(0)\psi_k^{-1}(0)p_3(x)G_k + h_k \theta D_k^{-1}(1)\psi_k^{-1}(1)p_4(x)G_{k+1}] \]  

(3.13a)
Again, for the moment, $\psi_k^*(0)$ and $\psi_k^*(1)$ and their adjoints are not allowed to be zero in any region.

The forms of these trial functions impose both flux and current continuity since:

\begin{align*}
U_k^*(z_k) &= \psi_k^*(x)[\psi_k^{*^{-1}}(0) p_1(x) F_k^* + \psi_k^{*^{-1}}(1) p_2(x) F_k^{*}]
+ h_k \theta \mathbb{D}_k^{-1}(0) \psi_k^{*^{-1}}(0) p_3(x) G_k^* + h_k \theta \mathbb{D}_k^{-1}(1) \psi_k^{*^{-1}}(1) p_4(x) G_{k+1}^* ] \quad (3.13b) \\
V_k^*(z_k) &= \eta_k^*(x)[\psi_k^{*^{-1}}(0) p_1(x) F_k^* + \psi_k^{*^{-1}}(1) p_2(x) F_k^{*}]
+ h_k \theta \mathbb{D}_k^{-1}(0) \psi_k^{*^{-1}}(0) p_3(x) G_k^* + h_k \theta \mathbb{D}_k^{-1}(1) \psi_k^{*^{-1}}(1) p_4(x) G_{k+1}^* ] \quad (3.13c) \\
V_k^*(z_k) &= \eta_k^*(x)[\psi_k^{*^{-1}}(0) p_1(x) F_k^* + \psi_k^{*^{-1}}(1) p_2(x) F_k^{*}]
+ h_k \theta \mathbb{D}_k^{-1}(0) \psi_k^{*^{-1}}(0) p_3(x) G_k^* + h_k \theta \mathbb{D}_k^{-1}(1) \psi_k^{*^{-1}}(1) p_4(x) G_{k+1}^* ] \quad (3.13d)
\end{align*}

Again, for the moment, $\psi_k^*(0)$ and $\psi_k^*(1)$ and their adjoints are not allowed to be zero in any region.

The forms of these trial functions impose both flux and current continuity since:

\begin{align*}
U_k(z_k) &= U_{k-1}(z_{k-1} + h_{k-1}) = F_k \\
V_k^*(z_k) &= V_{k-1}(z_{k-1} + h_{k-1}) = F_k^* \\
V_k(z_k) &= V_{k-1}(z_{k-1} + h_{k-1}) = G_k \\
V_k^*(z_k) &= V_{k-1}(z_{k-1} + h_{k-1}) = -G_k^*
\end{align*}

(3.14a) (3.14b) (3.14c) (3.14d)
where it is assumed that at region mesh points the detailed current solutions are zero:

\[ \eta_k(z_k) = \eta_k(z_k + h_k) = 0 \quad (3.15a) \]

\[ \eta_k^*(z_k) = \eta_k^*(z_k + h_k) = 0 \quad (3.15b) \]

for all regions \( k = 1 \) to \( K \).

Also, by comparison to Eqs. 2.31, it is evident that this approximation reduces to the cubic Hermite finite element method when the \( \psi_k \)'s are constant, and the \( \eta_k \)'s are correspondingly zero.

The insertion of the above trial function forms into variation equation 2.12a results in a lengthy equation which can be simplified to the following form:

\[
\begin{align*}
\delta F_1^* & \begin{bmatrix} b_{11}F_1 + b_{21}G_1 + c_{11}F_2 + c_{21}G_2 \end{bmatrix} \\
+ \delta G_1^* & \begin{bmatrix} b_{31}F_1 + b_{41}G_1 + c_{31}F_2 + c_{41}G_2 \end{bmatrix} \\
+ \sum_{k=2}^{K} & \left\{ \begin{array}{l}
\delta F_k^* \begin{bmatrix} a_{1k}F_{k-1} + a_{2k}G_{k-1} + b_{1k}F_k + b_{2k}G_k + c_{1k}F_{k+1} + c_{2k}G_{k+1} \end{bmatrix} \\
+ \delta G_k^* \begin{bmatrix} a_{3k}F_{k-1} + a_{4k}G_{k-1} + b_{3k}F_k + b_{4k}G_k + c_{3k}F_{k+1} + c_{4k}G_{k+1} \end{bmatrix} \\
+ \delta F_{K+1}^* \begin{bmatrix} a_{1K+1}F_K + a_{2K+1}G_K + b_{1K+1}F_{K+1} + b_{2K+1}G_{K+1} \end{bmatrix} \\
+ \delta G_{K+1}^* \begin{bmatrix} a_{3K+1}F_K + a_{4K+1}G_K + b_{3K+1}F_{K+1} + b_{4K+1}G_{K+1} \end{bmatrix} = 0 
\end{array} \right\}
\end{align*}
\]

(3.16)

where the detailed definitions of the twelve integral \( G \times G \) matrix coefficients \( \{a_{1k}, \ldots, c_{4k}\} \) of the form \( A - \frac{1}{\lambda} B \) for all \( k \) are given in section 2 of Appendix C.
Boundary conditions for either zero flux or symmetry are easily imposed by setting either $F_k$ or $G_k$, respectively, to zero with $k=1$ for the conditions on the left at $z_1$ or $k=K+1$ for conditions on the right at $z_{K+1}$. The corresponding variations for $k=1$ and $k=K+1$ then vanish. Allowing arbitrary variations in the remaining $F_k^*$ and $G_k^*$ in Eq. 3.16 results in a system of $2K \times G$ matrix equations relating $2K \times G$ column vector unknowns, as illustrated in Figure 2.7.

Explicit zero flux boundary conditions imposed by $\psi_1(z_1) = 0$ or $\psi_K(z_{K+1}) = 0$ can be incorporated into the approximation by modifying the trial function definitions in the boundary regions. The modified flux trial functions in the first region, for example, are

\[
U_1(x) = \psi_1(x)\left[ \psi^{-1}_1(1)F_2 + h_1 \theta \mathcal{D}_1^{-1}(0)\psi^{-1}_1(0)p_3(x)G_1 + h_1 \theta \mathcal{D}_1^{-1}(1)\psi^{-1}_1(1)p_4(x)G_2 \right]
\]

\[
U_1^*(x) = \psi_1^*(x)\left[ \psi^{-1}_1^*(1)F_2^* + h_1 \theta \mathcal{D}_1^{-1}(0)\psi^{-1}_1^*(0)p_3(x)G_1^* + h_1 \theta \mathcal{D}_1^{-1}(1)\psi^{-1}_1^*(1)p_4(x)G_2^* \right]
\]

(3.17)

where the current trial functions are again given by Fick's laws. Use of modified trial function forms of this type in the boundary regions results in $2K$ equations with different definitions of $c_{31}$, $a_{22}$, $b_{12}$, $b_{22}$, and $b_{32}$ as well as $b_{1K}$, $b_{2K}$, $c_{2K}$, and $a_{3K+1}$, which are also included in Appendix C.

Other boundary condition restrictions may be imposed on this approximation, but the matrix form of the resulting difference equations will remain unchanged. Only the coefficients defined for $k = 1, 2, K, \text{ and } K+1$ will in general be altered.
The matrix equations resulting from this approximation can always be written as

\[ A \vec{F} = \frac{1}{\lambda} B \vec{F} \]  

(3.18)

where \( A \) and \( B \) are \((G \times 2K)\) by \((G \times 2K)\) matrices, independent of \( \lambda \), and \( \vec{F} \) is the \( G \times K \) column vector of unknowns containing both \( F_k \) and \( G_k \) column vectors for each \( k \). The matrix properties and solution methods of the matrix equations derived in these proposed approximation methods are discussed in the next chapter.
The matrix properties of the difference equations resulting from both the proposed approximations and the finite element methods in one dimension, as well as the solution schemes used to solve these equations, are summarized in the following section. Various calculational and programming techniques used in conjunction with these approximation methods and their solution schemes are presented and discussed in section 4.2.

4.1 Solution Methods and Matrix Properties

The matrix equations which result from the approximations given in this report are of the form

\[
A_F = \frac{1}{\lambda} B_F
\]  

(4.1)

and are solved using the fission source power iteration method without fission source renormalization.\(^7\),\(^44\) The method of solution is illustrated schematically in Figure 4.1. Other definitions of the iteration eigenvalue \(\lambda^{(i)}\) can be found elsewhere.\(^45\)

Figure 4.1 illustrates that an outer iteration solution scheme\(^46\) is used, and that the geometry and nuclear properties of the reactor are not altered. Since the fission source is not normalized by the iteration eigenvalue during the iterations, \(\lambda^{(i)}\) converges to the effective multiplication factor, \(k_{\text{eff}}\), of the problem. Had fission source renormalization been included, by \(\Sigma^{(i)} = B_F^{(i)}/\lambda^{(i-1)}\) for example, then \(\lambda^{(i)}\)
i = 0

Guess $F^{(1)} > 0$

$i = i + 1$  \hspace{1cm} \text{Outer Iteration}

$S^{(i)} = IB \, F^{(i)}$

$\tilde{F}^{(i+1)} = A^{-1} \, S^{(i)}$  \hspace{1cm} \text{Matrix Inversion}

$\lambda^{(i)} = \frac{(1, IB \, \tilde{F}^{(i+1)})}{(1, IB \, F^{(i)})}$

$e^{(i)} = \frac{1}{\lambda} \, \tilde{F}^{(i+1)} - F^{(i)}$

$F^{(i+1)} = F^{(i)} + \omega \, e^{(i)}$

$|e^{(i)}| < \text{Tol. } \epsilon_1$

$|\lambda^{(i)} - \lambda^{(i-1)}| < \text{Tol. } \epsilon_3$

Convergence

Figure 4.1. Solution of $A \, F = \frac{1}{\lambda} \, IB \, F$ Using the Fission Source Power Iteration Method Without Fission Source Renormalization.
would converge to unity. The $k_{\text{eff}}$ of the problem would then be simply the product of all of the iteration eigenvalues.\textsuperscript{46,47,48}

The matrix inversions required within the iteration scheme were performed directly. Although overrelaxation methods are usually employed only in iterative matrix inversion schemes (or inner iterations),\textsuperscript{49} an overrelaxation parameter $w$, $1 \leq w \leq 2$, is available in the outer iteration in order to hasten the convergence of the solution vector.

The power method is very appealing to neutron diffusion flux calculations because it converges to the largest or fundamental eigenvalue $|\lambda_0| > |\lambda_1|$, $i \neq 0$, and the corresponding eigenvector $\mathbf{F}_0$ of the given matrix problem. The convergence rate is governed by the dominance ratio, defined as $\max_{i \neq 0} |\lambda_i/\lambda_0|$, in such a way that smaller ratios result in faster convergence. Although the power method will always converge when $\lambda_0$ is positive and unique, specific matrix properties of $A$ and $B$ are sufficient but not always necessary to insure convergence to a positive $k_{\text{eff}}$ and everywhere positive neutron flux approximation.\textsuperscript{50}

In many problems the order of $A$ may be quite large, and solution methods which require the direct inversion of $A$ may not be practical. For the purposes of this report, as in most multigroup calculational schemes, neutron up-scattering will not be permitted. The inversion of $A$ is then performed by successive group-iteration techniques.

The equations given in Eq. 4.1 have been defined as ordered first by spatial indexing followed by group indexing within each spatial index. It is convenient to reorder these equations so that they are ordered first by group indexing followed by spatial indexing within each group.
After reordering, Eq. 4.1 can be written as

\[(IL + IM)F = TF + \frac{1}{\lambda} IBF\]  \hspace{1cm} (4.2a)

where

\[A = IL + IM - T\]  \hspace{1cm} (4.2b)

and: IL, the stiffness matrix, results from leakage; IM, the mass matrix, results from absorption; T is the group-to-group scattering transfer matrix; and IB is the fission source production matrix.

Assuming K spatial unknowns in each of the G groups, IL and IM are \(G \times K\) block diagonal matrices composed of \(G^2 K \times K\) matrices \(IL_g\) and \(IM_g\) of the form

\[IL = \text{Diag}[IL_1, \ldots, IL_G]\]  \hspace{1cm} (4.3a)

\[IM = \text{Diag}[IM_1, \ldots, IM_G]\]  \hspace{1cm} (4.3b)

and T and IB are in general full block matrices composed of \(G^2 K \times K\) matrices \(T_{gg'}\) and \(IB_{gg'}\), respectively. Since only downscattering is permitted, T becomes lower block triangular; \(T_{gg'} = 0\) whenever \(g' > g\). The matrix inversion, \(F^{(i+1)} = A^{-1} S^{(i)}\), can then be solved for the GK unknowns

\[F^{(i+1)} = \text{Col}[F_1^{(i+1)}, \ldots, F_G^{(i+1)}]\]  \hspace{1cm} (4.4)

by solving successively the following system of group equations:

\[
\begin{aligned}
\text{Do for } g = 1 \text{ to } G:\n\sum_{g' = 1}^{G} (T_{gg'} + IB_{gg'}) F^{(k)} &= S^{(i)}_g \\
\text{where: } k &= \begin{cases} 
    i + 1; & g' < g \\
    i; & g' \geq g
\end{cases} \\
F^{(i+1)} &= (IL + IM)^{-1} S^{(i)}_g
\end{aligned}
\]  \hspace{1cm} (4.5)
where the updating of the group fission source by the iteration index \( k = i + 1 \) for \( g' < g \) generally enables a faster rate of convergence of the outer iteration than \( k = i \).

The desirable convergence properties of a positive eigenvalue and everywhere positive flux solution when using the above group iteration method depend upon the properties of the \( K \times K \) spatial matrices for each group \( g \): \( I_L g \), \( M_g \), and \( T_{gg'} \) and \( B_{gg'} \), for \( g' = 1 \) to \( G \). Using the Perron-Frobenius theorem\(^{50}\), it can be shown that if \( T_{gg'} \) and \( B_{gg'} \) are all nonnegative for each group \( g \) and \( I_L g \) and \( M_g \) are both Stieltjes or S-type matrices\(^\dagger\) for each group \( g \), then the power method will converge to a positive eigenvalue, \( \lambda O > 0 \), and a corresponding positive eigenvector, \( F_O > 0 \). These matrix properties naturally depend upon the form of the spatial approximations employed and generally differ for different approximation schemes.

The conventional finite difference approximation has become popular because the spatial matrices which arise from its use exhibit these desirable properties regardless of the size of the mesh regions chosen. The spatial matrices resulting from the linear finite element method, however, are known to exhibit these properties only if the mesh size is restricted by

\[
h_k \leq \max_{g = 1 \text{ to } G} \left\{ \sqrt{6} \ell_{g,k} \right\}
\]

(4.6)

where \( \ell_{g,k} \) is the diffusion length, \( \ell_{g,k}^2 = D_{g,k}/\Sigma_{g,k} \), for group \( g \) in mesh region \( k \). The spatial matrices resulting from the cubic Hermite

\(^\dagger\)A Stieltjes matrix is a real, irreducible, positive definite matrix with nonpositive off-diagonal elements.\(^{50}\)
finite element method do not exhibit these "desirable" characteristics. Since the eigenvector \( F \) contains current as well as flux unknowns, convergence to an all-positive solution vector is not desirable.

The properties of the spatial matrices for each group \( g \) resulting from the proposed approximation methods can be found by generalizing the proposed trial function forms in each group as

\[
U_g(z) = \sum_{k=1}^{K} \left[ P_{g,k}(x) F - g,k + P_{g,k}^T F - g,k+1 \right]
\]

\[
U_g^*(z) = \sum_{k=1}^{K} \left[ P_{g,k}^*(x) F - g,k^* + P_{g,k}^{*T} F - g,k^*+1 \right]
\]

where the \( F_{g,k} \) are in general column vectors of length \( N \) given by:

\[
F_{g,k} = F_{g,k} (N = 1)
\]

for the linear basis function approximations, and

\[
F_{g,k} = \text{Col} [ F_{g,k}, G_{g,k} ] (N = 2)
\]

for the cubic Hermite basis function approximations. Similar definitions hold for the \( F_{g,k}^* \). The \( P_{g,k}^\pm(x) \) are column vectors of length \( N \) whose elements are functions of \( z \) (or \( x \)) defined as nonzero only within region \( k \) which provide the basis for the approximations. The definitions of the \( P_{g,k}^\pm(x) \) for the proposed approximations are given as follows:

\[
N = 1; \text{ Linear Basis Functions:}
\]

\[
P_{g,k}^-(x) = (1-x) \psi_{g,k}^{-1}(0) \psi_{g,k}(x)
\]

\[
P_{g,k}^+(x) = x \psi_{g,k}^{-1}(1) \psi_{g,k}(x)
\]
N = 2; Cubic Hermite Basis Functions:

\[ P_{g,k}^- = \text{Col}[p_1(x)\psi_{g,k}^{-1}(0)\psi_{g,k}(x), h_k\theta p_3(x)D_{g,k}^{-1}(0)\psi_{g,k}^{-1}(0)\psi_{g,k}(x)] \]

(4.10a)

\[ P_{g,k}^+ = \text{Col}[p_2(x)\psi_{g,k}^{-1}(1)\psi_{g,k}(x), h_k\theta p_4(x)D_{g,k}^{-1}(1)\psi_{g,k}^{-1}(1)\psi_{g,k}(x)] \]

(4.10b)

where \( \psi_{g,k}(x) \) and \( D_{g,k}(x) \) are the detailed flux solutions and diffusion coefficients of group \( g \), the polynomials \( p_1(x) \) through \( p_4(x) \) are defined in Eqs. 3.11, and \( 0 \leq x \leq 1 \) within each region \( k \). Similar definitions hold for the \( P_{g,k}^{\pm}(x) \).

Equations 4.7 can be written in matrix form as

\[ U_g(z) = IP_g(x)F_g \]

(4.11a)

\[ U_g^*(z) = IP_g^*(x)F_g^* \]

(4.11b)

where:

\[ F_g = \text{Col}(F_{g,1}, \ldots, F_{g,K+1}) \]

(4.12a)

and \( IP_g(x) \) is the K by N(K+1) matrix defined by

\[ IP_g(x) = \begin{bmatrix}
P_{g,1}^{-T}(x) & P_{g,1}^{+T}(x) & 0 \\
0 & P_{g,K}^{-T}(x) & P_{g,K}^{+T}(x)
\end{bmatrix} \]

(4.12b)

\( IP_g^*(x) \) is defined similarly. Insertion of these trial function forms into variation equation 2.12b for each group \( g \) results in
\[ \begin{align*}
\delta F^T_{\text{g}} & \int_{K} \dot{\mathbf{I}}^T_{\text{g}} \mathbf{D}_{\text{g}} \dot{\mathbf{I}}_{\text{g}} \mathbf{F}_{\text{g}} + \mathbf{P}^T_{\text{g}} \sum_{g'_{1}=1}^{G} \mathbf{A}_{\text{gg'}} \mathbf{P}_{\text{g'}} \mathbf{F}_{\text{g'}} \, dz = 0 
\end{align*} \]  
\hspace{1cm} (4.13)

where \( \mathbf{D}_{\text{g}} \) and \( \mathbf{A}_{\text{gg'}} \) are \( K \times K \) diagonal matrices of the form

\[ \mathbf{D}_{\text{g}} = \mathbf{D}_{\text{g}}(x) = \text{Diag}[D_{\text{g,1}}(x), \ldots, D_{\text{g,K}}(x)] \]  
\hspace{1cm} (4.14a)

and

\[ \mathbf{A}_{\text{gg'}} = \mathbf{A}_{\text{gg'}}(x) = \text{Diag}[A_{\text{gg',1}}(x), \ldots, A_{\text{gg',K}}(x)] \]  
\hspace{1cm} (4.14b)

The quantity \( \dot{\mathbf{I}}_{\text{g}} \) represents the derivative of \( \mathbf{I}^T_{\text{g}} \) with respect to \( z \), and the integration over \( K \) denotes integration over the entire range of \( z; \ z_1 \leq z \leq z_{K+1} \).

\( D_{\text{g,k}} \) and \( \mathbf{A}_{\text{gg',k}} \),

\[ \mathbf{A}_{\text{gg',k}} = \left( \Sigma_{\text{tg}} - \Sigma_{\text{gg'}} \right) - \frac{1}{\lambda} \chi_{\text{g}} \nu \Sigma_{fg'} \)  
\hspace{1cm} (4.14c)

are the group material constants in mesh region \( k \), and are usually dependent on \( x \). \( \mathbf{A}_{\text{gg'}} \) can thus be conveniently expanded as

\[ \mathbf{A}_{\text{gg'}} = \mathbf{A}^{A}_{\text{g}} - \mathbf{A}^{S}_{\text{gg'}} - \frac{1}{\lambda} \mathbf{A}^{F}_{\text{gg'}} \]  
\hspace{1cm} (4.14d)

Allowing arbitrary variations in each element of \( F^*_{\text{g}} \) for each group \( g \) in Eq. 4.13 results in the matrix equations

\[ \left( \mathbf{I}^T_{\text{g}} \mathbf{M} \right) \mathbf{F} = \sum_{g'_{1}=1}^{G} \left( \mathbf{T}_{\text{gg'}} + \frac{1}{\lambda} \mathbf{B}_{\text{gg'}} \right) \mathbf{F} \right)_{\text{g'}}; \ g = 1 \text{ to } G \]  
\hspace{1cm} (4.15)

as described in Eqs. 4.1 through 4.5, where:

\[ \mathbf{I}^T_{\text{g}} = \int_{K} \dot{\mathbf{I}}^T_{\text{g}} \mathbf{D}_{\text{g}} \dot{\mathbf{I}}_{\text{g}} \, dz \]  
\hspace{1cm} (4.16a)

\[ \mathbf{M}_{\text{g}} = \int_{K} \mathbf{P}^T_{\text{g}} \mathbf{A}_{\text{g}} \mathbf{P}_{\text{g}} \, dz \]  
\hspace{1cm} (4.16b)
These matrices are \(N(K+1) \times N(K+1)\) block tridiagonal of similar form whose \(N \times N\) submatrices are integrals of \(N \times N\) dyads. The \(k^{\text{th}}\) row of the product \(\mathbf{A}_{gg'} F_{g'}\) is, for example:

\[
\begin{align*}
[\mathbf{A}_{gg'} F_{g'}]_k &= \left\{ \int_0^1 h_{k-1} \mathbf{A}_{gg',k-1}(x) P^+ \mathbf{P}^T_{g,k-1}(x) P^-_{g',k-1}(x) dx \right\} F_{g',k-1} \\
&+ \left\{ \int_0^1 h_{k-1} \mathbf{A}_{gg',k-1}(x) P^+ \mathbf{P}^T_{g,k-1}(x) P^-_{g',k-1}(x) dx \right\} F_{g',k} \\
&+ \left\{ \int_0^1 h_{k} \mathbf{A}_{gg',k}(x) P^+ \mathbf{P}^T_{g,k}(x) P^-_{g',k}(x) dx \right\} F_{g',k+1}
\end{align*}
\]

These matrix relationships allow presentation of the following matrix properties.

**Theorem 1:** \(\mathbf{I}_{g}\) and \(\mathbf{M}_{g}\) are guaranteed to be positive definite whenever the detailed weighting functions \(\psi^*_{g,k}(z)\) have a similar shape to that of the detailed flux solutions \(\psi_{g,k}(z)\), as given by:

\[
\psi^*_{g,k}(z) = C_{g,k} \psi_{g,k}(z)
\]  

where \(C_{g,k}\) is a positive constant for each energy group \(g\) and each region \(k\).
Proof: Under these conditions, \( \mathbf{P}_{g,k}^{\pm} = \mathbf{C}_{g,k} \mathbf{P}_{g,k}^{\pm} \);

hence,

\[
\mathbf{I} \mathbf{P}_{g}^{*} = \mathbf{C}_{g} \mathbf{I} \mathbf{P}_{g}^{*}
\]  

(4.19a)

where

\[
\mathbf{C}_{g} = \text{Diag}(\mathbf{C}_{g,1}, \ldots, \mathbf{C}_{g,K})
\]  

(4.19b)

First consider \( \mathbf{M}_{g} \). Given any arbitrary constant nonzero vector \( \mathbf{q} \),

\[
\mathbf{q}^{T} \mathbf{M}_{g} \mathbf{q} = \int_{K} (\mathbf{I} \mathbf{P}_{g} \mathbf{q})^{T} \mathbf{C}_{g} \mathbf{A}_{g}^{T} \mathbf{I} \mathbf{P}_{g} \mathbf{q} \, dz
\]

(4.20)

and since \( \mathbf{C}_{g} \) and \( \mathbf{A}_{g}^{T} \) are both positive block diagonal matrices with diagonal submatrices, their product can be factored into

\[
\mathbf{C}_{g}^{T} \mathbf{A}_{g}^{T} = \left[ (\mathbf{A}_{g}^{T} \mathbf{C}_{g}^{T})^{\frac{1}{2}} \right]^{T} \left( \mathbf{C}_{g}^{T} \mathbf{A}_{g}^{T} \right)^{\frac{1}{2}}
\]

(4.21)

Therefore,

\[
\mathbf{q}^{T} \mathbf{M}_{g} \mathbf{q} = \int_{K} \left[ (\mathbf{C}_{g}^{T} \mathbf{A}_{g}^{T})^{\frac{1}{2}} \mathbf{I} \mathbf{P}_{g} \mathbf{q} \right]^{T} \left[ (\mathbf{C}_{g}^{T} \mathbf{A}_{g}^{T})^{\frac{1}{2}} \mathbf{I} \mathbf{P}_{g} \mathbf{q} \right] \, dz
\]

(4.22a)

\[
= \int_{K} \mathbf{I} \mathbf{R}^{T} \mathbf{I} \mathbf{R} \, dz
\]

(4.22b)

which is always greater than zero for arbitrary nonzero \( \mathbf{q} \). Hence, by definition, \( \mathbf{M}_{g} \) is positive definite. A similar proof holds for \( \mathbf{L}_{g} \) using Eq. 4.16a.

The following corollaries immediately result.

Corollary 1: If Rayleigh-Ritz Galerkin weighting, \( \mathbf{U}^{*} = \mathbf{U} \), is used in the approximation, then \( \mathbf{L}_{g} \) and \( \mathbf{M}_{g} \) are positive definite.
**Corollary 2:** $\mathbb{IL}_g$ and $\mathbb{IM}_g$ resulting from the finite element methods using linear and cubic Hermite basis functions are both positive definite.

**Corollary 3:** If $\mathbb{IL}_g$ and $\mathbb{IM}_g$ are positive definite, then so is the matrix $(\mathbb{IL}_g + \mathbb{IM}_g)$. These three matrices are then also symmetric.

It is also interesting to note the properties of $\mathbb{IL}_g$ and $\mathbb{IM}_g$ for cases of symmetry; that is, when the material properties and detailed flux solutions are symmetric about the center of each coarse mesh region $k$. Such symmetry occurs in regular repeating reactor geometries, and is denoted by:

\[
D_{g,k}(x) = D_{g,k}(1-x)
\]  
(4.23a)

and

\[
\Lambda_{g,k}(x) = \Lambda_{g,k}(1-x)
\]  
(4.23b)

Hence,

\[
\psi_{g,k}(x) = \psi_{g,k}(1-x)
\]  
(4.23c)

and

\[
\eta_{g,k}(x) = -\eta_{g,k}(1-x)
\]  
(4.23d)

and similarly for the weighting fluxes and currents. Under such conditions, the $P_{g,k}^+$ and $P_{g,k}^-$ support functions can be found by inspection of Eqs. 4.9 and 4.10 to obey the following symmetries:

For $N = 1$:

\[
P_{g,k}^+(x) = P_{g,k}^-(1-x)
\]  
(4.24a)

\[
P_{g,k}^-(x) = -P_{g,k}^+(1-x)
\]  
(4.24b)
For $N = 2$:

\[
\hat{\mathbf{P}}_{g,k}^\mp(x) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{P}_{g,k}^\pm(1-x) \tag{4.24c}
\]

\[
\hat{\mathbf{P}}^\mp_{g,k}(x) = -\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \hat{\mathbf{P}}^\pm_{g,k}(1-x) \tag{4.24d}
\]

where the following symmetries of the polynomials defined in Eqs. 3.11 have been used:

\[
p_1(x) = p_2(1-x)
\]
\[
p_3(x) = -p_4(1-x)
\]
\[
q_1(x) = -q_2(1-x)
\]
\[
q_3(x) = q_4(1-x)
\]

Similar identities with identical signs hold for the weighting quantities $\hat{\mathbf{p}}_{g,k}^\pm(x)$.

**Theorem 2:** For cases of symmetry, as given above, the matrices $\mathbf{L}_g$, $\mathbf{M}_g$, $\mathbf{T}_{g g'}$, and $\mathbf{B}_{g g'}$ are all symmetric regardless of the relation of $\psi_{g,k}^*(x)$ to $\psi_{g,k}(x)$.

**Proof:** Referring to Eq. 4.17, $\mathbf{L}_g$, for example, is symmetric only if

\[
\int_0^1 h_k \mathbf{D}_{g,k}(x) \hat{\mathbf{p}}^\pm_{g,k}(x) \hat{\mathbf{p}}^\pm_{g,k}(x) dx = \int_0^1 h_k \mathbf{D}_{g,k}(x) \hat{\mathbf{p}}^\pm_{g,k}(x) \hat{\mathbf{p}}^\pm_{g,k}(x) dx
\]

This can be shown for any $N$ by changing variables in one of the integrals from $x$ to $1-x'$ and using the symmetry properties of Eqs. 4.23 and 4.24. Similar proofs hold for the other matrices.
It is unfortunate that the above symmetry conditions do not allow direct proof that $L_g$ and $M_g$ have positive diagonal elements and are also diagonally dominant (for at least one row) for arbitrary positive and symmetric detailed flux solutions. Under such conditions, $L_g$ and $M_g$ would then be positive definite, since they are block tridiagonal with nonzero diagonal elements and hence irreducible. Instead, these conditions can be used to obtain a set of algebraic equations which, for completely arbitrary detailed flux solutions, must be satisfied in order that $L_g$ and $M_g$ be positive definite.

The requirement that $L_g$ and $M_g$ be positive definite is useful only in the inversion of $(L_g + M_g)$. Although the inversion can always be performed using Gaussian elimination techniques, the property of positive definiteness allows the use of Cholesky's method, discussed in the next section, which is faster and requires less computer storage.

4.2 Calculational and Programming Techniques

Calculation of the one-dimensional subassembly detailed fluxes, currents, and adjoint solutions, as well as detailed "exact" or reference solutions, were performed using the program REF2G described in section 1 of Appendix D. Assuming subassembly $k$ to be divided into $N$ homogeneous intervals at nodes $t_i$ and of width $h_{s_i}$, the program uses fine mesh linear finite element approximations to calculate the detailed flux solutions for each group. Omitting group subscripts, the detailed flux solution for each group in subassembly $k$ is represented by a set of $N+1$ points

$$\psi_k(x) = \{\psi_{k,i} : i = 1, N+1\} \quad (4.27)$$
where $\psi_k(x)$ is linear between points. Detailed group current solutions $\eta_k(x)$ are represented by a set of $N$ points

$$\eta_k(x) = \{\eta_{k,i} : i = 1, N\} \quad (4.28)$$

which are found from the converged flux solutions by Fick's law

$$\eta_{k,i} = \frac{1}{h_{s_i} D_{k,i}} \psi_{k,i} - \psi_{k,i+1}; \quad i = 1, N \quad (4.29)$$

where $D_{k,i}$ is the diffusion constant homogeneous in interval $i$. $\eta_k(x)$ is of constant value, $\eta_{k,i}$, within each interval. The forms of these solutions are illustrated in Figure 4.2.

In order to approximate the symmetry boundary conditions imposed on the detailed subassembly flux solutions, small intervals $h_{s_1}$ and $h_{s_N}$ are defined at the edges of each subassembly. The detailed current solutions can then be made to have zero boundary values by setting $\eta_{k,1}$ and $\eta_{k,N}$ to zero. However, since the currents in each interval are defined as inversely proportional to the mesh size, the calculated boundary currents using this scheme may not be small enough to be negligible.

Explicit zero current boundary conditions can be imposed on the detailed current solutions by transforming the above discontinuous current $\eta_k(x)$ into a continuous current solution $\tilde{\eta}_k(x)$ represented by a set of $N+1$ points

$$\tilde{\eta}_k(x) = \{\tilde{\eta}_{k,i} : i = 1 \text{ to } N+1\} \quad (4.30)$$

where $\tilde{\eta}_k(x)$ is linear between points, as also illustrated in Figure 4.2. By seeking to minimize the mean square error between $\eta_k(x)$ and $\tilde{\eta}_k(x)$ within each interval $i$, variational techniques yield the following set of $N-1$ equations for each group:
Figure 4.2. Subassembly Notations and Detailed Solutions
\[
\left( \frac{1}{6} h s_{i-1} \right) \tilde{\eta}_{k,i-1} + \frac{1}{3} \left( h s_{i-1} + h s_i \right) \tilde{\eta}_{k,i} + \left( \frac{1}{6} h s_i \right) \tilde{\eta}_{k,i+1} = \left( \frac{1}{2} h s_{i-1} \right) \eta_{k,i-1} + \left( \frac{1}{2} h s_i \right) \eta_{k,i} ; \quad i = 2 \text{ to } N \quad (4.31)
\]

These equations, given the \( \eta_{k,i} \) from Eq. 4.29, are easily solved for the \( \tilde{\eta}_{k,i} \), \( i = 2 \) to \( N \), where \( \tilde{\eta}_{k,1} \) and \( \tilde{\eta}_{k,N+1} \) are set to zero. Both forms of the detailed currents, \( \eta_k(x) \) and \( \tilde{\eta}_k(x) \), are allowed for use in the proposed approximation methods.

The proposed methods using linear and cubic Hermite basis functions have been programmed into computer codes LINEAR and CUBIC which are described respectively in sections 2 and 3 of Appendix D.

The matrix elements required for use in the approximation methods are integrals of products of subassembly detailed solutions and polynomial functions. These integrals are calculated, for each index \( k \), from the basic integral unit

\[
\text{BIU}_k = \int_0^1 f_k(x) g_k(x) C_k(x) x^n h_k \, dx \quad (4.32)
\]

where the functions \( f_k(x) \) and \( g_k(x) \) represent flux and/or current solutions for same or different groups. These functions may be either constant within each interval

\[
f_k(x) = \left\{ f_{k,i} : i = 1 \text{ to } N \right\} \quad (4.33)
\]

or of linear form within each interval

\[
f_k(x) = \left\{ (1-y)f_{k,i} + y f_{k,i+1} : i = 1 \text{ to } N \right\} \quad (4.34)
\]

where \( y = \frac{1}{h s_i} (t-t_i) \), as defined in Figure 4.2. \( C_k(x) \) represents a
group nuclear constant which is homogeneous in each interval

\[ C_k(x) = \{ C_{k,i} : i = 1 \text{ to } N \} \quad (4.35) \]

and \( n \) is a positive integer exponent in the range \( 0 \leq n \leq 6 \). Since the following remarks concern only subassembly \( k \), the index \( k \) is dropped for simplicity.

The basic integral unit can be broken into integrals over each interval by transforming variables from \( x \) to \( y \). The result

\[ \text{BIU}_k = \frac{1}{h^*_k n} \sum_{i=1}^{N} h s_i \int_0^1 f_i(y) g_i(y) C_i(t_i + h s_i y)^n dy \quad (4.36) \]

can be integrated analytically by expanding \( (t_i + h s_i y)^n \) into a binomial series. The results of these integrations for any \( n \) depend only on the given forms of \( f(x) \) and \( g(x) \), and are summarized in Table 4.1.

The coarse mesh flux-weighting homogenization calculations were performed using the above basic integral unit with \( n = 0 \). In these calculations a linear form of \( f(x) \), representing the detailed subassembly flux solutions from REF2G, and a constant value of \( g(x) = 1 \) were used.

Once the elements of the matrices of the approximation methods have been formed, considerable computer storage can be saved by collapsing the sparse band-structured matrices into full matrix form using row index transformations. In this way, a \( N \times N \) tridiagonal matrix \( \mathbf{IL} \) resulting from the use of linear basis functions can be stored as the \( N \times 3 \) matrix \( \mathbf{IL}' \) by

\[ (\mathbf{IL}')_{ik} = (\mathbf{IL})_{ij} \quad (4.37) \]

where \( k = j + 2 - i \), and \( k \) values outside \( 1 \leq k \leq 3 \) are omitted.
Table. 4.1. Calculation of the Basic Integral Unit in Subassembly $k$.

1. Constant $f(x)$ and constant $g(x)$ in each interval $i$:

$$ BIU_k = \frac{1}{h_k} \sum_{i=1}^{N} \left\{ \sum_{\ell=0}^{n} b_{\ell}^n \frac{t_i^{n-\ell} h_s^\ell}{(\ell+1)} \frac{1}{(\ell+1)} \right\} $$

2. Linear $f(x)$ and constant $g(x)$ in each interval $i$:

$$ BIU_k = \frac{1}{h_k} \sum_{i=1}^{N} \left\{ \sum_{\ell=0}^{n} b_{\ell}^n \frac{t_i^{n-\ell} h_s^\ell}{(\ell+1)} \frac{f_i}{(\ell+1)(\ell+2)} + \frac{f_{i+1}}{(\ell+2)} \right\} $$

3. Linear $f(x)$ and linear $g(x)$ in each interval $i$:

$$ BIU_k = \frac{1}{h_k} \sum_{i=1}^{N} \left\{ \sum_{\ell=0}^{n} b_{\ell}^n \frac{t_i^{n-\ell} h_s^\ell}{(\ell+1)} \frac{2f_i g_i}{(\ell+1)(\ell+2)(\ell+3)} \right. $$

$$ + \frac{f_i g_{i+1} + f_{i+1} g_i}{(\ell+2)(\ell+3)} + \frac{f_{i+1} g_{i+1}}{(\ell+3)} \left\} $$

where

$$ b_{\ell}^n = \frac{n!}{\ell! (n-\ell)!} $$

is the binomial series coefficient.
Similarly, a N×N matrix \( IL \) of half-band width equal to three, which results from the use of cubic Hermite basis functions, can be stored as the N×6 matrix \( IL' \), as given above, where in this case:

\[
k = j - i + 3 + \begin{cases} 
1 & \text{for } i \text{ odd} \\
0 & \text{for } i \text{ even}
\end{cases}
\]  \hspace{1cm} (4.38)

and \( k \) values outside \( 1 \leq k \leq 6 \) are omitted.

In those cases where the mass and stiffness matrices are both positive definite, Cholesky's method of matrix factorization,

\[
IL = G G^T
\]  \hspace{1cm} (4.39)

where \( IL \) is positive definite and \( G \) is lower triangular, can be used to solve the matrix inversion for each group in the power method. The matrix elements \( g_{ij} = (G)_{ij} \) are calculated from the elements \( \ell_{ij} = (IL)_{ij} \) by the following algorithm:

\[
\begin{align*}
\text{For each } j = 1 \text{ to } N: \\
g_{jj} &= \left( \ell_{jj} - \sum_{k=1}^{j-1} g_{jk} \right)^{1/2} \\
\text{For each } i = j+1 \text{ to } N: \\
g_{ij} &= \left( \ell_{ij} - \sum_{k=1}^{j-1} g_{ik} g_{jk} \right) / g_{jj}
\end{align*}
\]  \hspace{1cm} (4.40)

Similar algorithms of a more complex form are used in the computer codes in conjunction with the matrix collapsing schemes given above.
Chapter 5
NUMERICAL RESULTS

5.1 Nuclear Constants and Subassembly Geometry

The effectiveness and accuracy of the proposed approximation methods were examined using one-dimensional, one- and two-group reactor configurations composed of representative PWR fuel subassemblies. Four separate subassemblies with identical geometry but different number constants are considered. Each subassembly is represented as an 18-cm, homogeneous fuel region of low, medium, or high enrichment, surrounding a 1-cm centrally located absorption rod or water channel. Two-group regional nuclear constants used to represent such PWR subassembly geometries are given in Table 5.1, where all fission neutrons are assumed to be born in the fast group. These constants were collapsed into representative one-group constants using the standard infinite medium group reduction procedure for two groups:

\[ \langle \Sigma \rangle_{1G} = \frac{1}{1+\alpha} \left( \Sigma_1 + \alpha \Sigma_2 \right) \]  

(5.1)

where \( \Sigma_1 \) and \( \Sigma_2 \) are macroscopic cross sections for the fast and thermal groups, respectively, and \( \alpha \) is the infinite medium thermal to fast flux ratio. The resulting one-group regional constants for the fuel and rod regions are given in Table 5.2, where the flux ratios of the three fuel regions have been averaged in order to collapse the absorption rod constants.
Table 5.1. Representative Two-Group, 18-cm, PWR Subassembly Regional Nuclear Constants.

\[ \chi_1 = 1.0; \chi_2 = 0.0. \]

<table>
<thead>
<tr>
<th>Region Material</th>
<th>( \Sigma_T )</th>
<th>( \nu \Sigma_f )</th>
<th>( D )</th>
<th>( \Sigma_{21} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel A: Low w/o</td>
<td>.0259</td>
<td>.00485</td>
<td>1.396</td>
<td>.0179</td>
</tr>
<tr>
<td></td>
<td>.0532</td>
<td>.0636</td>
<td>.388</td>
<td></td>
</tr>
<tr>
<td>Fuel B: Medium w/o</td>
<td>.0260</td>
<td>.00553</td>
<td>1.397</td>
<td>.0172</td>
</tr>
<tr>
<td></td>
<td>.0710</td>
<td>.102</td>
<td>.389</td>
<td></td>
</tr>
<tr>
<td>Fuel C: High w/o</td>
<td>.0261</td>
<td>.00659</td>
<td>1.399</td>
<td>.0168</td>
</tr>
<tr>
<td></td>
<td>.0832</td>
<td>.129</td>
<td>.387</td>
<td></td>
</tr>
<tr>
<td>Absorption Rod</td>
<td>.0452</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>.959</td>
<td>0.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td>.0383</td>
<td>0.0</td>
<td>1.63</td>
<td>.0380</td>
</tr>
<tr>
<td></td>
<td>.0108</td>
<td>0.0</td>
<td>.275</td>
<td></td>
</tr>
</tbody>
</table>

Fast group constants appear first for each region material, followed by thermal group constants. Fission neutrons are assumed to be born in the fast group only.

Table 5.2. Representative One-Group, 18-cm, PWR Subassembly Regional Nuclear Constants.

<table>
<thead>
<tr>
<th>Region Material</th>
<th>( \Sigma_T )</th>
<th>( \nu \Sigma_f )</th>
<th>( D )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuel A: Low w/o</td>
<td>.0329</td>
<td>.0199</td>
<td>1.14</td>
</tr>
<tr>
<td>Fuel B: Medium w/o</td>
<td>.0348</td>
<td>.0244</td>
<td>1.20</td>
</tr>
<tr>
<td>Fuel C: High w/o</td>
<td>.0357</td>
<td>.0272</td>
<td>1.23</td>
</tr>
<tr>
<td>Absorption Rod</td>
<td>.235</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Water</td>
<td>.0136</td>
<td>0.0</td>
<td>.414</td>
</tr>
</tbody>
</table>
Four subassembly configurations, labeled A through D, were used in the one- and two-group test configurations, and are illustrated in Figure 5.1. Subassemblies labeled A, B, and C contain homogeneous fuel of low, medium, and high enrichment, respectively, surrounding the 1-cm absorption rod while subassembly D contains low enriched homogeneous fuel surrounding a 1-cm water channel.

5.2 Subassembly Detailed Solutions and Homogenized Nuclear Constants

The detailed flux and current solutions for each subassembly were found using the computer code REF2G with symmetry boundary conditions and a 68-mesh region per subassembly geometry as indicated in Figure 5.2. The resulting one-group detailed flux solutions for each subassembly are shown in Figure 5.3. The resulting two-group detailed flux and adjoint flux solutions for each subassembly are shown in Figures 5.4 and 5.5, respectively.

Homogeneous subassembly group constants for use in the finite element approximations were found by flux weighting the group cross sections in each subassembly by the corresponding subassembly detailed group flux solutions. The resulting homogenized one-group constants for each subassembly are given in Table 5.3, and the resulting homogenized two-group constants are given in Table 5.4. The results of homogenizing the diffusion coefficient as the transport cross section, $1/(1/D)$, as well as by direct homogenization, $\langle D \rangle$, are included in the tables. The results of both schemes were found to differ at most by only 2%. The directly homogenized diffusion coefficients, $\langle D \rangle$, were used in the finite element approximations.
Subassembly Type A:

Subassembly Type B:

Subassembly Type C:

Subassembly Type D:

Figure 5.1. Subassembly Configuration Geometries
Symmetric Partitioning:

1 (1/16 cm) + 1 (15/16) + 4 (1) + 2 (1/2) + 6 (1/4) + 4 (1/8) + 8 (1/16) + 8 (1/16)

Figure 5.2. Mesh Geometry in Half a Subassembly. Detailed flux and current solution calculations use this 68 intervals/subassembly geometry in each subassembly type.
Figure 5.3. Subassembly Detailed Flux Solutions for the One-Group Case

Subassembly Type A ——— (lower curve)
Subassembly Type B - - - - -
Subassembly Type C ·······
Subassembly Type D ——— (upper curve)
Figure 5.4. Subassembly Detailed Flux Solutions for the Two-Group Case

The fluxes are normalized by fast flux values so that the thermal fluxes appear in the lower portion of the figure.

Subassembly Type A ———— (lower curves)
Subassembly Type B - - - -
Subassembly Type C ·······
Subassembly Type D ———— (upper curves)
Figure 5.5. Subassembly Detailed Adjoint Flux Solutions for the Two-Group Case

Subassembly Type A ——— (lower curves)
Subassembly Type B ————
Subassembly Type C ————
Subassembly Type D ———— (upper curves)
Table 5.3. Homogenized Subassembly One-Group Nuclear Constants.

<table>
<thead>
<tr>
<th>Subassembly</th>
<th>$\langle \Sigma_T \rangle$</th>
<th>$\langle \nu \Sigma_f \rangle$</th>
<th>$\langle D \rangle$</th>
<th>$1/\langle 1/D \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.04149392</td>
<td>0.01905379</td>
<td>1.134047</td>
<td>1.133253</td>
</tr>
<tr>
<td>B</td>
<td>0.04341140</td>
<td>0.02335046</td>
<td>1.191397</td>
<td>1.189765</td>
</tr>
<tr>
<td>C</td>
<td>0.04431869</td>
<td>0.02602374</td>
<td>1.220054</td>
<td>1.217887</td>
</tr>
<tr>
<td>D</td>
<td>0.03184731</td>
<td>0.01881458</td>
<td>1.100401</td>
<td>1.040479</td>
</tr>
</tbody>
</table>

Table 5.4. Homogenized Subassembly Two-Group Nuclear Constants.

\[ x_1 = 1.0; \quad x_2 = 0.0. \]

<table>
<thead>
<tr>
<th>Subassembly</th>
<th>$\langle \Sigma_T \rangle$</th>
<th>$\langle \nu \Sigma_f \rangle$</th>
<th>$\langle D \rangle$</th>
<th>$\langle \Sigma_{21} \rangle$</th>
<th>$1/\langle 1/D \rangle$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.02688787</td>
<td>0.004601752</td>
<td>1.379526</td>
<td>0.01698379</td>
<td>1.371911</td>
</tr>
<tr>
<td></td>
<td>0.06812834</td>
<td>0.06255182</td>
<td>0.3980863</td>
<td></td>
<td>0.3919533</td>
</tr>
<tr>
<td>B</td>
<td>0.02698495</td>
<td>0.005246314</td>
<td>1.379480</td>
<td>0.01631765</td>
<td>1.37185</td>
</tr>
<tr>
<td></td>
<td>0.08647802</td>
<td>0.1002221</td>
<td>0.3996499</td>
<td></td>
<td>0.3931874</td>
</tr>
<tr>
<td>C</td>
<td>0.02708207</td>
<td>0.006251160</td>
<td>1.379433</td>
<td>0.01593619</td>
<td>1.371787</td>
</tr>
<tr>
<td></td>
<td>0.09893614</td>
<td>0.1266822</td>
<td>0.3980142</td>
<td></td>
<td>0.391310</td>
</tr>
<tr>
<td>D</td>
<td>0.02657213</td>
<td>0.004587110</td>
<td>1.412467</td>
<td>0.0189895</td>
<td>1.410790</td>
</tr>
<tr>
<td></td>
<td>0.05034835</td>
<td>0.05932253</td>
<td>0.3804001</td>
<td></td>
<td>0.3775656</td>
</tr>
</tbody>
</table>

Fast group constants appear first for each subassembly, followed by thermal group constants.
Before applying the proposed approximation methods to complex reactor geometries, test runs were performed in order to evaluate the differences between using either flux or adjoint flux weighting, and using current solutions of either constant or linear form in each subassembly interval, as described in section 4.2. The test problem consisted of three consecutive Type A subassemblies with symmetry boundary conditions imposed on each end so that the converged eigenvalue \( \lambda (k_{\text{eff}}) \) should be identical to that of the detailed flux solution of subassembly A. Entire subassemblies were chosen as the mesh regions so that the proposed synthesis methods should converge to flux values of unity, and current values of zero. The numerical results of these tests for the one- and two-group cases are summarized in Table 5.5. Although the choice of weighting function did not influence the results for either approximation, use of current solutions of the linear form enables better eigenvalue accuracy. In addition, the results when using currents of linear form converged to flux values of unity and current values of zero, as expected, while results using the constant current form produced errors of about 0.5% in the converged flux and 0.01% in the converged current at interior points. Although the difference in accuracy between the use of these different current forms is small, the small flux and current errors resulting from the use of the constant current form may lead to larger errors in larger and more complex problems. For the above reasons, the linear current form was used in the following case studies. Adjoint weighting was also used. Although the use of adjoint weighting has not been shown to guarantee the success of Cholesky's method in the numerical solution scheme, no difficulties with its use were ever encountered.
Table 5.5. Test Results Using Three Consecutive Type A Subassemblies.

\[
\% \lambda = \left( \frac{\lambda_{\text{Sub. } A} - \lambda_{\text{Conv.}}}{\lambda_{\text{Sub. } A}} \right) \times 100%.
\]

<table>
<thead>
<tr>
<th>Synthesis Approximation</th>
<th>Weighting Function</th>
<th>Form of Currents</th>
<th>Converged ( \lambda )</th>
<th>% ( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ONE GROUP:</strong> ( \lambda_{\text{Sub. } A} = 0.459194 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear</td>
<td>FLUX</td>
<td>Constant</td>
<td>.459363</td>
<td>-.036%</td>
</tr>
<tr>
<td>Linear</td>
<td>FLUX</td>
<td>Linear</td>
<td>.459254</td>
<td>-.013%</td>
</tr>
<tr>
<td>Cubic</td>
<td>FLUX</td>
<td>Constant</td>
<td>.459363</td>
<td>-.036%</td>
</tr>
<tr>
<td>Cubic</td>
<td>FLUX</td>
<td>Linear</td>
<td>.459254</td>
<td>-.013%</td>
</tr>
<tr>
<td><strong>TWO GROUPS:</strong> ( \lambda_{\text{Sub. } A} = 0.751095 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear</td>
<td>FLUX</td>
<td>Linear</td>
<td>.751284</td>
<td>-.025%</td>
</tr>
<tr>
<td>Linear</td>
<td>ADJOINT</td>
<td>Constant</td>
<td>.7513818</td>
<td>-.038%</td>
</tr>
<tr>
<td>Linear</td>
<td>ADJOINT</td>
<td>Linear</td>
<td>.751284</td>
<td>-.025%</td>
</tr>
<tr>
<td>Cubic</td>
<td>FLUX</td>
<td>Linear</td>
<td>.751284</td>
<td>-.025%</td>
</tr>
<tr>
<td>Cubic</td>
<td>ADJOINT</td>
<td>Constant</td>
<td>.7513818</td>
<td>-.038%</td>
</tr>
<tr>
<td>Cubic</td>
<td>ADJOINT</td>
<td>Linear</td>
<td>.751284</td>
<td>-.025%</td>
</tr>
</tbody>
</table>
5.3 Case Studies and Results

Four one-dimensional reactor configurations, each made up of different combinations of types of subassemblies, are considered in the case studies below. One-group calculations were performed only for the first case, and two-group calculations were performed in all cases. Entire 18-cm subassemblies were used as coarse mesh regions in each case, while the effect of using half-subassembly mesh regions was also included in Case 1. The geometry and subassembly configurations of the case studies are shown in Figure 5.6.

Three separate approximation methods were used to calculate converged detailed flux solutions for comparison in each case. They are:

1. The proposed approximation methods using heterogeneous nuclear constants and subassembly detailed flux solutions for coarse mesh solutions.
2. The finite element methods using subassembly homogenized nuclear constants for coarse mesh solutions.
3. The linear finite element method for fine mesh reference solutions.

Calculations of both the proposed approximation and the coarse mesh finite element method using linear basis functions were performed using program LINEAR, while the corresponding cubic Hermite basis function approximations were performed using program CUBIC. The fine mesh reference solutions were calculated using program REF2G, and the results of these approximations were compared and analyzed by program ANALYZE. Descriptions of these programs are given in Appendix D.
Case 1:

<table>
<thead>
<tr>
<th>Symmetry</th>
<th>Subassembly Type A</th>
<th>Subassembly Type B</th>
<th>Subassembly Type C</th>
<th>Symmetry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.</td>
<td>18.</td>
<td>36.</td>
<td>54.</td>
<td></td>
</tr>
</tbody>
</table>

Case 2:

<table>
<thead>
<tr>
<th>Symmetry</th>
<th>Subassembly Type D</th>
<th>Subassembly Type B</th>
<th>Subassembly Type C</th>
<th>Symmetry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.</td>
<td>18.</td>
<td>36.</td>
<td>54.</td>
<td></td>
</tr>
</tbody>
</table>

Case 3:

<table>
<thead>
<tr>
<th>Zero Flux</th>
<th>Water</th>
<th>C</th>
<th>C</th>
<th>C</th>
<th>A</th>
<th>A</th>
<th>A</th>
<th>D</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.18.36.54.72.90.108.126.144.153.cm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Case 4:

<table>
<thead>
<tr>
<th>Zero Flux</th>
<th>Water</th>
<th>D</th>
<th>D</th>
<th>D</th>
<th>C</th>
<th>D</th>
<th>D</th>
<th>A</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.18.36.54.72.90.108.126.144.153.cm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.6. Geometry of the Four Case Studies Composed of Types of Subassemblies
The results of each case study are divided into the two approximation method categories as defined below.

1. The linear basis function approximation
   A. Linear FEM:
      (The linear finite element method using homogenized coarse mesh nuclear constants)
   B. Linear Synth:
      (The proposed approximation method using heterogeneous coarse mesh nuclear constants and detailed coarse mesh solutions)

2. The cubic Hermite basis function approximations
   A. Cubic FEM
   B. Cubic Synth

The results of the approximations in each category are compared to the reference solution by examining:

1. The converged eigenvalues \( \lambda (k_{\text{eff}}) \) and their percent normalized eigenvalue error,
   \[ \% \lambda = \frac{\lambda_{\text{Ref}} - \lambda_{\text{Conv}}}{\lambda_{\text{Ref}}} \times 100\% \]

2. Composite graphs of the converged detailed group flux solutions \( U_g(z) \) normalized to equivalent power levels

3. The fractional normalized power levels \( P(k) \) calculated for each 18-cm subassembly \( k \) by
\[
P(k) = \frac{\int \text{Sub. k} \sum_{g=1}^{G} \nu \Sigma_{fg}(z) U_{g}(z) \, dz}{\int \text{All Subs.} \sum_{g=1}^{G} \nu \Sigma_{fg}(z) U_{g}(z) \, dz}
\] (5.2)

and their percent normalized errors

\[
\%P(k) = \frac{(P(k)_{\text{Ref}} - P(k)_{\text{Conv}})}{P(k)_{\text{Ref}}} \times 100\%
\] (5.3)

5.3.1. **Case 1:** Three different subassemblies of Types A, B, and C with symmetry boundary conditions.

The graphical results of the one-group approximation methods for this case are shown in Figures 5.7 and 5.8, while the results of the two-group approximation methods are presented in Figures 5.9 - 5.12. Only the coarse mesh boundaries are labeled in the figures, which indicates that entire 18-cm subassemblies were used as the coarse mesh regions. Two-group results using only half-subassemblies as the coarse mesh regions are shown in Figures 5.13 - 5.16. The reference solutions were calculated using 150-mesh regions, as defined by the symmetric partitioning

\[5(1 \text{ cm}) + 4(.5 \text{ cm}) + 4(.25 \text{ cm}) + 4(.125 \text{ cm}) + 8(.0625 \text{ cm})\]

in each of the three subassemblies. The converged approximation eigenvalues and fractional normalized subassembly power levels for the one- and two-group calculations are summarized in Table 5.6. The fractional powers, \(P(k)\), for each subassembly are listed in the reference solution column, while the percent errors, \(\%P(k)\) are listed in the approximation columns.
Figure 5.7. Case 1: One-Group Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.559045</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>.556943</td>
</tr>
<tr>
<td>Linear Synth</td>
<td>.557154</td>
</tr>
</tbody>
</table>
Figure 5.8. Case 1: One-Group Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.559045</td>
</tr>
<tr>
<td>Cubic FEM</td>
<td>.558647</td>
</tr>
<tr>
<td>Cubic Synth</td>
<td>.558761</td>
</tr>
</tbody>
</table>
Figure 5.9. Case 1: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.917267</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>.914489</td>
</tr>
<tr>
<td>Linear Synth</td>
<td>.915221</td>
</tr>
</tbody>
</table>
Figure 5.10. Case 1: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>⋯ ⋯ ⋯</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>- - - -</td>
</tr>
<tr>
<td>Linear Synth</td>
<td>_______</td>
</tr>
</tbody>
</table>
Figure 5.11. Case 1: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.917267</td>
</tr>
<tr>
<td>Cubic FEM</td>
<td>0.916717</td>
</tr>
<tr>
<td>Cubic Synth</td>
<td>0.917059</td>
</tr>
</tbody>
</table>
Figure 5.12. Case 1: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.917267</td>
</tr>
<tr>
<td>Cubic FEM</td>
<td>.916717</td>
</tr>
<tr>
<td>Cubic Synth</td>
<td>.917059</td>
</tr>
</tbody>
</table>
Figure 5.13. Case 1: Two-Group Fast Results Using Linear Basis Function Approximations and 9-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.917267</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>0.916356</td>
</tr>
<tr>
<td>Linear Synth</td>
<td>0.916427</td>
</tr>
</tbody>
</table>
Figure 5.14. Case 1: Two-Group Thermal Results Using Linear Basis Function Approximations and 9-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.........</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>.........</td>
</tr>
<tr>
<td>Linear Synth</td>
<td></td>
</tr>
</tbody>
</table>
Figure 5.15. Case 1: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 9-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.917267</td>
</tr>
<tr>
<td>Cubic FEM</td>
<td>.916669</td>
</tr>
<tr>
<td>Cubic Synth</td>
<td>.917294</td>
</tr>
</tbody>
</table>
Figure 5.16. Case 1: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 9-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.917267</td>
</tr>
<tr>
<td>Cubic FEM</td>
<td>0.916669</td>
</tr>
<tr>
<td>Cubic Synth</td>
<td>0.917294</td>
</tr>
</tbody>
</table>
Table 5.6. Results of Case 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>Linear FEM</th>
<th>Linear Synth</th>
<th>Cubic FEM</th>
<th>Cubic Synth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ONE-GROUP RESULTS:**

<table>
<thead>
<tr>
<th>λ</th>
<th>.559045</th>
<th>.556943</th>
<th>.557154</th>
<th>.558647</th>
<th>.558761</th>
</tr>
</thead>
<tbody>
<tr>
<td>%λ</td>
<td>--</td>
<td>.376%</td>
<td>.338%</td>
<td>.072%</td>
<td>.051%</td>
</tr>
<tr>
<td>P(1)</td>
<td>.084</td>
<td>-12.1%</td>
<td>-11.9%</td>
<td>-3.07%</td>
<td>-2.44%</td>
</tr>
<tr>
<td>P(2)</td>
<td>.294</td>
<td>-4.29%</td>
<td>-3.93%</td>
<td>-2.41%</td>
<td>-0.434%</td>
</tr>
<tr>
<td>P(3)</td>
<td>.622</td>
<td>+3.67%</td>
<td>+3.47%</td>
<td>+5.28%</td>
<td>+5.34%</td>
</tr>
</tbody>
</table>

**TWO-GROUP RESULTS:**

<table>
<thead>
<tr>
<th>λ</th>
<th>.917267</th>
<th>.914489</th>
<th>.915221</th>
<th>.916717</th>
<th>.917059</th>
</tr>
</thead>
<tbody>
<tr>
<td>%λ</td>
<td>--</td>
<td>.302%</td>
<td>.223%</td>
<td>.060%</td>
<td>.023%</td>
</tr>
<tr>
<td>P(1)</td>
<td>.134</td>
<td>-6.93%</td>
<td>-5.63%</td>
<td>-1.43%</td>
<td>-1.13%</td>
</tr>
<tr>
<td>P(2)</td>
<td>.315</td>
<td>-1.63%</td>
<td>-1.39%</td>
<td>-0.072%</td>
<td>-0.120%</td>
</tr>
<tr>
<td>P(3)</td>
<td>.549</td>
<td>+2.63%</td>
<td>+2.17%</td>
<td>+0.391%</td>
<td>+0.347%</td>
</tr>
</tbody>
</table>

Two-Group Results Using Half-Subassembly Mesh Regions

<table>
<thead>
<tr>
<th>λ</th>
<th>.917267</th>
<th>.916356</th>
<th>.916427</th>
<th>.916669</th>
<th>.917294</th>
</tr>
</thead>
<tbody>
<tr>
<td>%λ</td>
<td>--</td>
<td>.093%</td>
<td>.092%</td>
<td>.065%</td>
<td>.003%</td>
</tr>
<tr>
<td>P(1)</td>
<td>.134</td>
<td>-2.38%</td>
<td>-2.69%</td>
<td>-1.51%</td>
<td>-0.461%</td>
</tr>
<tr>
<td>P(2)</td>
<td>.315</td>
<td>-0.475%</td>
<td>-0.602%</td>
<td>-0.063%</td>
<td>-0.047%</td>
</tr>
<tr>
<td>P(3)</td>
<td>.549</td>
<td>+0.851%</td>
<td>-2.63%</td>
<td>-3.22%</td>
<td>+1.40%</td>
</tr>
</tbody>
</table>
It is apparent from these results that the proposed approximation methods, and in particular the method utilizing the cubic Hermite basis functions, approximate to a high degree of accuracy the detailed reference spatial flux. Comparison of the eigenvalue and fractional power results in Table 5.6 indicates that comparable if not superior measurements are obtained using the proposed methods in this case.

It is interesting to note the effects of employing the given sub-assembly heterogeneous nuclear constants rather than subassembly homogenized nuclear constants for use in the finite element method calculations. Under such conditions, the finite element method becomes identical to the proposed methods in which the heterogeneous nuclear constants and constant or flat subassembly solutions are used. Two-group calculations using the cubic Hermite approximation method were performed for Case 1 and are presented in Figures 5.17 and 5.18. This scheme was found to give very poor detailed flux results, converge to an eigenvalue 21% in error, and yield an average of 20% error in the fractional normalized power levels in each subassembly. This example clearly illustrates the necessity for the use of homogenized constants in the finite element method, or equivalently, the importance of the subassembly detailed solutions in the proposed approximations.
Figure 5.17. Case 1: Two-Group Fast Results Using Cubic Hermite Finite Element Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>0.917267</td>
</tr>
<tr>
<td>Cubic FEM + Homogenized Consts.</td>
<td>0.916717</td>
</tr>
<tr>
<td>Cubic FEM + Detailed Consts.</td>
<td>0.720422</td>
</tr>
</tbody>
</table>
Figure 5.18. Case 1: Two-Group Thermal Results Using Cubic Hermite Finite Element Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.917267</td>
</tr>
<tr>
<td>Cubic FEM + Homogenized Consts.</td>
<td>.916717</td>
</tr>
<tr>
<td>Cubic FEM + Detailed Consts.</td>
<td>.720422</td>
</tr>
</tbody>
</table>
5.3.2. **Case 2**: Three different subassemblies of Types D, B, and C with symmetric boundary conditions.

The results of the two-group approximations for Case 2 are presented in Figures 5.19 - 5.22, where entire subassemblies were taken as the coarse mesh regions. The reference solutions were calculated using the same reference mesh geometry as in Case 1. The converged eigenvalues and fractional normalized power levels in each subassembly are summarized in Table 5.7. These results better illustrate the superiority of the cubic Hermite basis function approximations over the linear basis function approximations, and the superiority of the proposed approximations over the finite element method in all aspects.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>Linear FEM</th>
<th>Linear Synth</th>
<th>Cubic FEM</th>
<th>Cubic Synth</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda )</td>
<td>.969986</td>
<td>.965260</td>
<td>.970236</td>
<td>.966816</td>
<td>.969578</td>
</tr>
<tr>
<td>( % \lambda )</td>
<td>--</td>
<td>.487%</td>
<td>-.026%</td>
<td>.326%</td>
<td>.042%</td>
</tr>
<tr>
<td>( P(1) )</td>
<td>.381</td>
<td>+11.26%</td>
<td>+3.46%</td>
<td>+6.07%</td>
<td>+.643%</td>
</tr>
<tr>
<td>( P(2) )</td>
<td>.296</td>
<td>-11.59%</td>
<td>-5.59%</td>
<td>-3.08%</td>
<td>-.157%</td>
</tr>
<tr>
<td>( P(3) )</td>
<td>.322</td>
<td>-2.66%</td>
<td>+1.03%</td>
<td>-4.34%</td>
<td>-.616%</td>
</tr>
</tbody>
</table>
Figure 5.19. Case 2: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.969986</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>.965260</td>
</tr>
<tr>
<td>Linear Synth</td>
<td>.970236</td>
</tr>
</tbody>
</table>
Figure 5.20. Case 2: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.969986</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>.965260</td>
</tr>
<tr>
<td>Linear Synth</td>
<td>.970236</td>
</tr>
</tbody>
</table>
Figure 5.21. Case 2: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>( \ldots \ldots )</td>
</tr>
<tr>
<td>Cubic FEM</td>
<td>( - - - - )</td>
</tr>
<tr>
<td>Cubic Synth</td>
<td>_______________</td>
</tr>
</tbody>
</table>
Figure 5.22. Case 2: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions

<table>
<thead>
<tr>
<th>Method</th>
<th>λ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.969986</td>
</tr>
<tr>
<td>Cubic FEM</td>
<td>.966816</td>
</tr>
<tr>
<td>Cubic Synth</td>
<td>.969578</td>
</tr>
</tbody>
</table>
5.3.3. **Case 3:** Half-core reflected PWR composed of an 18-cm water reflector, the seven subassemblies C,C,C,A,A,A,D, and half of subassembly D. Zero flux boundary conditions are imposed outside the reflector, and symmetry is imposed in the center of the last D-type subassembly.

The Case 3 results of the two-group approximations using full 18-cm coarse mesh regions in all but the last 9-cm region are presented in Figures 5.23 - 5.26, and summarized in Table 5.8. The reference solutions were obtained using 198 mesh regions given by the symmetric partitioning

\[2(2 \text{ cm}) + 2(1 \text{ cm}) + 4(0.5 \text{ cm}) + 2(0.25 \text{ cm}) + 2(0.25 \text{ cm})\]

in each of the subassemblies, and 18 (1 cm) regions in the reflector.

The use of many subassemblies containing absorption rods throughout the reactor, except in the center subassemblies where water channels are present, results in central peaked fluxes with large gradients and, by comparison, a relatively small thermal neutron peak in the reflector.

Both coarse mesh methods were found to overestimate the flux in the subassemblies near the reflector, and underestimate the flux in the central subassembly regions regardless of the type of basis function approximations used. The larger inaccuracies of the linear basis function methods can be in part attributed to the fact that these methods cannot approximate the peaked thermal flux in the reflector, and result in large flux values in the subassemblies nearest the reflector. The cubic Hermite basis function approximations, however, are better able to approximate both the thermal flux reflector peak and the complex
<table>
<thead>
<tr>
<th>Method</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>.941640</td>
</tr>
<tr>
<td>Linear FEM</td>
<td>.931429</td>
</tr>
<tr>
<td>Linear Synth</td>
<td>.935873</td>
</tr>
</tbody>
</table>

Figure 5.23. Case 3: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions
Figure 5.24. Case 3: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions
Figure 5.25. Case 3: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions
Figure 5.26. Case 3: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions
Table 5.8. Results of Case 3.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>Linear FEM</th>
<th>Linear Synth</th>
<th>Cubic FEM</th>
<th>Cubic Synth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\lambda$</td>
<td>.941640</td>
<td>.931429</td>
<td>.938561</td>
<td>.935873</td>
<td>.928020</td>
</tr>
<tr>
<td>$%\lambda$</td>
<td>--</td>
<td>1.08%</td>
<td>.32%</td>
<td>.61%</td>
<td>1.44%</td>
</tr>
<tr>
<td>P(1)</td>
<td>.01699</td>
<td>-461.0%</td>
<td>-514.0%</td>
<td>-37.4%</td>
<td>-85.1%</td>
</tr>
<tr>
<td>P(2)</td>
<td>.02513</td>
<td>-192.0%</td>
<td>-179.0%</td>
<td>-32.8%</td>
<td>-98.6%</td>
</tr>
<tr>
<td>P(3)</td>
<td>.02890</td>
<td>-149.0%</td>
<td>-133.0%</td>
<td>-26.0%</td>
<td>-79.6%</td>
</tr>
<tr>
<td>P(4)</td>
<td>.0224</td>
<td>-50.2%</td>
<td>-36.9%</td>
<td>-13.5%</td>
<td>-31.5%</td>
</tr>
<tr>
<td>P(5)</td>
<td>.04969</td>
<td>58.52%</td>
<td>8.24%</td>
<td>-2.78%</td>
<td>-7.63%</td>
</tr>
<tr>
<td>P(6)</td>
<td>.1465</td>
<td>7.47%</td>
<td>13.7%</td>
<td>-4.94%</td>
<td>.259%</td>
</tr>
<tr>
<td>P(7)</td>
<td>.4362</td>
<td>26.4%</td>
<td>22.9%</td>
<td>5.01%</td>
<td>8.40%</td>
</tr>
<tr>
<td>P(8)</td>
<td>.2740</td>
<td>18.6%</td>
<td>20.1%</td>
<td>1.99%</td>
<td>13.1%</td>
</tr>
</tbody>
</table>

neutron leakage across the core, and give better results. Table 5.8 indicates that the cubic Hermite basis function approximations better approximate the detailed reference solutions, and that results obtained using the cubic Hermite finite element method were for this case better than those obtained using either of the proposed approximations. The ability of these methods to approximate large thermal flux peaks in the reflector regions is considered in the next case.
5.3.4. Case 4: Half-core reflected PWR composed of an 18-cm water reflector, the seven subassemblies D,D,D,C,D,D,A, and half of subassembly Type A. Zero flux boundary conditions are imposed in the center of the last Type A subassembly.

The Case 4 geometry produces a large but detailed thermal flux in the half-core region and a large thermal peak in the reflector region, as seen from the results in Figures 5.27 - 5.30. The reference solutions were calculated using the reference mesh geometry as given in Case 3. The results of the approximations are summarized in Table 5.9.

The results show that the linear basis function approximations cannot approximate accurately the thermal flux reflector peak and result in large flux and fractional power errors in the subassemblies near the reflector. The cubic Hermite basis function approximations, on the other hand, are better able to approximate this thermal peak and result in much more accurate power levels, especially in the first subassembly region.

The Case 4 results typify the approximation accuracy of both the finite element method and the proposed approximation method. In general, the cubic Hermite basis function approximations are superior to the linear basis function approximations, and the proposed methods give comparable or superior results as compared to those obtained from the finite element method using the same class of basis functions. In this case, the proposed method using cubic Hermite basis functions was able to estimate the reference eigenvalue within 0.04%, closely approximate the detailed reference flux solution to within a few percent at all spatial points, and result in fractional normalized power levels in each subassembly with less than 5% error.
Figure 5.27. Case 4: Two-Group Fast Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions
Figure 5.28. Case 4: Two-Group Thermal Results Using Linear Basis Function Approximations and 18-cm Coarse Mesh Regions
Figure 5.29. Case 4: Two-Group Fast Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions
Figure 5.30. Case 4: Two-Group Thermal Results Using Cubic Hermite Basis Function Approximations and 18-cm Coarse Mesh Regions
Table 5.9. Results of Case 4.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>Linear FEM</th>
<th>Linear Synth</th>
<th>Cubic FEM</th>
<th>Cubic Synth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>λ</td>
<td>.979108</td>
<td>.980618</td>
<td>.985849</td>
<td>.976896</td>
<td>.978689</td>
</tr>
<tr>
<td>%λ</td>
<td>--</td>
<td>-.15%</td>
<td>-.69%</td>
<td>.22%</td>
<td>.04%</td>
</tr>
<tr>
<td>P(1)</td>
<td>.098</td>
<td>-23.4%</td>
<td>-21.4%</td>
<td>-13.3%</td>
<td>5.54%</td>
</tr>
<tr>
<td>P(2)</td>
<td>.160</td>
<td>-1.53%</td>
<td>2.39%</td>
<td>-.893%</td>
<td>-1.08%</td>
</tr>
<tr>
<td>P(3)</td>
<td>.193</td>
<td>11.5%</td>
<td>7.11%</td>
<td>1.83%</td>
<td>-3.35%</td>
</tr>
<tr>
<td>P(4)</td>
<td>.211</td>
<td>-19.5%</td>
<td>-10.6%</td>
<td>-7.23%</td>
<td>-1.60%</td>
</tr>
<tr>
<td>P(5)</td>
<td>.168</td>
<td>17.4%</td>
<td>10.3%</td>
<td>3.86%</td>
<td>3.79%</td>
</tr>
<tr>
<td>P(6)</td>
<td>.118</td>
<td>10.6%</td>
<td>4.96%</td>
<td>5.97%</td>
<td>-1.01%</td>
</tr>
<tr>
<td>P(7)</td>
<td>.039</td>
<td>1.45%</td>
<td>3.44%</td>
<td>1.65%</td>
<td>3.25%</td>
</tr>
<tr>
<td>P(8)</td>
<td>.010</td>
<td>18.1%</td>
<td>11.8%</td>
<td>2.30%</td>
<td>-2.72%</td>
</tr>
</tbody>
</table>
Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1. Characteristics of the Proposed Approximation Methods

The use of detailed subassembly flux solutions or other a priori flux shapes directly in the spatial shape or trial function form of flux approximations in reactor physics has resulted in many coarse mesh approximation schemes which are classified in the broad area of overlapping multichannel synthesis. The proposed approximation methods are similar to existing synthesis methods of this kind, but are unique in that they reduce to conventional and well understood approximation methods in regions where little or no spatial flux information is given, or in completely homogeneous regions. In contrast, the overlapping synthesis methods proposed to date do not. This characteristic is especially important in calculations involving homogeneous regions, of which reflector regions are a prime example.

The proposed approximations are very similar to coarse mesh finite element method approximations in which detailed flux behavior has been used to flux-weight the nuclear constants in each region. The methods are conceptually different and become equivalent only when all of the coarse mesh regions are homogeneous.

The matrix equations resulting from the use of the proposed methods are identical in form to those resulting from the finite element method utilizing similar basis functions. In addition, the matrix
elements of the proposed methods are curiously different from those of the finite element methods using detailed flux-weighted nuclear constants. Although the spatial mass and stiffness matrices of the proposed methods for each group have been proven to be positive definite only for the case of Galerkin flux weighting, the use of adjoint weighting in all of the cases considered did not alter these properties. In addition, the proposed methods were found always to converge to a positive eigenvalue and to flux shapes which were everywhere positive.

The numerical results indicate that the proposed methods are able to predict accurate criticality or $k_{\text{eff}}$ measurements and regional power levels as well as to approximate the reference detailed flux shapes for each group with a high degree of accuracy. The results indicate that in general, use of the proposed methods results in superior criticality estimates over those obtained by the use of the finite element method with flux-weighted constants; this behavior was observed for each type of basis function approximation. Moreover, each of the proposed methods is in general vastly superior to its finite element method counterparts in approximating the actual detailed flux behavior and regional as well as total power levels.

Detailed flux behavior could be reintroduced into the results of the homogenized finite element methods by normalizing the detailed subassembly solutions in each coarse mesh region to match the power levels of the converged results in each region. The detailed solutions resulting from such a procedure would be discontinuous at the region boundaries and may, to some extent, exhibit the fine flux structure present in the results of the proposed methods. However, the results
are not expected to be as good as an approximation as those of the proposed methods, since the current coupling or diffusion approximation is not made until after the coarse mesh homogenization procedure.

6.2 Applicability and Limitations

Because the matrix forms of the equations which result from the use of the proposed methods are identical to those which result from the use of the finite element methods, the proposed approximations can be incorporated into existing finite element approximation schemes. Although additional integrations must be performed in the proposed methods, they can be reduced to sums of known products so that little additional computation time is required.

As in any coarse mesh approximation method, inaccurate results can occur when the coarse mesh region sizes chosen are too large. For a given region size, the accuracy of the results for any approximation scheme is unknown. The accuracy of the finite element methods is known to improve geometrically as the mesh size is decreased, resulting in a useful error criterion for the method. A disadvantage of the proposed methods is that no such error criterion has been developed. The inability to predict error estimates has always been a major drawback of synthesis techniques. However, the use of such methods, and use of the proposed methods, has been shown to be justified through proper physical insight and experience.
6.3. Recommendations for Future Work

Obviously the next step is the application of these proposed methods to two-dimensional diffusion problems. However, the one-dimensional problem still contains areas which may deserve closer attention. One such area is the examination of the matrix properties of both the finite element method and the proposed approximation methods which are necessary in order to guarantee convergence to a positive eigenvalue and an everywhere positive flux solution. Another area is the development of error criteria for the proposed methods. The close similarity between the proposed methods and the finite element methods may allow an extension or generalization of characteristics which hitherto have belonged only to the finite element methods.

The usefulness of the proposed methods depends on their applicability and accuracy in two- and three-dimensional diffusion problems. Just as the finite element approximations can be derived in two- or three-dimensions using variational modal-nodal techniques, so can the proposed methods for multidimensional problems. The proposed trial functions could be defined as continuous at mesh nodes, but may in general be discontinuous along mesh line interfaces. In order that the flux and current trial functions not be allowed to be discontinuous at identical spatial points, the current trial functions would then have to be defined as continuous across these interfaces. The use of the proposed class of trial function forms in the two-dimensional problem will raise the challenge of extending the spatial overlapping synthesis methods of this type to multidimensional reactor problems.
REFERENCES


BIOGRAPHICAL NOTE

The author is a native Californian. Born in San Francisco August 19, 1945, he was raised in Orange, and upon graduation from Orange High School in 1963, he received the Bank of America Award in Mathematics and the Orange County Industrial Education Association Award in Electronics.

He then attended the University of California at Berkeley, and supported by special as well as Regents' Scholarship awards, graduated in 1967 from the School of Engineering with a Bachelor of Science degree in Physics. He was elected to Tau Beta Pi, Phi Beta Kappa, and received a four-year letter in varsity gymnastics.

The author then enrolled at the Massachusetts Institute of Technology. Supported by an Atomic Energy Commission Special Fellowship in Nuclear Engineering, he received the Master of Science degree from the Department of Nuclear Engineering in 1969.

As a research associate in the Space-Time Kinetics Project in the same department, the author completed this work and obtained the degree of Doctor of Philosophy in 1972.

While at M.I.T. the author was active in many organizations, most notable of which was the M.I.T. Rugby Football Club.
APPENDICES
Appendix A

TABLE OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g$</td>
<td>Energy group index which runs from the highest to the lowest energy group as $g = 1$ to $G$.</td>
</tr>
<tr>
<td>$\phi_g(r)$</td>
<td>Scalar neutron flux in energy group $g$ (neutrons/cm$^2$·sec).</td>
</tr>
<tr>
<td>$J_g(r)$</td>
<td>Vector neutron current in energy group $g$ (neutrons/cm$^2$·sec).</td>
</tr>
<tr>
<td>$D_g(r)$</td>
<td>Diffusion coefficient for neutrons in energy group $g$ (cm).</td>
</tr>
<tr>
<td>$\Sigma_g(r)$</td>
<td>Macroscopic total removal cross section in energy group $g$ (cm$^{-1}$).</td>
</tr>
<tr>
<td>$\nu\Sigma_{fg}(r)$</td>
<td>Macroscopic fission-production cross section in energy group $g$ (cm$^{-1}$).</td>
</tr>
<tr>
<td>$\Sigma_{gg'}(r)$</td>
<td>Macroscopic transfer cross section from energy group $g'$ to energy group $g$ (cm$^{-1}$).</td>
</tr>
<tr>
<td>$\chi_g$</td>
<td>Fission spectrum yield in energy group $g$.</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>The eigenvalue or criticality of the diffusion problem.</td>
</tr>
<tr>
<td>$\Phi(r), \Phi^*(r)$</td>
<td>Scalar group flux column vector of length $G$ and its adjoint.</td>
</tr>
<tr>
<td>$\underline{J}(r), \underline{J}^*(r)$</td>
<td>Vector group current column vector of length $G$ and its adjoint.</td>
</tr>
</tbody>
</table>
\( \Lambda(r) \) \ G \times G \) group material removal, scattering, and production matrix.

\( D(r) \) \ \( G \times G \) diagonal group diffusion coefficient matrix.

\( U(r), U^*(r) \) Scalar group flux and weighting flux trial function column vectors of length \( G \).

\( V(r), V^*(r) \) Vector group current and weighting current trial function column vectors of length \( G \).

\( k \) One-dimensional spatial index which runs from the leftmost first region to the rightmost \( K \)-th region, as \( k = 1 \) to \( K \).

\( z \) The one-dimensional axis variable divided into \( K \) regions such that each region \( k \) is bounded by nodes \( z_k \) and \( z_{k+1} \).

\( x \) A dimensionless variable defined in each region \( k \) as \( x = (z - z_k)/(z_{k+1} - z_k) \), such that \( 0 \leq x \leq 1 \) as \( z_k \leq z \leq z_{k+1} \).

\( F_k \) Approximate one-dimensional group flux solution at node \( z_k \).

\( G_k \) Approximate one-dimensional group current solution at node \( z_k \).

\( \psi_k(z), \psi^*_k(z) \) Detailed one-dimensional subassembly flux and weighting flux solutions in coarse mesh region \( k \) whose form is linear within each homogeneous subassembly interval.

\( \eta_k(z), \eta^*_k(z) \) Detailed one-dimensional subassembly current and weighting current solutions in coarse mesh region \( k \) whose form is constant within each homogeneous subassembly interval.
\[ \tilde{\eta}_k(z), \quad \tilde{\eta}_k^w(z) \]

Detailed one-dimensional subassembly current and weighting current solutions in coarse mesh region \( k \) whose form is linear within each homogeneous subassembly interval.

\[ A \]
Discretized matrix form of the GXG group diffusion, absorption, and scattering matrices.

\[ B \]
Discretized matrix form of the GXG group fission-production matrix.

\( F \)
The unknown approximate group flux solution vector which may contain group current unknowns.

\[ \% \lambda \]
Normalized eigenvalue percent error:
\[ \% \lambda = \left( \lambda_{\text{Reference}} - \lambda_{\text{Method}} \right) / \lambda_{\text{Reference}} \times 100\% . \]

\( P(k) \)
Fractional power produced in coarse mesh region \( k \) when the total power produced has been normalized to unity.

\[ \% P(k) \]
Normalized fractional power percent error:
\[ \% P(k) = \left( P(k)_{\text{Reference}} - P(k)_{\text{Method}} \right) / P(k)_{\text{Reference}} \times 100\% . \]
Appendix B

DIFFERENCE EQUATION COEFFICIENTS RESULTING FROM USE OF THE FINITE ELEMENT APPROXIMATION METHODS

The $G \times G$ matrix coefficients resulting from the conventional finite difference approximation, the linear finite element approximation, and the cubic Hermite finite element approximation in one-dimensional multigroup diffusion theory are defined below in sections B.1, B.2, and B.3, respectively. The coefficients are given in terms of assumed homogeneous regional nuclear constants through the use of the $G \times G$ group matrices $D_k$ and $A_k$, where $A_k = M_k - T_k - \frac{1}{\lambda} B_k$, which are defined in Chapter 2 and are constant for each region $k$, where $k = 1$ to $K$.

More general definitions of these coefficients may be found from the coefficients resulting from the use of the proposed approximations, given in Appendix C, by requiring that $\psi_k(z)$ be constant and $\eta_k(z)$ be zero in each region $k$.

B.1. Coefficients of the Conventional Finite Difference Equations (as defined by Eqs. 2.16)

Interior Coefficients; $k = 2$ to $K$:

\[
\begin{align*}
    a_k &= - \frac{D_{k-1}}{h_{k-1}} \\
    b_k &= \frac{1}{2} \left( (A_{k-1} h_{k-1} + A_k h_k) + \frac{D_{k-1}}{h_{k-1}} + \frac{D_k}{h_k} \right) \\
    c_k &= - \frac{D_k}{h_k}
\end{align*}
\]
Symmetry Boundary Condition Coefficients:

\[ b_1 = \frac{1}{2} \Lambda_1 h_1 + D_1/h_1 \]
\[ c_1 = -D_1/h_1 \]
\[ a_{K+1} = -D_K/h_K \]
\[ b_{K+1} = \frac{1}{2} \Lambda_K h_K + D_K/h_K \]

B. 2. Coefficients of the Linear Finite Element Method Equations
(as defined by Eqs. 2.30)

Interior Coefficients; \( k = 2 \) to \( K \):

\[ a_k = \frac{1}{6} \Lambda_{k-1} h_{k-1} - D_{k-1}/h_{k-1} \]
\[ b_k = \frac{1}{3} \left[ \Lambda_{k-1} h_{k-1} + \Lambda_k h_k \right] + D_{k-1}/h_{k-1} + D_k/h_k \]
\[ c_k = \frac{1}{6} \Lambda_k h_k - D_k/h_k \]

Symmetry Boundary Condition Coefficients:

\[ b_1 = \frac{1}{3} \Lambda_1 h_1 + D_1/h_1 \]
\[ c_1 = \frac{1}{6} \Lambda_1 h_1 - D_1/h_1 \]
\[ a_{K+1} = \frac{1}{6} \Lambda_K h_K - D_K/h_K \]
\[ b_{K+1} = \frac{1}{3} \Lambda_K h_K + D_K/h_K \]
B.3. Coefficients of the Cubic Hermite Finite Element Method Equations (as defined by Eq. 2.33)

Interior Coefficients; \( k = 2 \) to \( K \):

\[
a_{1k} = \frac{9}{70} A_{k-1} h_{k-1} - \frac{6}{5} D_{k-1} / h_{k-1}
\]

\[
a_{2k} = \left( -\frac{13}{420} A_{k-1} h_{k-1}^2 D_{k-1}^{-1} + \frac{1}{10} \right) \theta
\]

\[
b_{1k} = \frac{13}{35} \left( A_{k-1} h_{k-1}^2 + A_{k-1} h_k \right) + \frac{6}{5} \left( D_{k-1} / h_{k-1} + D_k / h_k \right)
\]

\[
b_{2k} = \frac{11}{210} \left( A_{k-1} h_{k-1}^2 D_{k-1}^{-1} - A_k h_k D_k^{-1} \right) \theta
\]

\[
c_{1k} = \frac{9}{70} A_k h_k - \frac{6}{5} D_k / h_k
\]

\[
c_{2k} = \left( \frac{13}{420} A_k h_k^2 D_k^{-1} - \frac{1}{10} \right) \theta
\]

\[
a_{3k} = \left( \frac{13}{420} D_{k-1}^{-1} h_{k-1}^2 A_{k-1} - \frac{1}{10} \right) \theta
\]

\[
a_{4k} = \left( -\frac{1}{140} D_{k-1}^{-1} h_{k-1}^3 A_{k-1} D_{k-1}^{-1} - \frac{1}{30} h_{k-1} D_{k-1}^{-1} \right) \theta^2
\]

\[
b_{3k} = \frac{11}{210} \left( D_{k-1}^{-1} h_{k-1}^2 A_{k-1} - D_k^{-1} h_k^2 D_k \right) \theta
\]

\[
b_{4k} = \left[ \frac{1}{105} \left( D_{k-1}^{-1} h_{k-1}^3 A_{k-1} D_{k-1}^{-1} + D_k^{-1} h_k^3 A_k D_k^{-1} \right) + \frac{2}{15} \left( h_{k-1} D_{k-1}^{-1} + h_k D_k^{-1} \right) \right] \theta^2
\]

\[
c_{3k} = \left( -\frac{13}{420} D_k^{-1} h_k^2 A_k + \frac{1}{10} \right) \theta
\]

\[
c_{4k} = \left( -\frac{1}{140} D_k^{-1} h_k^3 A_k D_k^{-1} - \frac{1}{30} h_k D_k^{-1} \right) \theta^2
\]
Zero Flux Boundary Condition Coefficients:

\[ b_{41} = \left( \frac{1}{105} \, D_{1}^{-1} h_{1}^{3} A_{1} D_{1}^{-1} + \frac{2}{15} \, h_{1} D_{1}^{-1} \right) \theta^{2} \]

\[ c_{31} = \left( - \frac{13}{420} \, D_{1}^{-1} h_{1}^{2} A_{1} + \frac{1}{10} \right) \theta \]

\[ c_{41} = \left( - \frac{1}{140} \, D_{1}^{-1} h_{1}^{3} A_{1} D_{1}^{-1} - \frac{1}{30} \, h_{1} D_{1}^{-1} \right) \theta^{2} \]

\[ a_{3K+1} = \left( \frac{13}{420} \, D_{K}^{-1} h_{K}^{2} A_{K} - \frac{1}{10} \right) \theta \]

\[ a_{4K+1} = \left( - \frac{1}{140} \, D_{K}^{-1} h_{K}^{3} A_{K} D_{K}^{-1} - \frac{1}{30} \, h_{K} D_{K}^{-1} \right) \theta^{2} \]

\[ b_{4K+1} = \left( \frac{1}{105} \, D_{K}^{-1} h_{K}^{3} A_{K} D_{K}^{-1} + \frac{2}{15} \, h_{K} D_{K}^{-1} \right) \theta^{2} \]

Symmetry Boundary Condition Coefficients:

\[ b_{11} = \frac{13}{35} A_{1} h_{1} + \frac{6}{5} D_{1}/h_{1} \]

\[ c_{11} = \frac{9}{70} A_{1} h_{1} - \frac{6}{5} D_{1}/h_{1} \]

\[ c_{21} = \left( \frac{13}{420} A_{1} h_{1} D_{1}^{-1} - \frac{1}{10} \right) \theta \]

\[ a_{1K+1} = \frac{9}{70} A_{K} h_{K} - \frac{6}{5} D_{K}/h_{K} \]

\[ a_{2K+1} = \left( - \frac{13}{420} A_{K} h_{K} D_{K}^{-1} + \frac{1}{10} \right) \theta \]

\[ b_{1K+1} = \frac{13}{35} A_{K} h_{K} + \frac{6}{5} D_{K}/h_{K} \]
Appendix C

DIFFERENCE EQUATION COEFFICIENTS RESULTING FROM USE OF THE PROPOSED APPROXIMATION METHODS

The G\times G matrix coefficients resulting from the proposed approximation methods using (1) linear basis functions, and (2) cubic Hermite basis functions, in one-dimensional multigroup diffusion theory are defined below in sections C.1 and C.2, respectively.

The coefficients are given as integrands of functions of x where the integration of every coefficient-integrand over a region

\[
\text{Coefficient} = \int_0^1 \left[ \text{Coefficient-Integrand}(x) \right] dx
\]

is understood.

In order to simplify the forms of the coefficient-integrands, it is convenient to define the following G\times G matrices:

\[
\begin{align*}
\mathbf{I}_k(x) &= \psi_k^T(x) \mathbf{A}_k(x) h_k \psi_k(x) \\
\mathbf{II}_k(x) &= \eta_k^T(x) \mathbf{D}_k^{-1}(x) h_k \eta_k(x) \\
\mathbf{IP}_k(x) &= \psi_k^T(x) \eta_k(x) \\
\mathbf{IQ}_k(x) &= \eta_k^T(x) \psi_k(x) \\
\mathbf{IR}_k(x) &= \frac{1}{h_k} \psi_k^T(x) \mathbf{D}_k(x) \psi_k(x)
\end{align*}
\]

for each region k. In each approximation below, two sets of polynomial functions \(p_1(x) \ldots p_{2N}(x)\) and \(q_1(x) \ldots q_{2N}(x)\) are given
which represent the basis functions of the approximation and their negative derivatives, where $N = 1$ for the linear basis function approximations and $N = 2$ for the cubic Hermite basis function approximations.

The $G \times G$ coefficient-integrands are then listed in terms of these matrices and polynomials by the $G \times G$ collapsed matrices $\mathbb{E}_{k}^{i,j}(x)$ defined as

$$
\mathbb{E}_{k}^{i,j}(x) = p_{i}(x)p_{j}(x)\mathbb{I}_{k}(x) - p_{i}(x)p_{j}(x)\mathbb{L}_{k}(x) + q_{i}(x)p_{j}(x)\mathbb{I}_{k}(x) - p_{i}(x)q_{j}(x)\mathbb{Q}_{k}(x) + q_{i}(x)q_{j}(x)\mathbb{R}_{k}(x)
$$

for given values of $i$ and $j$ for each region $k$, where $k = 1$ to $K$. It should be noted that $\mathbb{E}_{k}^{i,j}(x)$ is not symmetric about $i$ and $j$; i.e.:

$$
\mathbb{E}_{k}^{i,j}(x) \neq \mathbb{E}_{k}^{j,i}(x), \text{ for } i \neq j.
$$

C.1. Coefficient-Integrands of the Proposed Approximation Method Equations Using Linear Basis Functions
(as defined by Eq. 3.6)

These coefficients are given in terms of the polynomial functions

$$
p_{1}(x) = (1-x)
$$

$$
p_{2}(x) = x
$$

$$
q_{1}(x) = 1
$$

$$
q_{2}(x) = -1
$$

for use in the $\mathbb{E}_{k}^{i,j}(x)$ below.
Interior Coefficient-Integrands; \( k = 2 \) to \( K \):

\[
a_k(x) = \psi_k^*(1) \mathbb{E}_{k-1}^2,1(x) \psi_{k-1}^{-1}(0)
\]

\[
b_k(x) = \psi_k^*(1) \mathbb{E}_{k-1}^2,2(x) \psi_{k-1}^{-1}(1) + \psi_k^*(0) \mathbb{E}_{k}^1,1(x) \psi_{k}^{-1}(0)
\]

\[
c_k(x) = \psi_k^*(0) \mathbb{E}_{k}^1,2(x) \psi_{k}^{-1}(1)
\]

Symmetry Coefficient-Integrands:

\[
b_1(x) = \psi_1^*(0) \mathbb{E}_{1}^1,1(x) \psi_{1}^{-1}(0)
\]

\[
c_1(x) = c_k(x); \text{ where } k = 1
\]

\[
a_{K+1}(x) = a_k(x); \text{ where } k = K + 1
\]

\[
b_{K+1}(x) = \psi_{K}^*(1) \mathbb{E}_{K-1}^2,2(x) \psi_{K}^{-1}(1)
\]

Implied Zero Flux Boundary Condition Coefficient-Integrands

(Corresponding with the modified trial functions of the type in Eqs. 3.9)

\[
b_2(x) = \psi_1^*(1) \[ \mathbb{K}_1(x) - \mathbb{L}_1(x) \] \psi_{1}^{-1}(1) + \psi_2^*(0) \mathbb{E}_{2}^1,1(x) \psi_{2}^{-1}(0)
\]

\[
b_K(x) = \psi_{K-1}^*(1) \mathbb{E}_{K-1}^2,2(x) \psi_{K-1}^{-1}(1) + \psi_K^*(0) \[ \mathbb{K}_K - \mathbb{L}_K \] \psi_{K}^{-1}(0)
\]
C.2. **Coefficient-Integrands of the Proposed Approximation Method**

**Equations Using Cubic Hermite Basis Functions** (as defined in Eq. 3.16)

These coefficients are given in terms of the polynomials $p_1(x)$ through $p_4(x)$ and $q_1(x)$ through $q_4(x)$, previously defined in Eqs. 3.11 and 3.12, for use in the $I_k^{i,j}(x)$ below.

**Interior Coefficient-Integrands; $k = 2$ to $K$:**

- $a_{1_k}(x) = \psi_k^{-1} \mathcal{I}_k^{2,1}(x) \psi_k^{-1}(0)$

- $a_{2_k}(x) = \psi_k^{-1} \mathcal{I}_k^{2,3}(x) \psi_k^{-1}(0) \mathcal{D}_k^{-1}(x)$

- $b_{1_k}(x) = \psi_k^{-1} \mathcal{I}_k^{2,2}(x) \psi_k^{-1}(1) + \psi_k^{-1} \mathcal{I}_k^{1,1}(x) \psi_k^{-1}(0) \mathcal{D}_k^{-1}(x)$

- $b_{2_k}(x) = \psi_k^{-1} \mathcal{I}_k^{2,4}(x) \psi_k^{-1}(1) + \psi_k^{-1} \mathcal{I}_k^{1,3}(x) \psi_k^{-1}(0) \mathcal{D}_k^{-1}(x)$

- $c_{1_k}(x) = \psi_k^{-1} \mathcal{I}_k^{1,2}(x) \psi_k^{-1}(1)$

- $c_{2_k}(x) = \psi_k^{-1} \mathcal{I}_k^{1,4}(x) \psi_k^{-1}(1) \mathcal{D}_k^{-1}(x)$

- $a_{3_k}(x) = \psi_k^{-1} \mathcal{I}_k^{4,1}(x) \psi_k^{-1}(0) \mathcal{D}_k^{-1}(x)$

- $a_{4_k}(x) = \psi_k^{-1} \mathcal{I}_k^{4,3}(x) \psi_k^{-1}(0) \mathcal{D}_k^{-1}(x)$
\[ b_{3, k}(x) = \psi_k^{*T}(1) D_{k-1}^{-1}(1) W_k^{-1}(x) \psi_{k-1}^{-1}(1) \theta \]

\[ + \psi_k^{*T}(0) D_k^{-1}(0) W_k^{3, 1}(x) \psi_k^{-1}(0) \theta \]

\[ b_{4, k}(x) = \psi_k^{*T}(1) D_{k-1}^{-1}(1) W_k^{4, 4}(x) \psi_{k-1}^{-1}(1) D_{k-1}^{-1}(1) \theta^2 \]

\[ + \psi_k^{*T}(0) D_k^{-1}(0) W_k^{3, 3}(x) \psi_k^{-1}(0) D_k^{-1}(0) \theta^2 \]

\[ c_{3, k}(x) = \psi_k^{*T}(0) D_k^{-1}(0) W_k^{3, 2}(x) \psi_k^{-1}(1) \theta \]

\[ c_{4, k}(x) = \psi_k^{*T}(0) D_k^{-1}(0) W_k^{3, 4}(x) \psi_k^{-1}(1) D_k^{-1}(1) \theta^2 \]

Zero Flux Boundary Condition Coefficient-Integrands:

\[ b_{4, 1}(x) = \psi_1^{*T}(0) D_1^{-1}(0) W_1^{3, 3}(x) \psi_1^{-1}(0) D_1^{-1}(0) \theta^2 \]

\[ c_{3, 1}(x) = c_{3, k}(x); \text{ where } k = 1 \]

\[ c_{4, 1}(x) = c_{4, k}(x); \text{ where } k = 1 \]

\[ a_{3, k+1}(x) = a_{3, k}(x); \text{ where } k = K + 1 \]

\[ a_{4, K+1}(x) = a_{4, K+1}(x); \text{ where } k = K + 1 \]

\[ b_{4, K+1}(x) = \psi_K^{*T}(1) D_K^{-1}(1) W_K^{4, 4}(x) \psi_K^{-1}(1) D_K^{-1}(1) \theta^2 \]
Symmetry Boundary Condition Coefficient-Integrands:

\[
\begin{align*}
-1 \\
b_{11} (x) &= \psi_1^T (0) E_{11}^{-1,1}(x) \psi_1^{-1}(0) \\
c_{11} (x) &= c_{1k} (x); \text{ where } k = 1 \\
c_{21} (x) &= c_{2k} (x); \text{ where } k = 1 \\
a_{1K+1} (x) &= a_{1k} (x); \text{ where } k = K+1 \\
a_{2K+1} (x) &= a_{2k} (x); \text{ where } k = K+1 \\
-1 \\
b_{1K+1} (x) &= \psi_K^T (1) E_{K1}^{2,2}(x) \psi_K^{-1}(1) \\
\end{align*}
\]

Implied Zero Flux Boundary Condition Coefficient-Integrands (corresponding with the modified trial functions of the type in Eq. 3.17):

\[
\begin{align*}
\frac{c_{31}}{c_{3k}} (x); \text{ where } k = 1 \\
\frac{a_{22}}{a_{2k}} (x); \text{ where } k = 2 \\
\frac{b_{12}}{b_{1k}} (x); \text{ where } k = 2 \\
\frac{b_{22}}{b_{2k}} (x); \text{ where } k = 2 \\
\frac{b_{32}}{b_{3k}} (x); \text{ where } k = 2 \\
\frac{b_{1K}}{b_{1k}} (x); \text{ where } k = K \\
\frac{b_{2K}}{b_{2k}} (x); \text{ where } k = K \\
\frac{c_{2K}}{c_{2k}} (x); \text{ where } k = K \\
\frac{b_{3K}}{b_{3k}} (x); \text{ where } k = K \\
\frac{a_{3K+1}}{a_{3k}} (x); \text{ where } k = K+1 \\
\end{align*}
\]

\[
\begin{align*}
\begin{cases}
p_1 (x) = 0 \\
p_2 (x) = 1 \\
q_1 (x) = 0 \\
q_2 (x) = 0 \\
p_1 (x) = 1 \\
p_2 (x) = 0 \\
q_1 (x) = 0 \\
q_2 (x) = 0 \\
\end{cases}
\end{align*}
\]
Appendix D

DESCRIPTION OF THE COMPUTER PROGRAMS

The computer programs REF2G, LINEAR, CUBIC, and ANALYZE are described respectively in the following four sections. The programs are written in FORTRAN IV, allow double precision calculations, and were used with the I.B.M. 360/65 and 370/155 FORTRAN G compilers at the M.I.T. Information Processing Center. Sample storage requirements and execution times of the programs are summarized in Table D.1.

The power method employed in the first three programs allows a maximum of 300 iterations to converge, and program execution continues after this limit. Initial group flux shapes are sinusoidal or flat, depending upon the boundary conditions chosen.

The input and output data of each program are divided into data blocks for ease of representation as described below.

D.1. Description of Program REF2G

REF2G finds the reference solutions of the one-dimensional, two-group diffusion equations of each case study, or the detailed sub-assembly solutions of each subassembly, using the linear finite element approximation method. The program allows up to a total of two hundred homogeneous fine mesh regions and employs combinations of both zero flux and symmetry boundary conditions. Identical material regions can be automatically repeated with no additional input.
Table D.1. Sample Storage Requirements and Execution Times of the Programs for Two-Group Results. Obtained using the M.I.T. I.B.M. 360/155.

<table>
<thead>
<tr>
<th>Storage Requirements in Bytes (without overlays):</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF2G: 260 K</td>
</tr>
<tr>
<td>LINEAR: 200 K</td>
</tr>
<tr>
<td>CUBIC: 250 K</td>
</tr>
<tr>
<td>ANALYZE: 205 K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C.P.U. Execution Times in Minutes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF2G: Detailed Subassembly Solutions (68 regions):</td>
</tr>
<tr>
<td>Case 1 Reference Solution (150 regions): 0.120</td>
</tr>
<tr>
<td>Case 4 Reference Solution (198 regions): 0.238</td>
</tr>
<tr>
<td>LINEAR: Case 1 Synthesis (Homogenized Method) (3 regions): 0.284 (0.157)</td>
</tr>
<tr>
<td>Case 4 Synthesis (Homogenized Method) (9 regions): 0.296 (0.209)</td>
</tr>
<tr>
<td>CUBIC: Case 1 Synthesis (Homogenized Method) (3 regions): 1.227 (0.183)</td>
</tr>
<tr>
<td>Case 4 Synthesis (Homogenized Method) (9 regions): 1.464 (0.328)</td>
</tr>
<tr>
<td>ANALYZE: Case 1 Linear (Cubic Hermite) Basis Functions: 0.087 (0.108)</td>
</tr>
<tr>
<td>Case 4 Linear (Cubic Hermite) Basis Functions: 0.122 (0.139)</td>
</tr>
</tbody>
</table>

---

a. Including adjoint flux and current calculations.
b. Not including adjoint calculations.
c. Including 0.126 minutes for calculation of the two-group homogenized constants.
Options for plotting graphically the history of the converging spatial flux as well as the converging eigenvalue are also available. In addition, the program allows the calculation of the adjoint flux and current solutions.

The approximate current solutions are linear within each mesh region and are calculated from the converged flux solutions using Eqs. 4.29 and 4.31. The converged flux and current solutions and the converged adjoint solutions can be punched out for future use as described below.

A. Reference Solution Input Block

Card Type 1: Format (20A4)

An Appropriate Problem Title

Card Type 2: Format (2I5, 3E10.3, 5I5)

KR  Total number of homogeneous fine mesh regions.  KR ≤ 200.

IBC  Boundary Condition Option

1. Zero flux on both boundaries
2. Zero flux on the left, symmetry on the right
3. Symmetry on the left, zero flux on the right
4. Symmetry on both boundaries

EPS1  Iteration tolerance to be met by differences between elements of successive iteration solution vectors:

\[ |F_j^{(i)} - F_j^{(i-1)}| < \epsilon_1; \text{ for all } j \]

EPS2  Iteration tolerance to be met by the mean square error between successive iteration solution vectors:

\[ \left\{ \sum_j \left[ F_j^{(i)} - F_j^{(i-1)} \right]^2 \right\}^{\frac{1}{2}} < \epsilon_2 \]
EPS3  Iteration tolerance to be met by the difference between successive iteration eigenvalues:
\[ |\lambda^{(i)} - \lambda^{(i-1)}| < \epsilon_3 \]

I PLOT  Allows printed graphical display of the converging flux solution:
0  No display
1  Plot only the resultant normalized flux
2  Plot a normalized history of the converging flux

J PLOT  Allows printed graphical display of the history of the converging eigenvalue when J PLOT = 1.

I PUNCH  Allows punched output when I PUNCH = 1.

I SEE  Allows printing of storage information:
0  No information printed
1  Input regional properties are printed
2  Input regional properties as well as the Common/B5/ storage arrays and the Common/B3/ power method matrices are printed.

N OADJ  Adjoint calculations are performed when NOADJ = 0, and bypassed if NOADJ = 1.

Card Type 3: Format (25I2)

ITF(k)  The consecutive type-number of each region from left to right as k = 1 to KR. Allows for repeating identical regions with no additional input.

Card Type 3 is repeated KR/25 times (rounded off to the next highest integer).

Card Type 4: Format (2F10.5)

CHI(1), CHI(2)  The fission yields \( \chi_1 \) and \( \chi_2 \) for the fast and thermal groups, respectively.
* An Input Region Data Block:
  Repeated for each different material region; max \([\text{ITF}(k)]\) times.

Card Type 5:  Format (I5)

  \(k\)  The consecutive mesh region number (counting from from left to right) for identification purposes.

Card Types 6, 7:  Format (3F10.5, 4E10.3, /, 30X, 3E10.3)

The geometry and nuclear constants for region \(k\):

- \(z(1)\): Beginning spatial coordinate of region \(k\) (cm)
- \(z(2)\): Ending spatial coordinate of region \(k\) (cm)
- \(H\): Width of region \(k\) (cm)
- \(A(1)\): Fast-group macroscopic total cross section in region \(k\) (cm\(^{-1}\))
- \(F(1)\): Fast-group macroscopic production cross section, \(\nu\Sigma_f\), in region \(k\) (cm\(^{-1}\))
- \(D(1)\): Fast-group diffusion coefficient in region \(k\) (cm)
- \(S\): Fast-to-thermal macroscopic scattering cross section in region \(k\) (cm\(^{-1}\))
- \(A(2)\): Thermal-group macroscopic total cross section in region \(k\) (cm\(^{-1}\))
- \(F(2)\): Thermal-group macroscopic production cross section, \(\nu\Sigma_f\), in region \(k\) (cm\(^{-1}\))
- \(D(2)\): Thermal-group diffusion coefficient in region \(k\) (cm)

* End of an Input Region Data Block.

** Power Method Input Block: Optional

Card Type 8:  Format (F10.5)

  \(\omega\): Outer iteration overrelaxation parameter \(1 \leq \omega \leq 2\). Default is \(\omega = 1.25\).

Card Type 9:  Format (D25.14)

  \(\lambda^{(0)}\): Initial eigenvalue guess. Default is \(\lambda^{(0)} = 1.0\).
Card Type 10: Format (4E20.10)

\[ ((F(g,i), i = 1 \text{ to } N), g = 1 \text{ to } 2) \]

Initial group flux solution guess without zero flux boundary values. Default is \( F = 1.0 \).

** End of Power Method Input Block.**

B. Reference Solution Output Block

When IPUNCH = 1, REF2G punches out the number of fine mesh regions, KR, under Format (I5) followed by the converged flux solutions \( \psi(g,k) \) and corresponding current solutions \( \tilde{\eta}(g,k) \) for each group \( g \) and spatial node \( k \) including boundary conditions. When adjoint calculations are included, the results are punched out under Format (4D20.10) as

\[ ((\psi(g,k), \tilde{\eta}(g,k), \psi^*(g,k), \tilde{\eta}^*(g,k), k = 1 \text{ to } KR + 1), g = 1 \text{ to } 2) \]

where the notation denotes case reference solutions as well as detailed subassembly solutions. When the adjoint calculations have been bypassed, the results are punched out under Format (2D20.10) as

\[ ((\psi(g,k), \tilde{\eta}(g,k), k = 1 \text{ to } KR + 1), g = 1 \text{ to } 2) \]

A total of 2 \( KR + 3 \) cards are punched out.

D.2. Description of Program LINEAR

Program LINEAR forms and solves the difference equations resulting from the proposed approximation method using the linear basis functions. The program allows up to twenty-five coarse mesh regions, each of which is allowed to be broken into not more than one hundred homogeneous intervals. Combinations of both zero flux and
symmetry boundary conditions as well as use of the modified trial function forms in the boundary regions are allowed. Spatial flux and eigenvalue iteration history plots are also available.

The program allows a choice of the type of weighting, Galerkin or adjoint, to be used in the approximation. Also, either form of the detailed subassembly current solutions $\eta_k(x)$ or $\tilde{\eta}_k(x)$ is allowed. In addition, identical coarse mesh regions with identical detailed subassembly solutions can be repeated implicitly.

LINEAR also calculates results of the linear finite element method when suitable input is used. Such results can be obtained by using homogenized coarse mesh region nuclear constants and defining the detailed group flux solutions to be constant and the detailed currents to be zero (or by setting $ITC = 0$).

Punched results using detailed subassembly solutions constitute a Synthesis Method Output Block, while punched output resulting from the reduction to the finite element method with homogenized regional constants constitutes a Homogenized Method Output Block.

A. Homogenized or Synthesis Method Input Block

Undefined input parameters are identical to those previously defined in the REF2G input.

Card Type 1: Format (20A4)

An Appropriate Problem Title

Card Type 2: Format (215, 3E10, 3, 6I5)

$KR$ Total number of coarse mesh regions. $KR \leq 25.$
IBC  1 - 4  As previously defined
      5  Modified trial function (no tilting) in the first region, symmetry on the right
      6  Zero flux on the left, modified trial function in the last region
      7  Modified trial functions in both boundary regions

EPS1
EPS2
EPS3
IPLOT
JPLOT
IPUNCH
ISEE

ITW  Type of approximation weighting desired:
      0  Flux (Galerkin)
      1  Adjoint

ITC  Form of the detailed current solutions in all sub-assemblies:
      0  $\eta_k(x)$, $\eta_k^*(x)$ as calculated by Fick's laws
      1  $\overset{\sim}{\eta}_k(x)$, $\overset{\sim}{\eta}_k(x)$ as given from REF2G output

Card Type 3:  Format (2512)

ITF(k)  The consecutive type-number of each coarse mesh region from left to right as $k = 1$ to KR. Allows for repeating identical subassemblies with no additional input.

Card Type 3 is repeated KR/25 times (rounded off to the next highest integer).

Card Type 4:  Format (2F10.5)

CHI(1), CHI(2)
* An Input Subassembly Region Data Block:
Repeated for each different coarse mesh region; max \( \text{ITF}(k) \) times.

**Card Type 5:** Format (2I5)
- \( k \) The consecutive coarse mesh region number (from left to right).
- \( N \) The number of homogeneous intervals in subassembly \( k \). \( N \leq 100 \).

**Card Types 6, 7:** Format (3F10.5, 4E10.3, /, 30X, 3E10.3)
The subassembly geometry and nuclear constants within each interval corresponding to the detailed subassembly solutions.
Repeated for each interval as \( i = 1 \) to \( N \):
- \( z(i) \) Beginning spatial coordinate of interval \( i \) (cm)
- \( z(i+1) \) Ending spatial coordinate of interval \( i \) (cm)
- \( H(i) \) Width of interval \( i \) (cm)
- \( A(1,i) \) Fast-group macroscopic total cross section in interval \( i \) (cm\(^{-1}\))
- \( F(1,i) \) Fast-group macroscopic production cross section, \( \nu \Sigma_f \), in interval \( i \) (cm\(^{-1}\))
- \( D(1,i) \) Fast-group diffusion coefficient in interval \( i \) (cm)
- \( S(i) \) Fast-to-thermal macroscopic scattering cross section in interval \( i \) (cm\(^{-1}\))
- \( A(2,i) \) Thermal-group macroscopic total cross section in interval \( i \) (cm\(^{-1}\))
- \( F(2,i) \) Thermal-group macroscopic production cross section, \( \nu \Sigma_f \), in interval \( i \) (cm\(^{-1}\))
- \( D(2,i) \) Thermal-group diffusion coefficient in interval \( i \) (cm)
Card Type 8: Format (4D20, 10)

The detailed subassembly solutions.

\((\psi(g, k), \tilde{\eta}(g, k), \psi^*(g, k), \tilde{\eta}^*(g, k), k = 1 \text{ to } KR + 1, g = 1 \text{ to } 2)\)

A subassembly's Reference Solution Output Block without the first card.

* END of an Input Subassembly Region Data Block.

** Expected Solution Input Block: Optional

Card Type 9: Format (D25, 14)

\(\lambda_{\text{REF}}\) Expected eigenvalue solution. Default is \(\lambda_{\text{REF}} = 1.0\).

Card Type 10: Format (4E20, 10)

\((F(i, g), i = 1 \text{ to } N), g = 1 \text{ to } 2)\) Expected group flux solution without zero flux boundary values. Default is \(F = 1.0\).

** END of the Expected Solution Input Block.

*** Power Method Input Block: Optional

As previously defined in the REF2G input.

*** END of the Power Method Input Block.

B. Homogenized or Synthesis Method Output Block

When IPUNCH = 1, LINEAR punches out the total number of coarse mesh regions, KR, under Format (I5) followed by the resultant flux solutions including boundary conditions. The flux solutions are punched out under Format (2E20, 7) as

\((F(1, k), F(2, k), k = 1 \text{ to } KR + 1)\)

These cards represent either a Homogenized or Synthesis Method Output Block, depending upon the type and form of input data used.
D.3 Description of Program CUBIC

Program CUBIC forms and solves the difference equations resulting from the proposed approximation method using the cubic Hermite basis functions. The program is very similar in form to program LINEAR and uses similar input.

A. Homogenized or Synthesis Method Input Block

The input to CUBIC is identical to that of LINEAR except for the following:

1. The boundary condition options are restricted by $1 \leq IBC \leq 4$.

2. The normalization constant $\theta$ can be included on Card Type 4 after CHI(2) under Format (3F10.5). Default is $\theta = 1.0$.

3. Both the expected group solutions and the initial group solutions of the Expected Solution and Power Method Input Blocks, respectively, are of the form $((F(g,i), i=1 \text{ to } N), g=1 \text{ to } 2)$ without either zero flux or zero current (or symmetry) boundary conditions. The solution vector is made up of alternating flux and current values as described in section 3.3 of Chapter 3. Default values are flux values of unity and current values of zero.

B. Homogenized or Synthesis Method Output Block

When IPUNCH = 1, CUBIC punches out the total number of coarse mesh regions, KR, under Format (15) followed by the resultant flux and current solutions including boundary conditions. The solutions are punched out under Format (4E20.7) as

$$(F(1,k), F(2,k), G(1,k), G(2,k), k = 1 \text{ to } KR+1)$$
where $F(g,k)$ represents the flux, and $G(g,k)$ the current solution of group $g$ at node $k$.

As in the case of LINEAR, these KR+2 output cards represent either a Synthesis or Homogenized Method Output Block, depending upon the type and form of input data used.

D.4 Description of Program ANALYZE

ANALYZE compares the results of the reference solution, homogenized finite element method, and the proposed synthesis method for each case study where either linear or cubic Hermite basis functions have been used in the latter methods. For each of these three methods, the program first forms the complete detailed flux solution and then normalizes the flux distributions for each method such that their total power levels are unity. The fractional (normalized) power levels produced in each coarse mesh region are then calculated, compared, and listed. Finally, the detailed group fluxes of each method are plotted graphically relative to one another using the Stromberg-Carlson Computer Recorder, SC-4020, facility at M.I.T.54 The graphic results for each group are normalized by the largest group-flux value such that the equivalent total power levels are preserved.

A. ANALYZE Input

The input to ANALYZE is read from five device units: 1, 2, 3, 11, 12, 13, and 5. Input and output data of the reference and approximation programs are read from the former six units while the standard input unit, 5, is reserved for SC-4020 plotting information.
The input is described by "Header Cards" and previously defined Input and Output Blocks. Header cards consist of one or more cards defined as follows:

**Header Card 1:** Format (4I5)

- **Method**: Indicates the type of basis function approximation:
  - 1  Linear
  - 2  Cubic Hermite

- **NK**: Total number of coarse mesh regions involved.

- **NR**: Total number of fine mesh regions involved. NR = NK except for reference solution calculations.

- **NAP**: Number of additional points to be plotted within each coarse mesh region. Used with the homogenized finite element method calculations. NAP < 0 denotes that the additional points are to be used in the first region (reflector) only.

**Header Card 2:** For use in device unit 3 input when NR ≠ NK. Format (6I5).

- **NRNK(k)**: The number of fine mesh regions which make up each coarse mesh region k, as k = 1 to NK.

The program is dimensioned to accept up to 200 fine mesh regions (or intervals) per coarse mesh region, up to 25 coarse mesh regions, and up to a grand total of 1000 fine mesh regions in each case study.

The form of the ANALYZE input is given as follows:
Input Data for Unit 1:

Header Cards

Homogenized Method Input Block

Input Data for Unit 2:

Header Cards

Synthesis Method Input Block

Input Data for Unit 3:

Header Cards

Reference Solution Input Block

Input Data for Unit 11:

Homogenized Method Output Block

Input Data for Unit 12:

Synthesis Method Output Block

Input Data for Unit 13:

Reference Solution Output Block

Input Data for Unit 5:

No SC-4020 plots are generated if this data is omitted.

Card 1: Format (20A4)

An appropriate title written above each plotted graph.
Card 2: Format (2F10.5)

XINCH: Total width of the graph in inches including labels (limited to 7.45"").

YINCH: Total height of the graph in inches including labels (limited to 7.45"").

Card 3: Format (I10, F10.5)

NCELL: Total number of coarse mesh regions. NCELL ≤ 25. (NCELL < 0 indicates that the last region is of width \( \frac{1}{2} \) WCELL.)

WCELL: Width of each coarse mesh region in cm.

Card 4: Optional. Format (I10, 7F10.5)

NLL: Number of vertical light lines to be added to the plotted graphs. NLL ≤ 100.

XL(i): Spatial location (cm) of the light lines; i = 1 to 7.

Card 5: Format (8F10.5)

XL(i): As above when NLL > 7.
Appendix E

SAMPLE INPUT AND OUTPUT DATA BLOCKS
FOR PROGRAMS REF2G, LINEAR, CUBIC, AND ANALYZE

(Included in only the first six copies of this report.)
E.1. REF2G SAMPLE INPUT AND OUTPUT DATA BLOCKS

SAMPLE REF2G REFERENCE SOLUTION INPUT BLOCK:

CASE 1 STUDY: THREE DIFFERENT SUBASSEMBLIES. 150 FINE MESH REFERENCE SOLUTION.
150 4 1.E-5 1.E-5 1.E-8 1 1 1 0 1
1 1 1 1 1 2 2 2 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5
5 5 5 5 5 5 5 4 4 4 4 4 3 3 3 3 2 2 2 2 1 1 1 1
6 6 6 6 6 7 7 7 8 8 8 8 9 9 9 9 9 101010101010101010
1010101010101010 9 9 9 9 8 8 8 8 7 7 7 6 6 6 6
111111111111212121213131314141414141515151515151515151515
1515151515151515151515151515151515151515151515151515151515151515
1.0 0.0
1 1
0.0 1.0 1.0 2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2
5.32 D-2 6.36 D-2 3.88 D-1
6 1
0.0 0.5 0.5 2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2
5.32 D-2 6.36 D-2 3.88 D-1
10 1
0.0 0.25 0.25 2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2
5.32 D-2 6.36 D-2 3.88 D-1
14 1
0.0 0.125 0.125 2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2
5.32 D-2 6.36 D-2 3.88 D-1
18 1
0.0 0.0625 0.0625 4.52 D-2 0.0 D 0 1.0 D 0 0.0 D 0
9.59 D-1 0.0 D 0 1.0 D 0
51 1
0.0 1.0 1.0 2.60 D-2 5.53 D-3 1.397 D 0 1.72 D-2
7.10 D-2 1.02 D-1 3.89 D-1
56 1
0.0 0.5 0.5 2.60 D-2 5.53 D-3 1.397 D 0 1.72 D-2
7.10 D-2 1.02 D-1 3.89 D-1

PAGE 169
| 60 | 1 | 0.0 | 0.25 | 0.25 | 2.60 D-2 5.53 D-3 1.397 D 0 1.72 D-2 7.10 D-2 1.02 D-1 3.89 D-1 |
| 64 | 1 | 0.0 | 0.125 | 0.125 | 2.60 D-2 5.53 D-3 1.397 D 0 1.72 D-2 7.10 D-2 1.02 D-1 3.89 D-1 |
| 68 | 1 | 0.0 | 0.0625 | 0.0625 | 4.52 D-2 0.0 D 0 1.0 D 0 0.0 D 0 9.59 D-1 0.0 D 0 1.0 D 0 |
| 101 | 1 | 0.0 | 1.0 | 1.0 | 2.61 D-2 6.59 D-3 1.399 D 0 1.68 D-2 8.32 D-2 1.29 D-1 3.87 D-1 |
| 106 | 1 | 0.0 | 0.5 | 0.5 | 2.61 D-2 6.59 D-3 1.399 D 0 1.68 D-2 8.32 D-2 1.29 D-1 3.87 D-1 |
| 110 | 1 | 0.0 | 0.25 | 0.25 | 2.61 D-2 6.59 D-3 1.399 D 0 1.68 D-2 8.32 D-2 1.29 D-1 3.87 D-1 |
| 114 | 1 | 0.0 | 0.125 | 0.125 | 2.61 D-2 6.59 D-3 1.399 D 0 1.68 D-2 8.32 D-2 1.29 D-1 3.87 D-1 |
| 118 | 1 | 0.0 | 0.0625 | 0.0625 | 4.52 D-2 0.0 D 0 1.0 D 0 0.0 D 0 9.59 D-1 0.0 D 0 1.0 D 0 |

PAGE 170
SAMPLE REF2G SUBASSEMBLY SOLUTION INPUT BLOCK:

<table>
<thead>
<tr>
<th>CELL A TWO GROUP CELL SOLUTION: FUEL A + CRUCIFORM ROD. TWO GROUP CONSTANTS.</th>
</tr>
</thead>
<tbody>
<tr>
<td>68 4 1.E-8 1.E-8 1.E-8 1 1 1 0 0</td>
</tr>
<tr>
<td>1 2 3 3 3 3 4 4 5 5 5 5 5 5 6 6 6 6 7 7 7 7 7 7</td>
</tr>
<tr>
<td>7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 7 7 7 7 7 7 7</td>
</tr>
<tr>
<td>6 6 6 6 5 5 5 5 5 5 4 4 3 3 3 3 2 1</td>
</tr>
<tr>
<td>1.0 0.0</td>
</tr>
<tr>
<td>1 1</td>
</tr>
<tr>
<td>0.0 0.0525 0.0625</td>
</tr>
<tr>
<td>2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2</td>
</tr>
<tr>
<td>5.32 D-2 6.36 D-2 3.88 D-1</td>
</tr>
<tr>
<td>0.0 0.9375 0.9375</td>
</tr>
<tr>
<td>2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2</td>
</tr>
<tr>
<td>5.32 D-2 6.36 D-2 3.88 D-1</td>
</tr>
<tr>
<td>3 1</td>
</tr>
<tr>
<td>0.0 1.0 1.0</td>
</tr>
<tr>
<td>2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2</td>
</tr>
<tr>
<td>5.32 D-2 6.36 D-2 3.88 D-1</td>
</tr>
<tr>
<td>7 1</td>
</tr>
<tr>
<td>0.0 0.5 0.5</td>
</tr>
<tr>
<td>2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2</td>
</tr>
<tr>
<td>5.32 D-2 6.36 D-2 3.88 D-1</td>
</tr>
<tr>
<td>9 1</td>
</tr>
<tr>
<td>0.0 0.25 0.25</td>
</tr>
<tr>
<td>2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2</td>
</tr>
<tr>
<td>5.32 D-2 6.36 D-2 3.88 D-1</td>
</tr>
<tr>
<td>15 1</td>
</tr>
<tr>
<td>0.0 0.125 0.125</td>
</tr>
<tr>
<td>2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2</td>
</tr>
<tr>
<td>5.32 D-2 6.36 D-2 3.88 D-1</td>
</tr>
<tr>
<td>19 1</td>
</tr>
<tr>
<td>0.0 0.0625 0.0625</td>
</tr>
<tr>
<td>2.59 D-2 4.85 D-3 1.396 D 0 1.79 D-2</td>
</tr>
<tr>
<td>5.32 D-2 6.36 D-2 3.88 D-1</td>
</tr>
<tr>
<td>27 1</td>
</tr>
<tr>
<td>0.0 0.0525 0.0625</td>
</tr>
<tr>
<td>4.52 D-2 0.0 0 1.0 0 0 0.0 0 0</td>
</tr>
<tr>
<td>9.59 D-1 0.0 0 1.0 0 0</td>
</tr>
</tbody>
</table>
**SAMPLE REF2G REFERENCE SOLUTION OUTPUT BLOCK (SAMPLE SUBASSEMBLY SOLUTION):**

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>68.000000 01 0.0 D 0</td>
<td>0.6840883D 00 0.0 D 0</td>
<td>0.9976040D 00 0.6824493D 00 -0.4344942D-02</td>
<td></td>
</tr>
<tr>
<td>0.9999906D 00 0.2062319D-02</td>
<td>0.6840819D 00 -0.1410808D-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9976040D 00 0.6351434D-02</td>
<td>0.6824493D 00 -0.4344942D-02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**63 ADDITIONAL FAST GROUP DATA CARDS**

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9976040D 00 -0.6351434D-02</td>
<td>0.6824493D 00 0.4344942D-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9999906D 00 -0.2062319D-02</td>
<td>0.6840819D 00 0.1410808D-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1000000D 01 0.0 D 0</td>
<td>0.6840883D 00 0.0 D 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3090145D 00 0.0 D 0</td>
<td>0.1000000D 01 0.0 D 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3090072D 00 0.4484312D-03</td>
<td>0.9999762D 00 -0.1451165D-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3071103D 00 0.1418028D-02</td>
<td>0.9938377D 00 -0.4588871D-02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**63 ADDITIONAL THERMAL GROUP DATA CARDS**

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3071103D 00 -0.1418028D-02</td>
<td>0.9938377D 00 0.4588871D-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3090072D 00 -0.4484312D-03</td>
<td>0.9999762D 00 0.1451165D-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3090145D 00 0.0 D 0</td>
<td>0.1000000D 01 0.0 D 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### E.2. Linear Sample Input and Output Data Blocks

**Sample Linear Jr Cubic Homogenized Method Input Block:**

CASE 1 Study: Three Different Subassemblies. Homogenized Finite Element Method

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.0</td>
<td>18.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Case 2 Study:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.0</td>
<td>18.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Case 3 Study:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>18.0</td>
<td>18.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
SAMPLE LINEAR METHOD OUTPUT BLOCK:
-------------------------------------

3

0.2444087D 00  0.6758041D-01
0.4060855D 00  0.9339887D-01
0.7838067D 00  0.1339626D 00
0.1000000D 01  0.1563839D 00
### E.3. CUBIC SAMPLE INPUT AND OUTPUT DATA BLOCKS

**SAMPLE LINEAR JR CUBIC PROPOSED SYNTHESIS METHOD INPUT BLOCK:**

**CASE 1 STUDY:** THREE DIFFERENT SUBASSEMBLIES. SUBASSEMBLY SYNTHESIS.

<table>
<thead>
<tr>
<th>1 2 3</th>
<th>I.E-5</th>
<th>I.E-5</th>
<th>I.E-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>68</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>0.06250</td>
<td>0.06250</td>
<td>0.259E01</td>
</tr>
<tr>
<td>0.532E-01</td>
<td>0.636E-01</td>
<td>0.388E00</td>
<td></td>
</tr>
<tr>
<td>0.06250</td>
<td>1.00000</td>
<td>0.93750</td>
<td>0.259E01</td>
</tr>
<tr>
<td>0.532E-01</td>
<td>0.636E-01</td>
<td>0.388E00</td>
<td></td>
</tr>
</tbody>
</table>

64 ADDITIONAL CARD PAIRS OF TYPE A SUBASSEMBLY INTERVAL MATERIAL INPUT:

| 0.1000000D01 | 0.0 | 0.2062319D-02 | 0.6840819D00 | -0.1410808D-02 |
| 0.9999999D00 | 0.6351434D-02 | 0.6824493D00 | -0.4344942D-02 |

63 ADDITIONAL FAST GROUP DATA CARDS:

| 0.9976040D00 | -0.6351434D-02 | 0.6824493D00 | -0.4344942D-02 |
| 0.9999999D00 | -0.2062319D-02 | 0.6840819D00 | -0.1410808D-02 |
| 0.1000000D01 | 0.0 | 0.6840883D00 | 0.0 |
| 0.3090145D00 | 0.0 | 0.1000000D01 | 0.0 |
| 0.3090072D00 | 0.4484312D-03 | 0.9999762D00 | -0.1451165D-02 |
| 0.3071103D00 | 0.1418028D-02 | 0.9938377D00 | -0.4588871D-02 |

63 ADDITIONAL THERMAL GROUP DATA CARDS
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3071103</td>
<td>0.00</td>
<td>-0.1418028</td>
<td>-0.260E-1</td>
<td>0.553E-2</td>
<td>0.140E+01</td>
<td>0.172E+01</td>
</tr>
<tr>
<td>0.3090072</td>
<td>0.00</td>
<td>-0.4484312</td>
<td>-0.710E-1</td>
<td>0.553E-2</td>
<td>0.140E+01</td>
<td>0.172E+01</td>
</tr>
</tbody>
</table>

64 ADDITIONAL CARD PAIRS IF TYPE B SUBASSEMBLY INTERVAL MATERIAL INPUT

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17.999912</td>
<td>0.00</td>
<td>0.1938698</td>
<td>-0.5978934</td>
<td>-0.1938698</td>
<td>-0.5978934</td>
<td>-0.1938698</td>
</tr>
</tbody>
</table>

63 ADDITIONAL FAST GROUP DATA CARDS

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2282630</td>
<td>0.00</td>
<td>-0.8868016</td>
<td>0.9948319</td>
<td>0.3864921</td>
<td>0.1217112</td>
<td>0.0000000</td>
</tr>
</tbody>
</table>

63 ADDITIONAL THERMAL GROUP DATA CARDS

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2282630</td>
<td>0.00</td>
<td>-0.8868016</td>
<td>0.9948319</td>
<td>0.3864921</td>
<td>0.1217112</td>
<td>0.0000000</td>
</tr>
</tbody>
</table>

3 68

0.0 0.06250 0.06250 0.261E+01 0.659E-02 0.140E+01 0.168E+01

PAGE 176
### 64 Additional Card Pairs of Type C Subassembly Interval Material Input

| Start Time | End Time | Temperature | Type | Material | Interval | Data
|------------|----------|-------------|------|----------|----------|------
| 17.000000 | 17.937500 | 0.937500 | 0.261E-01 | 0.659E-02 | 0.140E-01 | 0.168E-01 | 0.832E-01 | 0.129E 00 | 0.387E 00 |
| 17.937500 | 18.000000 | 0.062500 | 0.261E-01 | 0.659E-02 | 0.140E-01 | 0.168E-01 | 0.832E-01 | 0.129E 00 | 0.387E 00 |

### 63 Additional Fast Group Data Cards

| Start Time | End Time | Temperature | Type | Material | Interval | Data
|------------|----------|-------------|------|----------|----------|------
| 0.99786650 | 0.00000000 | 0.5673004D-02 | 0.660944D00 | -0.3752725D-02 |
| 0.99999163 | 0.00000000 | 0.1837899D-02 | 0.6615002D00 | -0.1215781D-02 |
| 0.99786650 | 0.00000000 | 0.5673004D-02 | 0.660944D00 | -0.3752725D-02 |

### 63 Additional Thermal Group Data Cards

| Start Time | End Time | Temperature | Type | Material | Interval | Data
|------------|----------|-------------|------|----------|----------|------
| 0.19288970 | 0.00000000 | 0.6627953D-03 | 0.9954074D00 | 0.3420356D-02 |
| 0.19377520 | 0.00000000 | 0.2081898D-03 | 0.9999824D00 | 0.10743640-02 |
| 0.19377960 | 0.00000000 | 0.00000000 | 0.10000000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 | 0.00000000 |

PAGE 177
SAMPLE CUBIC METHOD OUTPUT BLOCK:
-----------------------------------

3

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25031910</td>
<td>0.0</td>
<td>0.7648905D-01</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.40354800</td>
<td>-0.22028074D-01</td>
<td>0.10629320D 00</td>
<td>0.5636188D-03</td>
<td></td>
</tr>
<tr>
<td>0.77856490</td>
<td>-0.2731031D-01</td>
<td>0.1642353D 00</td>
<td>0.3579210D-03</td>
<td></td>
</tr>
<tr>
<td>0.10000000</td>
<td>0.0</td>
<td>0.1946404D 00</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>
E.4. ANALYZE SAMPLE INPUT AND OUTPUT

SAMPLE ANALYZE INPUT (CASE 1 CUBIC METHODS RESULTS):

//G.FT01F001 DD *
  2 3 3 35
CASE 1 HOMOGENIZED LINEAR FINITE ELEMENT METHOD INPUT BLOCK

/*

//G.FT02F001 DD *
  2 3 3 0
CASE 1 LINEAR SYNTHESIS METHOD INPUT BLOCK

/*

//G.FT03F001 DD *
  1 3 150 0
  50 50 50
CASE 1 REFERENCE SOLUTION INPUT BLOCK

/*

//G.FT11F001 DD *
CASE 1 HOMOGENIZED LINEAR FINITE ELEMENT METHOD OUTPUT BLOCK

/*
CASE 1 LINEAR SYNTHESIS METHOD OUTPUT BLOCK

CASE 1 REFERENCE SOLUTION OUTPUT BLOCK

TWO GROUP CASE 1 CUBIC RESULTS.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>18.3</td>
<td>9.5</td>
<td>26.5</td>
<td>27.5</td>
<td>44.5</td>
<td>45.5</td>
</tr>
</tbody>
</table>

PAGE 180
ANALYZE PRINTED OUTPUT: Case 1 with Linear Basis Functions.

RESULTS OF THE INTEGRATED POWER IN EACH OF THE 3 REGIONS:

<table>
<thead>
<tr>
<th>REGION:</th>
<th>HOMOGENIZED RESULTS:</th>
<th>SYNTHESIZED RESULTS:</th>
<th>REFERENCE RESULTS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.115E+00</td>
<td>0.108E+00</td>
<td>0.103E+00</td>
</tr>
<tr>
<td>2</td>
<td>0.258E+00</td>
<td>0.245E+00</td>
<td>0.245E+00</td>
</tr>
<tr>
<td>3</td>
<td>0.431E+00</td>
<td>0.411E+00</td>
<td>0.417E+00</td>
</tr>
<tr>
<td>TOTALS:</td>
<td>0.806E+00</td>
<td>0.765E+00</td>
<td>0.761E+00</td>
</tr>
</tbody>
</table>

FRACTIONAL POWER NORMALIZED PERCENT ERRORS:

<table>
<thead>
<tr>
<th>REGION:</th>
<th>(REF-HOMO)/REF %</th>
<th>(REF-SYNT)/REF %</th>
<th>(SYNTN-HOMO)/SYNT %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.693E-01</td>
<td>-0.563E-01</td>
<td>-0.122E+01</td>
</tr>
<tr>
<td>2</td>
<td>-0.163E-01</td>
<td>-0.139E-01</td>
<td>-0.239E+01</td>
</tr>
<tr>
<td>3</td>
<td>0.263E-01</td>
<td>0.217E-01</td>
<td>0.467E+01</td>
</tr>
</tbody>
</table>
ANALYZE PRINTED OUTPUT: Case 1 with Linear Basis Functions.

EXECUTING GENERAL ANALYSIS AND FLUX PLOTTING PROGRAM:

TITLE OF PLOTTING RUN IS: THREE DIFFERENT SUBASSEMBLIES PROBLEM.

REACTOR GEOMETRY PARAMETERS:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCELL</td>
<td>3</td>
</tr>
<tr>
<td>WCELL</td>
<td>18.000000</td>
</tr>
<tr>
<td>XMIN</td>
<td>0.0</td>
</tr>
<tr>
<td>XMAX</td>
<td>54.000000</td>
</tr>
<tr>
<td>YMIN</td>
<td>0.0</td>
</tr>
<tr>
<td>YMAX</td>
<td>1.000000</td>
</tr>
<tr>
<td>NLL</td>
<td>6</td>
</tr>
<tr>
<td>(XL(I),I=1,NLL) =</td>
<td>8.50000 9.50000 26.50000 27.50000 44.50000 45.50000</td>
</tr>
</tbody>
</table>
Appendix F

SOURCE LISTINGS OF THE PROGRAMS

FORTRAN source listings of programs REF2G, LINEAR, CUBIC, and ANALYZE are listed in only the first six copies of this report in the following four sections.

A figure of a subroutine overlay structure precedes each listing in order to indicate the construction of each program.
F.1. SOURCE LISTING of Program REF2G
Figure F.1. Structure of Program REF2G.
C PROGRAM REF2G:
C TWO GROUP DETAILED REFERENCE AND SUBASSEMBLY SOLUTION PROGRAM.
C
CALL TIMING(11)
CALL SYNTH
CALL TIMING(12)
CALL POWER
CALL TIMING(13)
CALL CURT
CALL TIMING(14)
CALL OUTPUT
CALL TIMING(15)
CALL POWER7
CALL TIMING(16)
CALL CURT7
CALL TIMING(17)
CALL OUTPUT7
CALL TIMING(18)
C TIMING EXECUTION
WRITE (6,30)
30 FORMAT (1HI, 'TIMING PROGRAM EXECUTION:',/) 
   J=12-11
   WRITE (6,701) J
   J=13-12
   WRITE (6,702) J
   J=14-13
   WRITE (6,703) J
   J=15-14
   WRITE (6,704) J
   J=16-15
   WRITE (6,705) J
   J=17-16
   WRITE (6,706) J
   J=18-17
   WRITE (6,707) J
701 FORMAT (1HI, ' SYNTH HAS TAKEN',I6, '/100 SECONDS.',)
702 FORMAT (1H, 'POWER HAS TAKEN', I6, '/100 SECONDS.')
703 FORMAT (1H, 'CURT HAS TAKEN', I6, '/100 SECONDS.')
704 FORMAT (1H, 'OUTPUT HAS TAKEN', I5, '/100 SECONDS.')
705 FORMAT (1H, 'POWER7 HAS TAKEN', I6, '/100 SECONDS.')
706 FORMAT (1H, 'CURT7 HAS TAKEN', I6, '/100 SECONDS.')
707 FORMAT (1H, 'OUTPUT7 HAS TAKEN', I5, '/100 SECONDS.')
    CALL TIMING(120)
    J=120-I
    WRITE(6,720) J
720 FORMAT (1H, 'THIS RUN HAS TAKEN', I6, '/100 SECONDS TO RUN.')
STOP
END
SUBROUTINE SYNTH
C
LINEAR FINITE ELEMENT METHOD:
C
ADJOINT QUANTITIES OF VARIABLES ARE DENOTED BY \( \tilde{\gamma} \) RATHER THAN \( \gamma \). THUS:
C
\( \pi \tilde{\gamma} \) (RATHER THAN \( \pi \gamma \)) IS THE ADJOINT OF \( \pi \).
C
IMPLICIT REAL*8 (A-H,K-Z)
COMMON /B1/ IBC,IPLT,JPLT,IPUNCH,ISEE,NOADJ
COMMON /B2/ KR,NN
COMMON /B3/ L1(201,3), L2(201,3), F1(201,3), F2(201,3), T(201,3)
COMMON /B5/ KAU(2,200), KAI(2,200), KAU(2,200), KBO(2,200),
X KB1(2,200), KB2(2,200), LAO(2,200), LA2(2,200),
X SKU(2,200), SR1(2,200), SR2(2,200), P1(2,200), P1(2,200),
X Q(2,200), Q1(2,200), R(2,200), P0(2,200), P07(2,200), PH(2,200),
X PH7(2,200), AL(2,200), BL(2,200), CL(2,200), AF(2,200), BF(2,200),
X CF(2,200), AT(200), BT(200), CT(200),
X BL0(2), CL0(2), BFC(2), CFO(2), BTO(2),
X CF0(2),
X ALK(2), BLK(2), AFK(2), BFK(2), ATK(2),
X BTK(2)
COMMON /B7/ HH(200), DD(2,200)
COMMON /CHIF/ CHI(2)
COMMON /BH/ X(2), H(1)
COMMON /ER/ EPS1,EPS2,EPS3
DIMENSION PHI(2,2),PHI7(2,2),CUR(2,2),CUR7(2,2),
X A(2,1),F(2,1),D(2,1),S(2,1),D1(2,1),XU(2,2)
DIMENSION ITF(200), KTF(200)
REAL TITLE(20)
INTEGER KR,Ks,Ks1,KRo,NN
INTEGER NUMITF, KTF, NOADJ
READ (5,200) TITLE
200 FORMAT (20A4)
WRITE (6,201) TITLE
201 FORMAT (1H1,20A4,/)
1 FORMAT (2I5v,301D0.3,5I5)
  C     READ IN THE TYPE-NUMBER OF EACH TF REGION:
  READ (5,100) (ITF(I),I=1,KR)
100 FORMAT (25I2).
  C     READ IN THE FISSION YIELD FOR EACH GROUP:
  READ (5,101) CHI(1), CHI(2)
101 FORMAT (2F10.5)
  KRO=KR-1
  WRITE (6,2) KR, IBC
2 FORMAT ('JVARIATIONAL SYNTHESIS PROGRAM #2G(200):',5X,'USING ',I3, X  ' SUBREACTOR REGIONS, OR TRIAL FUNCTIONS.',/, X  ' OBVIOUS CONDITION NUMBER (IBC) IS ',I1,'.',/, X  ' MATERIAL PROPERTIES AND TRIAL FUNCTIONS FOR EACH SUBREGION FO XLLLOW:',/, X  ' MATERIAL PROPERTIES ARE HOMOGENEOUS IN THE INDICATED REGIONS. X',/, X  ' OFLUX TRIAL FUNCTIONS ARE LINEAR IN EACH SEGMENT OF THE SUBREG XIONS.',/, X  ' CURRENT TRIAL FUNCTIONS ARE FLAT IN EACH OF THE ', X  ' SUBREGIONS.\)\)
  WRITE (6,20) EPS1,EPS2,EPS3,IPLOT,JPILOT,IPUNCH,ISEE,NOADJ
20 FORMAT (//,'OTLERANCES TO POWER ARE: EPS1 = ',1PD10.3,/, X  28X,'EPS2 = ',1PD10.3,/,28X,'EPS3 = ',1PD10.3,/, X  'OUTPUT PARAMETERS TO POWER ARE: IPlot = ',I1,/, X  34X,'JPILOT = ',I1,/,34X,'IPUNCH = ',I1,/, X  34X,'ISEE = ',I1,/, X  34X,'NOADJ = ',I1,'\)\)
  WRITE (6,22) CHI(1), CHI(2)
22 FORMAT (//,'JFISSION YIELDS ARE: CHI(1) = ',F10.5,/, X  22X,'CHI(2) = ',F10.5)
  IF ((KK.LE.2).AND.(IBC.EQ.1)) CALL ERROR(1,KR)
  IF (KR.GT.200) CALL ERROR(2,KR)
  IF (EPS1.LT.1.0E-16) CALL ERROR(6,1)
  IF (EPS2.LT.1.0E-16) CALL ERROR(6,2)
  IF (EPS3.LT.1.0E-16) CALL ERROR(6,3)
  IF ((IBC.LT.1).OR.(IBC.GT.4)) CALL ERROR(7,IBC)
C DUMMY NORMAL VECTOR: XU = UNITY. (FOR THE INTEGRATION FUNCTIONS.)
DO 21 IG=1,2
   DO 21 II=1,2
21 XU(IG,II)=1.0
C SET FLUXES TO UNITY FOR SYNTH 2G:
DO 25 IG=1,2
   DO 25 II=1,2
   PHI(IG,II)=1.0
25 PHI(IG,II)=1.0
C COUNTER OF THE NUMBER OF TYPE-NUMBERS OF EACH TF REGION:
NUMITF=1
WRITE (6,9).
9 FORMAT ('I')
C BEGIN TO READ IN THE TF REGION DATA AND FILL THE ARRAYS,
C DEPENDING ON THE TYPE-NUMBER OF EACH TF REGION.
DO 50 I=1,KR
   IF (ITF(I).EQ.NUMITF) GO TO 110
50 CALL REPEAT(I,KTF(J))
   GO TO 50
C READ IN THE TF REGION'S DATA FOR NEW TF REGION TYPE-NUMBERS:
110 NUMITF=NUMITF+1
   KTF(NUMITF-1)=I
   READ (5,1) K
   KS=1
   IF (KS.GT.100) CALL ERROR(3,1)
   KS=KS+1
C CHECK FOR IMPROPER SEQUENCING OF INPUT DATA:
   IF (I.NE.K) CALL ERROR(4,1)
C READ THE GEOMETRY AND THE MATERIAL PROPERTIES OF THIS REGION:
   READ (5,3) (X(J),X(J+1),H(J),A(1,J),F(1,J),D(1,J),S(1,J),
   A(2,J),F(2,J),D(2,J),J=1,KS)
3 FORMAT (3F10.5,4E10.3,/,3JX,3E10.3)
C WRITING OUT THE INPUT INFORMATION:
IF (ISEE.EQ.J) GO TO 14
WRITE (6,10) X,KR,KS,(J,X(J),X(J+1),H(J),A(1,J),F(1,J),D(1,J)),
X S(1,J),A(2,J),F(2,J),D(2,J),J=1,KS
10 FORMAT ('INPUT MATERIAL PROPERTIES FOR SUBREGION NUMBER ',I3,
X ' OF THE ',I3,' USED.',//,
X ' THIS SUBREGION IS DIVIDED INTO ',I3,' HOMOGENEOUS SEGMENTS
X AS FULLMS:',//,
X ' FAST GROUP CONSTANTS APPEAR FIRST:',//,
X ' REGION #',5X,'INTERNAL BOUNDARIES',13X,'WIDTH',3X,
X ' ABSORB. CX (1/CM)',3X,'FISSION CX (1/CM)',6X,'DIFFUSION (CM)',
X 4X,'SCATT. CX (1/CM)',//,
X 5X,'I',11X,'X(I)',9X,'X(I+1)',11X,'H(I)',13X,'A(IG,I)',13X,
X 'F(I,G,I)',13X,'D(IG,I)',14X,'S(I,G)',//,
X (16,3F15.4,4D20.8,/,5I,3D20.8))
C END OF THE IN-OUT SECTION.
14 CONTINUE
C DEFINING MISC. ARRAYS FOR THE INTEGRATION FUNCTIONS:
C LENGTH OF THE SUBREGION: HT
HT=X(KSI)-X(1)
HH(K)=HT
DD(1,K)=D(1,J)
DD(2,K)=D(2,J)
C INVERSE OF D ARRAYS:
DO 13 J=1,KS
13 DI(J,J)=1./D(J,J)
C FORMATION OF THE INTEGRATION FUNCTIONS:
CALL BHSET(KS).
C DO FOR ALL ENERGY GROUPS:
DO 50 IG=1,2
KA0(IG,K)=GIFO(IG,PHI7,PHI,A,KS)
KA1(IG,K)=GIFO(IG,PHI7,PHI,A,KS)
KA2(IG,K)=GIF2(IG,PHI7,PHI,A,KS)
KB0(IG,K)=GIFO(IG,PHI7,PHI,F,KS)
KB1(IG,K)=GIFO(IG,PHI7,PHI,F,KS)
KB2(IG,K)=GIFO(IG,PHI7,PHI,F,KS)
R(IG,K) = GIF0(IG,PHI7,PHI,D,KS)/(HT*HT)

C NO SCATTERING IN THE LOWEST GROUP:
IF (IG.EQ.2) GO TO 50
SRO(IG,K) = GIF0(IG,PHI7,PHI,S,KS)
SRI(IG,K) = GIF1(IG,PHI7,PHI,S,KS)
SR2(IG,K) = GIF2(IG,PHI7,PHI,S,KS)
50 CONTINUE
NUMITF = NUMITF - 1
WRITE (6,51) NUMITF
51 FORMAT(*' THERE ARE ONLY', I3, ' DIFFERENT TRIAL FUNCTION REGIONS. ')
WRITE (6,52) (1, ITF(I), I = 1, KR)
52 FORMAT(*' TABLE OF THE TRIAL FUNCTION REGION TYPES: ', //,
      ' X ' 3X, ' TF REGION', 4X, ' REGION TYPE- NUMBER ', //,
      ' X ' (17, 12X, 11))

C DETERMINATION OF THE B.C. OPTION PARAMETERS:
C NN IS THE MM AND FF MATRIX BLOCK SIZE.
IF (IBC.EQ.1) NN = KR - 1
IF ((IBC.EQ.2) .OR. (IBC.EQ.3)) NN = KR
IF (IBC.EQ.4) NN = KR + 1

C FORMATION OF THE COEFFICIENT VECTORS:
C THE INTERIOR COEFS:
DO 60 IG = 1, 2
DO 60 K = 2, KR
J = K - 1
AL(IG,K) = KA1(IG, J) - KA2(IG, J) - R(IG, J)
BL(IG,K) = KA2(IG, J) + R(IG, J) + KA0(IG,K) - 2.0*KA1(IG,K) + KA2(IG,K)
X
CL(IG,K) = KA1(IG,K) - KA2(IG,K) - R(IG,K)
AF(IG,K) = KB1(IG,J) - KB2(IG,J)
BF(IG,K) = KB2(IG,J) + KB0(IG,K) - 2.0*KB1(IG,K) + KB2(IG,K)
CF(IG,K) = KB1(IG,K) - KB2(IG,K)
AT(K) = SRI(1,J) - SR2(1,J)
BT(K) = SR2(1,J) + SRO(1,K) - 2.0*SRI(1,K) + SR2(1,K)
CT(K) = SRI(1,K) - SR2(1,K)
60 CONTINUE
C THE ZERO FLUX COEFS:
DO 61 IG=1,2
BL(IG)=KA0(IG,1)-2.*KA1(IG,1)+KA2(IG,1)+R(IG,1)
BF0(IG)=KB0(IG,1)-2.*KB1(IG,1)+KB2(IG,1)
CLO(IG)=KA1(IG,1)-KA2(IG,1)-R(IG,1)
61 CFO(IG)=KB1(IG,1)-KB2(IG,1)
BT0(1)=SR0(1,1)-2.*SR1(1,1)+SR2(1,1)
CTO(1)=SR1(1,1)-SR2(1,1)

THE ZERO CURRENT COEFFS:
K=KR
DO 62 IG=1,2
ALK(IG)=KA1(IG,K)-KA2(IG,K)-R(IG,K)
BLK(IG)=KA2(IG,K)+R(IG,K)
AFK(IG)=KB1(IG,K)-KB2(IG,K)
62 BFKM(IG)=KB2(IG,K)
ATK(1)=SR1(1,K)-SR2(1,K)
BTK(1)=SR2(1,K)

ZERO MATRICES:
L1(1,1)=0.
L2(1,1)=0.
F1(1,1)=0.
F2(1,1)=0.
T (1,1)=0.
L1(NN,3)=0.
L2(NN,3)=0.
F1(NN,3)=0.
F2(NN,3)=0.
T (NN,3)=0.

FILL ALL THE MATRICES FOR POWER:
J=1

DETERMINE THE LEFT BOUNDARY CONDITIONS:
IF (IBC.LT.3) GO TO 67
L1(J,2)=BL0(1)
L2(J,2)=BL0(2)
F1(J,2)=BF0(1)
F2(J,2)=BF0(2)
T(J,2)=BT0(1)
DO 70 K = 2, KK
   IF (J.EQ.1) GO TO 69
   L1(J, 1) = AL(1, K)
   L2(J, 1) = AL(2, K)
   F1(J, 1) = AF(1, K)
   F2(J, 1) = AF(2, K)
   T(J, 1) = AT(K)
   L1(J, 2) = BL(1, K)
   L2(J, 2) = BL(2, K)
   F1(J, 2) = BF(1, K)
   F2(J, 2) = BF(2, K)
   T(J, 2) = BT(K)
   L1(J, 3) = CL(1, K)
   L2(J, 3) = CL(2, K)
   F1(J, 3) = CF(1, K)
   F2(J, 3) = CF(2, K)
   T(J, 3) = CT(K)
   J = J + 1
C
70 CONTINUE

C FOR ALL THE INTERIOR EQUATIONS:
67 DO 70 K = 2, KK
   IF (J.EQ.1) GO TO 69
   L1(J, 1) = AL(1, K)
   L2(J, 1) = AL(2, K)
   F1(J, 1) = AF(1, K)
   F2(J, 1) = AF(2, K)
   T(J, 1) = AT(K)
   L1(J, 2) = BL(1, K)
   L2(J, 2) = BL(2, K)
   F1(J, 2) = BF(1, K)
   F2(J, 2) = BF(2, K)
   T(J, 2) = BT(K)
   L1(J, 3) = CL(1, K)
   L2(J, 3) = CL(2, K)
   F1(J, 3) = CF(1, K)
   F2(J, 3) = CF(2, K)
   T(J, 3) = CT(K)
   J = J + 1
C
DETERMINE THE RIGHT BOUNDARY CONDITIONS:
70 CONTINUE

C IF ((IBC.EQ.1).OR.(IBC.EQ.3)) GO TO 80
   L1(J, 1) = ALK(1)
   L2(J, 1) = ALK(2)
   F1(J, 1) = AFK(1)
   F2(J, 1) = AFK(2)
   T(J, 1) = ATK(1)
   L1(J, 2) = BLK(1)
   L2(J, 2) = BLK(2)
   F1(J, 2) = BFK(1)
C
F2(J,2)=BFK(2)
T(J,2)=BTK(1)
80 CONTINUE
C PRINTS OUT THE SYNTH K ARRAYS, AND THE MATRICES GIVEN TO POWER
C FOR ISEE = 2:
IF (ISSE.EQ.2) CALL PRTOUT
RETURN
END
SUBROUTINE ERRDR(I,J)
C ANNDUNECtS INPUT ERRORS AND TERMINATES PROGRAM EXECUTION:
GO TO (1,2,3,4,5,6,7,8,9), I
1 WRITE (6,101)
GO TO 10
2 WRITE (6,102) J
GO TO 10
3 WRITE (6,103) J
GO TO 10
4 WRITE (6,104) J
GO TO 10
5 WRITE (6,105) J
GO TO 10
6 WRITE (6,106) J
GO TO 10
7 CONTINUE
8 CONTINUE
9 CONTINUE
10 WRITE (6,110)
101 FORMAT ('1MUST HAVE > 2 SUBREGIONS FOR ZERO FLUX B.C.S. INVALID.'
102 FORMAT ('1NUMBER OF SUBREGIONS =',13,' > 25. INVALID.'
103 FORMAT ('1SUBREGION NUMBER',13,' HAS > 25 SECTIONS. INVALID.'
104 FORMAT ('1INPUT ERROR IN REGION SEQUENCING AT REGION',5,.
105 FORMAT ('1REGION I =',13,'. INVALID.'
106 FORMAT ('1THE TOLERANCE: EPS',11,' IS < 1.0E-16. INVALID.'
107 FORMAT ('1BOUNDARY CONDITION OPTION =',12,' < 1 OR > 4. INVALID.'
108 FORMAT (1H0,'PROBLEM TERMINATED.'
CALL EXIT
RETURN
END
SUBROUTINE REPEAT(K,L)
C
SETS THE /BS/ ARRAYS (K) EQUAL TO PAST STORED ARRAYS (L):
IMPLICIT REAL*8 (A-Z)
COMMON /BS/ KAO(2,200),KA1(2,200),KA2(2,200),KB0(2,200),
X KB1(2,200),KB2(2,200),LA0(2,200),LA1(2,200),LA2(2,200),
X SR0(2,200),SR1(2,200),SR2(2,200),P(2,200),P1(2,200),
X Q1(2,200),Q1(2,200),R(2,200),P0(2,200),P07(2,200),PH(2,200),
X PH7(2,200)
COMMON /BS/ HH(200),DD(2,200)
INTEGER K,L,G
DO 10 G=1,2
KAO(G,K)=KAO(G,L)
KA1(G,K)=KA1(G,L)
KA2(G,K)=KA2(G,L)
KB0(G,K)=KB0(G,L)
KB1(G,K)=KB1(G,L)
KB2(G,K)=KB2(G,L)
LA0(G,K)=LA0(G,L)
LA1(G,K)=LA1(G,L)
LA2(G,K)=LA2(G,L)
IF (G.EQ.2) GO TO 5
SR0(G,K)=SR0(G,L)
SR1(G,K)=SR1(G,L)
SR2(G,K)=SR2(G,L)
5 CONTINUE
CONTINUE
P(G,K)=P(G,L)
P1(G,K)=P1(G,L)
Q(G,K)=Q(G,L)
Q1(G,K)=Q1(G,L)
R(G,K)=R(G,L)
P0(G,K)=P0(G,L)
P07(G,K)=P07(G,L)
PH(G,K)=PH(G,L)
PH7(G,K)=PH7(G,L)
10 CONTINUE
HH(K)=HH(L)
DD(1,K) = DD(1,L)
DD(2,K) = DD(2,L)
RETURN
END
SUBROUTINE BHSET(K)
C  FIRST OF 4 ANALYTICAL INTEGRATION ROUTINES.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /BH/ X(2),H(1),H2(1),H3(1),H4(1),H5(1)
DO 1 I=1,K
  H2(I)=X(I+1)**2-X(I)**2
  H3(I)=X(I+1)**3-X(I)**3
  H4(I)=X(I+1)**4-X(I)**4
  H5(I)=X(I+1)**5-X(I)**5
1 CONTINUE
RETURN
END
DOUBLE PRECISION FUNCTION GIFO(IG,Y,Z,C,K)

IMPLICIT REAL*8 (A-H,L-Z)

COMMON /BH/ X(2),H(1),H2(1),H3(1),H4(1),H5(1)

DIMENSION Y(2,2), Z(2,2), C(2,1)

SUM = 0.0

DO 1 I=1,K

SUM = C(IG,I)*(Y(IG,I)*Z(IG,I)*H(I)+H(I)*(Z(IG,I)*X(IG,I+1)+Y(IG,I)*(Z(IG,I+1)-Z(IG,I))/3.) + SUM

1 CONTINUE

GIFO = SUM

RETURN

END
DOUBLE PRECISION FUNCTION GIF1(I,G,Y,Z,C,K)
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /BH/ X(2),H(1),H2(1),H3(1),H4(1),H5(1)
DIMENSION Y(2,2), Z(2,2), C(2,1)
SUM = 0.0
DO 1 I=1,K
SUM = C(I,G,1)*((H2(I)/2.-X(I)*H(I))*Y(I,G,1)*Z(I,G,1))
  X +((Z(I,G,1)*((Y(I,G,1+1)-Y(I,G,1))*Y(I,G,1))*
  X  +((Z(I,G,1+1)-Z(I,G,1))*((H3(I)/3.-H2(I)*X(I)*X(I))/2.
  X  +(H4(I)*H(I))*Y(I,G,1+1)-Y(I,G,1)))*C(I,G,1+1)-Z(I,G,1))
  X *(H4(I)/4.*H3(I)*(2.*X(I)+X(I))/3.+H2(I)*X(I)*X(I)+2.*X(I))
  X *X(I))/2.-X(I)*X(I)*X(I)*H(I))/((H(I)*H(I)))+SUM
1 CONTINUE
GIF1 = SUM/(X(K+1)-X(1))
RETURN
END
DOUBLE PRECISION FUNCTION GIF2(IG,Y,Z,C,K)
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /B// X(2),H(I),H2(I),H3(I),H4(I),H5(I)
DIMENSION Y(2,2), Z(2,2), C(2,1)
SUM = 0.0
DO 1 I=1,K
  SUM = C(IG,1)*Z(IG,1)*H3(I)/3. - X(1)*H2(I) + X(1)*X(1)*H(I)
  X = (1.0/H(I))*Z(IG,1)*Y(IG,1)*H(I) + Y(IG,1)*Z(IG,1) + H4(I)*H2(I)*H(I)
  X = X + H5(I)*H(I)
  X = X + X**2. + X**3. + X**4. + X**5.
  X = X + X**2. + X**3. + X**4. + X**5.
  X = X + X**2. + X**3. + X**4. + X**5.
  X = X + X**2. + X**3. + X**4. + X**5.
1 CONTINUE
GIF2 = SUM/((X(K+1)-X(1))**2)
RETURN
END
SUBROUTINE PRINTOUT

PRINTS OUT THE /85/ ARRAYS AND THE MATRICES GIVEN TO POWER:

IMPLICIT REAL*8 (A-H,K-Z)

COMMON /B2/ KR, N
COMMON /B3/ L1(201,3), L2(201,3), F1(201,3), F2(201,3), T(201,3)
COMMON /B5/ KA0(2,200), KA1(2,200), KA2(2,200), KB0(2,200),
X KB1(2,200), KB2(2,200), LA0(2,200), LA1(2,200), LA2(2,200),
X SR0(2,200), SR1(2,200), SR2(2,200), PI(2,200), P(2,200),
X Q1(2,200), Q2(2,200), R(2,200), P0(2,200), P0(2,200),
X PH(2,200), PH(2,200)

INTEGER KR, G, N

C KA AND KB ARRAYS:

WRITE (6,10)
10 FORMAT ('1 G',4X,'I',12X,'KA0(G,I)',12X,'KA1(G,I)',12X,
X 'KA2(G,I)',12X,'KB0(G,I)',12X,'KB1(G,I)',12X,'KB2(G,I)')
DO 11 G=1,2
WRITE (6,12)
11 WRITE (6,15) (G,I),KA0(G,I),KA1(G,I),KA2(G,I),KB0(G,I),KB1(G,I),
X KB2(G,I),I=1,KR)
12 FORMAT (' ')
15 FORMAT (215,6D20.7)

C LA AND SR ARRAYS:

WRITE (6,20)
20 FORMAT ('1 G',4X,'I',12X,'LA0(G,I)',12X,'LA1(G,I)',12X,
X 'LA2(G,I)',12X,'SR0(G,I)',12X,'SR1(G,I)',12X,'SR2(G,I)')
G=1
WRITE (6,12)
WRITE (6,15) (G,I),LA0(G,I),LA1(G,I),LA2(G,I),SR0(G,I),SR1(G,I),
X SR2(G,I),I=1,KR)
G=2
WRITE (6,12)
WRITE (6,25) (G,I),LA0(G,I),LA1(G,I),LA2(G,I),I=1,KR)
25 FORMAT (215,3020.7)

C P, Q, AND R ARRAYS:

WRITE (6,30)
30 FORMAT ('1 G',4X,'I',14X,'P(G,I)',13X,'P1(G,I)',14X,'Q(G,I)',
X 'R(G,I)',14X,'R1(G,I)',14X,'R2(G,I)')
DO 31 G=1,2
WRITE (6,12)
31 WRITE (6,35) (G,IP(G),PI(G),Q(G),Q1(G),R(G),I=1,MR)
35 FORMAT (215,5D20.7)

C PRINT OUT THE /B3/ MATRICES:
WRITE (6,50)
50 FORMAT (*1MATRIX L1:*,/)
WRITE (6,55) ((L1(I,J),J=1,3),I=1,N)
55 FORMAT (3E12.3,7X,3E12.3,7X,3E12.3)
WRITE (6,60)

C PRINT OUT THE /B3/ MATRICES:
WRITE (6,50)
50 FORMAT (*1MATRIX L2:*,/)
WRITE (6,55) ((L2(I,J),J=1,3),I=1,N)
55 FORMAT (3E12.3,7X,3E12.3,7X,3E12.3)
WRITE (6,60)

C PRINT OUT THE /B3/ MATRICES:
WRITE (6,50)
50 FORMAT (*1MATRIX T:*,/)
WRITE (6,55) ((T(I,J),J=1,3),I=1,N)
RETURN
END
SUBROUTINE POWER
C SOLVES THE 2*N MULTIGROUP EQUATIONS: M*PHI = (1/LAMDA)*F*PHI
C BY THE FISSION SOURCE POWER METHOD
C USING SIMULTANEOUS OVERRELAXATION.
C WHERE: M AND F ARE DOUBLE PRECISION 2N BY 2N BLOCK MATRICES;
C AND: PHI IS THE 2N FLUX (FAST AND THERMAL) VECTOR.
C L1*PHI1 = CHI1*(F1*PHI1 + F2*PHI2)
C -T*PHI1 + L2*PHI2 = CHI2*(F1*PHI1 + F2*PHI2)
C METHOD FOLLOWS WACHPRESS, PAGE 83. SOLUTION BY GROUP ITERATION.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /81/ IBC,PLOT,JPLOT,PUNCH,ISEE
COMMON /82/ KR,N
COMMON /83/ L1(201,3), L2(201,3), F1(201,3), F2(201,3), T(201,3)
COMMON /84/ PHI(2,201), PSI(2,201), LAMDA, ICOUT
COMMON /85/ S(201), ERROR(2,201), Z(201)
COMMON /86/ TE1(2,5),TE2(2,5),TE3(5),IN(5)
COMMON /87/ HH(200)
COMMON /CHI1/ CHI(2)
COMMON /ER/ EPS1,EPS2,EPS3
COMMON /I/ I1,I4
COMMON /FSTR/ PHISTR(2,201,6)
COMMON /ESTR/ LAMISTR(300), EFSTR(2,300), EFSISTR(2,300), ERLAM(300)
COMMON /READS/ R5
DIMENSION PSI1(201), PSI2(201), SQ(2), DPHI(2), ERRMAX(2)
INTEGER N
R5=1.
C DEFAULT OPTIONS FOR POWER PARAMETERS:
ALPHA=1.25
LAMDA=1.0
HX=0.0
DO 505 I=1,KR
505 HX=HX+HH(I)
DO 555 IG=1,2
IF (IBC.NE.4) GO TO 551
DO 550 I=1,N
550 PHI(IG,I)=1.0
551   X=3.1415926/HX
      IF (IBC.NE.1) X=X/2.0
      SUM1=0.0
      DO 552 K=1,KR
        SUM1=SUM1+H(K)
      552   PHI(IG,K)=USIN(SUM1*X)
      CONTINUE

C     READ IN: OVERRELAXATION PARAMETERS ; ALPHA (OUTER ITERATION)
C     INITIAL GUESS AT EIGENVALUE; LAMDA
C     INITIAL NORMALIZED FLUX ; PHI(1-N)
     READ (5,506,END=510) ALPHA
     READ (5,502,END=510) LAMDA
     READ (5,503) (PHI(1,I),I=1,N)
     READ (5,503) (PHI(2,I),I=1,N)
     506  FORMAT (F10.5)
     502  FORMAT (E25.14)
     503  FORMAT ((4E20.10))
     GO TO 511
     510  R5=0.
     511  CONTINUE
C     STORING FOR PRINTING THE MULTIGROUP FLUX SHAPE.
     DO 11 IG=1,2
     DO 10 I=1,N
       10 PHISTR(IG,1,2)=PHI(IG,1)
     C     FILL RUNNING COORD IN PHISTR
       KR1=KR+1
       DO 11 I=1,KR1
       11 PHISTR(IG,1,I)=DFLOAT(I)
C     IK IS THE FLUX PLOTTING COUNTER.
       IK=1
C     STORES THE ITERATION NUMBER FOR FLUX HISTORY PLOTTING:
       IN(1)=0
C     STORES TEMPORARY ERRORS FOR FLUX HISTORY PLOTTING:
       TE1(1,1)=0.
       TE1(2,1)=0.
TE2(1,1)=0.
TE2(2,1)=0.
TE3(1)=0.0

C EIGENVALUE OF THE PREVIOUS ITERATION:
LAMBDA=LAMDA
C THE MAXIMUM NUMBER OF ALLOWED ITERATIONS: ICMA
ICMAX=300
C
PRINT OUT THE POWER METHOD PARAMETER INFORMATION:
WRITE (6,700) ICMA, ALPH, LAMDA, (PHI(1,I),I=1,N)
WRITE (6,701) (PHI(2,1),I=1,N)
700 FORMAT ('EXECUTING MULTIGROUP FISSION SOURCE POWER ITERATION METH
XOD...',/,
X 5X,'MAXIMUM NUMBER OF ALLOWABLE ITERATIONS: ',/,
X 10X,'ICMAX = ',I4,/,,
X 5X,'OUTER ITERATION RELAXATION PARAMETER: ',/,
X 10X,'ALPHA = ',F7.3,/,,
X 5X,'INITIAL GUESS AT EIGENVALUE: ',/,
X 10X,'LAMDA = ',E22.14,/,,
X 5X,'INITIAL GUESS AT THE GROUP FLUX SHAPE CONNECTION POINTS: ',
X //,8X,'FAST GROUP: ',/,
X 10X,'F(K)**S = ',4E25.14,/(18X,4E25.14))
701 FORMAT ('*',7X,'THERMAL GROUP: ',/,
X 10X,'F(K)**S = ',4E25.14,/(18X,4E25.14))

C BEGIN ITERATION LOOP.
ICOUT=0
C ICOUT IS THE OUTER ITERATION COUNTER.
20 ICOUT=ICOUT+1
IF (ICOUT.GT.ICMA) GO TO 100
C
FORM THE ITERATION SOURCE VECTOR, S; AND ITS L-2 NORM, SUM1:
SUM1=0.
DO 15 I=1,N
S(I)=0.
10=1
I1=3
IF (I.EQ.1) 10=2
IF (I.EQ.N) I1=2
PAGE 207
DO 14 J=10,11
K=I-2+J
14 S(I)=S(I)+F1(I,J)*PHI(1,K)+F2(I,J)*PHI(2,K)
15 SUM1=SUM1+S(I)**2
SUM1=DSQRT(SUM1).
SUM1=SUM1*(CHI(1)+CHI(2))
C SOLVE FOR THE NEW GROUP FLUX VECTORS: PSI:
C FAST GROUP; SOURCE VECTOR:
DO 25 I=1,N
25 Z(I)=CHI(1)*S(I)
C FAST FLUX:
CALL SOLV3D(N,L1,PSI1,Z)
C THERMAL GROUP; SOURCE VECTOR:
DO 27 I=1,N
Z(I)=0.
I0=1
II=3
IF (I.EQ.1) I0=2
IF (I.EQ.4) II=2
DO 26 J=10,11
K=I-2+J
26 Z(I)=Z(I)+T(I,J)*PSII(K)
27 Z(I)=Z(I)+CHI(2)*S(I)
C THERMAL FLUX:
CALL SOLV3D(N,L2,PSI2,Z)
C FORM NEW SOURCE VECTOR FROM THE NEW UNNORMALIZED FLUXES; PSI:
SUM2=0.
DO 29 I=1,N
S(I)=0.
I0=1
II=3
IF (I.EQ.1) I0=2
IF (I.EQ.4) II=2
DO 28 J=10,11
K=I-2+J
28 S(I)=S(I)+F1(I,J)*PSII(K)+F2(I,J)*PSI2(K)
29 SUM2=SUM2+S(I)**2
SUM2=DSQRT(SUM2)
SUM2=SUM2*(CHI(1)+CHI(2))

C CALCULATION OF THE EIGENVALUE:
LAMDA=SUM2/SUM1
LAMSTR(IGOUT)=LAMDA
ERRLAM=DABS(LAMDA-LAMB4)

C PUT PSI1 AND PSI2 INTO BIGGER PSI:
DO 30 I=1,N
PSI1(I,1)=PSI1(I)
PSI2(I,1)=PSI2(I)
30

C POINT BY POINT SIMULTANEOUS RELAXATION FLUX ITERATION:
X=ALPHA

C DO NOT RELAX DURING THE FIRST THREE ITERATIONS:
IF (ICOUT.LE.3) X=1.0

C CALCULATE THE NEW GROUP FLUX ITERATES AND GROUP ERRORS:
DO 40 IG=1,2
SQ(IG)=0.
DO 40 I=1,N
ERROR(IG,I)=PSI(IG,I)/LAMDA-PHI(IG,I)
SQ(IG)=SQ(IG)+ERROR(IG,I)**2
PHI(IG,1)=PHI(IG,1)+X*ERROR(IG,I)
40 CONTINUE

C AND FOR PLOTTING PURPOSES:
PSI(IG,1)=PHI(IG,1)

C 40 CONTINUE

C DO 34 IG=1,2
SQ(IG)=DSQRT(SQ(IG))

C NORMALIZE PSI:

C NORMALIZES BOTH ARRAY GROUPS TO 1.0:
CALL NURM2(PSI,N)

C ERRMAX(IG) = THE MAX ERROR BETWEEN THE GROUP ITERATION FLUXES:
ERRMAX(IG)=ERROR(IG,1)
DO 36 I=2,N
IF (DABS(ERROR(IG,I)).GT.ERRMAX(IG)) ERRMAX(IG)=DABS(ERROR(IG,I))
36 CONTINUE
IF (IPLON.NE.2) GO TO 45
C THE FOLLOWING IS FOR NICELY PLOTTING THE GROUP FLUX HISTORY.
DO 41 IG=1,2
DO 41 I=1,N
41 ERROR(IG,I)=PSI(IG,I)
C ERROR NOW CONTAINS THE NEW NORMALIZED FLUX ITERATE PHI.
JK=IK
IF (IK.EQ.0) JK=5
DO 42 IG=1,2
DO 42 I=1,N
IF (DABS(ERROR(IG,I)-PHISTR(IG,I,JK+1)).GE.0.01) GO TO 43
42 CONTINUE
C FLUX HAS NOT CHANGED ENOUGH FOR PLOTTING.
GO TO 45
C SAVE THE NORMALIZED FLUX FOR PLOTTING:
43 IK=IK+1
IN(IK)=ICOUT
TE3(IK)=ERRLAM
DO 44 IG=1,2
TE1(IG,IK)=ERRMAX(IG)
TE2(IG,IK)=SQ(IG)
DO 44 I=1,N
44 PHISTR(IG,IK,JK+1)=ERROR(IG,I)
IF (IK.NE.5) GO TO 45
C PLOT THE LAST FIVE SAVED FLUXES:
CALL PHIPLT(5)
IK=0
45 CONTINUE
C ERROR CRITERIA FOR ACCEPTANCE OF CONVERGENCE.
IFLAG1=0
IFLAG2=0
IFLAG3=0
C STORE THE ERRORS FOR COMPARISON:
C ERROR BETWEEN ITERATION EIGENVALUES:
ERRLAM(ICOUT)=ERRLAM
DO 46 IG=1,2
MAXIMUM ERROR BETWEEN ITERATION FLUXES:
\[ E_{F\text{MAX}}(IG,IC\text{OUT}) = \text{ERMAX}(IG) \]

MEAN SQUARE ERROR BETWEEN ITERATION FLUXES:
\[ E_{M\text{FSTR}}(IG,IC\text{OUT}) = \text{SQ}(IG) \]

CONTINUE
\[
\text{IF (ERMAX(1).LT.}\text{EPS1).AND. (ERMAX(2).LT.}\text{EPS1))} \quad \text{IFLAG1}=1
\]
\[
\text{IF ((SQ(1).LT.}\text{EPS2).AND. (SQ(2).LT.}\text{EPS2))} \quad \text{IFLAG2}=1
\]
\[
\text{IF (ERLAM.LT.}\text{EPS3} \quad \text{IFLAG3}=1
\]
\[
\text{IFLAG4=IFLAG1}\times\text{IFLAG2}\times\text{IFLAG3}
\]
\[
\text{IF (IFLAG4.EQ.1) GO TO 50}
\]

OTHERWISE CONTINUE THE ITERATION.

LAMB4=LAMDA
GO TO 20

CONVERGENCE ACCOMPLISHED.

NORMALIZE THE CONVERGED FLUX VECTOR:
CALL NORMAL(PHI,N)

PLOT ANY LEFT OVER FLUX HISTORY PLOTS:
IF ((IPLJ.EQ.2).AND.(IK.NE.0)) CALL PHIPLT(IK)

BOUNDARY CONDITION INSERTIONS.
IER=0
IER ALLOWS B.C. INSERTIONS FOR YES AND NO CONVERGENCE:

IF (IBC.EQ.4) GO TO 90
IF (IBC.NE.3) GO TO 60
PHI(1,KR+1)=0.
PHI(2,KR+1)=0.
GO TO 90

DO 70 I=1,N
J=N+1-I
PHI(1,J+1)=PHI(1,J)

70 PHI(2,J+1)=PHI(2,J)
PHI(1,1)=0.
PHI(2,1)=0.
IF (IBC.NE.1) GO TO 90
PHI(1,KR+1)=0.
PHI(2,KR+1)=0.
90 IF (IER.EQ.1) GO TO 102
RETURN
C NO CONVERGENCE ACCOMPLISHED:
100 CONTINUE
C NORMALIZE THE UNCONVERGED FLUX:
    CALL NORMAL(PHI,N)
    ICOUT=ICOUT-1
    WRITE (6,101) ICOUT
101 FORMAT (1HI,'POWER METHOD DID NOT CONVERGE FOR THIS CASE AFTER',
         /X '14, ' ITERATIONS. ',/X,'EXECUTION TERMINATED ')
    IER=1
    GO TO 55
102 CONTINUE
C FOR PRINTING OUT THE EIGENVALUE HISTORY AND THE FINAL FLUX SHAPE:
    IF (IPLOT.EQ.0) IPLOT=1
    IF (JPLOT.EQ.0) JPLOT=1
RETURN
END
SUBROUTINE CURT
C SOLVES FOR THE CURRENT FROM THE INPUT H(K)'S AND D(K)'S
C USING F(K)'S FROM POWER:
C CURRENT IS LINEAR (LEAST SQUARES - VARIATIONAL) AND PUT INTO ARRAY C.
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /82/ KR
COMMON /84/ F(2,201), C(2,201)
COMMON /85/ T(201,3), S1(201), S2(201), C1(201), C2(201)
COMMON /87/ H(200), D(2,200)
C FROM THE MATRIX PROBLEM FOR LINEAR FIT OF STEP DATA:
M=KR
N=KR+1
T(1,1)=0.
T(N,3)=0.
T(1,2)=H(1)/3.
T(1,3)=H(1)/6.
T(N,1)=H(M)/6.
T(N,2)=H(M)/3.
S1(I)=D(1,1)*(F(1,I)-F(1,2))/2.
S2(I)=D(2,1)*(F(2,I)-F(2,2))/2.
S1(N)=D(1,M)*(F(1,M)-F(1,M+1))/2.
S2(N)=D(2,M)*(F(2,M)-F(2,M+1))/2.
DO 20 I=2,M
    J=I-1
    T(I,1)=H(J)/6.
    T(I,2)=(H(J)+H(I))/3.
    T(I,3)=H(I)/6.
    S1(I)=(D(1,I)*F(1,J)-F(1,I)+D(1,I)*F(1,I)-F(1,I+1))/2.
    S2(I)=(D(2,I)*F(2,J)-F(2,I)+D(2,I)*F(2,I)-F(2,I+1))/2.
20 CONTINUE
CALL SOLV3D(N,T,C1,S1)
CALL SOLV3D(N,T,C2,S2)
DO 30 I=1,N
    C(I,1)=C1(I)
30 C(2,I)=C2(I)
RETURN
SUBROUTINE OUTPUT
C PRINTS THE RESULTS OF THE METHOD.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /B1/ IBG, IPILOT, JPILOT, IPUNCH, ISEE, NOADJ
COMMON /B2/ KR, N
COMMON /B4/ PHI(2,201), CUR(2,201), LAMDA, ICOUT
COMMON /B5/ EPS1, EPS2, EPS3
COMMON /ESTR/ LAMSTR(300), EFSTR(2,300), EFMSTR(2,300), ERLAM(300)
INTEGER N, NOADJ
KRO=KR-1
KR1=KR+1
WRITE (6,1) ICOUT
1 FORMAT ('RESULTS OF THE MULTIGROUP METHOD:')
WRITE (6,13) ICOUT
10 FORMAT (' PROBLEM TERMINATED AFTER ', I5,
X ' OUTER (POWER) ITERATIONS TO:')
WRITE (6,20) LAMDA
20 FORMAT (' LAMDA = ', 1PE21.14)
C PRINT OUT EIGENVALUES.
CALL PLOT
WRITE (6,30)
30 FORMAT (' RESULTS AFTER PROBLEM TERMINATION:',//,
X 'ON NUMBER', 9X, ' THERMAL FLUX', 4X, ' THERMAL CURRENT', 12X,
X 'FAST FLUX', 7X, 'FAST CURRENT',//)
WRITE (6,50) (K, PHI(2,K), CUR(2,K), PHI(1,K), CUR(1,K), K=1, KR1)
50 FORMAT (17, 1PE21.7, 1PE19.7, 1PE21.7, 1PE19.7)
C PRINT OUT THE STORED ITERATION ERRORS:
WRITE (6,110) EPS1, (EFSTR(2,I), I=1, ICOUT)
WRITE (6,111) EPS1, (EFSTR(1,I), I=1, ICOUT)
WRITE (6,112) EPS2, (EFMSTR(2,I), I=1, ICOUT)
WRITE (6,113) EPS2, (EFMSTR(1,I), I=1, ICOUT)
WRITE (6,114) EPS3, (ERLAM(I), I=1, ICOUT)
110 FORMAT (' MAXIMUM ERRORS BETWEEN THE THERMAL FLUX ITERATIONS:',
X 25X, ' TOLERANCE USED = ', 1PE12.4, '//, (1PE5E20.5))
111 FORMAT (' MAXIMUM ERRORS BETWEEN THE FAST FLUX ITERATIONS:',
X 25X, ' TOLERANCE USED = ', 1PE12.4, '//, (1PE5E20.5))
112 FORMAT ('MEAN SQUARE ERROR BETWEEN THE THERMAL FLUX ITERATIONS:',
X     18X,'TOLERANCE USED = ',1PE12.4,/, (1P5E20.5))
113 FORMAT ('MEAN SQUARE ERROR BETWEEN THE FAST FLUX ITERATIONS:',
X     18X,'TOLERANCE USED = ',1PE12.4,/, (1P5E20.5))
114 FORMAT ('ERROR BETWEEN THE ITERATION EIGENVALUES:',
X     28X,'TOLERANCE USED = ',1PE12.4,/, (1P5E20.5))
     IF (NUADJ.EQ.0) RETURN
C OTHERWISE ADJOINT CALCULATIONS ARE NOT EXECUTED:
     WRITE (6,120)
120 FORMAT ('ADJOINT CALCULATIONS HAVE BEEN BYPASSED.',//,
X     ' PROGRAM TERMINATED.')</C
     IF (IPUNCH.EQ.1) WRITE (7,124) KR
     IF (IPUNCH.EQ.1) WRITE (7,125) (PHI(1,I),CUR(1,I),I=1,KR1)
     IF (IPUNCH.EQ.1) WRITE (7,125) (PHI(2,I),CUR(2,I),I=1,KR1)
124 FORMAT (15)
125 FORMAT (2D20.10)
     CALL EXIT
     RETURN
C
     END
SUBROUTINE PLOT

C PLOTTING OUT THE EIGENVALUE HISTORY AS A TABLE AND A GRAPH,
AS WELL AS PLOTTING OUT THE FINAL MULTIGROUP FLUX SHAPES.

IMPLICIT REAL*8 (A-H,L-Z)
COMMUN /81/ IBC, IPLT, JPLT, IPUNCH
COMMUN /82/ KR
COMMUN /84/ PHI(2,201), PSI(2,201), LAMDA, ICOUT
COMMUN /85/ B(300,2)
COMMUN /ESTR/ LAMSTR(300)
DIMENSION C(201,3)

C IN ORDER TO SAVE SOME SPACE:
EQUIVALENCE (81,1), C(1)
INTEGER ND=201
WRITE (6,1) (LAMSTR(I),I=ICOUT)
1 FORMAT ('OTABLE OF EIGENVALUES DURING THE POWER ITERATION:
X //,(1P5E25.14))
IF (JPLT.EQ.0) GO TO 20
DO 10 I=1,ICOUT
   B(I,1)=I
10 B(I,2)=LAMSTR(I)
   CALL PRTPHL(1,B,ICOUT,2,ICOUT,0,300,2,1)
   WRITE (6,11)
11 FORMAT ('OPLUT OF THE EIGENVALUE HISTORY THROUGH THE ITERATIONS.'
   X 5X,'FAST FLUX: .',//,5X,'THERMAL FLUX: -'
   RETURN
20 IF (IPLT.EQ.0) RETURN
   KR1=KR+1
   DO 30 I=1,KR1
      C(I,1)=I
      C(I,2)=PHI(1,I)
30   C(I,3)=PHI(2,I)
   CALL PRTPHL(2,C,KR1,3,KR1,0,ND,3,2)
   WRITE (6,31)
31 FORMAT ('OFINAL CONVERGED CONNECTING FLUX POINTS; F(K).
X 5X,'FAST FLUX: .',//,5X,'THERMAL FLUX: -'
RETURN
END
SUBROUTINE PUWERT
C *** ADJOINT PROBLEM ***
C SOLVES THE 2*N MULTIGROUP ADJOINT EQUATIONS:
C M*PHI = (1/LAMDA)*F*PHI
C BY THE FISSION SOURCE POWER METHOD
C USING SIMULTANEOUS OVERRELAXATION.
C WHERE: M AND F ARE DOUBLE PRECISION 2N BY 2N BLOCK MATRICES;
C AND: PHI IS THE 2N ADJOINT (FAST AND THERMAL) VECTOR.
C
L1*PHI1 - T*PHI2 = CHI1*F1*PHI1 + CHI2*F1*PHI2
L2*PHI2 = CHI1*F2*PHI1 + CHI2*F2*PHI2
C
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /B1/ IBC,IPLOT,JPLUT,IPUNCH,ISEE
COMMON /B2/ KR,N
COMMON /B3/ L1(201,3), L2(201,3), F1(201,3), F2(201,3), T(201,3)
COMMON /B4/ PHI(2,201), PSI(2,201), LAMDA, ICOUT
COMMON /B5/ S(201), ERROR(2,201), Z(201)
COMMON /B6/ TE1(2,5), TE2(2,5), TE3(5), IN(5)
COMMON /B7/ HH(200)
COMMON /CHIF/ CHI(2)
COMMON /ER/ EPS1,EPS2,EPS3
COMMON /F/ I1,I4
COMMON /FSTR/ PHISTR(2,201,6)
COMMON /ESTR/ LAMSTR(300), EFSSTR(2,300), EFMMSTR(2,300), ERLAM(300)
COMMON /READ5/ R5
DIMENSION PSI1(201), PSI2(201), SQ(2), DPHI(2), ERRMAX(2)
INTEGER N
C DEFAULT OPTIONS FOR POWER PARAMETERS:
ALPHA=1.25
LAMDA=1.0
HX=0.0
DO 505 I=1,KR
505 HX=HX+HH(I)
DO 555 IG=1,2
IF (IBC.NE.4) GO TO 551
DO 550 I=1,N
550 PHI(IG,I)=1.0
GO TO 555
  551  X=3.1415926/DX
  IF (IBC.NE.1) X=X/2.0
  SUM1=0.0
  DO 552 K=1,KR
       SUM1=SUM1+HH(K)
  552  PHI(IG,K)=DSIN(SUM1*X)
  555  CONTINUE
  IF (R5.EQ.0.) GO TO 510
C    READ IN: OVERRELAXATION PARAMETERS ; ALPHA (OUTER ITERATION)
C    INITIAL GUESS AT EIGENVALUE; LAMDA
C    INITIAL NORMALIZED FLUX ; PHI(1-N)
READ (5,506,END=510) ALPHA
READ (5,502,END=510) LAMDA
READ (5,503) (PHI(1,I),I=1,N)
READ (5,503) (PHI(2,I),I=1,N)
  506  FORMAT (F10.5)
  502  FORMAT (E25.14)
  503  FORMAT ((4E20.10))
  510  CONTINUE
C    STORING FOR PRINTING THE MULTIGROUP FLUX SHAPE.
   DO 10 IG=1,2
      DO 10 I=1,N
 10   PHISTR(IG,1,2)=PHI(IG,I)
C    FILL RUNNING COORD IN PHISTR
   KR1=KR+1
   DO 11 I=1,KR1
 11   PHISTR(IG,1,1)=DFLOAT(I)
C    IK IS THE FLUX PLOTTING COUNTER.
   IK=1
C    STORES THE ITERATION NUMBER FOR FLUX HISTORY PLOTTING:
   IN(1)=0.
C    STORES TEMPORARY ERRORS FOR FLUX HISTORY PLOTTING:
   TE1(1,1)=0.
   TE1(2,1)=0.
   TE2(1,1)=0.
\[ TE2(2,1) = 0. \]
\[ TE3(1) = 0.0 \]

C EIGENVALUE OF THE PREVIOUS ITERATION:
LAMB4=LAMDA
C 
C THE MAXIMUM NUMBER OF ALLOWED ITERATIONS: ICMAX
ICMAX=300
C 
C PRINT OUT THE POWER METHOD PARAMETER INFORMATION:
WRITE (6,700) ICMAX,ALPHA,LAMDA,(PHI(1,I),I=1,N)
WRITE (6,701) (PHI(2,I),I=1,N)

700 FORMAT ('EXECUTING MULTIGROUP ADJOINT FISSION SOURCE POWER ITERATION',//
X 'EXECUTING MULTIGROUP ADJOINT FISSION SOURCE POWER ITERATION',/)
X 5X,'MAXIMUM NUMBER OF ALLOWABLE ITERATIONS:',/,//
X 10X,'ICMAX = ',I4,///
X 5X,'OUTER ITERATION RELAXATION PARAMETER:',/,//
X 10X,'ALPHA = ',F7.3,///
X 5X,'INITIAL GUESS AT ADJOINT EIGENVALUE:',/,//
X 10X,'LAMDA = ',E22.14,///
X 5X,'INITIAL GUESS AT THE GROUP FLUX SHAPE CONNECTION POINTS:',
X 8X,'FAST ADJOINT GROUP:',//
X 10X,'FK1*S = ',4E25.14,/,18X,4E25.14))

701 FORMAT ('OUTER ITERATION RELAXATION PARAMETER:',//
X 10X,'FK1*S = ',4E25.14,/,18X,4E25.14))

C BEGIN ITERATION LOOP.

ICOUT=0
C ICOUT IS THE OUTER ITERATION COUNTER.
20 ICOUT=ICOUT+1
IF (ICOUT.GT.ICMAX) GO TO 100
C
C FORM THE GROUP TOTAL SOURCE S, AND ITS L-2 NORM SUM1:
C
SUM1=0.
DO 15 I=1,N
Z(I)=0.
S(I)=0.
I0=1
I1=3
IF (I.EQ.1) I0=2
DO 15 I=1,N
S(I)=0.
Z(I)=0.
END

PAGE 220
IF (I.EQ.N) II=2
DO 14 J=10,11
   K=1-2+J
   Z(I)=Z(I)+F2(I,J)*(CHI(1)*PHI(1,K)+CHI(2)*PHI(2,K))
14  S(I)=S(I)+F1(I,J)*(CHI(1)*PHI(1,K)+CHI(2)*PHI(2,K))
   S(I)=S(I)+L(I)
15 SUM1=SUM1+S(I)**2
   SUM1=DSQRT(SUM1)
C SOLVE FOR THE NEW GROUP ADJOINT FLUX VECTORS; PSI:
C THERMAL ADJOINT FLUX:
   CALL SOLV3D(N,L2,PSI2,Z)
C FAST ADJOINT GROUP; SOURCE VECTOR:
DO 27 I=1,N
   S(I)=0.
   Z(I)=0.
   IO=1
   II=3
   IF (I.EQ.1) IO=2
   IF (I.EQ.N) II=2
   DO 26 J=10,11
      K=1-2+J
      S(I)=S(I)+CHI(1)*F1(I,J)*PHI(1,K)+CHI(2)*F1(I,J)*PSI2(K)
26   Z(I)=Z(I)+T(I,J)*PSI2(K)
27   Z(I)=Z(I)+S(I)
C FAST ADJOINT FLUX:
   CALL SOLV3D(N,L1,PSI1,Z)
C FORM NEW GROUP TOTAL SOURCE S FROM PSI'S, AND ITS L-2 NORM SUM2:
   SUM2=0.
   DO 29 I=1,N
      S(I)=0.
      IO=1
      II=3
      IF (I.EQ.1) IO=2
      IF (I.EQ.N) II=2
      DO 28 J=10,11
         K=1-2+J
   28   S(I)=S(I)+L(I)
29   SUM2=SUM2+S(I)**2
   SUM2=DSQRT(SUM2)
28 \[ S(I) = S(I) + (F1(I,J) + F2(I,J)) \times (\chi(1) \times \psi(I,K) + \chi(2) \times \psi(2,K)) \]
29 \[ \text{SUM2} = \text{SUM2} + S(I)^2 \]
30 \[ \text{SUM2} = \text{DSQRT} \left( \text{SUM2} \right) \]

C CALCULATION OF THE EIGENVALUE:
31 \[ \lambda = \text{SUM2} / \text{SUM1} \]
32 \[ \lambda_{\text{STR}}(ICUT) = \lambda \]
33 \[ \text{ERRLAM} = \text{DABS} \left( \lambda - \lambda_{84} \right) \]

C PUT \( \psi_1 \) AND \( \psi_2 \) INTO BIGGER PSI:
34 \[ \text{DO} \ 30 \quad I = 1, N \]
35 \[ \psi_{1}(1, I) = \psi_{11}(I) \]
36 \[ \psi_{2}(1, I) = \psi_{12}(I) \]

C POINT BY POINT SIMULTANEOUS RELAXATION FLUX ITERATION:
37 \[ X = \text{ALPHA} \]
38 \[ \text{IF} \ (ICOUT .LE. 3) \quad X = 1.0 \]

C CALCULATE THE NEW GROUP FLUX ITERATES AND GROUP ERRORS:
39 \[ \text{DO} \ 40 \quad IG = 1, 2 \]
40 \[ \text{SQ}(IG) = 0. \]
41 \[ \text{DO} \ 40 \quad I = 1, N \]
42 \[ \text{ERROR}(IG, I) = \psi(IG, I) / \lambda - \phi(IG, I) \]
43 \[ \text{SQ}(IG) = \text{SQ}(IG) + \text{ERROR}(IG, I)^2 \]
44 \[ \phi(IG, I) = \phi(IG, I) + X \times \text{ERROR}(IG, I) \]

C AND FOR PLOTTING PURPOSES:
45 \[ \psi(IG, I) = \phi(IG, I) \]
46 \[ \text{CONTINUE} \]
47 \[ \text{DO} \ 34 \quad IG = 1, 2 \]
48 \[ \text{SQ}(IG) = \text{DSQRT} \left( \text{SQ}(IG) \right) \]

C NORMALIZE PSI:
49 \[ \text{NORMALIZES BOTH ARRAY GROUPS TO 1.0:} \]
50 \[ \text{CALL NORM2} \left( \text{PSI}, N \right) \]
51 \[ \text{DO} \ 36 \quad IG = 1, 2 \]
52 \[ \text{ERRMAX}(IG) = \text{THE MAX ERROR BETWEEN THE GROUP ITERATION FLUXES:} \]
53 \[ \text{ERRMAX}(IG) = \text{ERROR}(IG, 1) \]
54 \[ \text{DO} \ 36 \quad I = 2, N \]
55 \[ \text{IF} \ (\text{DABS} \left( \text{ERROR}(IG, I) \right) \times \text{ERRMAX}(IG)) \]
56 \[ \text{ERRMAX}(IG) = \text{DABS} \left( \text{ERROR}(IG, I) \right) \]
57 \[ \text{CONTINUE} \]
IF (IPLT.NE.2) GO TO 45

C THE FOLLOWING IS FOR NICELY PLOTTING THE GROUP Flux HISTORY.
DO 41 IG=1,2
DO 41 I=1,N
41 ERROR(IG,I)=PSI(IG,I)
C ERROR NOW CONTAINS THE NEW NORMALIZED FLUX ITERATE PHI.
JK=IK
IF (IK.EQ.0) JK=5
DO 42 IG=1,2
DO 42 I=1,N
IF (DABS(ERROR(IG,I)-PHISTR(IG,I,JK+1)).GE.0.01) GO TO 43
42 CONTINUE
C FLUX HAS NOT CHANGED ENOUGH FOR PLOTTING.
GO TO 45
C SAVE THE NORMALIZED FLUX FOR PLOTTING:
43 IK=IK+1
IN(IK)=ICOUT
TE3(IK)=ERRLAM
DO 44 IG=1,2
TE1(IG,IK)=ERRMAX(IG)
TE2(IG,IK)=SQ(IG)
DO 44 I=1,N
44 PHISTR(IG,1,IK+1)=ERROR(IG,I)
IF (IK.NE.5) GO TO 45
C PLOT THE LAST FIVE SAVED FLUXES:
CALL PHIPLT(5)
IK=0
45 CONTINUE
C ERROR CRITERIA FOR ACCEPTANCE OF CONVERGENCE.
IFLAG1=0
IFLAG2=0
IFLAG3=0
C STORE THE ERRORS FOR COMPARISON:
C ERROR BETWEEN ITERATION EIGENVALUES:
ERRLAM(1CDJT)=ERRLAM
DO 46 IG=1,2

PAGE 223
MAXIMUM ERROR BETWEEN ITERATION FLUXES:
\[ \text{EFSTR}(I,G,ICOUT) = \text{ERRMAX}(IG) \]

MEAN SQUARE ERROR BETWEEN ITERATION FLUXES:
\[ \text{EFMSTR}(I,G,ICOUT) = \text{SQ}(IG) \]

CONTINUE
\[
\begin{align*}
\text{IF} & \ ((\text{ERRMAX}(1) \cdot \text{LT.} \cdot \text{EPS1}) \\text{AND} \ (\text{ERRMAX}(2) \cdot \text{LT.} \cdot \text{EPS1})) & \text{IFLAG1} = 1 \\
\text{IF} & \ ((\text{SQ}(1) \cdot \text{LT.} \cdot \text{EPS2}) \\text{AND} \ (\text{SQ}(2) \cdot \text{LT.} \cdot \text{EPS2})) & \text{IFLAG2} = 1 \\
\text{IF} & \ (\text{ERRLAM} \cdot \text{LT.} \cdot \text{EPS3}) & \text{IFLAG3} = 1 \\
\text{IFLAG4} & = \text{IFLAG1} \cdot \text{IFLAG2} \cdot \text{IFLAG3} \\
\text{IF} & \ (\text{IFLAG4} \cdot \text{EQ}.1) & \text{GO TO 50} \\
\text{OTHERWISE CONTINUE THE ITERATION.} \\
\text{LAMB4} & = \text{LAMDA} \\
\text{GO TO 20} \\
\text{CONTINUE} \\
\text{CONVERGENCE ACCOMPLISHED.} \\
\text{CALL NORMAL(PHI,N)} \\
\text{PLOT ANY LEFT OVER FLUX HISTORY PLOTS:} \\
\text{IF} & \ ((\text{IPLT}. \text{EQ}.2) \\text{AND} \ (\text{IK} \cdot \text{NE}.0)) & \text{CALL PHIPLT(IK)} \\
\text{BOUNDARY CONDITION INSERTIONS.} \\
\text{IER} & = 0 \\
\text{IER ALLOWS B.C. INSERTIONS FOR YES AND NO CONVERGENCE:} \\
\text{IF} & \ (\text{IBC}. \text{EQ}.4) & \text{GO TO 90} \\
\text{IF} & \ (\text{IBC} \cdot \text{NE}.3) & \text{GO TO 60} \\
\text{PHI}(1,\text{KR}+1) & = 0. \\
\text{PHI}(2,\text{KR}+1) & = 0. \\
\text{GO TO 90} \\
\text{DO} & \ 70 \ \text{I}=1,N \\
\text{J} & = \text{N}+1-1 \\
\text{PHI}(1,J+1) & = \text{PHI}(1,J) \\
\text{PHI}(2,J+1) & = \text{PHI}(2,J) \\
\text{IF} & \ (\text{IBC}. \text{NE}.1) & \text{GO TO 90} \\
\text{PHI}(1,\text{KR}+1) & = 0. \\
\text{PHI}(2,\text{KK}+1) & = 0. \\
\text{CONTINUE} \\
\text{C} \\
\text{NORMALIZE THE CONVERGED FLUX VECTOR:} \\
\text{C} \\
\text{PLOT ANY LEFT OVER FLUX HISTORY PLOTS:} \\
\text{IF} & \ ((\text{IPLT}. \text{EQ}.2) \\text{AND} \ (\text{IK} \cdot \text{NE}.0)) & \text{CALL PHIPLT(IK)} \\
\text{C} \\
\text{BOUNDARY CONDITION INSERTIONS.} \\
\text{IER} & = 0 \\
\text{IER ALLOWS B.C. INSERTIONS FOR YES AND NO CONVERGENCE:} \\
\text{IF} & \ (\text{IBC}. \text{EQ}.4) & \text{GO TO 90} \\
\text{IF} & \ (\text{IBC} \cdot \text{NE}.3) & \text{GO TO 60} \\
\text{PHI}(1,\text{KR}+1) & = 0. \\
\text{PHI}(2,\text{KR}+1) & = 0. \\
\text{GO TO 90} \\
\text{DO} & \ 70 \ \text{I}=1,N \\
\text{J} & = \text{N}+1-1 \\
\text{PHI}(1,J+1) & = \text{PHI}(1,J) \\
\text{PHI}(2,J+1) & = \text{PHI}(2,J) \\
\text{IF} & \ (\text{IBC}. \text{NE}.1) & \text{GO TO 90} \\
\text{PHI}(1,\text{KR}+1) & = 0. \\
\text{PHI}(2,\text{KK}+1) & = 0. \\
\text{CONTINUE}
90 IF (IER.EQ.1) GO TO 102
   RETURN
C NO CONVERGENCE ACCOMPLISHED:
100 CONTINUE
C NORMALIZE THE UNCONVERGED FLUX:
   CALL NORMAL(PHI,N)
   ICOUT=ICOUT-1
   WRITE (6,1J1) ICOUT
   FORMAT (1*1, 'POWER METHOD DID NOT CONVERGE FOR THIS CASE AFTER',
          14,'ITERATIONS. ',//,1X,'EXECUTION TERMINATED '),
          IER=1
   GO TO 55
102 CONTINUE
C FOR PRINTING OUT THE EIGENVALUE HISTORY AND THE FINAL FLUX SHAPE:
   IF (IPLJT.EQ.0) IJPLT=1
   IF (JPLUT.EQ.0) JPLOUT=1
   RETURN
END
SUBROUTINE CUR7
C
SOLVES FOR THE ADJOINT CURRENT FROM THE INPUT H(K)'S AND D(K)'S
USING F(K)'S FROM POWER7:
CURRENT IS LINEAR (LEAST SQUARES - VARIATIONAL) AND PUT INTO ARRAY C.
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /B2/ KR
COMMON /B47/ F(2,201), C(2,201)
COMMON /B5/ T(201,3), S1(201), S2(201), C1(201), C2(201)
COMMON /B7/ H(200), D(2,200)
C
FROM THE MATRIX PROBLEM FOR LINEAR FIT OF STEP DATA:
M=KR
N=KR+1
T(1,1)=0.
T(N,3)=0.
T(1,2)=H(1)/3.
T(1,3)=H(1)/6.
T(N,1)=H(M)/6.
T(N,2)=H(M)/3.
S1(1)=D(1,1)*(F(1,1)-F(1,2))/2.
S2(1)=D(2,1)*(F(2,1)-F(2,2))/2.
S1(N)=D(1,M)*(F(1,M)-F(1,M+1))/2.
S2(N)=D(2,M)*(F(2,M)-F(2,M+1))/2.
DO 20 1=2,M
J=1-1
T(I,1)=H(J)/6.
T(I,2)=(H(J)+H(I))/3.
T(I,3)=H(I)/6.
S1(I)=(D(1,J)*(F(1,J)-F(1,1))+D(1,1)*(F(1,1)-F(1,I+1)))/2.
S2(I)=(D(2,J)*(F(2,J)-F(2,1))+D(2,1)*(F(2,1)-F(2,I+1)))/2.
20 CONTINUE
CALL SOLV3D(N,T,C1,S1)
CALL SOLV3D(N,T,C2,S2)
DO 30 I=1,N
C(I,1)=-C1(I)
30 C(2,I)=-C2(I)
RETURN
SUBROUTINE OUTPT7
C PRINTS THE RESULTS OF THE ADJOINT METHOD:
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /81/ IBC,IPLOT,JPLOT,IPUNCH
COMMON /82/ KR,N
COMMON /847/ P,K(2,201),CUR(2,201),LAMDA,ICOUT
COMMON /ER/ EPS1,EPS2,EPS3
COMMON /ESTR/ LAMSTR(300),EFSTR(2,300),EFMSTR(2,300),ERLAM(300)
INTEGER N
KRO=KR-1
KRI=KR+1
WRITE (6,11)
FORMAT ('IRESULTS OF THE MULTIGROUP ADJOINT METHOD:')
WRITE (6,10)
FORMAT ('PROBLEM TERMINATED AFTER',I5,
X 'OUTER (POWER) ITERATIONS TO:')
WRITE (6,20)
FORMAT ('ADJOINT LAMBDA = ',1PE21.14)
C PRINT OUT EIGENVALUES.
CALL PLOT7
WRITE (6,30)
FORMAT ('IRESULTS AFTER PROBLEM TERMINATION:',/,,
X 'NUMBER',9X,'THERMAL FLUX',4X,'THERMAL CURRENT',12X,
X 'FAST FLUX',7X,'FAST CURRENT',/)
WRITE (6,50) (K,PHI(2,K),CUR(2,K),PHI(1,K),CUR(1,K),K=1,KR1)
FORMAT (I7,IPE21.7,OPE19.7,1PE21.7,OPE19.7)
C PRINT OUT THE STORED ITERATION ERRORS:
WRITE (6,110) EPS1,(EFSTR(2,I),I=1,ICOUT)
WRITE (6,111) EPS1,(EFSTR(1,I),I=1,ICOUT)
WRITE (6,112) EPS2,(EFMSTR(2,I),I=1,ICOUT)
WRITE (6,113) EPS3,(EFMSTR(1,I),I=1,ICOUT)
WRITE (6,114) EPS3,(ERLAM(I),I=1,ICOUT)
FORMAT ('Imaximum errors between the thermal flux iterations:',
X 'Tolerance used = ',1PE12.4,/, (1P5E20.5))
FORMAT ('Imaximum errors between the fast flux iterations:',
X 'Tolerance used = ',1PE12.4,/, (1P5E20.5))
X 25X,'TOLERANCE USED = ',1PE12.4,, (1P5E20.5))
112 FORMAT ('MEAN SQUARE ERROR BETWEEN THE THERMAL FLUX ITERATIONS:','
X 18X,'TOLERANCE USED = ',1PE12.4,, (1P5E20.5))
113 FORMAT ('MEAN SQUARE ERROR BETWEEN THE FAST FLUX ITERATIONS:','
X 18X,'TOLERANCE USED = ',1PE12.4,, (1P5E20.5))
114 FORMAT ('ERROR BETWEEN THE ITERATION EIGENVALUES:','
X 28X,'TOLERANCE USED = ',1PE12.4,, (1P5E20.5))
C CHECK FOR CALL TO PUNCH:
IF (IPUNCH.EQ.1) CALL PUNCH
RETURN
END
SUBROUTINE PLOT7
C
C PLOTS OUT THE EIGENVALUE HISTORY AS A TABLE AND A GRAPH,
C AS WELL AS PLOTTING OUT THE FINAL MULTIGROUP FLUX SHAPES.
C FOR THE ADJOINTS:
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /B1/ IBC,IPLOT,JPLUT,IPUNCH
COMMON /B2/ KR
COMMON /B47/ PHI(2,201), CUR(2,201), LAMDA, ICOUT
COMMON /B5/ B(300,2)
COMMON /ESTR/ LAMSTR(300)
DIMENSION C(201,3)
C
IN ORDER TO SAVE SOME SPACE:
EQUIVALENCE (B(1),C(1))
INTEGER ND
ND=201
WRITE (6,1)
1 FORMAT ('OTABLE OF EIGENVALUES DURING THE POWER ITERATION:',
X //,(1P5E25.14))
C
IF (JPLUT.EQ.0) GO TO 20
DO 10 I=1,ICOUT
B(1,1)=I
10 B(1,2)=LAMSTR(I)
CALL PRPTLT(1,B,ICOUT,2,ICOUT,0,300,2,1)
WRITE (6,11)
11 FORMAT ('OPLOT OF THE EIGENVALUE HISTORY THROUGH THE ITERATIONS.')
20 IF (JPLUT.EQ.0) RETURN
KR1=KR+1
DO 30 I=1,KR1
C(1,1)=I
C(1,2)=PHI(1,1)
30 C(1,3)=PHI(2,1)
CALL PRPTLT(2,C,KR1,3,KR1,0,ND,3,2)
WRITE (6,31)
31 FORMAT ('QFINAL CONVERGED CCNNECTING FLUX POINTS; F(K).',//,
X 5X,'FAST FLUX: .',//,5X,'THERMAL FLUX: -')
SUBROUTINE SOLV3D(N, A, X, Y)
C SOLVES THE N DOUBLE PRECISION MATRIX EQUATIONS: A*X = Y,
C FOR X - GIVEN THE N BY N TRIDIAGONAL MATRIX A
C AND THE SOURCE VECTOR Y.
C METHOD IS FORWARD ELIMINATION FOLLOWED BY BACKWARD SUBSTITUTION.
C CF - WACHPRESS, PAGE 23.
REAL*8 A, X, Y, H, P, D
DIMENSION A(201,3), X(201), Y(201), H(201), P(201)
IF (A(1,2).EQ.0.0) GO TO 10
H(1)=-A(1,3)/A(1,2)
P(1)=Y(1)/A(1,2)
DO 1 M=2,N
D=A(M,2)+A(M, 1)*H(M-1)
IF (D.EQ.0.0) GO TO 20
P(M)=(Y(M)-A(M, 1)*P(M-1))/D
IF (M.EQ.N) GO TO 1
H(M)=-A(M, 3)/D
1 CONTINUE
X(N)=P(N)
DO 2 I=2,N
M=N+1-I
2 X(M)=P(M)+H(M)*X(M+1)
RETURN
C IN CASE OF ANY IMPENDING ZERO DIVISORS:
10 WRITE (6,11)
11 FORMAT ('0FIRST ELEMENT OF A, A(1,1), IS ZERO.',/,X
5X,'BETTER FIX IT BOSS.'))
GO TO 30
20 WRITE (6,21) M
21 FORMAT ('0ZERO DIVISOR ENCOUNTERED IN EQUATION M =',I3,'.',/,X
5X,'BETTER FIX IT BOSS.'))
30 WRITE (6,31)
31 FORMAT ('0EXECUTION TERMINATED.')
CALL EXIT
RETURN
END
SUBROUTINE NORMAL(PHI,N)

NORMALIZES THE GROUP FLUXES TO ONE. NOT BOTH GROUPS.

REAL*8 PHI(2,201), A

A=DABS(PHI(1,1))
DO 1 IG=1,2
DO 1 I=1,N

IF (DABS(PHI(IG,I)).GT.A) A=DABS(PHI(IG,I))

1 CONTINUE
DO 2 IG=1,2
DO 2 I=1,N

2 PHI(IG,I)=PHI(IG,I)/A
RETURN
END
SUBROUTINE NORM2(PSI,N)
C
NORMALIZES BOTH ENERGY GROUPS OF PSI TO 1.0.
REAL*8 PSI(2,2D1), A(2)
DO 1 IG=1,2
   A(IG)=DABS(PSI(IG,1))
DO 1 I=1,N
   IF (DABS(PSI(IG,I)).GT.A(IG)) A(IG)=DABS(PSI(IG,I))
1 CONTINUE
DO 2 IG=1,2
   A(IG)=DABS(PSI(IG,I))
2 PSI(IG,I)=PSI(IG,I)/A(IG)
RETURN
END
SUBROUTINE PHIPLT(L)
C      PLOTS THE GROUP FLUX HISTORY, WITH UP TO 5 GROUP FLUXES PER PLOT.
C      FAST AND THERMAL GROUP FLUXES ARE PLOTTED SEPARATELY.
C      L IS THE NUMBER OF FLUXES TO BE PLOTTED.
C      L IS BETWEEN 1 AND 5.
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /B1/ IBC
COMMON /B2/ KR,N
COMMON /B5/ S(201), A(201,6), B(201,6)
COMMON /B6/ TEL(2,5),TE2(2,5),TE3(5),IN(5)
COMMON /E3/ EPS1,EPS2,EPS3
COMMON /FSTR/ PHISTR(2,201,6)
DIMENSION SYMBOL(5)
INTEGER SYMaOL
ND=201
KR1=KR+1
C      SET UP B.C. CONDITIONS
IF (IBC.EQ.4) GO TO 5
IF (IBC.EQ.3) GO TO 3
DO 2 IG=1,2
DO 2 K=1,L
DO 1 I=1,N
J=N+1-I
1 PHISTR(IG,J+1,K+1)=PHISTR(IG,J,K+1)
2 PHISTR(IG,1,K+1)=0.
3 IF (IBC.EQ.2) GO TO 5
DO 4 IG=1,2
DO 4 K=1,L
4 PHISTR(IG,KRI,K+1)=0.
5 CONTINUE
C      FLUXES IN PHISTR HAVE BEEN NORMALIZED IN POWER.
C      PUT THE FAST FLUX IN A, AND THE THERMAL FLUX IN B:
   LI=L+1
   DO 10 K=1,LI
   DO 10 I=1,KRI
   A(I,K)=PHISTR(1,I,K)

PAGE 235
10 B(I,K)=PHISTR(I,2,I,K)
C PLOT THE L FAST FLUX SHAPES ON ONE GRAPH:
   CALL PRTPLT(0,1,KR1,L1,KR1,0,ND,6,2)
   WRITE (6,20)
20 FORMAT (/,'FAST FLUX ITERATION HISTORY PLOT.',/)
   WRITE (6,30)
30 FORMAT (X 'KEY:',5X,'SYMBOL',5X,'ITERATION NUMBER:',7X,'ERROR CRITERIA',
           X 11X,'ERROR',13X,'TOLERANCE')
   DO 35 I=1,L
35 WRITE (6,40) SYMBOL(I),IN(I),TE1(1,I),EPS1,TE2(1,I),EPS2,
            TE3(I),EPS3
40 FORMAT (/,12X,A1,15X,I3,16X,'FLUX',14X,1PD15.5,5X,1PD15.5,/,X 47X,'MEAN SQ. FLUX',5X,1PD15.5,5X,1PD15.5,/,X 47X,'EIGENVALUE',8X,1PD15.5,5X,1PD15.5)
C C PLOT THE L THERMAL FLUX SHAPES ON THE OTHER GRAPH:
   CALL PRTPLT(0,1,KR1,L1,KR1,0,ND,6,2)
   WRITE (6,50)
50 FORMAT (/,'OTHERMAL FLUX ITERATION PLOT.',/)
   WRITE (6,30)
   DO 55 I=1,L
55 WRITE (6,40) SYMBOL(I),IN(I),TE1(2,I),EPS1,TE2(2,I),EPS2,
            TE3(I),EPS3
   RETURN
END
SUBROUTINE PRTPLT(NO,B,N,M,NL,NS,KX,JX,ISP)
C*****MODIFIED VERSION FROM THAT OF SSP OR ANY OTHER SOURCE *****
C CONVERTS DOUBLE PRECISION B ARRAY TO REAL*4.
C PLOT SEVERAL CROSS-VARIABLES VERSUS A BASE VARIABLE
C NO - CHART NUMBER (3 DIGITS MAXIMUM)
C B - MATRIX OF DATA TO BE PLOTTED. FIRST COLUMN REPRESENTS
C BASE VARIABLE AND SUCCESSIVE COLUMNS ARE THE CROSS-
C VARIABLES (MAXIMUM IS 9).
C N - NUMBER OF ROWS IN MATRIX B
C M - NUMBER OF COLUMNS IN MATRIX B (EQUAL TO THE TOTAL
C NUMBER OF VARIABLES). MAXIMUM IS 10.
C NL - NUMBER OF LINES IN THE PLOT. IF 0 IS SPECIFIED, 50
C LINES ARE USED. THE NUMBER OF LINES MUST BE EQUAL TO
C OR GREATER THAN N
C (USUALLY USE NL=N, AND ISP FOR SPACING.)
C NS - CODE FOR SORTING THE BASE VARIABLE DATA IN ASCENDING
C ORDER
C 0 SORTING IS NOT NECESSARY (ALREADY IN ASCENDING
C ORDER).
C 1 SORTING IS NECESSARY.
C KX- DIMENSION OF B MATRIX FROM DIMENSION STATEMENT.
C IT MUST BE OF THE FORM B(KX,JX)
C JX- DIMENSION OF B MATRIX FROM DIMENSION STATEMENT.
C IT MUST BE OF THE FORM B(KX,JX)
C ISP- CODE FOR SPACING LINES WHILE PLOTTING:
C 1 SINGLE SPACE
C 2 DOUBLE SPACE
C 3 TRIPLE SPACE
C ...ETC.
REAL*8 B
DIMENSION OUT(101),YPR(11),IANG(9),A(1500),B(KX,JX)
INTEGER IDJM/*1*/,IANG/'.','-','+','*','A','B','C','D'/
INTEGER OUT
I=1
DO 39 J=1,M
DO 39 K=1,N
A(I) = B(K, J)
I = I + 1
39 CONTINUE
1 FORMAT(1H1, 60X, 7H CHART , 13, //)
2 FORMAT(1H, F11.4, 5X, 101A1)
3 FORMAT(1H )
7 FORMAT(1H, 16X, 101H+ + + + + + + + + )
8 FORMAT(1H, 9X, 11F10.4)
NLL = NL
IF(NS) 16, 16, 10
C SORT BASE VARIABLE DATA IN ASCENDING ORDER
10 DO 15 I = 1, N
14 DO 12 J = 1, N
IF(A(I) - A(J)) 14, 14, 11
11 L = I - N
LL = J - N
DO 12 K = 1, M
L = L + N
LL = LL + N
F = A(L)
A(L) = A(LL)
12 A(LL) = F
14 CONTINUE
15 CONTINUE
C TEST NLL
16 IF(NLL) 20, 18, 20
18 NLL = 50
C PRINT TITLE
20 WRITE(6, 1) ND
C DEVELOP BLANK AND DIGITS FOR PRINTING
BLANK = 0
C FIND SCALE FOR BASE VARIABLE
XSCAL = (A(N) - A(1)) / (FLOAT(NLL - 1))
C FIND SCALE FOR CROSS-VARIABLES
YMIN = 1.0E75
YMAX=-1.0E75
M1=N+1
M2=M*N
DO 40 J=M1,M2
   IF (A(J).GT.YMAX) YMAX=A(J)
   IF (A(J).LT.YMIN) YMIN=A(J)
40 CONTINUE
YSCAL=(YMAX-YMIN)/100.0
C CHECK TO SEE IF THE SPREAD IN Y IS TOO SMALL FOR PLOTTING:
   IF (YSCAL.EQ.0.0) GO TO 100
C OTHERWISE, A DIVIDE CHECK WILL OCCUR AFTER STATEMENT 56.
C FIND BASE VARIABLE PRINT POSITION
XB=A(1)
L=1
MY=M-1
IT=ISP-1
DO 80 I=1,NLL
   F=I-1
   XPR=XB+F*XSCAL
   IF(A(L)-XPR-XSCAL*0.5) 50,50,70
50 CONTINUE
C FIND CROSS-VARIABLES
55 DO 55 IX=1,101
   OUT(IX)=BLANK
57 CONTINUE
56 LL=L+J*N
   JP=((A(LL)-YMIN)/YSCAL)+1.0
   OUT(JP)=1ANG(J)
60 CONTINUE
C PRINT LINE AND CLEAR, OR SKIP
   IF(L.EQ.N) GO TO 61
   L=L+1
   IF(A(L)-XPR-XSCAL*0.5) 57,57,61
61 CONTINUE
WRITE(6,2)XPR,(OUT(IZ),IZ=1,101)
   IF (IT.EQ.0) GO TO 65
DO 64 IV=1,IT
 64 WRITE (6,3)
 65 GO TO 80
 70 WRITE(6,3)
 80 CONTINUE
C PRINT CROSS-VARIABLES NUMBERS
  WRITE(6,7)
  YPR(1)=YMIN
  DO 90 KN=1,9
  90 YPR(KN+1)=YPR(KN)+YSCALE*10.0
  10 YPR(11)=YMAX
  WRITE(6,8)(YPR(IP),IP=1,11)
  RETURN
100 WRITE (6,101)
101 FORMAT ('UNPLOT IS GENERATED BECAUSE THE SPREAD IN THE Y VARIABLE
X IS TOO SMALL.',/10X,'(I.E. - EQUALS ZERO UNDER REAL*4.)',/,
X 5X, 'EXECUTION CONTINUING.')
  RETURN
END
SUBROUTINE PUNCH
  C    PUNCHES THE FLUX AND CURRENT AND ADJOINTS OUT AFTER CONVERGENCE.
  C    CALLED BY IPUNCH=1.
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /B2/ KR
  COMMON /B4/ F(2,201), C(2,201)
  COMMON /B47/ F7(2,201), C7(2,201)
  WRITE (7,1) KR
  1 FORMAT (15)
      N=KR+1
  C    PUNCH OUT THE FAST FLUX:
  WRITE (7,10) (F(1,J), C(1,J), F7(1,J), C7(1,J), J=1,N)
  C    PUNCH OUT THE THERMAL FLUX:
  WRITE (7,10) (F(2,J), C(2,J), F7(2,J), C7(2,J), J=1,N)
  10 FORMAT (4E20.7)
RETURN
END
F.2. SOURCE LISTING of Program LINEAR
Figure F.2. Structure of Program LINEAR.
C PROGRAM LINEAR:
C TWO GROUP PROPOSED METHOD USING LINEAR BASIS FUNCTIONS.
CALL TIMING(11)
CALL SYNT
CALL TIMING(14)
CALL POWER
CALL TIMING(16)
CALL TIMING(17)
CALL OUTPUT
CALL TIMING(18)
C TIMING EXECUTION
WRITE (6,30)
30 FORMAT (1H1,'TIMING PROGRAM EXECUTION:','/)
J=14-11
WRITE(6,701) J
J=16-14
WRITE(6,704) J
J=17-16
WRITE(6,706) J
J=18-17
WRITE(6,707) J
701 FORMAT (1H1,' SYNT HAS TAKEN',I6,' /100 SECONDS. ')
704 FORMAT (1H1,' POWER HAS TAKEN',I6,' /100 SECONDS. ')
706 FORMAT (1H1,' CURENT HAS TAKEN',I5,' /100 SECONDS. ')
707 FORMAT (1H1,' OUTPUT HAS TAKEN',I5,' /100 SECONDS. ')
CALL TIMING(120)
J=120-11
WRITE(6,720) J
720 FORMAT (1H1,' THIS RUN HAS TAKEN',I6,' /100 SECONDS TO RUN. ')
STOP
END
SUBROUTINE SYNTH
C
C  PROPOSED LINEAR SYNTHESIS METHOD:
C
C  ADJOINT QUANTITIES OF VARIABLES ARE DENOTED BY \( \tilde{\text{7}} \) RATHER THAN *.
C
C  THUS: \( \tilde{\phi} \) (RATHER THAN \( \phi^* \)) IS THE ADJOINT OF \( \phi \). ETC.
IMPLICIT REAL*8 (A-H,K-Z)
COMMON /B1/ IBC, IPLOT, JPLLOT, IPUNCH, ISEE
COMMON /B2/ KR, NN
COMMON /B3/ L1(26,26), L2(26,26), F1(26,26), F2(26,26), X
  F3(26,26), F4(26,26), T(26,26),
   COMMON /B5/ KA0(2,25), KA1(2,25), KA2(2,25), KB0(2,25), KB1(2,25),
   X  KB2(2,25), LA0(2,25), LA1(2,25), LA2(2,25), SR0(1,25),
   X  SR1(1,25), SR2(1,25), KC0(1,25), KC1(1,25), KC2(1,25),
   X  KD0(1,25), KD1(1,25), KD2(1,25),
   X  P(1,2,25), PT(1,2,25), P(1,2, 25),
   X  PH7(2,25), AL(2,25), BL(2,25), CL(2,25), AF(4,25),
   X  BF(4,25), CF(4,25), AT(25), BT(25), CT(25),
   X  AL(2,25), BL(2,25), AF(4,25), BF(4,25), ATK(2),
   X  CO(2), CH(2),
   COMMON /CHIF/ CHI(2),
   COMMON /XAXIS/ HX, HR(25),
   COMMON /BH/ X(101), HI(101),
   COMMON /ER/ EPS1, EPS2, EPS3
   DIMENSION PHI(2,101), PHI7(2,101), CUR(2,101), CUR7(2,101),
   X  A(2,100), F(2,100), D(2,100), S(2,100), D(2,100),
   X  XU(2,100)
   DIMENSION V(2), V1(2), V2(2), V3(2)
   DIMENSION ITF(25), KTF(25)
   IN ORDER TO SAVE SPACE:
   EQUIVALENCE (PHI(1), L1(1)), (PHI7(1), L1(301)),
   X  (CUR(1), L2(1)), (CUR7(1), L2(301)),
   X  (XU(1), F1(1)), (A(1), F1(301)),
   X  (F1, F2(1)), (D(1), F2(301)),
   X  (S(1), T(1)), (D(1), T(301))
REAL TITLE(20) INTEGER KR,K,KS,KS1,KRO,NN,NUMITF,KTF READ (5,200) TITLE 200 FORMAT (20A4) WRITE (6,201) TITLE 201 FORMAT (1HL,20A4,/) C READ IN THE NUMBER OF REGION TRIAL FUNCTIONS AND TYPE OF B.C.S. C AS WELL AS THE TOLERANCES AND THE OUTPUT TYPES DESIRED: READ (5,1) KR,IBC,EPS1,EPS2,EPS3,IPLCT,JPLOT,IPUNCH,ISEE,ITW,ITC 1 FORMAT (215,3D10.3,6I5) IF (IBC.EQ.3) IBC=2 C READ IN THE TYPE-NUMBER OF EACH TF REGION: READ (5,100) (ITF(I),I=1,1,1) 100 FORMAT (25I2) C READ IN THE FISSION YIELDS FOR EACH GROUP: READ (5,101) CHI(1), CHI(2) 101 FORMAT (2F10.5) KRO=KR-1 WRITE (6,2) KR, IBC, ISEE, ITW, ITC 2 FORMAT ('DUNE DIMENSIONAL TWO GROUP LINEAR SYNTHESIS PROGRAM:',,,) X 5X,'NUMBER OF COARSE MESH REGIONS: KR = ',I2,,/ X 5X,'BOUNDARY CONDITION NUMBER: IBC = ',I2,,/ X 5X,'AMOUNT OF OUTPUT REQUESTED: ISEE = ',I2,,/ X 5X,'TYPE OF WEIGHTING FUNCTIONS: ITW = ',I2,,/ X 5X,'TYPE OF CURRENT FUNCTIONS: ITC = ',I2,,/ X 5X,'REGIONAL INPUT MATERIAL PROPERTIES AND FLUX SHAPES FOLLOW', X /*5X,'IF ISEE > 0:','/ X 5X,'FLUX SHAPES ARE LINEAR IN EACH INDICATED SUBREGION.') IF (ITC.EQ.0) WRITE (6,16) IF (ITC.EQ.1) WRITE (6,17) 16 FORMAT (5X,'CURRENTS ARE CONSTANT IN EACH INDICATED SUBREGION.') 17 FORMAT (5X,'CURRENTS ARE LINEAR IN EACH INDICATED SUBREGION.') IF (ITW.EQ.0) WRITE (6,116) IF (ITW.EQ.1) WRITE (6,117) 116 FORMAT (5X,'WEIGHTING FLUX = FLUX;',5X,'WEIGHTING CURRENT = - XCURRENT.')
FORMAT (/,5X,'WEIGHTING FLUX = ADJOINT FLUX;',/5X,'WEIGHTING CURR XENT = ADJOINT CURRENT.')
WRITE (6,20) EPS1,EPS2,EPS3,IPLT,JPLT,IPUNCH
20 FORMAT (/,'TOLERANCES TO POWER ARE: EPS1 = ',1PD10.3,/,,
     28X,'EPS2 = ',1PD10.3,/,28X,'EPS3 = ',1PD10.3,/
     34X,'IPLOT = ',II,/,34X,'JPLT = ',II)
WRITE (6,22) CHI(1), CHI(2)
22 FORMAT (/,,'OFISSION YIELDS ARE: CHI(1) = ',F10.5,/,,
     22X,'CHI(2) = ',F10.5)
IF ((KR.LE.2).AND.(IBC.EQ.1)) CALL ERROR(1,KR)
IF (KR.GT.25) CALL ERROR(2,KR)
IF (EPS1.LT.1.0E-16) CALL ERROR(6,1)
IF (EPS2.LT.1.0E-16) CALL ERROR(6,2)
IF (EPS3.LT.1.0E-16) CALL ERROR(6,3)
IF (IBC.LT.1).OR.(IBC.GT.7)) CALL ERROR(7,IBC)
C DUMMY NORMAL VECTOR XU = UNITY. (FOR THE INTEGRATION FUNCTIONS)
DO 21 IG=1,2
DO 21 II=1,100
21 XU(IG,II)=1.0
ITC0=2
ITC1=2
IF (ITC.EQ.1) GO TO 23
ITC0=0
ITC1=1
C COUNTER OF THE NUMBER OF TYPE-NUMBERS OF EACH TF REGION:
NUMITF=1
HX=0.0
C BEGIN TO READ IN THE TF REGION DATA AND FILL THE ARRAYS,
C DEPENDING ON THE TYPE-NUMBER OF EACH TF REGION.
DO 50 I=1,KR
IF (ITF(I).EQ.NUMITF) GO TO 110
C FILL THE ARRAYS FROM OLD TF REGION TYPES:
J=ITF(I)
CALL REPEAT(I,KTF(J))
GO TO 50
C READ IN THE TF REGION'S DATA FOR NEW TF REGION TYPE-NUMBERS:
110 NUMITF=NUMITF+1
   KTF(NUMITF-1)=1
C READ THE SUBREGION NUMBER AND THE NUMBER OF REGIONS IN THE SUBREGION:
   READ (5,1) K, KS
   IF (KS.GT.100) CALL ERROR(3,1)
C CHECK FOR IMPROPER SEQUENCING OF INPUT DATA:
   IF (I.NE.K) CALL ERROR(4,1)
C READ IN THE GEOMETRY AND THE MATERIAL PROPERTIES:
   READ (5,3) (X(J),X(J+1),H(J),A(1,J),F(1,J),D(1,J),S(1,J),
   X A(2,J),F(2,J),D(2,J),J=1,KS)
3 FORMAT (3F10.5,4D10.3,/,3JX,3010.3)
C READ IN THE REGIONAL GROUP TRIAL FUNCTIONS:
   KS1=KS+1
   READ (5,4) (PHI(1,J),CUR(1,J),PHI7(1,J),CUR7(1,J),J=1,KS1)
   READ (5,4) (PHI(2,J),CUR(2,J),PHI7(2,J),CUR7(2,J),J=1,KS1)
4 FORMAT (4D20.7)
   IF (ITW.EQ.1) GO TO 120
C FORM WEIGHTING FUNCTIONS FROM THE GIVEN FUNCTIONS:
   DO 119 IG=1,2
   DO 119 J=1,KS1
   PHI7(IG,J)=PHI(IG,J)
119   CUR7(IG,J)=-CUR(IG,J)
   IF (ITG.EQ.1) GO TO 5
C FORM THE REGION CONSTANT CURRENTS:
   DO 7 IG=1,2
   DO 6 J=1,KS
   CUR(IG,J)=-D(IG,J)*(-PHI(IG,J)+PHI(IG,J+1))/H(J)
6   CUR7(IG,J)=+D(IG,J)*(-PHI7(IG,J)+PHI7(IG,J+1))/H(J)
   CUR(IG,KS1)=0.0
7   CUR7(IG,KS1)=0.0
C WRITING OUT THE INPUT INFORMATION:
   IF (ISEE.EQ.0) GO TO 14
5   WRITE (6,10) K,KR,KS,(J,X(J),X(J+1),H(J),A(1,J),F(1,J),D(1,J),
   X S(1,J),A(2,J),F(2,J),D(2,J),J=1,KS)
10   FORMAT (*1INPUT MATERIAL PROPERTIES FOR REGION NUMBER *',I3,
X 'OF THE ',I3,' USED.',//,
X 5X,'THIS REGION IS DIVIDED INTO ',I3,' HOMOGENEOUS SUBREGIONS A
X5X,'FAST GROUP CONSTANTS APPEAR FIRST:',//,
X 1 'SUBREGION #',5X,'INTERNAL BOUNDARIES',10X,'WIDTH',3X,
X 1 'TOTAL CX (1/CM)',3X,'FISSENC CX (1/CM)',6X,'DIFFUSION (CM)',
X 4X,'SCATT.CX (1/CM)',//,
X 5X,'I',11X,'X(I)',9X,'X(I+1)',,11X,'H(I)',13X,'A(IG,I)',13X,
X 'F(IG,1)',13X,'D(IG,1)',14X,'S(1,1)',,//,
X (16,3F15.4,4D20.8,/,51X,3D20.8))
C
DO 15 IG=1,2
15 WRITE (6,11) IG,K,KR(J,X(J),PHI(IG,J),CUR(IG,J),PH17(IG,J),
X CUR7(IG,J),J=1,KS1)
11 FORMAT ('INPUT TRIAL FUNCTIONS FOR GROUP',I2, ' FOR REGION',I3,
X ' OUT OF THE',I3,' USED:',//,
X ' INDEX',5X,'COORD',16X,'FLUX',13X,'CURRENT',8X,' WEIGHT FLUX',,
X 5X, 'WEIGHT CURRENT',,//,(16,F10.5,4D20.7))
14 CONTINUE
C
END OF THE IN-OUT SECTION:
C DEFINING MISC. ARRAYS FOR THE INTEGRATION FUNCTIONS:
C LENGTH OF THE SUBREGION: HT
HT=X(KS1)-X(1)
HR(K)=HT
HX=HX+HR(K)
C INVERSE JF THE D ARRAYS:
DO 13 J=1,KS
D1(1,J)=1./D(1,J)
13 D1(2,J)=1./D(2,J)
C FORMATION OF THE INTEGRATION GROUPS:
CALL BHSET(KS)
C DO FOR ALL ENERGY GROUPS:
DO 50 IG=1,2
KA0(IG,K)=GIF(0,IG,PHI7,IG,A,PHI,KS,2)
KA1(IG,K)=GIF(1,IG,PHI7,IG,A,PHI,KS,2)
KA2(IG,K)=GIF(2,IG,PHI7,IG,A,PHI,KS,2)
STORE THE TERMINAL POINTS FOR LATER USE:

P0(IG,K) = PHI(IG,1)
P07(IG,K) = PHI7(IG,1)
PHIG,K) = PHI(IG,KS1)
PH7(IG,K) = PHI7(IG,KS1)
IF (K.EQ.1) CO(IG) = CUR(IG,1)
IF (NUMITF-1.EQ.ITF(KR).AND.ITC.EQ.1) CH(IG) = CUR(IG,KS)
IF (NUMITF-1.EQ.ITF(KR).AND.DTC.EQ.1) CH(IG) = CUR(IG,KS1)

FOR THE OOF DIAGONAL MATRIX ELEMENTS:
IF (IG.EQ.2) GO TO 50
SR0(IG,K) = GIF(0,2,PHI7,1,S,PHI,KS,2)
SR1(IG,K) = GIF(1,2,PHI7,1,S,PHI,KS,2)
SR2(IG,K) = GIF(2,2,PHI7,1,S,PHI,KS,2)
KC0(IG,K) = GIF(0,1,PHI7,2,F,PHI,KS,2)
KC1(IG,K) = GIF(1,1,PHI7,2,F,PHI,KS,2)
KC2(IG,K) = GIF(2,1,PHI7,2,F,PHI,KS,2)
KD0(IG,K) = GIF(0,2,PHI7,1,F,PHI,KS,2)
KD1(IG,K) = GIF(1,2,PHI7,1,F,PHI,KS,2)
KD2(IG,K) = GIF(2,2,PHI7,1,F,PHI,KS,2)

50 CONTINUE
NUMITF = NUMITF - 1
WRITE (0,51) NUMITF
51 FORMAT ('THERE ARE ONLY',I3,' DIFFERENT TRIAL FUNCTION REGIONS.' )
WRITE (0,52) (1,ITF(I),I=1,KR)
52 FORMAT ('T O TABLE OF THE TRIAL FUNCTION NUMBER TYPES:',/)
PRINTS OUT THE /B5/ ARRAYS:

IF (ISEE.GE.2) CALL PRTOOUT(1)

NN IS THE MATRIX BLOCK SIZE:

NN=KR
IF (IBC.EQ.1 .OR. IBC.GE.6) NN=KR-1
IF (IBC.EQ.4) NN=KR+1

FORMATION OF THE COEFFICIENT VECTORS:

THE INTERIOR COEFFICIENTS:

IX=2
IY=KR
IF (IBC.EQ.5 .OR. IBC.EQ.7) IX=3
IF (IBC.EQ.1 .OR. IBC.GE.6) IY=KR-1
DO 60 IG=1,2
DO 60 K=IX,IY
J=K-1
V1(IG)=1./(PH7(IG,J)*P0(IG,J))
V2(IG)=1./(PH7(IG,J)*PH(IG,J))
V3(IG)=1./(P0(IG,K)*P0(IG,K))

AL(IG,K)= (KA1(IG,J)-KA2(IG,J)-R(IG,J)+LA2(IG,J)-LA1(IG,J))
AF(IG,K)= (KB1(IG,J)-KB2(IG,J))*V1(IG)
BF(IG,K)=KB2(IG,J)*V1(IG)+(KB0(IG,K)-2.*KB1(IG,K)+KB2(IG,K))

IF (IG.EQ.2) GO TO 60
GO TO 60
BT(K) = SR2(1,J)/ (PH7(2,J)*PH(1,J))
X + (SR1(1,K)-2.*SR2(1,K)) / (P07(2,K)*P0(1,K))
BF(3,K) = KC2(IG,J) / (PH7(1,J)*PH(2,J))
X + (KC1(IG,K)-2.*KC2(IG,K)) / (P07(1,K)*P0(2,K))
BF(4,K) = KD2(IG,J) / (PH7(2,J)*PH(1,J))
X + (KD1(IG,K)-2.*KD2(IG,K)) / (P07(2,K)*P0(1,K))
CT(K) = (SR1(1,K)-SR2(1,K)) / (P07(2,K)*PH(1,K))
CF(3,K) = KC1(IG,K)-KC2(IG,K)) / (P07(1,K)*PH(2,K))
CF(4,K) = KD1(IG,K)-KD2(IG,K)) / (P07(2,K)*PH(1,K))

60 CONTINUE

C THE ZERO FLUX COEFFS:
C NONE NEEDED AS F(1,1) AND F(2,1) BOTH = 0.0.
C IF (IBC.EQ.1) GO TO 64
C
C IF (.NOT.(IBC.EQ.5.OR.IBC.EQ.7)) GO TO 66
C ZERO FLUX COEFFICIENTS FOR NO TILTING ON THE LEFT:
DO 61 IG=1,2
V1(IG) = 1./(PH7(IG,1)*PH(IG,1))
V2(IG) = 1./(P07(IG,2)*P0(IG,2))

61 CF0(IG) = (KB1(IG,2)-KB2(IG,2))V2(IG)

BTO(1) = SR0(1,1)/(PH7(2,1)*PH(1,1))
X + (SR0(1,2)-2.*SR2(1,2))/ (P07(2,2)*P0(1,2))
BF0(3) = KC0(1,1)/(PH7(1,1)*PH(2,1))
X + (KC0(1,2)-2.*KC2(1,2))/ (P07(1,2)*P0(2,2))
BF0(4) = KD0(1,1)/(PH7(2,1)*PH(1,1))
X + (KD0(1,2)-2.*KD2(1,2))/ (P07(2,2)*P0(1,2))
CT0(1) = (SR0(1,2)-SR2(1,2))/ (P07(2,2)*PH(1,2))
CF0(3) = (KC0(1,2)-KC2(1,2))/ (P07(1,2)*PH(2,2))
CF0(4) = (KD0(1,2)-KD2(1,2))/ (P07(2,2)*PH(1,2))
66 IF (IBC .LE. 5) GO TO 69
C FOR THE LAST REGION TO BE NOT TILTED:
K=KR-1
DO 67 IG=1,2
V(IG)=1./(PH7(IG,K)*PO(IG,K))
V1(IG)=1./(PH7(IG,K)*PH(IG,K))
V2(IG)=1./(PH7(IG,K)*PO(IG,K))
AL(IG,KR)=(KA1(IG,K)-KA2(IG,K)-RA1(IG,K)+LA2(IG,K)+P(IG,K))
B(IG,KR)=(KA0(IG,KR)-LA2(IG,KR))*V1(IG)+(KA2(IG,K)-LA2(IG,K))
X -P1(IG,K)-Q1(IG,K)-R(IG,K))*V1(IG)
AF(IG,KR)=(KB1(IG,K)-KB2(IG,K))*V1(IG)
BF(IG,KR)=KB0(IG,KR)*V1(IG)+KB2(IG,K)*V2(IG)
IF (IG.EQ.2) GO TO 67
AT( K)=SR1(IG,K)-SR2(IG,K)/(PH7(2,K)*PO(1,K))
AF(3,KR)=(KC1(IG,K)-KC2(IG,K))/PH7(1,K)*PO(2,K))
AF(4,KR)=(KD1(IG,K)-KD2(IG,K))/PH7(2,K)*PO(1,K))
BT(KR)=SR0(IG,KR)/(PO(2,K)*PO(1,K))
X +SR2(IG,K)/(PH7(2,K)*PH(1,K))
BF(3,KR)=KC0(IG,KR)/(PO(1,K)*PO(2,K))
X +KC2(IG,K)/(PH7(1,K)*PH(2,K))
BF(4,KR)=KD0(IG,KR)/(PO(2,K)*PO(1,K))
X +KD2(IG,K)/(PH7(2,K)*PH(1,K))
67 CONTINUE
GO TO 64
C THE ZERO CURRENT COEFFS:
69 K=KR
DO 62 IG=1,2
V(IG)=1./(PH7(IG,K)*PO(IG,K))
V1(IG)=1./(PH7(IG,K)*PH(IG,K))
AL(IG)=(KA1(IG,K)-KA2(IG,K)-R(IG,K)+P(IG,K)+LA2(IG,K))
X -LA1(IG,K)-Q1(IG,K))*V1(IG)
BL(IG)=(KA2(IG,K)+R(IG,K)-P(IG,K)-LA2(IG,K)+Q1(IG,K))*V1(IG)
X +CH(IG)/PH(IG,K)
AFK(IG)=(KB1(IG,K)-KB2(IG,K))*V1(IG)
BFK(IG)=KB2(IG,K)*V1(IG)
ATK(1) = (SR1(1,K) - SR2(1,K)) / (PH7(2,K) * PO(1,K))
AFK(3) = (KC1(1,K) - KC2(1,K)) / (PH7(1,K) * PO(2,K))
AFK(4) = (KD1(1,K) - KD2(1,K)) / (PH7(2,K) * PO(1,K))
BTK(1) = SR2(1,K) / (PH7(2,K) * PH(1,K))
BFK(3) = KC2(1,K) / (PH7(1,K) * PH(2,K))
BFK(4) = KD2(1,K) / (PH7(2,K) * PH(1,K))
IF {IBC = "NE-4"} GO TO 64
   ZERU CURRENT ON THE LEFT COEFFS:
   K = 1
   DO 63 I = 1, 2
   V2(I,G) = 1. / (PO7(G,K) * PO(I,G))
   V3(I,G) = 1. / (PO7(G,K) * PH(I,G))
   BLO(I,G) =
       X + (KA0(I,G,K) - 2.*KA1(I,G,K) + KA2(I,G,K) - LA0(I,G,K) + 2.*LA1(I,G,K))
       X - CO(I,G) / PO(I,G,1)
       X - Q1(I,G,K) + P1(I,G,K) + V2(I,G)
   BF0(I,G) =
       V2(I,G) * (KB0(I,G,K) - 2.*KB1(I,G,K) + KB2(I,G,K))
       V3(I,G)
   CF0(I,G) = (KB1(I,G,K) - KB2(I,G,K)) * V3(I,G)
   BT0(I) = (SR1(1,K) - 2.*SR1(1,K) + SR2(1,K)) / (PO7(2,K) * PO(1,K))
   BF0(3) = (KC0(1,K) - 2.*KC1(1,K) + KC2(1,K)) / (PO7(1,K) * PO(2,K))
   BF0(4) = (KD0(1,K) - 2.*KD1(1,K) + KD2(1,K)) / (PO7(2,K) * PO(1,K))
   CT0(I) = (SR1(1,K) - SR2(1,K)) / (PO7(2,K) * PH(1,K))
   CF0(3) = (KC1(1,K) - KC2(1,K)) / (PO7(1,K) * PH(2,K))
   CF0(4) = (KD1(1,K) - KD2(1,K)) / (PO7(2,K) * PH(1,K))
   C    ZERO MATRICES:
   64 DO 65 I = 1, NN
   DO 65 J = 1, NN
   L1(I, J) = 0.0
   L2(I, J) = 0.0
   F1(I, J) = 0.0
   F2(I, J) = 0.0
   F3(I, J) = 0.0
   F4(I, J) = 0.0
   65 T(I, J) = 0.0
FILLING THE MATRICES FOR POWER:

DETERMINE THE LEFT BC:

\[
\begin{align*}
J &= 1 \\
\text{IF } & (IBC \leq 2 \text{ OR } IBC = 6) \text{ GC TO 75} \\
L1(J,1) &= BL0(1) \\
L2(J,1) &= BL0(2) \\
F1(J,1) &= BF0(1) \\
F2(J,1) &= BF0(2) \\
F3(J,1) &= BF0(3) \\
F4(J,1) &= BF0(4) \\
T(J,1) &= BT0(1) \\
L1(J,2) &= CL0(1) \\
L2(J,2) &= CL0(2) \\
F1(J,2) &= CF0(1) \\
F2(J,2) &= CF0(2) \\
F3(J,2) &= CF0(3) \\
F4(J,2) &= CF0(4) \\
T(J,2) &= CT0(1) \\
J &= J+1
\end{align*}
\]

FOR ALL INTERNAL EQUATIONS:

\[
\begin{align*}
\text{DO 70 } & K = 1X,1Y \\
\text{IF } & (J \equiv 1) \text{ GO TO 76} \\
L1(J,J-1) &= AL(1,K) \\
L2(J,J-1) &= AL(2,K) \\
F1(J,J-1) &= AF(1,K) \\
F2(J,J-1) &= AF(2,K) \\
F3(J,J-1) &= AF(3,K) \\
F4(J,J-1) &= AF(4,K) \\
T(J,J-1) &= AT(K) \\
\text{DO 76 } & L1(J,J) = BL(1,K) \\
\text{DO 76 } & L1(J,J+1) = CL(1,K) \\
\text{DO 76 } & L2(J,J) = BL(2,K) \\
\text{DO 76 } & L2(J,J+1) = CL(2,K) \\
F1(J,J) &= BF(1,K) \\
F1(J,J+1) &= CF(1,K) \\
F2(J,J) &= BF(2,K)
\end{align*}
\]
F2(J, J+1)=CF(2, K)
F3(J, J)=BF(3, K)
F3(J, J+1)=CF(3, K)
F4(J, J)=BF(4, K)
F4(J, J+1)=CF(4, K)
T(J, J)=BT(K)
T(J, J+1)=CT(K)
J=J+1
70 CONTINUE
IF (IBC.EQ.1 .OR. IBC.EQ.5) GC TC 500
C FOR THE LAST EQUATION: (K=NN):
    K=NN
    L1(K, K -1)=ALK(1)
    L1(K, K )=BLK(1)
    L2(K, K -1)=ALK(2)
    L2(K, K )=BLK(2)
    F1(K, K -1)=AFK(1)
    F1(K, K )=BFK(1)
    F2(K, K -1)=AFK(2)
    F2(K, K )=BFK(2)
    F3(K, K -1)=AFK(3)
    F3(K, K )=BFK(3)
    F4(K, K -1)=AFK(4)
    F4(K, K )=BFK(4)
    T(K, K -1)=ATK(1)
    T(K, K )=BTK(1)
500 CONTINUE
C PRINTS OUT THE /B3/ MATRICES:
    IF (ISEE.GE.2) CALL PRTOUT(2)
RETURN
END
SUBROUTINE ERROR(I,J)
C ANNOUNCES INPUT ERRORS AND TERMINATES PROGRAM EXECUTION:
GO TO (1,2,3,4,5,6,7,8,9,1)
1 WRITE (6,101)
    GO TO 10
2 WRITE (6,102) J
    GO TO 10
3 WRITE (6,103) J
    GO TO 10
4 WRITE (6,104) J
    GO TO 10
5 WRITE (6,105) J
    GO TO 10
6 WRITE (6,106) J
    GO TO 10
7 CONTINUE
8 CONTINUE
9 CONTINUE
10 WRITE (6,110)
101 FORMAT ('IMUST HAVE > 2 SUBREGIONS FCR ZERO FLUX B.C.S. INVALID.')
102 FORMAT ('INUMBER OF SUBREGIONS =',I3,' > 25. INVALID.')
103 FORMAT ('SUBREGION NUMBER',I3,' HAS > 25 SECTIONS. INVALID.')
104 FORMAT ('INPUT ERROR IN REGION SEQUENCING AT REGION',I5,'.')
105 FORMAT ('121(I) = 0. IN REGION I =',I3,'. INVALID.')
106 FORMAT ('THE TOLERANCE: EPS',I1,' IS < 1.0E-16. INVALID.')
107 FORMAT ('BOUNDARY CONDITION OPTICN =',I2,' < 1 OR > 7. INVALID.')
110 FORMAT ('PROBLEM TERMINATED.')
CALL EXIT
RETURN
END
SUBROUTINE REPEAT(K, L)
C SETS THE /85/ ARRAYS (K) EQUAL TO PAST STORED ARRAYS (L):
IMPLICIT REAL*8 (A-Z)
COMMON /85/ KAO(2,25), KA1(2,25), KA2(2,25), KB0(2,25), KB1(2,25),
X KB2(2,25), LA0(2,25), LA1(2,25), LA2(2,25), SR0(1,25),
X SR1(1,25), SR2(1,25), KC0(1,25), KC1(1,25), KC2(1,25),
X KD0(1,25), KD1(1,25), KD2(1,25),
X P(2,25), P1(2,25), Q(2,25),
X Q1(2,25), R(2,25), P0(2,25), P07(2,25), PH(2,25),
X PH7(2,25)
COMMON /XAXIS/ HXHR(25)
INTEGER K, L, G
HR(K)=HR(L)
HX=HX+HR(K)
DO 10 G=1, 2
KAO(G,K)=KAO(G,L)
KA1(G,K)=KA1(G,L)
KA2(G,K)=KA2(G,L)
KB0(G,K)=KB0(G,L)
KB1(G,K)=KB1(G,L)
KB2(G,K)=KB2(G,L)
LA0(G,K)=LA0(G,L)
LA1(G,K)=LA1(G,L)
LA2(G,K)=LA2(G,L)
IF (G.EQ.2) GO TO 5
SR0(G,K)=SR0(G,L)
SR1(G,K)=SR1(G,L)
SR2(G,K)=SR2(G,L)
KC0(G,K)=KC0(G,L)
KC1(G,K)=KC1(G,L)
KC2(G,K)=KC2(G,L)
KD0(G,K)=KD0(G,L)
KD1(G,K)=KD1(G,L)
KD2(G,K)=KD2(G,L)
5 CONTINUE
P(G,K)=P(G,L)
P1(G,K) = P1(G,L)
Q(G,K) = Q(G,L)
Q1(G,K) = Q1(G,L)
R(G,K) = R(G,L)
PO(G,K) = PO(G,L)
PO7(G,K) = PO7(G,L)
PH(G,K) = PH(G,L)
PH7(G,K) = PH7(G,L)

10 CONTINUE
RETURN
END
SUBROUTINE BHSET(K)
C SETS UP THE /BH/ ARRAYS FOR GIF:
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /BH/ X(101), H(101), Z(101)
DO 1 I=1,K
  1 Z(I)=X(I)-X(1)
RETURN
END
DOUBLE PRECISION FUNCTION GIF(N,G1,F,G2,C,G,K,ITC)
C INTEGRATES: F(G1,J)*C(G2,J)*G(G2,J) * (Z/H)**N
C OVER ALL K SUBREGIONS J
C WHERE Z RUNS FROM 0 TO X(K+1)-X(1) IN THIS REGION.
C WHERE THE FORM OF F AND G IN REGION J IS GIVEN BY ITC:
C ITC = 0: F AND G ARE BOTH CONSTANT.
C ITC = 1: F IS LINEAR AND G IS CONSTANT.
C ITC = 2: F AND G ARE BOTH LINEAR.
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /BH/ X(101),H(101),Z(101)
DIMENSION F(2,101),G(2,101),C(2,100)
INTEGER G1,G2
N1=N+1
SUMJ=0.0D0
IF (ITC.EQ.0) GO TO 40
IF (ITC.EQ.1) GO TO 20
C LINEAR F AND G IN REGIONS J:
DO 10 J=1,K
A=F(G1,J)*G(G2,J)
B=F(G1,J)*G(G2,J+1)+F(G1,J+1)*G(G2,J)
D=F(G1,J+1)*G(G2,J+1)
SUML=SUML+A
DO 5 I=1,N1
L=I-1
M=N-L
IF (H(J).LEQ.0.0.AND.L.EQ.0) GO TO 1
EH=H(J)**L
GO TO 2
1 EH=1.0
2 IF (Z(J).LEQ.0.0.AND.M.EQ.0) GO TO 3
EZ=Z(J)**M
GO TO 4
3 EZ=1.0
4 SUML=SUML+FACT(N)/(FACT(M)*FACT(L))*EH*EZ*
X (2.DO*A/DFLOAT((L+3)*(L+2))*(L+1)+B/DFLOAT((L+3)*(L+2))
X +D/DFLOAT(L+3))
5 CONTINUE
  SUMJ=SUMJ+H(J)*C(G2,J)*SUML
10 CONTINUE
  GO TO 100
C LINEAR F AND CONSTANT G IN REGICNS J:
20 DO 30  J=1,K
  SUML=0.0D0
  DO 25  I=1,NI
    L=I-1
    M=N-L
    IF (H(J).EQ.0.0.AND.L.EQ.0) GO TO 21
    EH=H(J)**L
    GO TO 22
21 EH=1.0
22 IF (Z(J).EQ.0.0.AND.M.EQ.0) GO TO 23
    EZ=Z(J)**M
    GO TO 24
23 EZ=1.0
24 SUML=SUML+FACT(N)/(FACT(M)*FACT(L))*EH*EZ*
  X (F(G1,J)/DFLOAT((L+1)*(L+2))+F(G1,J+1)/DFLOAT(L+2))
  25 CONTINUE
  SUMJ=SUMJ+H(J)*C(G2,J)*G(G2,J)*SUML
30 CONTINUE
  GO TO 100
C CONSTANT F AND G IN REGICNS J:
40 DO 50  J=1,K
  SUML=0.0D0
  DO 45  I=1,NI
    L=I-1
    M=N-L
    IF (H(J).EQ.0.0.AND.L.EQ.0) GO TO 51
    EH=H(J)**L
    GO TO 52
51 EH=1.0
52 IF (Z(J).EQ.0.0.AND.M.EQ.0) GO TO 53
    EZ=Z(J)**M
53 CONTINUE
  GO TO 100
GO TO 54
53 EZ=1.0
54 SUML=SUML+FACT(N)/(FACT(M)*FACT(L))*EH*EZ*
  X (1./DFLOAT(L+1))
55 CONTINUE
   SUMJ=SUMJ+F(G1,J)*G(G2,J)*SUML
50 CONTINUE
100 GIF=SUMJ/(X(K+1)-X(1))**N
RETURN
END
DOUBLE PRECISION FUNCTION FACT(N)
C
COMPUTES N FACTORIAL:
FACT=1.0D0
IF (N.LE.1) RETURN
DO 1 I=2,N
  FACT=FACT*DFLOAT(I)
1 RETURN
END
SUBROUTINE PRTLUT(IP)

    C    IP = 1: PRINTS OUT THE /B5/ ARRAYS.
    C    IP = 2: PRINTS OUT THE /B3/ MATRICES GIVEN TO POWER.
    C
    IMPLICIT REAL*8 (A-H,K-Z)
    COMMON /B2/ KR, N
    COMMON /B3/ L1(26,26),L2(26,26),F1(26,26),F4(26,26),
               X F2(26,26),F3(26,26),T(26,26),
    COMMON /B5/ KAO(2,25),KA1(2,25),KA2(2,25),KB0(2,25),KB1(2,25),
               X KB2(2,25),LA0(2,25),LA1(2,25),LA2(2,25),SR0(1,25),
               X SR1(1,25),SR2(1,25),KC0(1,25),KC1(1,25),KC2(1,25),
               X KD0(1,25),KD1(1,25),KD2(1,25),
               X P(2,25),P1(2,25),Q(2,25),
               X Q1(2,25), R(2,25), PO(2,25), P07(2,25), PH(2,25),
               X PH7(2,25)
    COMMON /XAXIS/ HX,HR(25)
    INTEGER KR, G, N
    GO TO (1001,1002),IP

    KA AND KB ARRAYS:

    1001 WRITE (6,10)
        10 FORMAT (1,4X,'G',4X,'I',12X,'KA0(G,I)',12X,'KA1(G,I)',12X,
               X 'KA2(G,I)',12X,'KB0(G,I)',12X,'KB1(G,I)',12X,'KB2(G,I))
        DO 11 G=1,2
            WRITE (6,12)
            11 WRITE (6,15) (G,I,KAO(G,I),KA1(G,I),KA2(G,I),KB0(G,I),KB1(G,I),
                X KB2(G,I),I=1,KR)
  12 FORMAT (' ',*)
  15 FORMAT (215,6D20.7)

    KC AND KD ARRAYS:

    WRITE (6,16)
  16 FORMAT (1,4X,'G',4X,'I',12X,'KC0(G,I)',12X,'KC1(G,I)',12X,
               X 'KC2(G,I)',12X,'KD0(G,I)',12X,'KD1(G,I)',12X,'KD2(G,I))
    G=1
    WRITE (6,12)
    17 WRITE (6,15) (G,I,KC0(G,I),KC1(G,I),KC2(G,I),KD0(G,I),KD1(G,I),
               X KD2(G,I),I=1,KR)

    LA AND SR ARRAYS:

PAGE 265
WRITE (6,20)
20 FORMAT (*1 G',12X,'LA0(G,I)',12X,'LA1(G,I)',12X,'LA2(G,I)',12X,'SR0(G,I)',12X,'SR1(G,I)',12X,'SR2(G,I)'),
G=1
WRITE (6,12)
WRITE (6,15) (G,I,LA0(G,I),LA1(G,I),LA2(G,I),SR0(G,I),SR1(G,I),SR2(G,I),I=1,KR)
G=2
WRITE (6,12)
WRITE (6,25) (G,I,LA0(G,I),LA1(G,I),LA2(G,I),I=1,KR)
25 FORMAT (215,3D20.7)
C
P, Q, AND R ARRAYS:
WRITE (6,30)
30 FORMAT (*1 G',4X,'I',14X,'P(G,I)',14X,'P1(G,I)',14X,'R(G,I)'),
X = 14X,'Q1(G,I)',14X,'R(G,I)'
DO 31 G=1,2
WRITE (6,12)
31 WRITE (6,35) (G,I,P(G,I),P1(G,I),Q(G,I),Q1(G,I),R(G,I),I=1,KR)
35 FORMAT (215,5D20.7)
C
PO, PH, AND HR ARRAYS:
WRITE (6,40)
40 FORMAT (*1 G',4X,'I',13X,'PO(G,I)',13X,'PH7(G,I)',13X,'PH(G,I)'),
X = 13X,'PO7(G,I)',13X,'PH7(G,I)',15X,'HR(I)'
DO 41 G=1,2
WRITE (6,12)
41 WRITE (6,45) (G,I,PO(G,I),PO7(G,I),PH7(G,I),PH(G,I),PH7(G,I),HR(I),I=1,KR)
45 FORMAT (215,5D20.7)
GO TO 100
C
PRINT OUT THE /B3/ MATRICES:
1002 WRITE (6,50)
50 FORMAT (*1MATRIX L1:','/)
DO 51 I=1,N
51 WRITE (6,55) (L1(I,J),J=1,N)
55 FORMAT (10D12.3,2X,10D12.3)
WRITE (6,60)
60 FORMAT (*1MATRIX L2:','/)

PAGE 266
DO 61 I=1,N
61 WRITE (6,55) (L2(I,J),J=1,N)
WRITE (6,70)
70 FORMAT ('IMATRIX F1:',/)
DO 71 I=1,N
71 WRITE (6,55) (F1(I,J),J=1,N)
WRITE (6,80)
80 FORMAT ('IMATRIX F2:',/)
DO 81 I=1,N
81 WRITE (6,55) (F2(I,J),J=1,N)
WRITE (6,82)
82 FORMAT ('IMATRIX F3:',/)
DO 83 I=1,N
83 WRITE (6,55) (F3(I,J),J=1,N)
WRITE (6,84)
84 FORMAT ('IMATRIX F4:',/)
DO 85 I=1,N
85 WRITE (6,55) (F4(I,J),J=1,N)
WRITE (6,90)
90 FORMAT ('IMATRIX T:',/)
DO 91 I=1,N
91 WRITE (6,55) (T(I,J),J=1,N)
100 RETURN
END
SUBROUTINE POWER

C SOLVES THE 2*N MULTIGROUP EQUATIONS:  M*PHI = (1/LAMDA)*F*PHI
C BY THE FISSION SOURCE POWER METHOD
C USING SIMULTANEOUS OVERRELAXATION.
C WHERE:  M AND F ARE DOUBLE PRECISION 2N BY 2N BLOCK MATRICES;
C AND:  PHI IS THE 2N FLUX (FAST AND THERMAL) VECTOR.
C 
L1*PHI1 = CHI1*(F1*PHI1 + F2*PHI2)
C -T*PHI1 + L2*PHI2 = CHI2*(F3*PHI1 + F4*PHI2)
C METHOD FOLLOWS WACHPRESS, PAGE 83. SOLUTION BY GROUP ITERATION.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /B1/ IBC, I PLOT, J PLOT, IPUNCH, ISEE
COMMON /B2/ K R, N
COMMON /B3/ L1(26,26), L2(26,26), F1(26,26), F4(26,26),
F2(26,26), F3(26,26), T(26,26)
COMMON /B4/ PHI(2,26), PSI(2,26), LAMDA, ICOUT
COMMON /B5/ S(26), ERROR(2,26), Z(26)
COMMON /B6/ TE1(2,5), TE2(2,5), TE3(5), IN(5)
COMMON /CHI/ CHI(2)
COMMON /XAXIS/ HXHR(25)
COMMON /EP/ EPS1, EPS2, EPS3
COMMON /PHI/ PHISTR(2,26,6)
COMMON /LAM/ LAMSTR(300), EFSTR(2,300), EFMSTR(2,300), ERLAM(300)
COMMON /TRUE/ TRULAM, TRUPHI(2,26), PHICON(2,300), LAMCON(300), IFT
DIMENSION PSI1(26), PSI2(26), SQ(2), DPHI(2), ERRMAX(2)
INTEGER N

C DEFAULT OPTIONS FOR THE TRUE EIGENVALUE AND FLUXES:
TRULAM = 1.0
DO 5 IG = 1, 2
DO 5 I = 1, N
5 TRUPHI(IG, I) = 1.0
C DEFAULT OPTIONS FOR POWER PARAMETERS:
ALPHA = 1.25
LAMDA = 1.0
DO 555 IG = 1, 2
IF (IBC.NE.4) GO TO 551
DO 555 I = 1, N
550 PHI(IG,1)=1.0
    GO TO 555
551 X=3.1415926/HX
    IF (IBC.NE.1) X=X/2.0
    SUM1=0.0
    DO 552 K=1,KR
        SUM1=SUM1+HR(K)
    552 PHI(IG,K)=DSIN(SUM1*X)
555 CONTINUE
C    READ IN THE TRUE (EXPECTED) EIGENVALUE AND FLUX VECTOR (MINUS 0 BC'S):
C        IFT=0
    READ (5,500,END=501) TRULAM, (TRUPHI(1,I),I=1,N)
    READ (5,503,END=501) (TRUPHI(2,I),I=1,N)
    IFT=1
500 FORMAT (E25.14,/,(4E20.10))
C    READ IN: OVERRELAXATION PARAMETERS ; ALPHA (OUTER ITERATION)
C        INITIAL GUESS AT EIGENVALUE; LAMDA
C        INITIAL NORMALIZED FLUX ; PHI(1-N)
501 READ (5,506,END=510) ALPHA
    READ (5,502,END=510) LAMDA
    READ (5,503) (PHI(1,I),I=1,N)
    READ (5,503) (PHI(2,I),I=1,N)
506 FORMAT (F10.5)
502 FORMAT (E25.14)
503 FORMAT ((4E20.10))
510 CONTINUE
C    STORING FOR PRINTING THE MULTIGROUP FLUX SHAPE.
    DO 11 IG=1,2
    DO 10 I=1,N
10    PHISTR(IG,I,2)=PHI(IG,I)
C        FILL RUNNING COORD IN PHISTR
    KR1=KR+1
    DO 11 I=1,KR1
11    PHISTR(IG,I,1)=DFLOAT(I)
C        IK IS THE FLUX PLOTTING COUNTER.
    IK=1
STORES THE ITERATION NUMBER FOR FLUX HISTORY PLOTTING:

\[ \text{IN}(1)=0 \]

STORES TEMPORARY ERRORS FOR FLUX HISTORY PLOTTING:

\[ \text{TE}1(1,1)=0. \]
\[ \text{TE}1(2,1)=0. \]
\[ \text{TE}2(1,1)=0. \]
\[ \text{TE}2(2,1)=0. \]
\[ \text{TE}3(1)=0.0 \]

EIGENVALUE OF THE PREVIOUS ITERATION:

\[ \text{LAMB}4=\text{LAMDA} \]

THE MAXIMUM NUMBER OF ALLOWED ITERATIONS: \( \text{ICMAX} = 300 \)

PRINT OUT THE POWER METHOD PARAMETER INFORMATION:

WRITE (6,700) \( \text{ICMAX} \), \( \text{ALPHA} \), \( \text{LAMDA} \), (PHI(1,I),I=1,N)

WRITE (6,701) (PHI(2,I),I=1,N)

700 FORMAT ('IEXECUTING MULTIGROUP FISSION SOURCE POWER ITERATION METHOD',//,
X 5X,'MAXIMUM NUMBER OF ALLOWABLE ITERATIONS:',/,X 10X,'ICMAX = ',I4,///,
X 5X,'OUTER ITERATION RELAXATION PARAMETER:',/,X 10X,'\text{ALPHA} = ',F7.3,///,
X 5X,'INITIAL GUESS AT EIGENVALUE:',/,X 10X,'\text{LAMDA} = ',E22.14,///,
X 5X,'INITIAL GUESS AT THE GROUP FLUX SHAPE CONNECTION POINTS:',
X //,8X,'\text{FAST GROUP}:',/

701 FORMAT ('I0',7X,'THERMAL GROUP:',//,
X 10X,'\text{F}(K)'fS =',4E25.14,//(18X,4E25.14))

BEGIN ITERATION LOOP.

ICOUT=0

ICOUT IS THE OUTER ITERATION COUNTER.

20 ICOUT=ICOUT+1

IF (ICOUT.GT. ICMA) GO TO 100

SOLVE FOR THE NEW GROUP FLUX VECTORS: \( \text{PSI} \):

DO 25 I=1,N

\[ \text{DO 25 I=1,N} \]
S(I)=0.0
DO 24 J=1,N
24 S(I)=S(I)+F1(I,J)*PHI(1,J)+F2(I,J)*PHI(2,J)
25 S(I)=CHI(1)*S(I)
C  FAST FLUX:
CALL SOLV3D(N,L1,PSI1,S)
C  THERMAL GROUP; SOURCE VECTOR:
DO 27 I=1,N
27 S(I)=.U0
Z(I)=0.0
DO 26 J=1,N
26 S(I)=S(I)+F3(I,J)*PSI1(J)+F4(I,J)*PHI(2,J)
26 Z(I)=Z(I)+T(I,J)*PSI1(J)
27 Z(I)=Z(I)+CHI(2)*S(I)
C  THERMAL FLUX:
CALL SOLV3D(N,L2,PSI2,Z)
C  CALCULATION OF THE EIGENVALUE:
SUM1=0.000
SUM2=0.000
DO 28 I=1,N
28 SUM2=SUM2+PSI1(I)*PSI1(I)+PSI2(I)*PSI2(I)
28 SUM1=SUM1+PSI1(I)*PHI(1,I)+PSI2(I)*PHI(2,I)
LAMDA=SUM2/SUM1
LAMSTR(IOUT)=LAMDA
ERRLAM=DABS(LAMDA-LAMBA)
C  PUT PSI1 AND PSI2 INTO BIGGER PSI:
DO 30 I=1,N
30 PSI1(I)=PSI1(I)
30 PSI2(I)=PSI2(I)
C  POINT BY POINT SIMULTANEOUS RELAXATION FLUX ITERATION:
X=ALPHA
C  DO NOT RELAX DURING THE FIRST THREE ITERATIONS:
IF (ICDUT.LE.3) X=1.0
C  CALCULATE THE NEW GROUP FLUX ITERATES AND GROUP ERRORS:
DO 40 IG=1,B
DO 40 I=1,N
40 PSI(IG,1)=PHI(IG,1)+X*(PSI(IG,1)/LAMDA-PHI(IG,1))
CALL NORMAL(PSI,N)
DO 39 IG=1,2
ERRMAX(IG)=0.0
SQ(IG)=0.0
DO 38 IG=1,N
ERROR(IG,I)=DABS((PSI(IG,1)-PHI(IG,1))/PSI(IG,1))
IF (ERROR(IG,1).GT.ERRMAX(IG)) ERRMAX(IG)=ERROR(IG,1)
SQ(IG)=SQ(IG)+ERROR(IG,1)**2
C UPDATE THE FLUX ITERATE:
38 PHI(IG,1)=PSI(IG,1)
39 SQ(IG)=SQRT(SQ(IG))
C NORMALIZE PSI GROUPS TO UNITY:
CALL NORM2(PSI,TRUPHI,N)
IF (IFT.EQ.0) GO TO 37
DLAM=LAMDA-TRULAM
DO 36 IG=1,2
DPHI(IG)=0.0
DO 35 I=1,N
35 DPHI(IG)=DPHI(IG)+(PSI(IG,1)-TRUPHI(IG,1))**2
36 DPHI(IG)=DSQRT(DPHI(IG))
37 IF (IPLOT.NE.2) GO TO 45
C THE FOLLOWING IS FOR NICELY PLOTTING THE GROUP FLUX HISTORY.
DO 41 IG=1,2
41 DO 41 I=1,N
41 ERROR(IG,1)=PSI(IG,1)
C ERROR NOW CONTAINS THE NEW NORMALIZED FLUX ITERATE PHI.
JK=IK
IF (IK.EQ.0) JK=5
DO 42 IG=1,2
42 DO 42 I=1,N
42 IF (DABSERROR(IG,1)-PHISTR(IG,1,JK+1)).GE.0.01) GO TO 43
43 CONTINUE
C FLUX HAS NOT CHANGED ENOUGH FOR PLOTTING.
GO TO 45
C SAVE THE NORMALIZED FLUX FOR PLOTTING:
IK=IK+1
IN(IK)=ICOUT
TE3(IK)=ERRLAM
DO 44 IG=1,2
TE1(IG,IK)=ERRMAX(IG)
TE2(IG,IK)=SQ(IG)
DO 44 IG=1,N
44 PHISTR(IG,1,IK+1)=ERROR(IG,1)
   IF (IK.NE.5) GO TO 45
C   PLOT THE LAST FIVE SAVED FLUXES:
   CALL PHIPLT(5)
   IK=0
45 CONTINUE
C   ERROR CRITERIA FOR ACCEPTANCE OF CONVERGENCE.
   IFLAG1=0
   IFLAG2=0
   IFLAG3=0
C   STORE THE ERRORS FOR COMPARISON:
C   ERROR BETWEEN ITERATION EIGENVALUES:
   ERLAM(ICOUT)=ERRLAM
   DO 46 IG=1,2
      C MAXIMUM ERROR BETWEEN ITERATION FLUXES:
      EFSTR(IG,ICOUT)=ERRMAX(IG)
      C MEAN SQUARE ERROR BETWEEN ITERATION FLUXES:
      EFMSR(IG,ICOUT)=SQ(IG)
      C MEAN SQUARE ERROR BETWEEN THE ITERATION FLUX AND GIVEN TRUE FLUX:
      PHICUN(IG,ICOUT)=DPHI(IG)
   CONTINUE
C   ERROR BETWEEN THE ITERATION EIGENVALUE AND GIVEN TRUE EIGENVALUE:
   LAMCUN(ICOUT)=DLAM
   IF ((ERRMAX1.LT.EPS1).AND.(ERRMAX2.LT.EPS1)) IFLAG1=1
   IF (SQ1.LT.EPS2).AND.(SQ2.LT.EPS2)) IFLAG2=1
   IF (ERRLAM.LT.EPS3) IFLAG3=1
   IFLAG4=IFLAG1*IFLAG2*IFLAG3
   IF (IFLAG4.EQ.1) GO TO 50
C   OTHERWISE CONTINUE THE ITERATION.
LAMB4=LAMDA
GO TO 20
50 CONTINUE
C CONVERGENCE ACCOMPLISHED.
C NORMALIZE THE CONVERGED FLUX VECTOR:
CALL NORMAL(PHI,N)
C PLOT ANY LEFT OVER FLUX HISTORY PLOTS:
IF ((IPLT.EQ.2).AND.(IK.NE.0)) CALL PHIPLT(IK)
C BOUNDARY CONDITION INSERTIONS.
IER=0
C IER ALLOWS B.C. INSERTIONS FOR YES AND NO CONVERGENCE:
55 IF (IBC.EQ.4) GO TO 90
IF (IBC.NE.3) GO TO 60
PHI(1,KR+1)=0.
PHI(2,KR+1)=0.
GO TO 90
60 DO 70 I=1,N
J=N+1-I
PHI(1,J+1)=PHI(1,J)
70 PHI(2,J+1)=PHI(2,J)
IF (IBC.EQ.5 .OR. IBC.EQ.7) GO TO 71
PHI(1,1)=0.0
PHI(2,1)=0.0
GO TO 72
71 PHI(1,1)=PHI(1,2)
PHI(2,1)=PHI(2,2)
72 IF (IBC.NE.1) GO TO 73
PHI(1,KR+1)=0.0
PHI(2,KR+1)=0.0
GO TO 90
73 IF (IBC.LT.6) GO TO 90
PHI(1,KR+1)=PHI(1,KR)
PHI(2,KR+1)=PHI(2,KR)
90 IF (IER.EQ.1) GO TO 102
RETURN
C NO CONVERGENCE ACCOMPLISHED:
100 CONTINUE
C   NORMALIZE THE UNCONVERGED FLUX:
   CALL NORMAL PHI,N
   ICOUT=ICOUT-1
   WRITE (6,101) ICOUT
101 FORMAT (1H1,'POWER METHOD DID NOT CONVERGE FOR THIS CASE AFTER',
           X ' I4,' ITERATIONS.'),//,IX,'EXECUTION TERMINATED ')
   IER=1
   GO TO 55
102 CONTINUE
C   FOR PRINTING OUT THE EIGENVALUE HISTORY AND THE FINAL FLUX SHAPE:
   I PLOT=1
   J PLOT=1
   RETURN
END
SUBROUTINE SOLV3D(N,A,X,Y)
C SOLVES THE N DOUBLE PRECISION MATRIX EQUATIONS: A*X = Y,
C FOR X - GIVEN THE N BY N TRIDIAGONAL MATRIX A
C AND THE SOURCE VECTOR Y.
C METHOD IS FORWARD ELIMINATION FOLLOWED BY BACKWARD SUBSTITUTION.
C CF - WACHPRESS, PAGE 23.

REAL*8 A, X, Y, H, P, D
DIMENSION A(26,26), X(26), Y(26), H(26), P(26)

IF (A(1,1).EQ.0.0) GO TO 10
H(1)=-A(1,2)/A(1,1)
P(1)=Y(1)/A(1,1)
DO 1 M=2,N
D=A(M,M)+A(M,M-1)*H(M-1)
IF (D.EQ.0.0) GO TO 20
P(M)=(Y(M)-A(M,M-1)*P(M-1))/D
IF (M.EQ.N) GO TO 1
H(M)=A(M,M+1)/D
1 CONTINUE
X(N)=P(N)
DO 2 I=2,N
M=N+I-1
2 X(M)=P(M)+H(M)*X(M+1)
RETURN

C IN CASE OF ANY IMPENDING ZERO DIVISORS:
10 WRITE (6,11)
11 FORMAT ("O FIRST ELEMENT OF A, A(1,1), IS ZERO.",/,
X 5X,'BETTER FIX IT BOSS.")
GO TO 30
20 WRITE (6,21) M
21 FORMAT ("O ZERO DIVISOR ENCOUNTERED IN EQUATION M =',I3,'.",/,
X 5X,'BETTER FIX IT BOSS.")
30 WRITE (6,31)
31 FORMAT ("O EXECUTION TERMINATED.")
CALL EXIT
RETURN
END
SUBROUTINE NORMAL PHI(N)
C NORMALIZES THE GROUP FLUXES TO ONE. NOT BOTH GROUPS.
REAL*8 PHI(2,26), A
A=DABS PHI(1,1)
DO 1 IG=1,2
DO 1 I=1,N
IF (DABS PHI(IG,I).GT.A) A=DABS PHI(IG,I)
1 CONTINUE
DO 2 IG=1,2
DO 2 I=1,N
2 PHI(IG,I)=PHI(IG,I)/A
RETURN
END
SUBROUTINE PHIPLT(L)
C PLOTS THE GROUP FLUX HISTORY, WITH UP TO 5 GROUP FLUXES PER PLOT.
C FAST AND THERMAL GROUP FLUXES ARE PLotted SEPERATELY.
C L IS THE NUMBER OF FLUXES TO BE PLOTTED.
C L IS BETWEEN 1 AND 5.
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /B1/ IBC
COMMON /B2/ KR,N
COMMON /B5/ S(26), A(26,6), B(26,6)
COMMON /B6/ TE1(2,5),TE2(2,5),TE3(5),IN(5)
COMMON /ER/ EPS1,EPS2,EPS3
COMMON /FSTR/ PHISTR(2,26,6)
DIMENSION SYMBOL(5)
INTEGER SYMBOL ('.',',','-', ',', '+', '#', '*')
KR1=KR+1
C SET UP B.C. CONDITIONS
IF (IBC.EQ.4) GO TO 5
IF (IBC.EQ.3) GO TO 3
DO 2 IG=1,2
DO 2 K=1,L
DO 1 I=1,N
J=N+1-I
1 PHISTR(IG,J+1,K+1)=PHISTR(IG,J,K+1)
2 PHISTR(IG,1,K+1)=0.
3 IF (IBC.EQ.2) GO TO 5
   DO 4 IG=1,2
   DO 4 K=1,L
4 PHISTR(IG,KR1,K+1)=0.
5 CONTINUE
C FLUXES IN PHISTR HAVE BEEN NORMALIZED IN POWER.
C PUT THE FAST FLUX IN A, AND THE THERMAL FLUX IN B:
L1=L+1
DO 10 K=1,L1
   DO 10 I=1,KR1
   A(I,K)=PHISTR(1,I,K)
10 B(I,K)=PHISTR(2,I,K)

C PLOT THE L FAST FLUX SHAPES ON ONE GRAPH:
   CALL PRTPLT(0,A,KR1,L1,KR1,0,26,6,2)
   WRITE (6,20)
20 FORMAT (/,"FAST FLUX ITERATION HISTORY PLOT.",/)
   WRITE (6,30)
30 FORMAT (X,'KEY:',5X,'SYMBOL',5X,'ITERATION NUMBER:',7X,'ERRCR CRITERIA',
X '11X,'ERROR',13X,'TOLERANCE'
   DO 35 I=1,L
35 WRITE (6,40) SYMBOL(I),IN(I),TE1(I,1),EPS1,TE2(I,1),EPS2,
X TE3(I),EPS3
40 FORMAT (/,'MEAN SQ. FLUX*',1PD15.5,5X,'1PD15.5',/
X,'EIGENVALUE*',8X,1PD15.5,5X,1PD15.5)
C PLOT THE L THERMAL FLUX SHAPES ON THE OTHER GRAPH:
   CALL PRTPLT(0,B,KR1,L1,KR1,0,26,6,2)
   WRITE (6,50)
50 FORMAT (/,"OTHERMAL FLUX ITERATION PLOT.",/)
   WRITE (6,30)
   DO 55 I=1,L
55 WRITE (6,40) SYMBOL(I),IN(I),TE1(2,1),EPS1,TE2(2,1),EPS2,
X TE3(I),EPS3
   RETURN
END
SUBROUTINE OUTPUT
C PRINTS THE RESULTS OF THE METHOD.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /B1/ IBC, IPILOT, IPLOT, IPUNCT
COMMON /B2/ KR, N
COMMON /B4/ PHI(2,26), PSI(2,26), LAMDA, ICOUT
COMMON /ER/ EPS1, EPS2, EPS3
COMMON /ESTR/ LAMSTR(300), EFSTR(2,300), EFMSTR(2,300), ERLAM(300)
COMMON /TRUE/ TRULAM, TRUPHI(2,26), PHICON(2,300), LAMCON(300), IFT
INTEGER N
KRO=KR-1
KR1=KR+1
WRITE (6,1) ICOUT
1 FORMAT ('RESULTS OF THE MULTIGROUP METHOD:\n
WRITE (6,10) ICOUT
10 FORMAT (//, 'PROBLEM TERMINATED AFTER',I5, X ' OUTER (POWER) ITERATIONS TO:')
WRITE (6,20) LAMDA
20 FORMAT (/,10X,'LAMDA = ',1PE21.14)
C PRINT OUT EIGENVALUES.
CALL PLOT
WRITE (6,30)
30 FORMAT ('RESULTS AFTER PROBLEM TERMINATION:\n', X 'NUMBER',5X,'THERMAL FLUX POINTS',5X,'FAST FLUX POINTS')
WRITE (6,50) (K, PHI(2,K), PHI(1,K), K=1,KR1)
50 FORMAT (15,1PE26.7,1PE21.7)
IF (IPUNCH.EQ.1) CALL PUNCH
C CALCULATE THE FINAL TO EXPECTED FLUX RATIOS:
C NORMALIZE BOTH PHI GROUP FLUXES FOR TRUPHI COMPARISON:
CALL NORM2(PHI, TRUPHI, KR1)
K1=1
K2=KR1
IF (IBC.LE.2) K1=2
IF ((IBC.EQ.1).OR.(IBC.EQ.3)) K2=KR
DO 60 IG=1,2
60 IF (IBC.LE.2) PSI(IG,1)=1.0
IF ((IBC.EQ.1).OR.(IBC.EQ.3)) PSI(IG,KR1) = 1.0
I=0
DO 60 K=K1,K2
I=I+1
60 PSI(IG,K)=PHI(IG,K)/TRUPHI(IG,I)
WRITE (6,70) (I,PSI(2,I),PSI(1,I),I=1,KR1)
70 FORMAT (*1RATIOS OF THE TERMINATED GROUP FLUX TO THE EXPECTED GROUP
XP FLUX:*,,
X 10X,'- AN INDICATION OF THE ACCURACY OF THE CONVERGENCE -',//,,
X 'K',12X,'THERMAL RATIO',15X,'FAST RATIO',//,(I5,2E25.10))
WRITE (6,110) EPS1,(EFSTR(2,I),I=1,ICOUT)
WRITE (6,111) EPS1,(EFSTR(1,I),I=1,ICOUT)
WRITE (6,112) EPS2,(EFMSTR(2,I),I=1,ICOUT)
WRITE (6,113) EPS3,(EFMSTR(1,I),I=1,ICOUT)
WRITE (6,114) EPS3,(ERLAM(I),I=1,ICCUT)
110 FORMAT (*1MAXIMUM NORMALIZED ERRORS BETWEEN THE THERMAL FLUX ITERA
XTIONS:*,,
X 25X,'TOLERANCE USED = ',E12.4,///,
X '1PE12.4, , (1P5E20.5))
111 FORMAT (*1MAXIMUM NORMALIZED ERRORS BETWEEN THE FAST FLUX ITERATIO
XNS:*,,
X 25X,'TOLERANCE USED = ',E12.4,///,
X '1PE12.4, , (1P5E20.5))
112 FORMAT (*1MEAN SQUARE NORMALIZED ERRCR BETWEEN THE THERMAL FLUX IT
XERATIONS:*,,
X 18X,'TOLERANCE USED = ',E12.4,///,
X '1PE12.4, , (1P5E20.5))
113 FORMAT (*1MEAN SQUARE NORMALIZED ERRCR BETWEEN THE FAST FLUX ITERA
XTIONS:*,,
X 18X,'TOLERANCE USED = ',E12.4,///,
X '1PE12.4, , (1P5E20.5))
114 FORMAT (*1ERROR BETWEEN THE ITERATION EIGENVALUES:*,,
X 28X,'TOLERANCE USED = ',E12.4,///,
X '1PE12.4, , (1P5E20.5))
C PRINT OUT THE GIVEN TRUE EIGENVALUE AND FLUX:
IF (IFT.EQ.0) RETURN
WRITE (6,115) TRULAM,((TRUPHI(3-J,I),J=1,2),I=1,N)
115 FORMAT (*1THE GIVEN TRUE EIGENVALUE:*,,15X,
X 'TRULAM =',E22.14,///,,
X '0THE GIVEN MULTIGROUP FLUXES:*,,,
C PRINT OUT THE STORED CONVERGENCE ERRORS:
WRITE (6,120) (PHICON(2,I),I=1,ICCUT)
WRITE (6,121) (PHICON(1,I),I=1,ICCUT)
WRITE (6,122) (LAMCON(I),I=1,ICCUT)
120 FORMAT ('1MEAN SQUARE ERROR BETWEEN THE THERMAL ITERATION FLUX AND X THE GIVEN TRUE THERMAL FLUX:','//,(1P5E20.5))
121 FORMAT ('1MEAN SQUARE ERROR BETWEEN THE FAST ITERATION FLUX AND TH XE GIVEN TRUE FAST FLUX:','//,(1P5E20.5))
122 FORMAT ('1ERROR BETWEEN THE ITERATION EIGENVALUES AND THE GIVEN TR XUE EIGENVALUE:','//,(1P5E20.5))
RETURN
END
SUBROUTINE PLOT
C PLOTS OUT THE EIGENVALUE HISTORY AS A TABLE AND A GRAPH,
C AS WELL AS PLOTTING OUT THE FINAL MULTIGROUP FLUX SHAPES.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /B1/ IBC,IPLT,JPLT,IPUNCH
COMMON /B2/ KR
COMMON /B4/ PHI(2,26), PSI(2,26), LAMDA, ICOUT
COMMON /B5/ B(300,2)
COMMON /ESTR/ LAMSTR(300)
DIMENSION C(26,3)
C IN ORDER TO SAVE SOME SPACE:
EQUIVALENCE (B(1),C(1))
WRITE (6,1) (LAMSTR(I),I=1,ICOUT)
1 FORMAT ("TABLE OF EIGENVALUES DURING THE POWER ITERATION:",
X //,(1PE25.14))
IF (JPLUT.EQ.0) GO TO 20
DO 10 I=1,ICOUT
10 B(I,1)=1
B(I,2)=LAMSTR(I)
CALL PRTPLT(1,B,ICOUT,2,ICOUT,0,300,2,1)
WRITE (6,11)
11 FORMAT ("OF THE EIGENVALUE HISTORY THROUGH THE ITERATIONS.")
20 IF (JPLUT.EQ.0) RETURN
KR1=KR+1
DO 30 I=1,KR1
C(I,1)=1
C(I,2)=PHI(I,1)
30 C(I,3)=PHI(I,2)
CALL PRTPLT(2,C,KR1,3,KR1,0,26,3,2)
WRITE (6,31)
31 FORMAT ("FINAL CONVERGED CONNECTING FLUX POINTS; F(K).",//,
X 5X,"FAST FLUX: ".",//,5X,"THERMAL FLUX: -")
RETURN
END
SUBROUTINE PUNCH
        PUNCHES OUT INPUT AND OUTPUT DATA.
COMMON /B2/ KR
COMMON /B4/ F(2,26)
REAL*8 F
KR1=KR+1
WRITE (7,1) KR,(F(1,I),F(2,I),I=1,KR1)
1 FORMAT (I5,/(2E20.7))
WRITE (6,100)
100 FORMAT ('///,' THE OUTPUT HAS BEEN PUNCHED OUT ONTO CARDS ')
RETURN
END
SUBROUTINE NORM2(PSI,TRUPHI,N)
C NORMALIZES BOTH ENERGY GROUPS OF PSI TO 1.0.
C DITTO FOR TRUPHI ON THE FIRST CALL.
REAL*8 PSI(2,26), TRUPHI(2,26), A(2)
DATA K /0/
K=K+1
DO 1 IG=1,2
A(IG)=DABS(PSI(IG,1))
DO 1 I=1,N
IF (DABS(PSI(IG,I)).GT.A(IG)) A(IG)=DABS(PSI(IG,I))
1 CONTINUE
DO 2 IG=1,2
PSI(IG,1)=PSI(IG,1)/A(IG)
DO 2 I=1,N
2 CONTINUE
IF (K.NE.1) RETURN
DO 5 IG=1,2
A(IG)=0.
DO 5 I=1,N
IF (TRUPHI(IG,I).GT.A(IG)) A(IG)=TRUPHI(IG,I)
5 CONTINUE
DO 6 IG=1,2
TRUPHI(IG,1)=TRUPHI(IG,1)/A(IG)
RETURN
END

PAGE 285
SUBROUTINE PRTPLT(NO, B, N, M, NL, NS, KX, JX, ISP)

* IDENTICAL TO SUBROUTINE PRTPLT PREVIOUSLY LISTED IN PROGRAM REF2G.

RETURN
END
F.3. SOURCE LISTING of Program CUBIC
Figure F.3. Structure of Program CUBIC.
C PROGRAM CUBIC:
C TWO GROUP PROPOSED METHOD USING CUBIC HERMITE BASIS FUNCTIONS.
CALL TIMING (11)
CALL SYNTH
CALL TIMING (14)
CALL POWER
CALL TIMING (16)
CALL CURENT
CALL TIMING (17)
CALL OUTPUT
CALL TIMING (18)
C TIMING EXECUTION
WRITE (6,30)
30 FORMAT (1H1,'TIMING PROGRAM EXECUTION:',/)
J=14-11
WRITE(6,701) J
J=16-14
WRITE(6,704) J
J=17-16
WRITE(6,706) J
J=18-17
WRITE(6,707) J
701 FORMAT (1H1,' SYNTH HAS TAKEN',16,' /100 SECONDS. ')
704 FORMAT (1H1,' POWER HAS TAKEN',16,' /100 SECONDS. ')
706 FORMAT (1H1,' CURENT HAS TAKEN',15,' /100 SECONDS. ')
707 FORMAT (1H1,' OUTPUT HAS TAKEN',15,' /100 SECONDS. ')
CALL TIMING (120)
J=120-11
WRITE(6,720) J
720 FORMAT (1H1,' THIS RUN HAS TAKEN',16,' /100 SECONDS TO RUN. ')
STOP
END
SUBROUTINE SYNTH

PROPOSED CUBIC HERMITE SYNTHESIS METHOD:

ADJOINT QUANTITIES OF VARIABLES ARE DENOTED BY $\tilde{\text{7}}$ RATHER THAN $\star$.

THUS: $\tilde{\text{PHI}}_12$ (RATHER THAN $\text{PHI}_1$) IS THE ADJOINT OF $\text{PHI}_1$. ETC.

IMPLICIT REAL*8 (A-H,K-Z)

COMMON /81/ IBC, IJPLT, JJPLT, IPUNCH, ISEE

COMMON /82/ KR, NN

COMMON /83/ LT(50,6,2), FT(50,6,4), T(50,6)

COMMON /85/ KA0(2,25), KA6(2,25), K85(2,25), LA4(2,25), P3(2,25), Q2(2,25), R1(2,25), SR2(1,25), KC0(1,25), KD(1,25), K06(i,25), P0(2,25), BC, IPLOTJPLUT, IPUNCH, ISEE

RNN

T(50,6,2), FT(50,6,4), T(50,6)


KB5(2,25), KB6(2,25), KA7(2,25), LA2(2,25), LA3(2,25),


R1(2,25), R2(2,25), R3(2,25), R4(2,25), Q1(2,25),

Q2(2,25), Q3(2,25), Q4(2,25), Q5(2,25), Q6(2,25), R0(2,25),

KR, PP(2,25), PH(2,25), PO7(2,25), PH7(2,25), DO(2,25), DH(2,25),

C0(2), CH(2), TITLE(2), ITF(25), KTF(25)

COMMON /CHIF/ CHI(2)

COMMON /XAXIS/ HX, HR(25)

COMMON /XH/ X(101), H(101)

COMMON /ER/ EPS1, EPS2, EPS3

DIMENSION PHI(2,101), PH17(2,101), CUR(2,101), CUR7(2,101)

DIMENSION A(2,100), F(2,100), D(2,100), S(2,100), DI(2,100), XU(2,100)

IN ORDER TO SAVE SPACE:

EQUIVALENCE (XU(1), LT(1)), (A(1), LT(201)), (F(1), LT(401)),

X (DI(1), FT(1)), (D(1), FT(201)), (S(1), FT(401))

REAL TITLE

INTEGER KR, KS, KS1, KRS, NN, NUMITF, KTF, N

READ (5,200) TITLE

200 FORMAT (20A4)
WRITE (6,201) TITLE
201 FORMAT (1H1,20A4, //)
C READ IN THE NUMBER OF REGION TRIAL FUNCTIONS AND TYPE OF B.C.S.
C AS WELL AS THE TOLERANCES AND THE OUTPUT TYPES DESIRED:
READ (5,1) KR, IBC, EPS1, EPS2, EPS3, IPLCT, JPLLOT, IPUNCH, ISEE, ITW, ITC
1 FORMAT (2I5, 3D10.3, 6I5)
IF (IBC.EQ.3) IBC=2
C READ IN THE TYPE-NUMBER OF EACH TF REGION:
READ (5,100) (ITF(I), I=1,KR)
100 FORMAT (25I2)
C READ IN THE FISSION YIELDS FOR EACH GROUP:
C AND THE MATRIX NORMALIZATION PARAMETER: THETA (DEFAULT = 1.0):
READ (5,101) CHI(1), CHI(2), THETA
101 FORMAT (3F10.5)
IF (THETA.EQ.0.0) THETA=1.0
KRO=KR-1
WRITE (6,2) KR, IBC, ISEE, ITW, ITC
2 FORMAT ('ONE DIMENSIONAL TWO GROUP CUBIC SYNTHESIS PROGRAM:', //,
X 5X,'NUMBER OF COARSE MESH REGIONS: KR = ',I2, '/',
X 5X,'BOUNDARY CONDITION NUMBER: IBC = ',I2, ' /',
X 5X,'AMOUNT OF OUTPUT REQUESTED: ISEE = ',I2, ' /',
X 5X,'TYPE OF WEIGHTING FUNCTIONS: ITW = ',I2, ' /',
X 5X,'TYPE OF CURRENT FUNCTIONS: ITC = ',I2, ' /',
X 5X,'REGIONAL INPUT MATERIAL PROPERTIES AND FLUX SHAPES FOLLOW',
X 7X, 'IF ISEE > 0: ', //,
X 5X,'FLUX SHAPES ARE LINEAR IN EACH INDICATED SUBREGION.')
IF (ITC.EQ.0) WRITE (6,16)
IF (ITC.EQ.1) WRITE (6,17)
16 FORMAT (5X,'CURRENTS ARE CONSTANT IN EACH INDICATED SUBREGION.' )
17 FORMAT (5X,'CURRENTS ARE LINEAR IN EACH INDICATED SUBREGION.' )
IF (ITW.EQ.0) WRITE (6,116)
IF (ITW.EQ.1) WRITE (6,117)
116 FORMAT ('WEIGHTING FLUX = FLUX; ', //, 5X,'WEIGHTING CURRENT = -XCURRENT.' )
117 FORMAT ('WEIGHTING FLUX = ADJOINT FLUX; ', //, 5X,'WEIGHTING CURR
XENT = ADJOINT CURRENT.' )
WRITE (6,20) EPS1, EPS2, EPS3, IPLOT, JPLOT, IPUNCH
20 FORMAT (/,'OTOLERANCES TO POWER ARE: EPS1 = ',1PD10.3,/, X 28X,'EPS2 = ',1PD10.3,/, 28X,'EPS3 = ',1PD10.3,/, X 'OUTPUT PARAMETERS TO POWER ARE: IPLOT = ',I1,/, X 34X,'JPLOT = ',I1,/, 34X,'IPUNCH = ',I1)
WRITE (6,22) CHI(1), CHI(2), THETA
22 FORMAT (/,'OFISSION YIELDS ARE: CHI(1) = ',F10.5,/, X 22X,*CHI(2) = ',F10.5,/, X 'INPUT THETA PARAMETER (FOR MATRICES) = ',D15.7)
IF ((KR.LE.2).AND.(IBC.EQ.1)) CALL ERROR(1,KR)
IF (KR.GT.25) CALL ERROR(2,KR)
IF (EPS1.LT.1.0E-16) CALL ERROR(6,1)
IF (EPS2.LT.1.0E-16) CALL ERROR(6,2)
IF (EPS3.LT.1.0E-16) CALL ERROR(6,3)
IF ((IBC.LT.1).OR.(IBC.GT.4)) CALL ERROR(7,IBC)
C DUMMY NORMAL VECTOR XU = UNITY. (FOR THE INTEGRATION FUNCTIONS)
DO 21 IG=1,2
DO 21 II=1,100
21 XU(IG,II)=1.0
ITCO=2
ITCI=2
IF (ITC.EQ.1) GO TO 23
ITC0=0
ITC1=1
C COUNTER OF THE NUMBER OF TYPE-NUMBERS OF EACH TF REGION:
23 NUMITF=1
HX=0.0
C BEGIN TO READ IN THE TF REGION DATA AND FILL THE ARRAYS, DEPENDING ON THE TYPE-NUMBER OF EACH TF REGION.
DO 50 I=1,KR
IF (ITF(I).EQ.NUMITF) GO TO 110
C FILL THE ARRAYS FROM OLD TF REGION TYPES:
J=ITF(I)
CALL REPEAT(I,KTF(J))
GO TO 50
C READ IN THE TF REGION'S DATA FOR NEW TF REGION TYPE-NUMBERS:
110 NUMITF=NUMITF+1.
   KTF(NUMITF-1)=1
C READ THE SUBREGION NUMBER AND THE NUMBER OF REGIONS IN THE SUBREGION:
READ (5,1) K, KS
IF (KS.GT.100) CALL ERROR(3,I)
KS1=KS+1
C CHECK FOR IMPROPER SEQUENCING OF INPUT DATA:
IF (I.NE.K) CALL ERROR(4,I)
C READ IN THE GEOMETRY AND THE MATERIAL PROPERTIES:
READ (5,3) (X(J),X(J+1),H(J),A(1,J),F(1,J),D(1,J),S(1,J),
   X A(2,J),F(2,J),D(2,J),J=1,KS)
3 FORMAT (3F10.5,4D10.3,/) 3JX,3D10.6
C READ IN THE REGIONAL GROUP TRIAL FUNCTIONS:
READ (5,4) (PHI(1,J),CUR(1,J),PHI7(1,J),CUR7(1,J),J=1,KS1)
READ (5,4) (PHI(2,J),CUR(2,J),PHI7(2,J),CUR7(2,J),J=1,KS1)
4 FORMAT (4D20.7)
IF (ITW.EQ.1) GO TO 123
C FORM WEIGHTING FUNCTIONS FROM THE GIVEN FUNCTIONS:
DO 119 IG=1,2
   DO 119 J=1,KS1
      PHI7(IG,J)=PHI(IG,J)
      CUR7(IG,J)=CUR(IG,J)
119      IF (ITC.EQ.1) GO TO 5
C FORM THE REGION CONSTANT CURRENTS FROM THE FLUXES:
DO 120 IG=1,2
   DO 120 J=1,KS
      CUR(IG,J)=-D(IG,J)*(-PHI(IG,J)+PHI(IG,J+1))/H(J)
      CUR7(IG,J)=+(IG,J)*(-PHI7(IG,J)+PHI7(IG,J+1))/H(J)
      CUR(IG,KS1)=0.0
5      CUR7(IG,KS1)=0.0
C WRITE OUT THE INPUT INFORMATION IF ISEE .GE.2:
5 IF (ISEE.LE.1) GO TO 14
   WRITE (6,1J) K,KR,KS,(J,X(J),X(J+1),H(J),A(1,J),F(1,J),D(1,J),
   X S(1,J),A(2,J),F(2,J),D(2,J),J=1,KS)
10 FORMAT ('INPUT MATERIAL PROPERTIES FOR REGION NUMBER ',I3,
   X ' OF THE ',I3,' USED.',//,
X 5X,'THIS REGION IS DIVIDED INTO ',I3,' HOMOGENEOUS SUBREGIONS A
XS FOLLOWS:',//,
X 5X,'FAST GROUP CONSTANTS APPEAR FIRST:',//,
X 4X,'SUBREGION #',5X,'INTERNAL BOUNDARIES',10X,'WIDTH',3X,
X 4X,'TOTAL CX (1/CM)',3X,'FISSION CX (1/CM)',6X,'DIFFUSION (CM)',
X 4X,'SCATT.CX (1/CM)',/,X 5X,'I',11X,'X(I)',9X,'X(I+1)',11X,'H(I)',13X,'A(IG,I)',13X,
X 4X,'F(IG,I)',13X,'D(IG,I)',14X,'S(IG,I)',/,X 5X,'TOTAL CX (1/CM)',//,
X (16,3F15.4,4D20.8,/,51X,3D20.8))
DO 15 IG=1,2
15 WRITE (6,11) IG,K,KR,(J,X(J),PHI(IG,J),CUR(IG,J),PHI7(IG,J),
X CUR7(IG,J),J=1,KS1)
11 FORMAT ('INPUT TRIAL FUNCTIONS FOR GROUP',I2,' FOR REGION',I3,
X 'OUT OF THE',I3,' USED:',//,
X 'INDEX',5X,'COORD',16X,'FLUX',13X,'CURRENT',8X,'WEIGHT FLUX',
X 5X,'WEIGHT CURRENT',//,(I6,F10.5,4D20.7))
14 CONTINUE
C END OF THE IN-OUT SECTION:
C DEFINING MISC. ARRAYS FOR THE INTEGRATION FUNCTIONS:
C LENGTH OF THE SUBREGION: HT
HT=X(KS1)-X(1)
HR(K)=HT
HX=HX+HR(K)
C INVERSE OF THE D ARRAYS:
DO 13 J=1,KS
13 DI(1,J)=1./DI(1,J)
DO 13 J=1,KS
13 DI(2,J)=1./DI(2,J)
C FORMATION OF THE INTEGRATION FUNCTIONS:
CALL BHSET(KS)
C DO FOR ALL ENERGY GROUPS:
DO 50 IG=1,2
KA0(IG,K)=GIF(0,IG,PHI7,IG,A,PHI,KS,2)
KA1(IG,K)=GIF(1,IG,PHI7,IG,A,PHI,KS,2)
KA2(IG,K)=GIF(2,IG,PHI7,IG,A,PHI,KS,2)
KA3(IG,K)=GIF(3,IG,PHI7,IG,A,PHI,KS,2)
KA4(IG,K)=GIF(4,IG,PHI7,IG,A,PHI,KS,2)
STORE THE TERMINAL POINTS FOR LATER USE:

KA5 (IG, K) = GIF (5, IG, PHI, IG, A, PHI, KS, 2)
KA6 (IG, K) = GIF (6, IG, PHI, IG, A, PHI, KS, 2)
KB0 (IG, K) = GIF (0, IG, PHI, IG, F, PHI, KS, 2)
KB1 (IG, K) = GIF (1, IG, PHI, IG, F, PHI, KS, 2)
KB2 (IG, K) = GIF (2, IG, PHI, IG, F, PHI, KS, 2)
KB3 (IG, K) = GIF (3, IG, PHI, IG, F, PHI, KS, 2)
KB4 (IG, K) = GIF (4, IG, PHI, IG, F, PHI, KS, 2)
KB5 (IG, K) = GIF (5, IG, PHI, IG, F, PHI, KS, 2)
KB6 (IG, K) = GIF (6, IG, PHI, IG, F, PHI, KS, 2)
LA0 (IG, K) = GIF (0, IG, CUR, IG, DI, CUR, KS, ITC0)
LA1 (IG, K) = GIF (1, IG, CUR, IG, DI, CUR, KS, ITC0)
LA2 (IG, K) = GIF (2, IG, CUR, IG, DI, CUR, KS, ITC0)
LA3 (IG, K) = GIF (3, IG, CUR, IG, DI, CUR, KS, ITC0)
LA4 (IG, K) = GIF (4, IG, CUR, IG, DI, CUR, KS, ITC0)
LA5 (IG, K) = GIF (5, IG, CUR, IG, DI, CUR, KS, ITC0)
LA6 (IG, K) = GIF (6, IG, CUR, IG, DI, CUR, KS, ITC0)
P0 (IG, K) = GIF (0, IG, PHI, IG, XU, CUR, KS, ITC1) / HT
P1 (IG, K) = GIF (1, IG, PHI, IG, XU, CUR, KS, ITC1) / HT
P2 (IG, K) = GIF (2, IG, PHI, IG, XU, CUR, KS, ITC1) / HT
P3 (IG, K) = GIF (3, IG, PHI, IG, XU, CUR, KS, ITC1) / HT
P4 (IG, K) = GIF (4, IG, PHI, IG, XU, CUR, KS, ITC1) / HT
P5 (IG, K) = GIF (5, IG, PHI, IG, XU, CUR, KS, ITC1) / HT
P6 (IG, K) = GIF (6, IG, PHI, IG, XU, CUR, KS, ITC1) / HT
Q0 (IG, K) = GIF (0, IG, PHI, IG, XU, CUR7, KS, ITC1) / HT
Q1 (IG, K) = GIF (1, IG, PHI, IG, XU, CUR7, KS, ITC1) / HT
Q2 (IG, K) = GIF (2, IG, PHI, IG, XU, CUR7, KS, ITC1) / HT
Q3 (IG, K) = GIF (3, IG, PHI, IG, XU, CUR7, KS, ITC1) / HT
Q4 (IG, K) = GIF (4, IG, PHI, IG, XU, CUR7, KS, ITC1) / HT
Q5 (IG, K) = GIF (5, IG, PHI, IG, XU, CUR7, KS, ITC1) / HT
Q6 (IG, K) = GIF (6, IG, PHI, IG, XU, CUR7, KS, ITC1) / HT
R0 (IG, K) = GIF (0, IG, PHI, IG, D, PHI, KS, 2) / HT**2
R1 (IG, K) = GIF (1, IG, PHI, IG, D, PHI, KS, 2) / HT**2
R2 (IG, K) = GIF (2, IG, PHI, IG, D, PHI, KS, 2) / HT**2
R3 (IG, K) = GIF (3, IG, PHI, IG, D, PHI, KS, 2) / HT**2
R4 (IG, K) = GIF (4, IG, PHI, IG, D, PHI, KS, 2) / HT**2
for the off diagonal matrix elements:

IF (IG.EQ.2) GO TO 50
SR0( 1,k)=GIF(0,1,PHI7,1,s,PHI,KS,2)
SR1( 1,k)=GIF(1,1,PHI7,1,s,PHI,KS,2)
SR2( 1,k)=GIF(2,1,PHI7,1,s,PHI,KS,2)
SR3( 1,k)=GIF(3,1,PHI7,1,s,PHI,KS,2)
SR4( 1,k)=GIF(4,1,PHI7,1,s,PHI,KS,2)
SR5( 1,k)=GIF(5,1,PHI7,1,s,PHI,KS,2)
SR6( 1,k)=GIF(6,1,PHI7,1,s,PHI,KS,2)
KC0( 1,k)=GIF(0,1,PHI7,2,F,PHI,KS,2)
KC1( 1,k)=GIF(1,1,PHI7,2,F,PHI,KS,2)
KC2( 1,k)=GIF(2,1,PHI7,2,F,PHI,KS,2)
KC3( 1,k)=GIF(3,1,PHI7,2,F,PHI,KS,2)
KC4( 1,k)=GIF(4,1,PHI7,2,F,PHI,KS,2)
KC5( 1,k)=GIF(5,1,PHI7,2,F,PHI,KS,2)
KC6( 1,k)=GIF(6,1,PHI7,2,F,PHI,KS,2)
KD0( 1,k)=GIF(0,2,PHI7,1,F,PHI,KS,2)
KD1( 1,k)=GIF(1,2,PHI7,1,F,PHI,KS,2)
KD2( 1,k)=GIF(2,2,PHI7,1,F,PHI,KS,2)
KD3( 1,k)=GIF(3,2,PHI7,1,F,PHI,KS,2)
KD4( 1,k)=GIF(4,2,PHI7,1,F,PHI,KS,2)
KD5( 1,k)=GIF(5,2,PHI7,1,F,PHI,KS,2)
KD6( 1,k)=GIF(6,2,PHI7,1,F,PHI,KS,2)

50 CONTINUE
NUMITF=NUMITF-1
WRITE (6,51) NUMITF

51 FORMAT ('THERE ARE ONLY',13,' DIFFERENT TRIAL FUNCTION REGIONS.')
IF (ISEE.GE.2) CALL PRTOUT(I)

C DETERMATIONS OF THE B.C. OPTION PARAMETERS:

C NN IS THE BLOCK SIZE OF THE POWER MATRICES.
NN=2*KR

C FORMATION OF THE COEFFICIENT VECTORS:
C FILLING THE MATRICES FOR POWER:
C FOR BOTH ENERGY GROUPS:

DO 60 IG=1,2

C THE MATRIX ROW INDEX:
I=1
C FOR ALL THE INTERIOR COEFFICIENTS:

DO 60 J=K+1

V=1./($P7(I,G,J)*P0(I,G,J))
V1=1./($P7(I,G,J)*PH(I,G))
V2=1./($P7(I,G,K)*P0(I,G,K))
V3=1./($P7(I,G,K)*PH(I,G))


-6.*KA6(I,G,J)+6.*KA7(I,G,J)+6.*KA8(I,G,J)+6.*KA9(I,G,J))

LT(1,2,IG)=(-3.*KA3(I,G,J)-8.*KA4(I,G,J)-7.*KA5(I,G,J)+2.*KA6(I,G,J)
+4.*KA7(I,G,J)+6.*KA8(I,G,J)+6.*KA9(I,G,J))

-6.*KA6(I,G,J)+6.*KA7(I,G,J)+6.*KA8(I,G,J)+6.*KA9(I,G,J))

-6.*KA6(I,G,J)+6.*KA7(I,G,J)+6.*KA8(I,G,J)+6.*KA9(I,G,J))

-6.*KA6(I,G,J)+6.*KA7(I,G,J)+6.*KA8(I,G,J)+6.*KA9(I,G,J))


PAGE 297
\[
\begin{align*}
\text{LT}(1, 2, IG) &= \frac{1}{2}(3 \cdot KA4(IG, J) - 9 \cdot KA5(IG, J) - 12 \cdot KA5(IG, K) + 3 \cdot KA2(IG, K) - 2 \cdot KA3(IG, K) - 9 \cdot KA5(IG, K) + 12 \cdot KA5(IG, K)) \\
\text{LT}(1, 3, IG) &= \frac{1}{2}(3 \cdot KA4(IG, K) - 9 \cdot KA5(IG, K) - 12 \cdot KA5(IG, K) + 4 \cdot KB3(IG, J) - 9 \cdot KB4(IG, J) + 12 \cdot KB5(IG, J)) \\
\text{FT}(1, 1, IG) &= -3 \cdot KB3(IG, J) + 8 \cdot KB4(IG, J) - 7 \cdot KB5(IG, J) + 2 \cdot KB6(IG, J)) \\
\text{FT}(1, 2, IG) &= -3 \cdot KB3(IG, J) + 8 \cdot KB4(IG, J) - 7 \cdot KB5(IG, J) + 2 \cdot KB6(IG, J)) \\
\text{FT}(1, 3, IG) &= 18 \cdot Q3(IG, J) - 30 \cdot Q4(IG, J) + 12 \cdot Q5(IG, J) \\
\text{LT}(1, 4, IG) &= 18 \cdot Q3(IG, J) - 30 \cdot Q4(IG, J) + 12 \cdot Q5(IG, J) \\
\text{FT}(1, 4, IG) &= 18 \cdot Q3(IG, J) - 30 \cdot Q4(IG, J) + 12 \cdot Q5(IG, J) \\
\end{align*}
\]
\[
\begin{align*}
X &= -8.0K4(IG, K) + 7.0K5(IG, K) - 2.0K6(IG, K)) * HR(K) \\
X &= (P07(2, K) * P01(1, K) * D01(1, K)) \\
FT(1, 5, 3) &= [3.0K02(IG, K) - 2.0K3(IG, K) - 9.0K4(IG, K) + 12.0K5(IG, K)] \\
X &= -4.0K6(IG, K)) * HR(K) / (PO7(2, K) * PH(1, K)) \\
FT(1, 6, 3) &= K02(IG, K) - K03(IG, K) - 3.0K4(IG, K) + 5.0K5(IG, K) \\
X &= -2.0K6(IG, K)) * HR(K) / (PO7(2, K) * PH(1, K) * DH(1, K)) \\
FT(1, 1, 4) &= [3.0K2(IG, K) - 2.0K3(IG, K) - 9.0K4(IG, K) + 12.0K5(IG, K)] \\
X &= -4.0K6(IG, K)) / (PH(1, K) * PC(2, K)) \\
FT(1, 2, 4) &= [3.0K3(IG, K) + 8.0K4(IG, K) - 7.0K5(IG, K) + 2.0K6(IG, K)] \\
X &= HR(1, 1) / (PH(1, 1) * P02(1, K) * D01(1, K)) \\
FT(1, 3, 4) &= [9.0K3(IG, K) - 12.0K5(IG, K) + 4.0K6(IG, K)] \\
X &= (PH(1, 1) * PH(2, 2)) \\
X &= [K01(IG, K) + 6.0K2(IG, K) + 4.0K3(IG, K) + 9.0K4(IG, K) \\
X &= -12.0K5(IG, K) + 4.0K6(IG, K)] / (P07(1, K) * P02(2, K)) \\
FT(1, 4, 4) &= [3.0K4(IG, K) + 5.0K5(IG, K) + 2.0K6(IG, K)] \\
X &= HR(1, 1) / (PH(1, 1) * PH(2, 2)) \\
X &= [K01(IG, K) + 2.0K2(IG, K) + 2.0K3(IG, K)] \\
X &= -8.0K4(IG, K) + 7.0K5(IG, K) - 2.0K6(IG, K)) * HR(K) \\
X &= (P07(1, K) * P02(2, K) * D01(2, K)) \\
FT(1, 5, 4) &= [3.0K2(IG, K) - 2.0K3(IG, K) - 9.0K4(IG, K) + 12.0K5(IG, K)] \\
X &= -4.0K6(IG, K)) / (PH(1, K) * PH(2, K)) \\
FT(1, 6, 4) &= [K2(IG, K) - K03(IG, K) - 3.0K4(IG, K) + 5.0K5(IG, K)] \\
X &= -2.0K6(IG, K)) * HR(K) / (PH(1, K) * DH(2, K)) \\
X &= 57 \quad i = 1+1 \\
LT(1, 1, 1, 1) &= [K02(IG, J) - K03(IG, J) + 3.0K4(IG, J) + 5.0K5(IG, J) - 2.0K6(IG, J)] \\
X &= [K04(IG, J) - [K04(IG, J) - [LA2(IG, J) - LA3(IG, J) - 3.0LA4(IG, J) + 5.0LA5(IG, J) - 2.0LA6(IG, J)] \\
X &= -2.0P1(IG, J) + 3.0P2(IG, J) + 6.0P3(IG, J) - 13.0P4(IG, J) + 6.0P5(IG, J)] \\
X &= [30.0K3(IG, J) - 18.0R4(IG, J)] * VRHR(J) / (PH(1, K) * DH(IG, J)) \\
LT(1, 2, 1) &= [K01(IG, J) + 3.0K4(IG, J) - 3.0KA5(IG, J) + KA6(IG, J)] \\
X &= [-LA3(IG, J) + 3.0LA4(IG, J) - 3.0LA5(IG, J) + LA6(IG, J)] + 2.0P2(IG, J) \\
X &= 7.0P3(IG, J) + 8.0P4(IG, J) - 3.0P5(IG, J) - 5.0P6(IG, J) - 6.0P7(IG, J) - 7.0P8(IG, J) \\
X &= [7.0P9(IG, J) - 3.0Q5(IG, J)] - 2.0R1(IG, J) + 11.0R2(IG, J) - 18.0R3(IG, J) \\
X &= [R3(IG, J) + 9.0R4(IG, J)] * VRHR(J) / (PH(1, K) * DH(IG, J)) \\
LT(1, 3, 1) &= [3.0K02(IG, J) - 5.0K04(IG, J) + 2.0KA6(IG, J) - 3.0LA4(IG, J)] \\
X &= -5.0LA5(IG, J) + 2.0LA6(IG, J) - 6.0P3(IG, J) + 13.0P4(IG, J) - 6.0P5(IG, J) \\
X &= 57 \quad i = 1+1 \\
LT(1, 1, 1, 1) &= [K02(IG, J) - K03(IG, J) + 3.0K4(IG, J) + 5.0K5(IG, J) - 2.0K6(IG, J)] \\
X &= [K04(IG, J) - [K04(IG, J) - [LA2(IG, J) - LA3(IG, J) - 3.0LA4(IG, J) + 5.0LA5(IG, J) - 2.0LA6(IG, J)] \\
X &= -2.0P1(IG, J) + 3.0P2(IG, J) + 6.0P3(IG, J) - 13.0P4(IG, J) + 6.0P5(IG, J)] \\
X &= [30.0K3(IG, J) - 18.0R4(IG, J)] * VRHR(J) / (PH(1, K) * DH(IG, J)) \\
LT(1, 2, 1) &= [K01(IG, J) + 3.0K4(IG, J) - 3.0KA5(IG, J) + KA6(IG, J)] \\
X &= [-LA3(IG, J) + 3.0LA4(IG, J) - 3.0LA5(IG, J) + LA6(IG, J)] + 2.0P2(IG, J) \\
X &= 7.0P3(IG, J) + 8.0P4(IG, J) - 3.0P5(IG, J) - 5.0P6(IG, J) - 6.0P7(IG, J) - 7.0P8(IG, J) \\
X &= [7.0P9(IG, J) - 3.0Q5(IG, J)] - 2.0R1(IG, J) + 11.0R2(IG, J) - 18.0R3(IG, J) \\
X &= [R3(IG, J) + 9.0R4(IG, J)] * VRHR(J) / (PH(1, K) * DH(IG, J)) \\
LT(1, 3, 1) &= [3.0K02(IG, J) - 5.0K04(IG, J) + 2.0KA6(IG, J) - 3.0LA4(IG, J)] \\
X &= -5.0LA5(IG, J) + 2.0LA6(IG, J) - 6.0P3(IG, J) + 13.0P4(IG, J) - 6.0P5(IG, J)
\[
\begin{align*}
&\text{P5(IG, J)+6.*Q3(IG, J)-12.*Q4(IG, J)+6.*Q5(IG, J)-(12.*R2(IG, J)} \\
&+30.*R3(IG, J)-18.*R4(IG, J)) V_1*H(R(J)/D(H(IG, J)) \\
&+(-K41(IG, K)+2.*K42(IG, K)+2.*K43(IG, K)-8.*K44(IG, K)+7.*K45(IG, K)+7.*K46(IG, K)) \\
&-K51(IG, K)-2.*K52(IG, K)-1.*K53(IG, K)+2.*K54(IG, K)+2.*K55(IG, K)-8.*K56(IG, K)) \\
&*LA4(IG, K)+7.*LA5(IG, K)-1.*LA6(IG, K)+P0(IG, K)-4.*P1(IG, K)+14.*P2(IG, K)+18.*P3(IG, K) \\
&-LA5(IG, K)+17.*P4(IG, K)+6.*P5(IG, K)-L-6.*Q2(IG, K)+18.*Q3(IG, K) \\
&-LA6(IG, K)+18.*Q4(IG, K)+6.*Q5(IG, K)+6.*Q6(IG, K)+18.*R1(IG, K)-30.*R2(IG, K)+42.*R3(IG, K) \\
&+R3(IG, K)-18.*R4(IG, K)) V_2*H(R(K)/D(H(IG, K)) \\
LT(I, 4, IG)={K4A(IG, J)+2.*K4A2(IG, K)+2.*K4A3(IG, K)-8.*K4A4(IG, K)+7.*K4A5(IG, K)+7.*K4A6(IG, K)} \\
L(I, 4, IG)={L4A(IG, J)+2.*L4A2(IG, K)+2.*L4A3(IG, K)-8.*L4A4(IG, K)+7.*L4A5(IG, K)+7.*L4A6(IG, K)} \\
+K5A1(IG, K)+2.*K5A2(IG, K)+2.*K5A3(IG, K)-8.*K5A4(IG, K)+7.*K5A5(IG, K)+7.*K5A6(IG, K)) \\
-LA4A(IG, K)+7.*LA5A(IG, K)-1.*LA6A(IG, K)+P0A(IG, K)-4.*P1A(IG, K)+14.*P2A(IG, K)+18.*P3A(IG, K) \\
&-LA5A(IG, K)+17.*P4A(IG, K)+6.*P5A(IG, K)-L-6.*Q2A(IG, K)+18.*Q3A(IG, K) \\
&-LA6A(IG, K)+18.*Q4A(IG, K)+6.*Q5A(IG, K)+6.*Q6A(IG, K)+18.*R1A(IG, K)-30.*R2A(IG, K)+42.*R3A(IG, K) \\
&+R3A(IG, K)-18.*R4A(IG, K)) V_2A*H(R(K)/D(H(IG, K)) \\
LT(I, 5, IG)={L5A(IG, J)+8.*L5A4(IG, K)-7.*L5A5(IG, K)+2.*L5A6(IG, K)} \\
L(I, 5, IG)={L5B(IG, J)+8.*L5B4(IG, K)-7.*L5B5(IG, K)+2.*L5B6(IG, K)} \\
+K5B1(IG, K)+2.*K5B2(IG, K)+2.*K5B3(IG, K)-8.*K5B4(IG, K)+7.*K5B5(IG, K)+7.*K5B6(IG, K)) \\
-LA5B(IG, K)+17.*P4B(IG, K)+6.*P5B(IG, K)-L-6.*Q2B(IG, K)+18.*Q3B(IG, K) \\
&-LA6B(IG, K)+18.*Q4B(IG, K)+6.*Q5B(IG, K)+6.*Q6B(IG, K)+18.*R1B(IG, K)-30.*R2B(IG, K)+42.*R3B(IG, K) \\
&+R3B(IG, K)-18.*R4B(IG, K)) V_2B*H(R(K)/D(H(IG, K)) \\
LT(I, 6, IG)={L6A(IG, J)+3.*L6A4(IG, K)-3.*L6A5(IG, K)+2.*L6A6(IG, K)} \\
L(I, 6, IG)={L6B(IG, J)+3.*L6B4(IG, K)-3.*L6B5(IG, K)+2.*L6B6(IG, K)} \\
+K6A1(IG, K)+2.*K6A2(IG, K)+2.*K6A3(IG, K)-8.*K6A4(IG, K)+7.*K6A5(IG, K)+7.*K6A6(IG, K)) \\
-LA6A(IG, K)+17.*P4A(IG, K)+6.*P5A(IG, K)-L-6.*Q2A(IG, K)+18.*Q3A(IG, K) \\
&-LA6B(IG, K)+18.*Q4B(IG, K)+6.*Q5B(IG, K)+6.*Q6B(IG, K)+18.*R1B(IG, K)-30.*R2B(IG, K)+42.*R3B(IG, K) \\
&+R3B(IG, K)-18.*R4B(IG, K)) V_2B*H(R(K)/D(H(IG, K)) \\
LT(I, 7, IG)={L7A(IG, J)+12.*L7A4(IG, K)-12.*L7A5(IG, K)+12.*L7A6(IG, K)} \\
L(I, 7, IG)={L7B(IG, J)+12.*L7B4(IG, K)-12.*L7B5(IG, K)+12.*L7B6(IG, K)} \\
+K7A1(IG, K)+2.*K7A2(IG, K)+2.*K7A3(IG, K)-8.*K7A4(IG, K)+7.*K7A5(IG, K)+7.*K7A6(IG, K)) \\
-LA7A(IG, K)+17.*P4A(IG, K)+6.*P5A(IG, K)-L-6.*Q2A(IG, K)+18.*Q3A(IG, K) \\
&-LA7B(IG, K)+18.*Q4B(IG, K)+6.*Q5B(IG, K)+6.*Q6B(IG, K)+18.*R1B(IG, K)-30.*R2B(IG, K)+42.*R3B(IG, K) \\
&+R3B(IG, K)-18.*R4B(IG, K)) V_2B*H(R(K)/D(H(IG, K)) \\
FT(I, 1, IG)={K3B2(IG, J)-K3B3(IG, J)-3.*K3B4(IG, J)+5.*K3B5(IG, J)} \\
&-2.*K3B6(IG, J)+V*H(R(J)/D(H(IG, J)) \\
FT(I, 2, IG)={-K3B2(IG, J)+3.*K3B4(IG, J)-3.*K3B5(IG, J)+K3B6(IG, J)} \\
&-2.*K3B6(IG, J)+V*H(R(J)/D(H(IG, J)) \\
FT(I, 3, IG)={-K3B2(IG, J)+3.*K3B4(IG, J)-3.*K3B5(IG, J)+K3B6(IG, J)} \\
&-2.*K3B6(IG, J)+V*H(R(J)/D(H(IG, J)) \\
FT(I, 4, IG)={K3B2(IG, J)-2.*K3B5(IG, J)+K3B6(IG, J)} \\
&-2.*K3B6(IG, J)+V*H(R(J)/D(H(IG, J))
X  *(KD2(IG,K)-4.*KD3(IG,K)+6.*KD4(IG,K))
X  +4.*KD5(IG,K)+KD6(IG,K))**HR(K)**2
X  *(PR7(1,K)*PR0(1,K)*DO(2,K)*DO(1,K))
FT(I,5,3)=-3.*KD3(IG,K)+8.*KD4(IG,K)-7.*KD5(IG,K)+2.*KD6(IG,K))
X  *(HR(K)/(PR7(2,K)*PR0(1,K)*DO(2,K))
FT(I,6,3)=(-KD3(IG,K)+3.*KD4(IG,K)-3.*KD5(IG,K)+KD6(IG,K))
X  *(HR(K)**2/(PR7(2,K)*PR0(1,K)*DO(2,K)*DO(1,K))
FT(I,1,4)=(KC2(IG,J)-KC3(IG,J)+3.*KC4(IG,J)+5.*KC5(IG,J)
X  -2.*KC6(IG,J))**HR(K)/(PR7(1,J)*PC(2,J)*DO(2,J)*DO(1,J))
X  *(HR(K)/(PR7(1,J)**2/(PR0(2,J)*DO(1,J)*DO(2,J)*DO(1,J)))
FT(I,2,4)=(-KC3(IG,J)+3.*KC4(IG,J)-3.*KC5(IG,J)+KC6(IG,J))
X  *(HR(K)/PR7(1,J)**2/(PR0(2,J)*DO(1,J)*DO(2,J)*DO(1,J)))
FT(I,3,4)=(3.*KC4(IG,J)-5.*KC5(IG,J)+2.*KC6(IG,J))
X  *(HR(K)**2/(PR7(1,J)*PC(2,J)*DO(2,J)**2/(PR0(2,J)*DO(1,J)*DO(2,J)))
FT(I,4,4)=-3.*KC3(IG,J)+3.*KC4(IG,J)+7.*KC5(IG,J)-3.*KC6(IG,J))
X  *(HR(K)**2/(PR7(1,J)*PR0(2,J)*DO(1,J))
FT(I,5,4)=-3.*KC3(IG,J)+3.*KC4(IG,J)+7.*KC5(IG,J)-3.*KC6(IG,J))
X  *(HR(K)**2/(PR7(1,J)**2/(PR0(2,J)*DO(1,J)*DO(2,J)*DO(1,J)))
FT(I,6,4)=-3.*KC3(IG,J)+3.*KC4(IG,J)+7.*KC5(IG,J)-3.*KC6(IG,J))
X  *(HR(K)**2/(PR7(1,J))
60 CONTINUE
IF (IBCOE*4.4) GO TO 63
C
ZERO FLUX COEFFICIENTS ON THE LEFT:
DO 61 IG=1,2
V2=1./(PR7(IG,1)*PR0(IG,1))
V3=1./(PR7(IG,1)**2/(PR0(IG,1)))
LT(1,1,IG)=(KA2(IG,1)-4.*KA3(IG,1)+6.*KA4(IG,1)-4.*KA5(IG,1)
X  +KA6(IG,1)-LA2(IG,1)-4.*LA3(IG,1)+6.*LA4(IG,1)-4.*LA5(IG,1)
X  +LA6(IG,1)-P1(IG,1)+P2(IG,1)-12.*P3(IG,1)+10.*P4(IG,1)
X  -3.*P5(IG,1)-Q1(IG,1)+6.*Q2(IG,1)-12.*Q3(IG,1)+10.*Q4(IG,1)
X  -3.*Q5(IG,1)+R0(IG,1)-8.*R1(IG,1)+22.*R2(IG,1)-24.*R3(IG,1)
X  +24.*R4(IG,1)}
X +9.*R4(IG,1)*V2*(HR(1)/D0(IG,1))**2

LT(1,5,IG)=(-3.*KA3(IG,1)+8.*LA4(IG,1)-7.*LA5(IG,1)+2.*LA6(IG,1))

X -1.*L A3(IG,1)+8.*LA4(IG,1)-7.*LA5(IG,1)+2.*LA6(IG,1))

X +3.*P2(IG,1)-14.*P3(IG,1)+17.*F4(IG,1)-6.*F5(IG,1)+(-6.*

X Q2(IG,1)+18.*Q3(IG,1)+18.*Q4(IG,1)+6.*Q5(IG,1)-(-6.*R1(IG,1)

X -30.*R2(IG,1)+42.*R3(IG,1)-18.*R4(IG,1)))*V3*HR(1)/DO(IG,1)

LT(1,6,IG)=(-3.*KA3(IG,1)+3.*KA4(IG,1)+3.*KA5(IG,1)+KA6(IG,1)

X (-L A3(IG,1)+3.*LA4(IG,1)+LA6(IG,1))+P2(IG,1)

X -5.*P4(IG,1)+7.*P5(IG,1)-(-2.*Q2(IG,1)-7.*Q3(IG,1)

X +8.*Q4(IG,1)+3.*Q5(IG,1)-2.*R1(IG,1)+11.*R2(IG,1)-18.*R3(IG,1)

X +9.*R4(IG,1)+V3*HR(1)*HR(1)/D0(IG,1)

FT(1,5,IG)=(KB2(IG,1)+4.*KB3(IG,1)+6.*KB4(IG,1)+4.*KB5(IG,1)

X +KB6(IG,1))*V2*(HR(1)/D0(IG,1))**2

FT(1,6,IG)=(-3.*KB3(IG,1)+8.*KB4(IG,1)-7.*KB5(IG,1)+2.*KB6(IG,1)

X *V3*HR(1)*HR(1)/D0(IG,1)

IF (IG.EQ.2) GO TO 61

T(1,4)=(SR2(IG,1)+4.*SR3(IG,1)+6.*SR4(IG,1)-4.*SR5(IG,1)

X +SR6(IG,1))+HR(1)**2/(P07(2,1)*PC(1,1)*DO(2,1)*DO(1,1))

T(1,5)=(-3.*SR3(IG,1)+8.*SR4(IG,1)+7.*SR5(IG,1)+2.*SR6(IG,1)

X *HR(1)/(P07(2,1)*PH(1,1)*DO(2,1))

T(1,6)=(-SR3(IG,1)+3.*SR4(IG,1)+3.*SR5(IG,1)+SR6(IG,1)

X *HR(1)**2/(P07(2,1)*PH(1,1)*DO(2,1)*DH(1,1))

FT(1,4,3)=(KD2(IG,1)+4.*KD3(IG,1)+6.*KD4(IG,1)-4.*KD5(IG,1)

X +KD6(IG,1)))+HR(1)**2/(P07(2,1)*PC(1,1)*DO(2,1)*DO(1,1))

FT(1,5,3)=(-3.*KD3(IG,1)+8.*KD4(IG,1)-7.*KD5(IG,1)+2.*KD6(IG,1)

X *HR(1)/(P07(2,1)*PH(1,1)*DO(2,1))

FT(1,6,3)=(-KD3(IG,1)+3.*KD4(IG,1)+3.*KD5(IG,1)+KD6(IG,1)

X *HR(1)**2/(P07(2,1)*PH(1,1)*DO(2,1)*DH(1,1))

FT(1,4,4)=(KC2(IG,1)+4.*KC3(IG,1)+6.*KC4(IG,1)-4.*KC5(IG,1)

X +KC6(IG,1))+HR(1)**2/(P07(2,1)*PC(1,1)*DO(2,1)*DO(1,1))

FT(1,5,4)=(-3.*KC3(IG,1)+8.*KC4(IG,1)-7.*KC5(IG,1)+2.*KC6(IG,1)

X *HR(1)/(P07(1,1)*PH(2,1)*DO(1,1))

FT(1,6,4)=(-KC3(IG,1)+3.*KC4(IG,1)-3.*KC5(IG,1)+KC6(IG,1)

X *HR(1)**2/(P07(1,1)*PH(2,1)*DO(1,1))
61 CONTINUE
   GO TO 65

C  ZERO CURRENT COEFFICIENTS ON THE LEFT:

63  K=1
   I=1
   DO 64  IG=1,2
   V2=1.*((PO7(1G,K))*PO(1G,K))
   V3=1.*((PO7(1G,K))*PH(1G,K))
   LT(I,4,IG)=
      R3(1G,K)+36.*R4(1G,K))*V2 -CO(1G)/PC(1G,1)
   LT(I,6,IG)=
      Q5(1G,K)-12.*R2(1G,K)+36.*R4(1G,K))*V3*HR(K)/DH(1G,1)
   IF (IG.EQ.2) GO TO 64

T1(1,4)=
   +(KB0(1G,K)-6.*KB2(1G,K)+4.*KB3(1G,K)+9.*KB4(1G,K)
   -12.*KB5(1G,K)+4.*KB6(1G,K))*V2
   FT(I,5,IG)=(3.*KB2(1G,K)-2.*KB3(1G,K)-9.*KB4(1G,K)+12.*KB5(1G,K)
      -4.*KB6(1G,K))*V3
   FT(I,6,IG)=
      (3.*KB2(1G,K)-2.*KB3(1G,K)-9.*KB4(1G,K)+5.*KB5(1G,K)
      -4.*KB6(1G,K))*V3*HR(K)/DH(1G,1)
   IF (IG.EQ.2) GO TO 64

T1(1,4)=
   +(SR0(1G,K)-6.*SR2(1G,K)+4.*SR3(1G,K)+9.*SR4(1G,K)
   -12.*SR5(1G,K)+4.*SR6(1G,K))/(PO7(2,K)*PO(1,1,K))
\[ T(I,5) = (3 \cdot SR2(IG,K) - 2 \cdot SR3(IG,K) - 9 \cdot SR4(IG,K) + 12 \cdot SR5(IG,K) \]
\[ X = -4 \cdot SR6(IG,K) / (PO7(2,K) \cdot PH(1,K)) \]
\[ T(I,6) = (SR2(IG,K) - SR3(IG,K) - 3 \cdot SR4(IG,K) + 5 \cdot SR5(IG,K) \]
\[ X = -2 \cdot SR6(IG,K) / (PO7(2,K) \cdot PH(1,K) \cdot DH(1,K)) \]
\[ FT(I,4, 3) = (KU(IGK) - 6 \cdot K02(IG,K) + 4 \cdot K03(IG,K) + 9 \cdot KD4(IGK) \]
\[ X = + (KU(IGK) - 6 \cdot K02(IG,K) + 4 \cdot K03(IG,K) + 9 \cdot KD4(IGK) \]
\[ \]
\[ V_l = 1/(P_{17}(IG,J)*P_{18}(IG,J)) \]

\[ L_l(1,1,IG) = (K_{A2}(IG,J)-K_{A3}(IG,J)-K_{A4}(IG,J)+5*K_{A5}(IG,J)-2*) \]

\[ X \quad K_{A6}(IG,J)+L_{A2}(IG,J)-L_{A3}(IG,J)-L_{A4}(IG,J)+5*L_{A5}(IG,J)-2* \]

\[ X \quad L_{A6}(IG,J)-2*P_{11}(IG,J)+3*P_{2}(IG,J)+6*P_{3}(IG,J)-13*P_{4}(IG,J)+6* \]

\[ X \quad P_{5}(IG,J)-6*Q_{1}(IG,J)-12*Q_{4}(IG,J)+6*Q_{5}(IG,J)-12*R_{2}(IG,J) \]

\[ +30*R_{3}(IG,J)-18*R_{4}(IG,J)\]*V*HR(J)/DH(IG,J) \]

\[ L_l(1,2,IG) = (-K_{A3}(IG,J)+3*K_{A4}(IG,J)-3*K_{A5}(IG,J)+K_{A6}(IG,J) \]

\[ X \quad -L_{A3}(IG,J)+3*L_{A4}(IG,J)-3*L_{A5}(IG,J)+L_{A6}(IG,J)+2*P_{2}(IG,J) \]

\[ X \quad -7*P_{3}(IG,J)+8*P_{4}(IG,J)-3*P_{5}(IG,J)-Q_{2}(IG,J)-5*Q_{3}(IG,J) \]

\[ X \quad +7*Q_{4}(IG,J)-3*Q_{5}(IG,J)-2*R_{1}(IG,J)+11*R_{2}(IG,J)-18* \]

\[ X \quad R_{3}(IG,J)+9*R_{4}(IG,J)\]*V*HR(J)/HR(J)/(P_{0}(1,J)*P_{0}(1,J)) \]

\[ L_l(1,3,IG) = (K_{A4}(IG,J)-K_{A5}(IG,J)+K_{A6}(IG,J)-L_{A4}(IG,J)-2* \]

\[ X \quad L_{A5}(IG,J)+L_{A6}(IG,J)-2*P_{3}(IG,J)+5*P_{4}(IG,J)-3*P_{5}(IG,J)-(-2* \]

\[ X \quad Q_{3}(IG,J)+5*Q_{4}(IG,J)-3*Q_{5}(IG,J)+4*R_{2}(IG,J)-12*R_{3}(IG,J)+9* \]

\[ X \quad R_{4}(IG,J)\]*V*HR(J)/DH(IG,J) \]

\[ F_{l}(1,1,IG) = (K_{B2}(IG,J)-K_{B3}(IG,J)-3*KB_{4}(IG,J)+5*KB_{5}(IG,J) \]

\[ X \quad -2*K_{B6}(IG,J)\)*V*HR(J)/DH(IG,J) \]

\[ X \quad F_{l}(1,2,IG) = (-K_{B3}(IG,J)+3*K_{B4}(IG,J)-3*KB_{5}(IG,J)+KB_{6}(IG,J) \]

\[ X \quad +V*HR(J)/HR(J)/(P_{0}(1,J)*P_{0}(1,J)) \]

\[ X \quad F_{l}(1,3,IG) = (K_{B4}(IG,J)-2*KB_{5}(IG,J)+KB_{6}(IG,J) \]

\[ X \quad +V*HR(J)/DH(IG,J) \]

\[ X \quad *V*HR(J)/DH(IG,J)\]
\[ FT(1,2,4) = (-KC3(IG,J) + 3 \cdot KC4(IG,J) - 3 \cdot KC5(IG,J) + KC6(IG,J)) \]
\[ \times \frac{HR(J)^2}{PH7(1,J) \cdot PJ(2,J) \cdot DH(1,J) \cdot D0(2,J)} \]
\[ FT(13,4) = (-KC4(IG,J) - 2 \cdot KC5(IG,J) + KC6(IG,J)) \]
\[ \times \frac{HR(J)^2}{PH7(1,J) \cdot PJ(2,J) \cdot DH(1,J) \cdot DH(2,J)} \]

72 CONTINUE
C
FIX UP THE LAST TWO COLUMNS TO MATCH G(K+1) (NOT F(K+1)):
I1 = 2*KR - 2
I2 = I1 + 1
DO 73 IG = 1, 2
    DO 73 IG = 1, 12
    LT(1, 6, IG) = LT(1, 6, IG)
    FT(1, 5, IG) = FT(1, 6, IG)
    IF (IG.EQ.2) GO TO 73
    T(1, 5) = T(1, 6)
    FT(1, 5, 3) = FT(1, 6, 3)
    FT(1, 5, 4) = FT(1, 6, 4)
73 CONTINUE
GO TO 80
C
ZERO CURRENT COEFFICIENTS ON THE RIGHT:
74 DO 62 IG = 1, 2
    V = 1./(PH7(IG,K) \cdot PJ(IG,K))
    V1 = 1./(PH7(IG, K) \cdot PJ(IG, K))
    LT(1, 1, IG) = (3 \cdot KA2(IG, K) - 2 \cdot KA3(IG, K) - 9 \cdot KA4(IG, K) + 12 \cdot KA5(IG, K)) \cdot V
    X - 4 \cdot KA6(IG, K) - (3 \cdot LA2(IG, K) - 2 \cdot LA3(IG, K) - 9 \cdot LA4(IG, K) + 12 \cdot LA5(IG, K)) \times V
    X L5(IG, K) - 4 \cdot LA6(IG, K) - (6 \cdot PL1(IG, K) - 6 \cdot P2(IG, K) - 18 \cdot P3(IG, K)) \times V
    X + 30 \cdot P4(IG, K) - 12 \cdot P5(IG, K) - (18 \cdot Q3(IG, K) - 30 \cdot Q4(IG, K) + 12 \cdot Q5(IG, K)) \times V
    X Q5(IG, K) - (36 \cdot R2(IG, K) - 72 \cdot R3(IG, K) + 36 \cdot R4(IG, K)) \times V  
    LT(1, 2, IG) = (13 \cdot KA3(IG, K) + 8 \cdot KA4(IG, K) - 7 \cdot KA5(IG, K) + 2 \cdot KA6(IG, K)) \times V
    X - (3 \cdot LA3(IG, K) + 8 \cdot LA4(IG, K) - 7 \cdot LA5(IG, K) + 2 \cdot LA6(IG, K)) \times V
    X P21(IG, K) + 18 \cdot P31(IG, K) - 18 \cdot P41(IG, K) + 6 \cdot P51(IG, K) - (3 \cdot Q21(IG, K)) \times V
    X - 14 \cdot Q31(IG, K) + 17 \cdot Q41(IG, K) - 6 \cdot Q51(IG, K) - (6 \cdot R1(IG, K) - 30 \cdot R2(IG, K)) \times V
    X R21(IG, K) + 42 \cdot R31(IG, K) - 18 \cdot R41(IG, K)) \times V \times HR(K) / DO1(IG, K)
    LT(1, 3, IG) = (19 \cdot KA4(IG, K) - 12 \cdot KA5(IG, K) + 4 \cdot KA6(IG, K) - 9 \cdot LA4(IG, K)) \times V
    X - 12 \cdot LA5(IG, K) + 4 \cdot LA6(IG, K) - (18 \cdot P31(IG, K) - 30 \cdot P41(IG, K) + 12 \cdot P51(IG, K)) \times V
    X P51(IG, K) + (18 \cdot Q31(IG, K) - 30 \cdot Q41(IG, K) + 12 \cdot Q51(IG, K)) \times V
    X + 36 \cdot R21(IG, K) - 72 \cdot R31(IG, K) + 36 \cdot R41(IG, K)) \times V \times CH(IG) / PH1(IG, KR)
FT(I,1,IG)=(3.*KB2(IG,K)-2.*KB3(IG,K)-9.*KB4(IG,K)+12.*KB5(IG,K)) *V
X   -4.*KB6(IG,K))*V
FT(I,2,IG)=(-3.*KB3(IG,K)+8.*KB4(IG,K)-7.*KB5(IG,K)+2.*KB6(IG,K)) *V*HR(K)/0U(IGK)
FT(1,3,IGK)=(9.*KB4(IGK)-12.*KB5(IGK)+4.*KB6(IGK))*V*HR(K)/0U(IGK)
IF (IG.EQ.2) GO TO 62
T(I,1)=(-3.*SR2(IG,K)-2.*SR3(IG,K)-9.*SR4(IG,K)+12.*SR5(IG,K)
X   -4.*SR6(IG,K)))/(PH7(2,K)*PO(1,K)
T(I,2)=(-3.*SR3(IG,K)+8.*SR4(IG,K)-7.*SR5(IG,K)+2.*SR6(IG,K))
X   *HR(K)/(PH7(2,K)*PO(1,K)*DO(1,K)
T(I,3)=(9.*SR4(IG,K)-12.*SR5(IG,K)+4.*SR6(IG,K))
X   (PH7(2,K)*PH(1,K)
FT(1,1, 3)=(3.*KD2(IG,K)-2.*KD3(IG,K)-9.*KD4(IG,K)+12.*KD5(IG,K)
X   -4.*KD6(IG,K))/PH7(2,K)*PO(1,K)
FT(I,2, 3)=(-3.*KD3(IG,K)+8.*KD4(IG,K)-7.*KD5(IG,K)+2.*KD6(IG,K))
X   *HR(K)/(PH7(2,K)*PO(1,K)*DO(1,K)
FT(1,3, 3)=(9.*KD4(IG,K)-12.*KD5(IG,K)+4.*KD6(IG,K))
X   (PH7(2,K)*PH(1,K)
FT(I,1, 4)=(3.*KC2(IG,K)-2.*KC3(IG,K)-9.*KC4(IG,K)+12.*KC5(IG,K)
X   -4.*KC6(IG,K))/(PH7(2,K)*PO(2,K)
FT(I,2, 4)=(-3.*KC3(IG,K)+8.*KC4(IG,K)-7.*KC5(IG,K)+2.*KC6(IG,K))
X   *HR(K)/(PH7(1,K)*PO(2,K)*DO(2,K)
FT(1,3, 4)=(9.*KC4(IG,K)-12.*KC5(IG,K)+4.*KC6(IG,K))
X   (PH7(1,K)*PH(2,K)
62 CONTINUE
80 CONTINUE
C INCLUDE THETA AND PHI (PHI = -1) IN THE MATRIX FORMATIONS:
C THE ABOVE EQUATIONS ARE DERIVED USING PHI = -1.
C
PHIPHI=+1.000
IF (THETA.NE.1.0) CALL MATFIX(THETA,PHIPHI)
C
IF (ISEE.GE.2) CALL PRTOUT(2)
RETURN
END
SUBROUTINE ERRDR(I,J)
C ANNOUNCES INPUT ERRORS AND TERMINATES PROGRAM EXECUTION:
GO TO (1,2,3,4,5,6,7,8,9),1
1 WRITE (6,101)
GO TO 10
2 WRITE (6,102) J
GO TO 10
3 WRITE (6,103) J
GO TO 10
4 WRITE (6,104) J
GO TO 10
5 WRITE (6,105) J
GO TO 10
6 WRITE (6,106) J
GO TO 10
7 CONTINUE
8 CONTINUE
9 CONTINUE
10 WRITE (6,110)
101 FORMAT ('IMUST HAVE > 2 SUBREGIONS FOR ZERO FLUX B.C.S. INVALID.')
102 FORMAT ('INNUMBER OF SUBREGIONS =',I3,' > 25. INVALID.')
103 FORMAT ('ISUBREGION NUMBER',I3,' HAS > 100 SECTIONS. INVALID.')
104 FORMAT ('INPUT ERROR IN REGION SEQUENCING AT REGION',I5,'.')
105 FORMAT ('IREGION I =',I3,'. INVALID.')
106 FORMAT ('THE TOLERANCE: EPS',I1,' IS < 1.0E-16. INVALID.')
107 FORMAT ('BOUNDARY CONDITION OPTION =',I2,' < 1 OR > 4. INVALID.')
110 FORMAT (IHO,'PROBLEM TERMINATED.
CALL EXIT
RETURN
END

PAGE 310
SUBROUTINE REPEAT(K, L)
C SETS THE /85/ ARRAYS (K) EQUAL TO PAST STORED ARRAYS (L):
IMPLICIT REAL*8 (A-Z)
COMMON /85/
X KA(1,2,25), KA1(2,25), KA2(2,25), KA3(2,25), KA4(2,25), KA5(2,25),
X KA6(2,25), KB0(2,25), KB1(2,25), KB2(2,25), KB3(2,25), KB4(2,25),
X P3(2,25), P4(2,25), P5(2,25), P6(2,25), Q0(2,25), Q1(2,25),
X Q2(2,25), Q3(2,25), Q4(2,25), Q5(2,25), Q6(2,25), R0(2,25),
X R1(2,25), R2(2,25), R3(2,25), R4(2,25), SR0(1,25), SR1(1,25),
X SR2(1,25), SR3(1,25), SR4(1,25), SR5(1,25), SR6(1,25), KC0(1,25),
X KC1(1,25), KC2(1,25), KC3(1,25), KC4(1,25), KC5(1,25), KC6(1,25),
X KD0(1,25), KD1(1,25), KD2(1,25), KD3(1,25), KD4(1,25), KD5(1,25),
X KD6(1,25),
X P0(2,25), PH(2,25), P07(2,25), PH7(2,25), D0(2,25), DH(2,25)
COMMON /XAXIS/ HX, HR(25)
INTEGER K, L, G
HR(K)=HR(L)
HX=HX+HR(K)
DO 10 G=1, 2
KA0(G,K)=KA0(G,L)
KA1(G,K)=KA1(G,L)
KA2(G,K)=KA2(G,L)
KA3(G,K)=KA3(G,L)
KA4(G,K)=KA4(G,L)
KA5(G,K)=KA5(G,L)
KA6(G,K)=KA6(G,L)
KB0(G,K)=KB0(G,L)
KB1(G,K)=KB1(G,L)
KB2(G,K)=KB2(G,L)
KB3(G,K)=KB3(G,L)
KB4(G,K)=KB4(G,L)
KB5(G,K)=KB5(G,L)
KB6(G,K)=KB6(G,L)
LA0(G,K)=LA0(G,L)
IF (G.EQ.2) GO TO 5
KC3(G,K) = KC3(G, L)
KC4(G,K) = KC4(G, L)
KC5(G,K) = KC5(G, L)
KC6(G,K) = KC6(G, L)
KD0(G,K) = KD0(G, L)
KD1(G,K) = KD1(G, L)
KD2(G,K) = KD2(G, L)
KD3(G,K) = KD3(G, L)
KD4(G,K) = KD4(G, L)
KD5(G,K) = KD5(G, L)
KD6(G,K) = KD6(G, L)
CONTINUE
PO(G,K) = PO(G, L)
PO7(G,K) = PO7(G, L)
PH(G,K) = PH(G, L)
PH7(G,K) = PH7(G, L)
DO(G,K) = DO(G, L)
DH(G,K) = DH(G, L)
CONTINUE
RETURN
END
SUBROUTINE BHSET(K)
C
SETS UP THE /BH/ ARRAYS FOR GIF:
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /BH/ X(101), H(101), Z(101)
DO 1 I=1,K
1 Z(I)=X(I)-X(1)
RETURN
END
DOUBLE PRECISION FUNCTION GIF(N,G1,F,G2,C,G,K,ITC)

* IDENTICAL TO SUBROUTINE GIF PREVIOUSLY LISTED IN PROGRAM LINEAR.

RETURN
END
DOUBLE PRECISION FUNCTION FACT(N)
C COMPUTES N FACTORIAL:
FACT=1.00D0
IF (N .LE. 1) RETURN
DO 1 I=2,N
  1 FACT=FACT*DFLOAT(I)
RETURN
END
SUBROUTINE MATFIX(THETA, PHI)
C MODIFYES THE MATRIX ELEMENTS OF THE /B3/ MATRICES BY THETA AND PHI.
C PROPER CHOICE OF THETA PROVIDES EASIER INVERSION OF THE MATRICES.
C MATRIX SOLUTION SHOULD BE INDEPENDENT OF PHI. HOWEVER:
C USE OF PHI > 0 RESULTS IN POSITIVE DEFINITE MATRICES.
COMMON /B1/ IBC
COMMON /B2/ KR, NN
COMMON /B3/ A(50, 6, 7)
REAL*8 A, THEATA, PHI, X, Y, Z
NO = NN - 1
DO 10 L = 1, 7
DO 5 I = 2, NO
X = THEATA
Y = 1.000
IF (MOD(I, 2) .EQ. 0) GO TO 1
X = THEATA*PHI
Y = PHI
1 DO 2 M = 2, 6, 2
2 A(I, M, L) = X*A(I, M, L)
DO 3 M = 1, 5, 2
3 A(I, M, L) = Y*A(I, M, L)
5 CONTINUE
C BOUNDARY CONDITION EQUATIONS:
X = 1.000
Y = 1.000
Z = THEATA
IF (IBC .GT. 2) GO TO 6
X = THEATA*PHI
Y = PHI
Z = X
6 A(I, 4, L) = X*A(I, 4, L)
A(I, 5, L) = Y*A(I, 5, L)
A(I, 6, L) = Z*A(I, 6, L)
X = 1.000
Y = THEATA
Z = X
IF (IBC.EQ.2 .OR. IBC.EQ.4) GO TO 7
X = PHI
Y = THEATA*PHI
Z = Y
7 A(NN,1,L) = X*A(NN,1,L)
   A(NN,2,L) = Y*A(NN,2,L)
   A(NN,3,L) = Z*A(NN,3,L)
10 CONTINUE
RETURN
END
SUBROUTINE PRTOUT(IP)

IP = 1 PRINTS OUT THE /B5/ ARRAYS USED IN MATRIX FORMATIONS.
C
IP = 2 PRINTS OUT THE /B3/ Matrices Given to Power.
C
IMPLICIT REAL*8 (A-H,K-Z)
COMMON /B2/ KR, N
COMMON /B3/ L1(50,6), L2(50,6), F1(50,6), F4(50,6), F3(50,6),
X F2(50,6), T(50,6)
COMMON /B5/
X KA0(2,25), KA1(2,25), KA2(2,25), KA3(2,25), KA4(2,25), KA5(2,25),
X KA6(2,25), KB0(2,25), KB1(2,25), KB2(2,25), KB3(2,25), KB4(2,25),
X P3(2,25), P(2,25), P5(2,25), P6(2,25), Q0(2,25), Q1(2,25),
X Q2(2,25), Q3(2,25), Q4(2,25), Q5(2,25), Q6(2,25), RO(2,25),
X R(2,25), R2(2,25), R3(2,25), R4(2,25), SR0(1,25), SR1(1,25),
X SR2(1,25), SR3(1,25), SR4(1,25), SR5(1,25), SR6(1,25), KC0(1,25),
X KC1(1,25), KC2(1,25), KC3(1,25), KC4(1,25), KC5(1,25), KC6(1,25),
X KD0(1,25), KD1(1,25), KDA1(1,25), KD2(1,25), KD3(1,25), KD4(1,25),
X KD5(1,25),
COMMON /XAXIS/ HX, HR(25)
INTEGER KR, G, N, K
GO TO (1001,1002), IP
C
KA'S:
1001 WRITE (6,10)
DO 11 G=1,2
11 WRITE (6,100) (G,K,KA0(G,K),KA1(G,K),KA2(G,K),KA3(G,K),KA4(G,K),
X KA5(G,K),KA6(G,K),K=1,KR)
C KB'S:
WRITE (6,20)
DO 21 G=1,2
21 WRITE (6,100) (G,K,KB0(G,K),KB1(G,K),KB2(G,K),KB3(G,K),KB4(G,K),
X KB5(G,K),KB6(G,K),K=1,KR)
C KC'S:
WRITE (6,22)
G=1
PAGE 319
23 WRITE (6,100) (G,K,KC0(G,K),KC1(G,K),KC2(G,K),KC3(G,K),KC4(G,K),
    X KC5(G,K),KC6(G,K),K=1,KR)
C KD'S:
    WRITE (6,24)
    G=1
25 WRITE (6,100) (G,K,KD0(G,K),KD1(G,K),KD2(G,K),KD3(G,K),KD4(G,K),
    X KD5(G,K),KD6(G,K),K=1,KR)
C LA'S:
    WRITE (6,30)
    DO 31 G=1,2
31 WRITE (6,100) (G,K,LA0(G,K),LA1(G,K),LA2(G,K),LA3(G,K),LA4(G,K),
    X LA5(G,K),LA6(G,K),K=1,KR)
C SR'S:
    WRITE (6,40)
    G=1
41 WRITE (6,100) (G,K,SR0(G,K),SR1(G,K),SR2(G,K),SR3(G,K),SR4(G,K),
    X SR5(G,K),SR6(G,K),K=1,KR)
C P'S:
    WRITE (6,50)
    DO 51 G=1,2
51 WRITE (6,100) (G,K,PO(G,K),P1(G,K),P2(G,K),P3(G,K),P4(G,K),
    X P5(G,K),P6(G,K),K=1,KR)
C Q'S:
    WRITE (6,60)
    DO 61 G=1,2
61 WRITE (6,100) (G,K,Q0(G,K),Q1(G,K),Q2(G,K),Q3(G,K),Q4(G,K),
    X Q5(G,K),Q6(G,K),K=1,KR)
C R'S:
    WRITE (6,70)
    DO 71 G=1,2
71 WRITE (6,100) (G,K,RO(G,K),R1(G,K),R2(G,K),R3(G,K),R4(G,K),K=1,KR)
C BOUNDARY VALUES:
    WRITE (6,80)
    DO 81 G=1,2
81 WRITE (6,100) (G,K,PO(G,K),PH(G,K),PO7(G,K),PH7(G,K),DO(G,K),
    X DH(G,K),HR(K),K=1,KR)
RETURN
1002 WRITE (6,90)
WRITE (6,110) ((L(I,J), J=1,6), I=1,N)
WRITE (6,91)
WRITE (6,110) ((L2(I,J), J=1,6), I=1,N)
WRITE (6,92)
WRITE (6,110) ((F1(I,J), J=1,6), I=1,N)
WRITE (6,93)
WRITE (6,110) ((F2(I,J), J=1,6), I=1,N)
WRITE (6,94)
WRITE (6,110) ((F3(I,J), J=1,6), I=1,N)
WRITE (6,95)
WRITE (6,110) ((F4(I,J), J=1,6), I=1,N)
WRITE (6,96)
WRITE (6,110) ((T(I,J), J=1,6), I=1,N)
10 FORMAT ('1 G',4X,'K',7X,'KA0(G,K)',7X,'KA1(G,K)',7X,'KA2(G,K)',
X 7X,'KA3(G,K)',7X,'KA4(G,K)',7X,'KA5(G,K)',7X,'KA6(G,K)',/)
20 FORMAT ('1 G',4X,'K',7X,'KB0(G,K)',7X,'KB1(G,K)',7X,'KB2(G,K)',
X 7X,'KB3(G,K)',7X,'KB4(G,K)',7X,'KB5(G,K)',7X,'KB6(G,K)',/)
22 FORMAT ('1 G',4X,'K',7X,'KC0(G,K)',7X,'KC1(G,K)',7X,'KC2(G,K)',
X 7X,'KC3(G,K)',7X,'KC4(G,K)',7X,'KC5(G,K)',7X,'KC6(G,K)',/)
24 FORMAT ('1 G',4X,'K',7X,'KD0(G,K)',7X,'KD1(G,K)',7X,'KD2(G,K)',
X 7X,'KD3(G,K)',7X,'KD4(G,K)',7X,'KD5(G,K)',7X,'KD6(G,K)',/)
30 FORMAT ('1 G',4X,'K',7X,'LA0(G,K)',7X,'LA1(G,K)',7X,'LA2(G,K)',
X 7X,'LA3(G,K)',7X,'LA4(G,K)',7X,'LA5(G,K)',7X,'LA6(G,K)',/)
40 FORMAT ('1 G',4X,'K',7X,'SR0(G,K)',7X,'SR1(G,K)',7X,'SR2(G,K)',
X 7X,'SR3(G,K)',7X,'SR4(G,K)',7X,'SR5(G,K)',7X,'SR6(G,K)',/)
50 FORMAT ('1 G',4X,'K',8X,'P0(G,K)',8X,'P1(G,K)',8X,'P2(G,K)',8X,
X 8X,'P3(G,K)',8X,'P4(G,K)',8X,'P5(G,K)',8X,'P6(G,K)',/)
60 FORMAT ('1 G',4X,'K',8X,'Q0(G,K)',8X,'Q1(G,K)',8X,'Q2(G,K)',8X,
X 8X,'Q3(G,K)',8X,'Q4(G,K)',8X,'Q5(G,K)',8X,'Q6(G,K)',/)
70 FORMAT ('1 G',4X,'K',8X,'R0(G,K)',8X,'R1(G,K)',8X,'R2(G,K)',8X,
X 8X,'R3(G,K)',8X,'R4(G,K)',/)
80 FORMAT ('1 G',4X,'K',8X,'P0(G,K)',8X,'PH(G,K)',8X,'PH7(G,K)',7X,
X 8X,'D0(G,K)',8X,'DH(G,K)',10X,'HR(K)',/)
100 FORMAT (215,7DL=7)
101 FORMAT (215,5D15.7)
90 FORMAT (*1MATRIX L1:*/,
91 FORMAT (*1MATRIX L2:*/,
92 FORMAT (*1MATRIX F1:*/,
93 FORMAT (*1MATRIX F2:*/,
94 FORMAT (*1MATRIX F3:*/,
95 FORMAT (*1MATRIX F4:*/,
96 FORMAT (*1MATRIX T:*/,
110 FORMAT (6D20.10)
   RETURN
   END
SUBROUTINE POWER
C SOLVES THE 2*N MULTIGROUP EQUATIONS: M*PHI = (1/LAMDA)*F*PHI
C BY THE FISSION SOURCE POWER METHOD
C USING SIMULTANEOUS OVERRELAXATION.
C WHERE: M AND F ARE DOUBLE PRECISION 2N BY 2N BLOCK MATRICES;
C AND: PHI IS THE 2N FLUX (FAST AND THERMAL) VECTOR.
C L1*PHI1 = CHI1*(F1*PHI1 + F2*PHI2)
C -T*PHI1 + L2*PHI2 = CHI2*(F3*PHI1 + F4*PHI2)
C METHOD Follows WACHPRESS, PAGE 83. SOLUTION BY GROUP ITERATION.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /81/ IBCLIPLOT,JPLOT,IPUNCH,ISEE
COMMON /82/ KR,N
COMMON /83/ L1(50,6),L2(5J,6),F1(50,6),F4(50,6),F3(50,6),F2(50,6),
T(50,6)
COMMON /84/ PHI(2,52), PSI(2,52), LAMDA, ICOUT
COMMON /85/ S(52), ERROR(2,52), Z(52), G(50,6), GT(50)
COMMON /86/ TEL(2,5),TE2(2,5),TE3(5),IN(5)
COMMON /C!IF/ CHI(2)
COMMON /XAXIS/ HX,HR(25)
COMMON /ER/ EPS1,EPS2,EPS3
COMMON /FSTR/ PHISTR(2,26,6)
COMMON /ESTR/ LAMSTR(300), EFSTR(2,300), EFMSTR(2,300), ERLAM(300)
COMMON /TRJE/ TRULAM, TRPHI(2,52), PHICON(2,300), LAMCON(300),IFT
DIMENSION PSI1(52), PSI2(52), SQ(2), DPHI(2), ERRMAX(2)
INTEGER N, NN
C DEFAULT OPTIONS FOR THE TRUE EIGENVALUE AND FLUX-CURRENT VECTOR:
TRULAM=1.0
DO 5 IG=1,2
DO 4 I=1,N
TRPHI(IG,I)=1.0
IF (MOD(I,2).EQ.1) TRPHI(IG,I)=0.0
4 CONTINUE
IF (IBC.EQ.1) TRPHI(IG,N)=0.0
IF (IBC.EQ.4) TRPHI(IG,1)=1.0
5 CONTINUE
C DEFAULT OPTIONS FOR POWER PARAMETERS:
ALPHA=1.25
LAMDA=1.0
DO 555 IG=1,2
   IF (IBC.NE.4) GO TO 551
   DO 550 I=1,N
      PHI(IG,I)=0.0
      IF (MOD(1,2).EQ.1) PHI(IG,I)=0.0
   CONTINUE
   PHI(IG,1)=1.0
   GO TO 555
550 CONTINUE
   PHI(IG,1)=1.0
   GO TO 555
551 X=3.1415926/HX
   IF (IBC.NE.1) X=X/2.0
   PHI(IG,1)=-X
   SUM1=0.0
   DO 552 K=2,KR
      I=2*K-2
      J=I+1
      SUM1=SUM1+HR(K-1)
      PHI(IG,I)=DSIN(SUM1*X)
   CONTINUE
   PHI(IG,J)=-X*DCOS(SUM1*X)
   PHI(IG,N)=1.0
   IF (IBC.EQ.1) PHI(IG,N)=X
555 CONTINUE
C    READ IN THE TRUE (EXPECTED) EIGENVALUE AND FLUX VECTOR (MINUS 0 BC'S):
   IFT=0
   READ (5,500,END=501) TRULAM, (TRUPHI(1,I),I=1,N)
   READ (5,503,END=501) (TRUPHI(2,I),I=1,N)
   IFT=1
500 FORMAT (E25.14,/,(4E20.10))
C    READ IN: OVERRELAXATION PARAMETERS ; ALPHA (OUTER ITERATION)
C    INITIAL GUESS AT EIGENVALUE; LAMDA
C    INITIAL NORMALIZED FLUX ; PHI(1-N)
   CHANGE (5,506,END=510) ALPHA
   READ (5,502,END=510) LAMDA
   READ (5,503) (PHI(1,I),I=1,N)
   READ (5,503) (PHI(2,I),I=1,N)
506 FORMAT (F10.5)
502 FORMAT (E25.14)
503 FORMAT ((4E20.10))
510 CONTINUE

C STORING FOR PRINTING THE MULTIGROUP FLUX SHAPE.

K=0
IF (IBC.EQ.4) K=1
DO 11 IG=1,2
  DO 10 I=1,KR
  II=I+K
  10 PHISTR(IG,II,2)=PHI(IG,2*I)
  IF (IBC.EQ.4) PHISTR(IG,II,2)=PHI(IG,1)
C FILL RUNNING COORD IN PHISTR
  KR1=KR+1
  DO 11 I=1,KR1
  11 PHISTR(IG,1,1)=DFLOAT(I)
C IK IS THE FLUX PLOTTING COUNTER.
  IK=1
C STORES THE ITERATION NUMBER FOR FLUX HISTORY PLOTTING:
  IN(1)=0
C STORES TEMPORARY ERRORS FOR FLUX HISTORY PLOTTING:
  TE1(1,1)=0.
  TE1(2,1)=0.
  TE2(1,1)=0.
  TE2(2,1)=0.
  TE3(1)=0.0
C EIGENVALUE OF THE PREVIOUS ITERATION:
  LAMBDA=LAMDA
C THE MAXIMUM NUMBER OF ALLOWED ITERATIONS: ICMAX
  ICMAX=300
C PRINT OUT THE POWER METHOD PARAMETER INFORMATION:
  WRITE (6,700) ICMAX,ALPHA,LAMDA,(PHI(1,I),I=1,N)
  WRITE (6,701) (PHI(2,I),I=1,N)
700 FORMAT ('EXECUTING MULTIGROUP FISSION SOURCE POWER ITERATION METH
XOD.',///,
X 5X,'MAXIMUM NUMBER OF ALLOWABLE ITERATIONS:',//,
BEGIN ITERATION LOOP.

ICOUT = 0
C ICOUT IS THE OUTER ITERATION COUNTER.

20 ICOUT = ICOUT + 1
  IF (ICOUT.GT.ICMAX) GO TO 100
C SOLVE FOR THE NEW GROUP FLUX VECTORS: PSI:
C THE GROUP FLUX ITERATES:

DO 21 I = 1, N
  PSI1(I) = PHI(1, I)
21 PSI2(I) = PHI(2, I)
C THE FAST ITERATION SOURCE S:
CALL VPRDF1(5, PSI1, S, N)
CALL VPRDF2(5, PSI2, Z, N)
DO 22 I = 1, N
22 S(I) = CHI(1) * (S(I) + Z(I))
C FAST FLUX:
CALL SOLVE(L1, PSI1, S, N, G, GT)
C THE THERMAL ITERATION SOURCE S:
CALL VPRDF3(5, PSI1, S, N)
CALL VPRDF4(5, PSI2, Z, N)
DO 25 I = 1, N
25 S(I) = CHI(2) * (S(I) + Z(I))
CALL VPRDFT(L1, PSI1, S, N)
DO 27 I = 1, N
27 S(I) = S(I) + Z(I)
C THERMAL FLUX:
CALL SOLVE(L2,PSI2,S,N,G,GT)
C    CALCULATION OF THE EIGENVALUE:
SUM1=0.000
SUM2=0.000
DO 28 I=1,N
SUM1=SUM1+PSI1(I)*PHI1(I,I)
28 SUM2=SUM2+PSI1(I)*PSI1(I)
DO 29 I=1,N
SUM1=SUM1+PSI2(I)*PHI2(I,I)
29 SUM2=SUM2+PSI2(I)*PSI2(I)
LAMDA=SUM2/SUM1
LAMSTR(IGOUT)=LAMDA
ERRLAM=DABS(LAMDA-LAMBDA)
C    PUT PSI1 AND PSI2 INTO BIGGER PSI:
DO 30 I=1,N
PSI1(I)=PSI1(I)
30 PSI2(I)=PSI2(I)
C    POINT BY POINT SIMULTANEOUS RELAXATION FLUX ITERATION:
X=ALPHA
C    DO NOT RELAX DURING THE FIRST THREE ITERATIONS:
IF (ICOUT.LE.3) X=1.0
C    CALCULATE THE NEW GROUP FLUX ITERATES AND GROUP ERRORS:
DO 40 IG=1,2
DO 40 I=1,N
PSI(IG,I)=PHI(IG,I) + X*(PSI(IG,I)/LAMDA-PHI(IG,I))
40 CONTINUE
C    NORMALIZE THE FLUX ITERATE (ONE GROUP):
CALL NURMAL(PSI,N)
NN=N
IF (IBC.EQ.1) NN=N-2
C    NORMALIZED ERRORS OF THE FLUX CNLY:
DO 39 IG=1,2
ERRMAX(IG)=0.0
SQ(I)=0.0
IF (IBC.NE.4) GO TO 37
ERROR(IG,1)=DABS((PSI(IG,1)-PHI(IG,1))/PSI(IG,1) )
ERRMAX(IG) = ERROR(IG, 1)
SQ(IG) = ERROR(IG, 1)**2
37 DO 38 I=2, NN, 2
   ERROR(IG, I) = DABS((PSI(IG, I) - PHI(IG, I)) / PSI(IG, I))
   IF (ERROR(IG, I) .GT. ERRMAX(IG)) ERRMAX(IG) = ERROR(IG, I)
38 SQ(IG) = SQ(IG) + ERROR(IG, I)**2
   SQ(IG) = USQRT(SQ(IG))
C UPDATE THE FLUX ITERATE:
   DO 39 I=1, N
39 PHI(IG, I) = PSI(IG, I)
   IF (IFT.EQ.0) GO TO 31
   DLAM = LAMDA - TRULAM
   DO 36 IG=1, 2
      DPHI(IG) = 0.0
   DO 35 I=1, N
      DPHI(IG) = DPHI(IG) + (PSI(IG, I) - TRUPHI(IG, I))**2
   DO 35 I=1, N
   36 DPHI(IG) = USQRT(DPHI(IG))
   31 IF (IPLOT.NE.2) GO TO 45
C THE FOLLOWING IS FOR NICELY PLOTTING THE GROUP FLUX HISTORY.
   CALL NORM2(PSI, TRUPHI, N)
K=0
   IF (IBC.EQ.4) K=1
   KBC=KR
   IF (IBC.EQ.1) KBC=KR-1
   IF (IBC.EQ.4) KBC=KR+1
   DO 41 IG=1, 2
      DO 41 I=1, KBC
         II=I+K
   41 ERROR(IG, II) = PSI(IG, 2*I)
   IF (IBC.EQ.4) ERROR(1, 1) = PSI(1, 1)
   IF (IBC.EQ.4) ERROR(2, 1) = PSI(2, 1)
C ERROR NOW CONTAINS THE NEW NORMALIZED FLUX ITERATE PHI.
   JK=IK
   IF (IK.EQ.0) JK=5
   DO 42 IG=1, 2
      DO 42 I=1, KBC
IF (OABS(ERROR(IG,I)-PHISTR(IG,I,JKI)).GE.0.01) GO TO 43

CONTINUE

C  FLUX HAS NOT CHANGED ENOUGH FOR PLOTTING.
GOTO 45

C  SAVE THE NORMALIZED FLUX FOR PLOTTING:

IK=IK+1
IN(IK)=ICOUT
TE3(IK)=ERRLAM
DO 44 IG=1,2
TE1(IG,IK)=ERRMAX(IG)
TE2(IG,IK)=SQ(IG)
DO 44 I=1,KBC

PHISTR(IG,I,IK+1)=ERROR(IG,I)
IF (IK.NE.5) GO TO 45

C  PLOT THE LAST FIVE SAVED FLUXES:
CALL PHIPLT(5)

IK=0

CONTINUE

ERROR CRITERIA FOR ACCEPTANCE OF CONVERGENCE.
IFLAG1=0
IFLAG2=0
IFLAG3=0

C  STORE THE ERRORS FOR COMPARISON:
C  ERROR BETWEEN ITERATION EIGENVALUES:
ERLAM(ICOUT)=ERRLAM
DO 46 IG=1,2
C  MAXIMUM ERROR BETWEEN ITERATION FLUXES:
EFSTR(IG,ICOUT)=ERRMAX(IG)
C  MEAN SQUARE ERROR BETWEEN ITERATION FLUXES:
EFMSTR(IG,ICOUT)=SQ(IG)
C  MEAN SQUARE ERROR BETWEEN THE ITERATION FLUX AND GIVEN TRUE FLUX:
PHICON(IG,ICOUT)=DPHI(IG)
CONTINUE
C  ERROR BETWEEN THE ITERATION EIGENVALUE AND GIVEN TRUE EIGENVALUE:
LAMCON(ICOUT)=DLAM
IF ((ERRMAX(1).LT.EPS1).AND.(ERRMAX(2).LT.EPS1)) IFLAG1=1
IF (SQ(I).LT.EPS2).AND.(SQ(2).LT.EPS2)) IFLAG2=1
IF (ERRLAM.LT.EPS3) IFLAG3=1
IFLAG4=IFLAG1*IFLAG2*IFLAG3
IF (IFLAG4.EQ.1) GO TO 50
C OTHERWISE CONTINUE THE ITERATION.
LAMB4=LAMDA
GO TO 20
50 CONTINUE
C CONVERGENCE ACCOMPLISHED.
C NORMALIZE THE CONVERGED FLUX VECTOR:
CALL NORMAL(PHI,N)
C PLOT ANY LEFT OVER FLUX HISTORY PLOTS:
IF ((IPLJT.EQ.2).AND.(IK.NE.0)) CALL PHIPLT(IK)
C BOUNDARY CONDITION INSERTIONS.
IER=0
C IER ALLJWS B.C. INSERTIONS FOR YES AND NO CONVERGENCE:
55 DO 70 IG=1,2
PHI(IG,N+2)=0.0
DO 60 I=1,N
J=N+1-I
60 PHI(IG,J+1)=PHI(IG,J)
IF (IBC.NE.4) GO TO 65
PHI(IG,2)=0.0
GO TO 70
65 PHI(IG,1)=0.0
IF (IBC.NE.1) GO TO 70
PHI(IG,N+2)=PHI(IG,N+1)
PHI(IG,N+1)=0.0
70 CONTINUE
90 IF (IER.EQ.1) GO TO 102
RETURN
C NO CONVERGENCE ACCOMPLISHED:
100 CONTINUE
C NORMALIZE THE UNCONVERGED FLUX:
CALL NORMAL(PHI,N)
ICOUT=ICOUT-1
WRITE (6,101) ICOUT
101 FORMAT (1H1,'POWER METHOD DID NOT CONVERGE FOR THIS CASE AFTER',
   X 'ITERATIONS.',//,1X,'EXECUTION TERMINATED ')
IER=1
GO TO 55
102 CONTINUE
   C FOR PRINTING OUT THE EIGENVALUE HISTORY AND THE FINAL FLUX SHAPE:
   IPLUT=1
   JPLUT=1
RETURN
END
SUBROUTINE VPROD(A,X,S,N)
  FORMS THE VECTOR S = PRODUCT OF NXN MATRIX A AND VECTOR X:
  WHERE A IS THE CUBIC HERMITE (1,6) STORAGE MATRIX.
  REAL*8 A(50,6),X(50),S(50)
  DO I I=1,N
  S(I)=0.0
  DO M=1,6
    K=2*(I/2)+M-3
    IF ((K.LT.1).OR.(K.GT.N)) GO TO 1
    S(I)=S(I)+A(I,M)*X(K)
  1 CONTINUE
RETURN
END
SUBROUTINE SOLVE(A,X,Y,N,G,Z)
C SOLVES A*X = Y USING CHOLESKY'S METHOD OF FACTORIZATION
C FOR POSITIVE DEFINITE REAL AND SYMMETRIC MATRICES A.
C REFERENCE: FORSYTHE & MOLER.
C G AND Z ARE TEMPORARY WORK AREAS.
C MODIFIED FOR THE CRAZY CUBIC HERMITE (I,6) MATRICES:
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(50,6),G(50,6),X(50),Y(50),Z(50)
C FORM THE MATRIX FACTORIZATION TO G:
CALL FORMG(A,G,N)
C SOLVE: G*Z = Y:
CALL LOWTRILZ(Y,G,N)
C FORM G AS SYMMETRIC MATRIX:
CALL SYMG(G,N)
C SOLVE: G-TRANSPOSE*X = Z:
CALL UPPTRILG(X,Z,N)
RETURN
END
SUBROUTINE FORMG(A,G,N)
C
FURMS MATRIX G FROM A:
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION A(50,6),G(50,6)
        DO 20 J=1,N
            K=3+MOD(J,2)
            KO=K-1
            LO=1
            IF (J.LE.3) LO=2
            IF (J.EQ.1) LO=4
            SUM=0.0
            IF (LO.GT.KO) GO TO 2
            DO I=1,LO,KO
               1 SUM=SUM+G(J,L)**2
            2 SUM=A(J,K)-SUM
            IF (SUM.LT.0.0) GO TO 100
            G(J,K)=DSQRT(SUM)
            IF (J.EQ.N) GO TO 20
            I1=J+1
            I2=J+6-K
            IF (I2.GT.N) I2=N
            M=2
            IF (K.EQ.3) M=3
            DO 10 I=I1,I2
               3 SUM=0.0
               LO=1
               IF (I.LE.3) LO=2
               MO=M-1
               IF (LO.GT.MO) GO TO 7
               DO 5 L=LO,MO
                  JL=L
                  IF (M.EQ.2) JL=3
                  5 SUM=SUM+G(I,L)*G(J,JL)
               7 G(I,M)=(A(I,M)-SUM)/G(J,K)
               IF (M.EQ.3) M=1
            10 CONTINUE
        100 CONTINUE
        END
20 CONTINUE
RETURN
100 WRITE (6,101) J,K
101 FORMAT ('ERROR IN FORMG:',///,
  X 5X,'A(',I2,','I2,') < 0.0',///,
  X 5X,'CHOLESKY METHOD HAS FAILED.',///,
  X 5X,'MATRIX A MAY NOT BE POSITIVE DEFINITE OR SYMMETRIC ',///,
  X 'EXECUTION TERMINATED ')
CALL EXIT
RETURN
END
SUBROUTINE SYMG(G,N)
C FORMS SYMMETRIC G FROM G LOWER TRIANGULAR:
REAL*8 G(50,6)
N2=N-2
C FILL THE UPPER PORTION OF SYMMETRIC G:
10 DO 20 I=1,N2
   IF (MOD(I,2).EQ.1) GO TO 15
   C I IS EVEN:
   G(I,4)=G(I+1,3)
   G(I,5)=G(I+2,1)
   G(I,6)=G(I+3,1)
   GO TO 20
   C I IS ODD:
15   G(I,5)=G(I+1,2)
   G(I,6)=G(I+2,2)
20 CONTINUE
G(N-1,5)=G(N,2)
RETURN
END
SUBROUTINE L3WTRI(G,Z,Y,N)

SOLVES: \( G*Z = Y \); FOR Z

WHERE \( G \) IS NXN LOWER TRIANGULAR.

REAL*8 G(50,6),Z(50),Y(50),SUM

DO 10 I=1,N

K=3+MOD(I,2)

KO=K-1

SUM=0.0

LT=2*(I/2)

DO 5 M=1,KO

L=LT+M-3

IF ((L.LT.1).OR.(L.GT.N)) GO TO 5

SUM=SUM+G(IM)*Z(L)

CONTINUE

5 CONTINUE

Z(I)=(Y(I)-SUM)/G(I,K)

RETURN

END
SUBROUTINE UPTRI(G,X,Z,N)
C     SOLVES:  G*X = Z;  FOR X
C     WHERE G IS NXN UPPER TRIANGULAR.
REAL*8 G(50,6), X(50), Z(50), SUM
DO 10 J=1,N
I=N+1-J
K=3+MOD(I,2)
K1=K+1
SUM=0.0
LT=2*(I/2)
DO 5 M=K1,6
L=LT+M-3
IF (L.LT.1).OR.(L.GT.N) GO TO 5
SUM=SUM+G(I,M)*X(L)
CONTINUE
5 SUM=SUM+G(I,1)*X(L)
CONTINUE
10 X(I)=(Z(I)-SUM)/G(I,K)
RETURN
END
SUBROUTINE NORMAL(PHI,N)
NORMALIZES THE GROUP FLUXES AND CURRENTS BY THE LARGEST FLUX VALUE:
COMMON /B1/ IBC
REAL*8 PHI(2,52), A
NN=N
IF (IBC.EQ.1) NN=N-2 A=0.0
IF (IBC.NE.4) GO TO 5
A=DABS(PHI(1,1))
IF (DABS(PHI(2,1)).GT.A) A=DABS(PHI(2,1))
5 DO 1 IG=1,2
DO 1 I=2,NN,2
IF (DABS(PHI(IG,I)).GT.A) A=DABS(PHI(IG,I))
1 CONTINUE
DO 2 IG=1,2
DO 2 I=1,N
2 PHI(IG,I)=PHI(IG,I)/A
RETURN
END
SUBROUTINE PHIPLT(L)
C  PLots the group flux history, with up to 5 group fluxes per plot.
C  Fast and thermal group fluxes are plotted separately.
C  L is the number of fluxes to be plotted.
C  L is between 1 and 5.
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /B1/ IBC
COMMON /B2/ KR,N
COMMON /B5/ A(26,6), B(26,6)
COMMON /B6/ TE1(2,5), TE2(2,5), TE3(5), IN(5)
COMMON /ER/ EPS1, EPS2, EPS3
COMMON /FSTR/ PHISTR(2,26,6)
DIMENSION SYMBOL(5)
INTEGER SYMBOL
/KRI=KR+1
C  SET UP B.C. CONDITIONS
IF (IBC.EQ.4) GO TO 5
IF (IBC.EQ.3) GO TO 3
DO 2 IG=1,2
DO 2 K=1,L
DO 1 I=1,KR
J=KR+1-I
1 PHISTR(IG,J+1,K+1)=PHISTR(IG,J,K+1)
2 PHISTR(IG,1,K+1)=0.
3 IF (IBC.EQ.2) GO TO 5
DO 4 IG=1,2
DO 4 K=1,L
4 PHISTR(IG,KRI,K+1)=0.
5 CONTINUE
C  FLuxes in PHISTR have been normalized in power.
C  put the fast flux in A, and the thermal flux in B:
L1=L+1
DO 10 K=1,L1
DO 10 I=1,KR1
A(I,K)=PHISTR(1,I,K)
10 B(I,K)=PHISTR(2,I,K)
C PLOT THE FAST FLUX SHAPES ON ONE GRAPH:
CALL PRTPLT(0,A,KR1,L1,KR1,0,26,6,2)
WRITE (6,20)
20 FORMAT (/,'OFAST FLUX ITERATION HISTORY PLOT.',/)
WRITE (6,30)
30 FORMAT (/,'KEY:',5X,'SYMBOL',5X,'ITERATION NUMBER:',7X,'ERROR CRITERIA',
           X 11X,'ERROR',13X,'TOLERANCE')
DO 35 I=1,L
35 WRITE (6,40) SYMBOL(I),IN(I),TE1(1,I),EPS1,TE2(1,I),EPS2,
            X    TE3(I),EPS3
40 FORMAT (/,'MEAN SQ. FLUX',5X,1PD15.5,5X,1PD15.5,/,
            X 'EIGENVALUE',8X,1PD15.5,5X,1PD15.5)
C PLOT THE THERMAL FLUX SHAPES ON THE OTHER GRAPH:
CALL PRTPLT(0,B,KR1,L1,KR1,0,26,6,2)
WRITE (6,50)
50 FORMAT (/,'OTHERMAL FLUX ITERATION PLOT.',/)
WRITE (6,30)
DO 55 I=1,L
55 WRITE (6,40) SYMBOL(I),IN(I),TE1(2,I),EPS1,TE2(2,I),EPS2,
            X    TE3(I),EPS3
RETURN
END
SUBROUTINE CURENT
C      FORMS THE SEPERATE FLUX AND CURRENT VECTORS
C      FROM THE COMBINED ELEMENTS OF PHI.
C      THEN:  FLUX=PHI;  CURRENT=CUR.
COMMON /B2/  KR,N
COMMON /B4/  PHI(2,52),CUR(2,52)
REAL*8 PHI,CUR
KR1=KR+1
DO 2  IG=1,2
   DO 1  K=1,KR1
   1  CUR(IG,K)=PHI(IG,2*K)
   DO 2  K=1,KR1
   2  PHI(IG,K)=PHI(IG,2*K-1)
RETURN
END
SUBROUTINE OUTPUT
C PRINTS THE RESULTS OF THE METHOD.
C
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /81/ IBC,IPLOT,JPLLOT,IPUNCH
COMMON /82/ KR,N
COMMON /84/ PHI(2,52), CUR(2,52), LAMDA, ICOUT
COMMON /85/ PSI(2,26)
COMMON /ER/ EPS1,EPS2,EPS3
COMMON /ESTR/ LAMSTR(30),EFSTR(2,300),EFMSTR(2,300),ERLAM(300)
COMMON /TRUE/ TRULAM, TRUPHI(2,52), PHICCN(2,300), LAMCCN(300), IFT

INTEGER N
KRO=KR-1
KR1=KR+1

1 FORMAT ('RESULTS OF THE MULTIGROUP METHOD:*)
WRITE (6,10) ICOUT
10 FORMAT ('PROBLEM TERMINATED AFTER',I5,
X OUTER (POWER) ITERATIONS TO:*)
WRITE (6,20) LAMDA
20 FORMAT ('INDEX',8X,'THERMAL FLUX',11X,'FAST FLUX',5X,
X 'THERMAL CURRENT',8X,'FAST CURRENT',/)
WRITE (6,50) (K,PHI(2,K),PHI(1,K),CUR(2,K),CUR(1,K),K=1,KR1)
50 FORMAT (16,4020.7)

IF (IPUNCH.EQ.1) CALL PUNCH
C PRINT OUT EIGENVALUES.
C
CALL PLUT
WRITE (6,30)
30 FORMAT ('RESULTS AFTER PROBLEM TERMINATION:*/,
X 'INDEX',8X,'THERMAL FLUX',11X,'FAST FLUX',5X,
X 'THERMAL CURRENT',8X,'FAST CURRENT',/)
WRITE (6,50) (K,PHI(2,K),PHI(1,K),CUR(2,K),CUR(1,K),K=1,KR1)
50 FORMAT (16,4020.7)

DO 52 IG=1,2
A=0.0
DO 51 I=1,KR1
IF (PHI(IG,I).GT.A) A=PHI(IG,I)
51 CONTINUE
DO 52 I=1,KR1

PAGE 343
PHI(IG,1) = PHI(IG,1) / A

52 CUR(IG,1) = CUR(IG,1) / A

WRITE (6,55) (K, PHI(2,K), PHI(1,K), CUR(2,K), CUR(1,K), K=1, KR1)

55 FORMAT (/,'GROUP NORMALIZED RESULTS:','/(IG,4D20.7))

C   CALCULATE THE FINAL TO EXPECTED FLUX RATIOS:

K1 = 1
K2 = KR
IF (IBC .LE. 2) K1 = 2
IF (((IBC.EQ.1).OR.(IBC.EQ.3)) K2 = KR
DO 60 IG=1,2
IF (IBC .LE. 2) PSI(IG,1) = 1.0
IF (((IBC.EQ.1).OR.(IBC.EQ.3)) PSI(IG, KR1) = 1.0
I = 0
DO 60 K = K1, K2
I = I + 2
60 PSI(IG,K) = PHI(IG,K) / TRUPHI(IG,1)

WRITE (6,70) (1, PSI(2,I), PSI(1,I), I = 1, KR1)

70 FORMAT (*1RATIOS OF THE TERMINATED GROUP FLUX TO THE EXPECTED GROUP XP FLUX:*,//,
X 10X,'- AN INDICATION OF THE ACCURACY OF THE CONVERGENCE -',//,
X ' K', 12X,'THERMAL RATIO', 15X,'FAST RATIO',//, (15,2E25.10))

C   PRINT OUT THE STORED ITERATION ERRORS:

WRITE (6,110) EPS1, (EFSTR(2,I), I = 1, ICOUT)
WRITE (6,111) EPS1, (EFSTR(1,I), I = 1, ICOUT)
WRITE (6,112) EPS2, (EFMSTR(2,I), I = 1, ICOUT)
WRITE (6,113) EPS3, (EFMSTR(1,I), I = 1, ICOUT)
WRITE (6,114) EPS3, (ERLAM(I), I = 1, ICOUT)

110 FORMAT (*1MAXIMUM NORMALIZED ERRORS BETWEEN THE THERMAL FLUX ITERATIONS:*,
X 25X,'TOLERANCE USED = ', 1PE12.4, '//, (1P5E20.5))

111 FORMAT (*1MAXIMUM NORMALIZED ERRORS BETWEEN THE FAST FLUX ITERATIONS:*,
X 25X,'TOLERANCE USED = ', 1PE12.4, '//, (1P5E20.5))

112 FORMAT (*1MEAN SQUARE NORMALIZED ERROR BETWEEN THE THERMAL FLUX ITERATIONS:*,
X 18X,'TOLERANCE USED = ', 1PE12.4, '//, (1P5E20.5))
113 FORMAT (*1MEAN SQUARE NORMALIZED ERROR BETWEEN THE FAST FLUX ITERATIONS:*,
  X 18X,'TOLERANCE USED = ',1PE12.4,///,(1P5E20.5))
114 FORMAT (*1ERROR BETWEEN THE ITERATION EIGENVALUES:*,
  X 28X,'TOLERANCE USED = ',1PE12.4,///,(1P5E20.5))
IF (IFT.EQ.0) RETURN
C PRINT OUT THE GIVEN TRUE EIGENVALUE AND FLUX:
WRITE (6,115) TRULAM,((TRUPHI(3-J,I),J=1,2),I=1,N)
115 FORMAT (*1THE GIVEN TRUE EIGENVALUE:*,//,15X,
  X 'TRULAM =',E22.14,///,' THE GIVEN MULTIGROUP FLUXES:*,//,
  X 13X,'THERMAL',16X,'FAST',//,(2D2C.10))
C PRINT OUT THE STORED CONVERGENCE ERRORS:
WRITE (6,120) (PHICON(2,I),I=1,ICCUT)
WRITE (6,121) (PHICON(1,I),I=1,ICOUT)
WRITE (6,122) (LAMCON(I),I=1,ICCUT)
120 FORMAT (*1MEAN SQUARE ERROR BETWEEN THE THERMAL ITERATION FLUX AND
  X THE GIVEN TRUE THERMAL FLUX:*,//,(1P5E20.5))
121 FORMAT (*1MEAN SQUARE ERROR BETWEEN THE FAST ITERATION FLUX AND THE
  X GIVEN TRUE FAST FLUX:*,//,(1P5E20.5))
122 FORMAT (*1ERROR BETWEEN THE ITERATION EIGENVALUES AND THE GIVEN TRUE
  X EIGENVALUE:*,//,(1P5E20.5))
C RETURN
END
SUBROUTINE PLOT
C PLOTS OUT THE EIGENVALUE HISTORY AS A TABLE AND A GRAPH,
C AS WELL AS PLOTTING OUT THE FINAL MULTIGROUP FLUX SHAPES.
IMPLICIT REAL*8 (A-H,L-Z)
COMMON /81/ IBC,IPLIC,JPLIC,IPUNCH
COMMON /82/ KR
COMMON /B4/ PHI(2,52), PSI(2,52), LAMDA, ICOUT
COMMON /B5/ B(300,2)
COMMON /ESTR/ LAMSTR(300)
DIMENSION C(26,3)
C IN ORDER TO SAVE SOME SPACE:
EQUIVALENCE (B(1),C(1))
WRITE (6,1) (LAMSTR(I),I=1,ICCUT)
1 FORMAT ('TABLE OF EIGENVALUES DURING THE POWER ITERATION:',
X //,(IP5E25.14))
IF (JPLUT.EQ.0) GO TO 20
DO 10 I=1,ICOUT
B(1,I)=I
10 B(1,2)=LAMSTR(I)
CALL PRPPLT(1,B,ICOUT,2,ICOUT,0,300,2,1)
WRITE (6,11)
11 FORMAT ('PLOT OF THE EIGENVALUE HISTORY THROUGH THE ITERATIONS.')
20 IF (IPLOT.EQ.0) RETURN
KRI=KR+1
DO 30 I=1,KRI
C(I,1)=1
C(I,2)=PHI(1,I)
30 C(I,3)=PHI(2,I)
CALL PRPPLT(2,C,KRI,3,KRI,0,26,3,2)
WRITE (6,31)
31 FORMAT ('OFINAL CONNECTING FLUX POINTS; F(K).',//,
X 5X,'FAST FLUX: .',//,5X,'THERMAL FLUX: -')
RETURN
END
SUBROUTINE PUNCH
C PUNCHES OUT THE CUBIC RESULTS FOR PLOT2G ROUTINE INPUT:
REAL*8 F,C
COMMUN /B2/ KR
COMMUN /B4/ F(2,52), C(2,52)
KR1=KR+1
WRITE (7,1) KR,(F(1, I),CLI I),F(2, I),C(2,I),I=1,KR1)
1 FORMAT (15,/,(4D20.7))
WRITE (6,100)
100 FORMAT (///,'THE OUTPUT HAS BEEN PUNCHED OUT ONTO CARDS ')
RETURN
END
SUBROUTINE NORM2(PSI,TRUPHI,N)
C NORMALIZES BOTH ENERGY GROUP FLUXES IN PSI TO 1.0:
C DITTO FOR TRUPHI ON THE FIRST CALL.
COMMON /B1/ IBC
REAL*8 PSI(2,52), TRUPHI(2,52), A(2)
DATA K /0/
K=K+1
NN=N
IF (IBC.EQ.1) NN=NN-2
DO 1 IG=1,2
A(IG)=0.0
IF (IBC.NE.4) GO TO 7
A(IG)=DABS(PSI(IG,1))
7 DO 1 I=2,NN,2
IF (DABS(PSI(IG,I)).GT.A(IG)) A(IG)=DABS(PSI(IG,I))
1 CONTINUE
DO 2 IG=1,2
DO 2 I=1,N
IF (A(IG).EQ.0.0) GO TO 2
PSI(IG,I)=PSI(IG,I)/A(IG)
2 CONTINUE
IF (K.NE.1) RETURN
DO 5 IG=1,2
A(IG)=0.
IF (IBC.NE.4) GO TO 8
A(IG)=DABS(TRUPHI(IG,1))
8 DO 5 I=2,NN,2
IF (TRUPHI(IG,I).GT.A(IG)) A(IG)=TRUPHI(IG,I)
5 CONTINUE
DO 6 IG=1,2
DO 6 I=1,N
IF (A(IG).EQ.0.0) GO TO 6
TRUPHI(IG,I)=TRUPHI(IG,I)/A(IG)
6 CONTINUE
RETURN
END
SUBROUTINE PRTPLT(NO,B,N,M,NL,NS,KX,JX,ISP)

C
C
C
IDENTICAL TO SUBROUTINE PRTPLT PREVIOUSLY LISTED IN PROGRAM REF2G.

RETURN
END
F.4. SOURCE LISTING of Program ANALYZE
Figure F.4. Structure of Program ANALYZE.
Not including the M.I.T. SC-4020 Subroutine Package.54
C PROGRAM ANALYZE:
C DATA ANALYSIS AND COMPARISON PROGRAM:
DIMENSION NP(200),CHI(2),R(9,26),F(2,201),G(2,201),H(1000),
X X(1001),PHI(2,1001),SF(2,1000),D(2,1000),
X U(2,1001),XB(1001),UB(2,1001),XC(1001),UC(2,1001),
X XS(1001),NSR(3,26),NRNK(25)
C THETA IS DEFINED SUCH THAT THE CURRENT (0) = THETA * G, FOR EACH K.
C DO FOR THE 3 FLUX DATA BLOCKS:
DO 10 IT=1,3
C READ IN THE MATERIALS INPUT DATA BLOCK:
CALL DATAIN(IT,METHOD,NK,NR,NRNK,NP,N,XS,PHI,FS,DF,CHI,THETA,NAP,H)
C READ IN THE CONVERGED FLUX POINTS DATA BLOCK:
CALL SYNPTS(IT,METHOD,NK,NR,F,G)
C CALCULATE THE DETAILED FLUX SHAPES:
IF (METHOD.EQ.2) GO TO 4
CALL CALCUL(IT,METHOD,NK,NR,NRNK,NP,N,X,U,XS,PHI,FS,D,CHI,
X THETA,F,G)
GO TO 5
4 CALL CALCUC(IT,METHOD,NK,NR,NRNK,NP,N,X,U,XS,PHI,FS,D,CHI,
X THETA,F,G,NAP)
C TRANSFORM THE FLUX POINTS INTO NK DISTINCT REGIONS:
5 CALL REFORM(IT,METHOD,NK,NR,NRNK,NP,N,X,U,FS,D)
C CALCULATE THE POWER IN EACH OF THE NK REGIONS FOR EACH FLUX:
CALL KPOWER(IT,METHOD,NK,NP,N,X,UF,DS,CHI,THETA,NAP,F,G,H,
X NSR,R,TPOWER)
C NORMALIZE THE REGION POWERS AND THE FLUX POINTS BY TPOWER)
CALL POWN4R(IT,METHOD,NK,NP,N,X,U,NS,R,TPCWER)
C SAVE THESE RESULTS FOR PLOTTING:
CALL SAVE(IT,NK,N,X,U,NBXB,UB,NC,UC)
10 CONTINUE
C NORMALIZE THE FLUX RESULTS TOGETHER FOR EACH GROUP TO 1.0:
CALL KELNOR(NK,N,X,U,NBXB,UB,NC,UC)
C PRINT OUT THE POWER AND ARRAY RESULTS:
CALL OUTPUT(NK,N,X,U,NS,UB,NC,UC,NSR,R)
C READ IN THE PLOTTING INFORMATION AND PLOT THE RESULTS ON THE SC 4020:
CALL SCPLUT(NK,N,X,U,NBXB,UB,NC,UC)
WRITE (6,20)
20 FORMAT ('ONORMAL PROGRAM TERMINATION.')
STOP
END
SUBROUTINE DATAIN(IT,METHOD,NK,NR,NRNK,NP,N,X,PHI, SF,C,CHI,THETA,
X NAP,H)
C    READS IN THE MATERIAL DATA INPUT BLOCKS:
DIMENSION NP(200),K(200),L(200),ITF(200),KTF(200),CHI(2),
X X(1001),PHI(2,1001),SF(2,1000),D(2,1000),H(1000),
X NRNK(25)
C    DATA BLOCK ID CARDS:
READ (IT,20) METHOD, NK, NR, NAP
C    NRNK = NUMBER OF NR REGIONS PER EACH NK REGION:
   IF (NR.NE.NK) READ (IT,20) (NRNK(I),I=1,NK)
20 FORMAT (1b15)
   XSTART=0.0
   K(1)=1
   READ (IT,11) NR
   READ (IT,12) (ITF(I),I=1,NR)
   READ (IT,17) CHI(1),CHI(2), THETA
   IF (THETA.EQ.0.) THETA=1.
   NUMITF=1
   DO 10 I=1,NR
   IF (ITF(I).LT.NUMITF) GO TO 2
   C    NEW REGION DATA:
   NUMITF=NUMITF+1
   KTF(NUMITF-1)=1
   READ (IT,13) NS
   IF (NS.EQ.0.) NS=1
   NP(I)=NS+1
   IF (I.NE.1) K(I)=K(I-1)+NP(I-1)
   L(I)=NP(I)
   IF (I.NE.1) L(I)=L(I-1)+NP(I)
   K0=K(I)
   L0=L(I)
   L1=L0-1
   L2=L0-2
   IF (NS.GT.1) READ (IT,14) (X(J),H(J),SF(1,J),D(1,J),
X SF(2,J),D(2,J),J=K0,L2)
   READ (IT,15) X(L1),X(L0),H(L1),SF(1,L1),D(1,L1),SF(2,L1),D(2,L1)
XSTART=XSTART-X(KO)

DO 1 J=KO,LO

1 X(J)=X(J)+XSTART
XSTART=X(LO)

IF (IT.EQ.3) GO TO 10
READ (IT,16) (PHI(1,J),J=KO,LO)
READ (IT,16) (PHI(2,J),J=KO,LO)
GO TO 10

C OLD REPEATED REGION DATA:

2 M=ITF(I)
M=KTF(M)
NP(I)=NP(M)
K(I)=K(I-1)+NP(I-1)
L(I)=L(I-1)+NP(I)
KO=K(I)
LO=L(I)
L1=LO-1
KM=K(M)
IB=KO-KM
XSTART=XSTART-X(KM)

DO 5 J=KO,LO

NJ=J-IB
X(J)=X(NJ)+XSTART
H(J)=H(NJ)

DO 5 IG=1,Z
D(IG,J)=D(IG,NJ)
SF(IG,J)=SF(IG,NJ)
PHI(IG,J)=PHI(IG,NJ)

5 CONTINUE
XSTART=X(LO)

10 CONTINUE

11 FORMAT (/,15)
12 FORMAT (2512)
13 FORMAT (5X,15)
14 FORMAT (F10.5,10X,F10.5,10X,2E10.3,/,40X,2E10.3)
15 FORMAT (3F10.5,10X,2E10.3,/,40X,2E10.3)
16 FORMAT (E20.7)
17 FORMAT (3F10.5).
    N=L(NR)
    IF (IT.NE.3) RETURN
    DO 50 I=1,N
    DO 50 IG=1,2
50   PHI(IG,1)=1.0
    RETURN
    END
SUBROUTINE SYNPS(IT,METHOD,NK,NR,F,G)
C READS IN CONVERGED FLUX AND CURRENT POINTS:
DIMENSION F(2,201),G(2,201)
JT=IT+10
READ (JT,1) NR
1 FORMAT (15)
NR1=NR+1
IF (METHOD.NE.1) GO TO 10
C FOR LINEAR SYNTHESIS:
   IF (IT.NE.3) READ (JT,2) (F(1,I),F(2,I),I=1,NR1)
   IF (IT.EQ.3) READ (JT,3) (F(1,I),I=1,NR1),(F(2,I),I=1,NR1)
2 FORMAT (2E20.7)
3 FORMAT (E20.7)
GO TO 20
10 READ (JT,11) (F(1,I),G(1,I),F(2,I),G(2,I),I=1,NR1)
11 FORMAT (4E20.7)
20 RETURN
END
SUBROUTINE CALCUL(IT,METHOD,NK,NRNRNKNP,NU,XU,U,XS,PHI, SF,D,CHI, X THETA,F,G)

C
CALCULATES THE FLUX TRIAL FUNCTION U FOR LINEAR SYNTHESIS:
DIMENSION NP(200),XU(1001),U(2,1001),PHI(2,1001),SF(2,1000), X D(2,1000),CHI(2),F(2,201),G(2,201),XS(1001),NRNK(25)

ISYNTH=20

C FOR EACH K*TH REGION:
N=0
LL=0
DO 20 K=1,NR
L=LL+1
LL=LL+NP(K)
DO 7 IG=1,2
IF (PHI(IG,L).EQ.0.0) PHI(IG,L)=1.0E-6
IF (PHI(IG,LL).EQ.0.0) PHI(IG,LL)=1.0E-6
7 CONTINUE

C FOR ALL POINTS IN THIS REGION:
DO 10 I=LL,L+1
N=N+1
XU(N)=XS(I)
X=(XU(N)-XS(L))/(XS(LL)-XS(L))
IF ((K.EQ.1).AND.(ISYNTH.EQ.21.OR.ISYNTH.EQ.24)) X=1.0
IF ((K.EQ.NR).AND.(ISYNTH.EQ.22.OR.ISYNTH.EQ.24)) X=0.0
DO 1 IG=1,2
1 U(IG,N)=PHI(IG,I)*(F(IG,K)*(1.-X)/PHI(IG,L))
X = +F(IG,K+1)*X/PHI(IG,LL))
10 CONTINUE
20 CONTINUE
NU=N
RETURN
END
SUBROUTINE CALCUC(IT,METHID,NK,SR,SRK,SP,NU,UX,UX,PHI,SD,CHI,
X THETA,F,G,NAP)
C CALCULATES THE FLUX TRIAL FUNCTION U FOR CUBIC HERMITE SYNTHESIS:
C NAP = # OF ADDITIONAL POINTS TO BE CALCULATED IN EACH REGION
C AND IS NEGATIVE IF NAP APPLIES ONLY TO THE FIRST REGION.
DIMENSION NP(200),UX(1001),U(2,1001),Phi(2,1001),SF(2,1000),
X D(2,1000),CHI(2),F(2,201),G(2,201),US(1001),SRK(25)
X
NAP=NAPT
NL=0
DO 50 K=1,NR
NL=NL+NP(K)
50
C FOR EACH K'TH REGION:
N=0
L=0
DO 20 K=1,NR
L=LL+1
LL=LL+NP(K)
H=US(LL)-US(L)
DO 7 IG=1,2
IF (PHI(IG,L).EQ.0.0) PHI(IG,L)=1.0E-6
IF (PHI(IG,LL).EQ.0.0) PHI(IG,LL)=1.0E-6
7 CONTINUE
C FOR ALL POINTS IN THIS REGION:
DO 10 I=LL,LL
N=N+1
UX(I)=US(I)
X=(UX(I)-US(L))/(US(LL)-US(L))
DO 1 IG=1,2
1 U(IG,N)=PHI(IG,I)*(F(IG,K)*(1.-3.*X**2+2.*X**3)/PHI(IG,L)
X +F(IG,K+1)*(3.*X**2-2.*X**3)/PHI(IG,LL)
X +H*(G*(IG,K)/(D*(IG,L)*PHI(IG,L))(-X+2.*X**2-X**3)
X +G*(IG,K+1)/(D*(IG,LL-1)*PHI(IG,LL)*(X**2-X**3))*THETA)
NAPP=1ABS(NAP)
IF (NAP.EQ.0) GO TO 10
IF (I.EQ.LL) GO TO 10
DX=(US(I+1)-US(I))/FLOAT(NAPP+1)
DO 5 INP=1,NAPP
N=N+1
XU(N)=XU(N-1)+DX
X=(XU(N)-XS(LL))/(XS(LL)-XS(L))
DO 5 IG=1,2
5 U(IG,N)=PHI(IG,I)*(F(IG,K)*(-3.*X**2+2.*X**3)/PHI(IG,L))
   *F(IG,K+1)*(-3.*X**2-2.*X**3)/PHI(IG,L)
   +H*(D(IG,L)*PHI(IG,L)*(-X+2.*X**2-X**3))
   +H*(D(IG,L-1)*PHI(IG,L)*(X**2-X**3))*THETA)*THETA
10 CONTINUE
IF (NAPP.EQ.0) GO TO 15
IF (IT.EQ.1) GO TO 15
C SHIFTING SF AND D ARRAYS TO COMPENSATE FOR NAP .NE. 0:
NS=NP(K)-1
NNP=NS*NAPP
NPS=0
K1=K+1
IF (K1.GT.NR) GO TO 60
DO 51 I=K1,NK
51 NPS=NPS+NP(I)
DO 55 J=1,NPS
I=NL+1-J
II=I+NPP
DO 55 IG=1,2
SF(IG,II)=SF(IG,I)
55 D(IG,II)=D(IG,I)
60 NL=NL+NPP
L=NL-NPS
L1=L-NNP-NP(K)
DO 70 I=1,NS
II=NS+1-I+L1
NAPP1=NAPP+1
DO 65 J=1,NAPP1
L=L-1
65 IG=1,2
SF(IG,L)=SF(IG,II)
PAGE 360
65 D(IG,L)=D(IG,II)
70 CONTINUE
15 NP(K)=NP(K)+NAPP
    IF (NAP.LT.0) NAP=0
20 CONTINUE
    NU=N
    RETURN
END
SUBROUTINE REFORM(IT, METHOD, NK, NR, NRNK, NP, N, X, U, SF, D)
C REFORMS THE NR GIVEN REGIONS INTO NK DESIRED REGIONS:
DIMENSION NP(200), X(1001), U(2, 1001), SF(2, 1000), D(2, 1000), NRNK(25)
IF (NK.EQ.NR) RETURN
C IT SHOULD BE 3 ONLY.
C DELETE ALL DOUBLE ENTRIES WITHIN EACH NK DATA BLOCK:
M=-1
I=0
DO 5 K=1, NK
L=NRNK(K)
DO 4 J=1, L
I=I+1
M=M+2
X(I)=X(M)
DO 4 IG=1, 2
U(IG, I)=U(IG, M)
SF(IG, I)=SF(IG, M)
4 D(IG, I)=D(IG, M)
I=I+1
X(I)=X(M+1)
5 U(IG, I)=U(IG, M+1)
N=1
C NP(K) = # OF FLUX POINTS IN REGION K = # NS SUBREGIONS + 1:
DO 10 K=1, NK
10 NP(K)=NP(K)+NRNK(K)-1
NR=NK
RETURN
END
SUBROUTINE KPOWER(IT,METHOD,NK,NP,N,X,U,SF,D,CHI,THETA,NAF,F,G,
X
H,NSR,R,TPOWER)

C CALCULATES THE POWER IN EACH REGION BY INTEGRATIONS:

DIMENSION NP(200),X(1001),U(2,1001),SF(2,1000),D(2,1000),CHI(2),
X
F(2,201),G(2,201),H(1000),R(9,26),NSR(3,26)

TPOWER=0.0
IF (IT.NE.1) GO TO 20
C FOR THE HOMOGENEOUS CASE:
C ASSUMING THAT NP IS ALWAYS 2 AND THUS D*S ARE CONSTANT.
DO 10 K=1,NK
M=2*K-1
NSR(IT,K)=1
R(IT,K)=0.0
DO 8 IG=1,2
IF (METHOD.NE.1) GO TO 5
C LINEAR FLUX:
R(IT,K)=(F(IG,K)+F(IG,K+1))*H(M)*SF(IG,M)/2.0+R(IT,K)
GO TO 8
C CUBIC FLUX:
5 R(IT,K)=((F(IG,K)+F(IG,K+1))*H(M)/2.0+THETA*(-G(IG,K)
X /D(IG,M)+G(IG,K+1)/D(IG,M))*H(M)**2/12.0)*SF(IG,M)+R(IT,K)
CONTINUE
10 TPOWER=TPPOWER+R(IT,K)
RETURN
C SYNTHESIS OR REFERENCE DETAILED FLUX CASE:
20 M=-1
DO 50 K=1,NK
M=M+1
NS=NP(K)-1
NSR(IT,K)=NS
R(IT,K)=0.0
DO 40 J=1,NS
M=M+1
DO 40 IG=1,2
40 R(IT,K)=(U(IG,M)+U(IG,M+1))*(X(M+1)-X(M))*SF(IG,M)/2.0 + R(IT,K)
50 TPOWER=TPPOWER+R(IT,K)
SUBROUTINE POWNOR(IT, METHOD, NK, NP, N, X, U, R, TPOWER)
C
NORMALIZES THE FLUXES AND REGION POWERS BY TPOWER:
DIMENSION NP(200), X(1001), U(2,1001), R(9,26)
DO 1 K=1,NK
1 R(IT+3,K)=R(IT,K)/TPOWER
DO 2 IG=1,2
DO 2 K=1,N
2 U(IG,K)=U(IG,K)/TPOWER
RETURN
END
SUBROUTINE SAVE(ITN,K,N,X,U,NB,XB,UB,NC,XC,UC)
  C       SAVES THE FLUX DISTRIBUTIONS IN THE IT LOOP BY PLACING:
  C       IT = 1: HOMOGENEOUS RESULTS IN UC;
  C       IT = 2: SYNTHESIS RESULTS IN UB;
  C       IT = 3: REFERENCE RESULTS IN U.
  DIMENSION X(1001),U(2,1001),XB(1001),UB(2,1001),
                  XC(1001),UC(2,1001)
  IF (IT.NE.1) GO TO 2
  HOMOGENEOUS RESULTS:
    DO 1 I=1,N
    XC(I)=X(I)
    DO 1 IG=1,2
    1 UC(IG,I)=U(IG,I)
    NC=N
    RETURN
  2 IF (IT.EQ.3) RETURN
  SYNTHESIS RESULTS:
    DO 3 I=1,N
    XB(I)=X(I)
    DO 3 IG=1,2
    3 UB(IG,I)=U(IG,I)
    NB=N
    RETURN
END
SUBROUTINE RELNOR(NK,N,X,UB,NB,XB,UB,NC,XC,UC)
C
NORMALIZES THE SET OF U, UB, UC; TO UNITY FOR EACH GROUP:

DIMENSION X(1001),U(2,1001),XB(1001),UB(2,1001),
X
XC(1001),UC(2,1001)

DIMENSION A(2),B(2),C(2),Z(2,3)

CALL MAX(N,UB,A)
CALL MAX(NB,UB,B)
CALL MAX(NC,UC,C)

DO 1 IG=1,2
Z(IG,1)=A(IG)
Z(IG,2)=B(IG)
1 Z(IG,3)=C(IG)

CALL MAX(3,Z,A)
CALL DIV(N,UB,A)
CALL DIV(NB,UB,B)
CALL DIV(UC,UC,A)
RETURN
END
SUBROUTINE MAX(N,U,A)
C FINDS THE MAXIMUM POSITIVE ELEMENT OF U FOR EACH GROUP:
DIMENSION U(2,1001),A(2)
DO 1 IG=1,2
A(IG)=0.
DO 1 I=1,N
IF (U(IG,I).GT.A(IG)) A(IG)=U(IG,I)
1 CONTINUE
RETURN
END
SUBROUTINE DIV(N,U,A)

C DIVIDES A INTO U FOR EACH GROUP:
DIMENSION U(2,1001),A(2)
DO 1 IG=1,2
DO 1 I=1,N
1 U(IG,1)=U(IG,1)/A(IG)
RETURN
END
SUBROUTINE OUTPUT(NK,N,X,U,NB,XB,UB,NC,XC,UC,NSR,R)

C PRINTS OUT THE ANALYSIS RESULTS:

DIMENSION X(1001),U(2,1001),XB(1001),UB(2,1001),
X
XC(1001),UC(2,1001),R(9,26),NSR(3,26)

C SUM UP THE REGION RAW POWERS AND REGION FRACTIONAL POWERS:
DO 5 IT=1,6
IF (IT.LE.3) NSR(IT,26)=0
R(IT,26)=0.0
DO 5 K=1,NK
IF (IT.LE.3) NSR(IT,26)=NSR(IT,26)+NSR(IT,K)
5 R(IT,26)=R(IT,26)+R(IT,K)

C THE PERCENT NORMALIZED DIFFERENCES OF THE REGION FRACTIONAL POWERS:
DO 60 K=1,NK
DO 50 IT=7,9
50 R(IT,K)=0.0
IF (R(IT,K).EQ.0.0) GO TO 60
R(IT,K)=(R(IT,K)-R(IT,K))*100./R(IT,K)
R(IT,K)=(R(IT,K)-R(IT,K))*100./R(IT,K)
IF (R(IT,K).EQ.0.0) GO TO 60
R(IT,K)=(R(IT,K)-R(IT,K))*100./R(IT,K)
60 CONTINUE

C THE PERCENT NORMALIZED DIFFERENCES OF THE TOTAL RAW POWER PRODUCED:
R(7,26)=(R(3,26)-R(1,26))*100./R(3,26)
R(8,26)=(R(2,26)-R(1,26))*100./R(2,26)
R(9,26)=(R(2,26)-R(1,26))*100./R(2,26)
WRITE (6,10) NK,(K,(NSR(IT,K),R(IT,K),IT=1,3),K=1,NK)
10 FORMAT ('RESULTS OF THE INTEGRATED POWER IN EACH OF THE',13,
X 'REGIONS:','/,,X 'CALCULATED POWER LEVELS, AND NUMBER OF SUBREGIONS PER REGION:
X '','//,3X,'REGION:','10X,'HOMOGENIZED RESULTS:','5X,
X 'SYNTHESIZED RESULTS:','7X,'REFERENCE RESULTS:','//,
X (110,10X,13,E17.7,18,E17.7,110,E15.7))
WRITE (6,12) (NSR(IT,26),R(IT,26),IT=1,3)
12 FORMAT ('TOTALS:','10X,13,E17.7,18,E17.7,110,E15.7)
WRITE (6,20) (K,(R(IT,K),IT=4,6),K=1,NK)
20 FORMAT ('FRACTIONAL POWER LEVELS:','//,3X,'REGION:','10X,
'HOMOGENEOUS RESULTS:', 5X,
'SYNTHESIZED RESULTS:', 7X, 'REFERENCE RESULTS:', //,
(I10, 5X, 3E25.7))
WRITE (6, 11) (R(IT, 26), IT=4, 6)
11 FORMAT (/, 3X, 'TOTALS:', 5X, 3E25.7)
WRITE (6, 30) (K, (R(IT, K), IT=7, 9), K=1, NK)
30 FORMAT (//, 'FRACTIONAL POWER NORMALIZED PERCENT ERRORS:', //,
3X, 'REGION: ', 14X, '(REF-HCMO)/REF %',
8X, '(REF-SYNTH)/REF %', 5X, '(SYNTN-HCMO)/SYNTH %', //,
(I10, 5X, 3E25.7))
WRITE (6, 40) (R(IT, 26), IT=7, 9)
40 FORMAT (//, 'RAW PRODUCTION POWER NORMALIZED PERCENT ERRORS:', //,
24X, '(REF-HUMO)/REF %',
8X, '(REF-SYNTH)/REF %', 5X, '(SYNTN-HCMO)/SYNTH %', //,
(15X, 3E25.7))
RETURN
END
SUBROUTINE SCPLOT(NK, N, X, U, NE, XBY, UB, NC, XC, UC)
C READS IN PLOTTING INFORMATION AND PLOTS THE FLUX COMPARISONS:
COMMON /SC/ TITLE(20), XINCH, YINCH, NCELL, WCELL, NLL, XL(100)
COMMON /MAX/ XMIN, XMAX, YMIN, YMAX
COMMON /RASTER/ IXS, IYS, IXE, IYE, IXT, IYT, IXLX, IYLX,
X IXLY, IYLY(3), LH, LW, IS, IR
COMMON /MARGIN/ ML, MR, MB, MT
DIMENSION X(1001), U(2, 1001), XB(1001), UB(2, 1001), XC(1001),
X UC(2, 1001)
READ (5, 1, END=20) TITLE
READ (5, 4, END=20) XINCH, YINCH
C NEGATIVE NCELL SPECIFIES THAT THE LAST CELL IS A HALF CELL:
READ (5, 3, END=20) NCELL, WCELL
READ (5, 3, END=5) NLL, (XL(I), I=1, 7)
IF (NLL.GT.7) READ (5, 4) (XL(I), I=8, NLL)
I FORMAT (20A4)
3 FORMAT (110, 7F10.5)
4 FORMAT (8F10.5)
GO TO 10
5 NLL=0
10 WRITE (6, 11) TITLE
11 FORMAT ('EXECUTING GENERAL ANALYSIS AND FLUX PLOTTING PROGRAM:',
X '//' 5X, 'TITLE OF PLOTTING RUN IS:', '20A4', '///')
X XMIN=0.0
XMAX=WCELL*FLOAT(INCELL)
IF (NCCELL.LT.0) XMAX=WCELL*(FLOAT(-NCCELL)-0.5)
YMIN=0.
YMAX=1.0
WRITE (6, 13) NCELL, WCELL, XMIN, XMAX, YMIN, YMAX, NLL
13 FORMAT ('OREACTOR GEOMETRY PARAMETERS:', '/', 5X, 'NCCELL =', 'I5,/',
X '/5X,' 'WCELL =', 'F10.5,/, 5X, 'XMIN =', 'F10.5,/, 5X, 'XMAX =', 'F10.5,/,
X '/5X,' 'YMIN =', 'F10.5,/, 5X, 'YMAX =', 'F10.5,/,
X '/5X,' 'NLL =', 'I5')
IF (NLL.GT.0) WRITE (6, 14) (XL(I), I=1, NLL)
14 FORMAT (5X, '(XL(I), I=1, NLL) =', '///', (10X, 10F10.5))
CALL SETUP
CALL PLUT(N,X,U,NB,XB,UB,NC,XC,UC)
20 RETURN
END
SUBROUTINE SETUP
C FINDS THE RASTER LOCATIONS FOR THE GRID CORNERS, LABELS, AND TITLE
C ENTIRE PLOTTING AREA (INCLUDING LABELS) IS XINCH X YINCH.
COMMON /SC/ TITLE(20),XINCH,YINCH,NCELL,WCELL,NLL,XL(100)
COMMON /RASTER/ IXS,IYS,IXE,IYE,IXT,IYT,IXLX,IYLY,IXLY,IYLY(3),LH,LW,IS,IR
COMMON /MARGIN/ ML,MR,MB,MT
C SET THE LETTER SIZE FOR THE LABELS AND TITLES:
  LH=2
  LW=2
  IS=5*LW+3
  IR=7*LH+5
C PLOTTING AREA:
C ONE INCH = 137 RASTERS. PLOTTING AREA IS 1023 RASTERS SQUARE.
  IXE=XINCH*137.+0.5
  IYE=YINCH*137.+0.5
  IF (IXE.GT.1020) IXE=1020
  IF (IYE.GT.1020) IYE=1020
  IXS=(1023-IXE)/2.+1
  IYS=(1023-IYE)/2+1
  IXE=IXS+IXE-1
  IYE=IYS+IYLY-1
C COORDS FOR THE LETTERS ARE FOR THEIR CENTERS:
C TITLE (UPPER LEFT CORNER):
  IXT=8
  IYT=1014
C X AXIS LABEL
  IXLX=(IXS+60+IXE-6*IS)/2+1+8
  IYLX=IYS + 10
C Y AXIS LABELS (FOR EACH FLUX PLOT TYPE):
  IYLY=IXS + 10
  IYLY(1)=(IYS+43+IYE-20*IS)/2+1 +4
  IYLY(2)=(IYS+43+IYE-23*IS)/2+1 +4
  IYLY(3)=(IYS+43+IYE-26*IS)/2+1 +4
C SET MARGIN SPACING FOR GRID:
  ML=IXS + 2*18 -1
MR=1023-1XE
MB=1YS + 2*18 - 1
MT=1023-1YE
RETURN
END
SUBROUTINE PLOT(NAX,UA,NC,NCUB,UC)

USES MIT-IPC'S SC-4020 SUBROUTINE PLOTTING PACKAGE.

PLOTS THE COMPARISON FLUX DISTRIBUTIONS:

KEY: HOMOGENEOUS - DASHED LINE.
SYNTHESIS - SOLID LINE.
REFERENCE - DOTTED LINE.

COMMON /RASTER/ IXS,IYX,IXT,IYT,IXLY,IYLX,
X
 IXLY,IYLX(3)

DIMENSION XA(1001),UA(2,1001),XB(1001),UC(2,1001),
X
 XC(1001),UC(2,1001), X(1001),T(1001)

HARDCOPY INITIALIZATION:
CALL STUIDV('M7788-6571',9,2)

DO FOR EACH ENERGY GROUP:
DO 10 IG=1,2
NTH=1
CALL GRID
IF (IG.EQ.1) CALL RITE2V(IXLY,IXLY(1),1024,90,1,20,NTH,
X 'NORMAILED FAST FLUX',N)
IF (IG.EQ.2) CALL RITE2V(IXLY,IXLY(2),1024,90,1,23,NTH,
X 'NORMALIZED THERMAL FLUX',N)

HOMOGENEOUS RESULTS:
CALL PUT(IG,NC,NCUB,UC,N,X,T)
CALL ADJUST(N,X,T,1)
CALL LINE(N,X,T,1)

SYNTHESIS RESULTS:
CALL PUT(IG,NB,NCUB,UC,N,X,T)
CALL ADJUST(N,X,T,0)
CALL LINE(N,X,T,0)

REFERENCE RESULTS:
CALL PUT(IG,NB,NCUB,UC,N,X,T)
CALL ADJUST(N,X,T,2)
CALL LINE(N,X,T,2)

10 CONTINUE
CALL PLTND(NTH)
WRITE (6,20) NTH
20 FORMAT ('1',15,' SC 4020 PLOTS HAVE BEEN PLOTTED.')
SUBROUTINE PUT(IG, NA, XA, UA, N, X, T)
C TRANSFERS NA, XA, UA OF IG INTO N, X, T:
DIMENSION XA(1001), UA(2,1001), X(1001), T(1001)
N=NA
DO 1 I=1, N
  X(I)=XA(I)
  T(I)=UA(IG,I)
1 RETURN
END
SUBROUTINE GRID
C SETS UP A NEW FRAME AND PLOTS THE GRID
C AS WELL AS RUN TITLE, LABELS, AND LIGHT LINES.
COMMON /SC/ TITLE(20),XINCH,YINCH,NCELL,WCELL,NLL,XL(100)
COMMON /MAX/ XMIN, XMAX, YMIN, YMAX
COMMON /RASTER/ IXS, IYS, IXE, IYE, IXT, IYT, IXLX, IYLX,
X IXLY, IYLX(3), LH, LW, IS, IR
COMMON /MARGIN/ ML, MR, MB, MT
DATA IT /1/
IT=IT+1
IF (IT.GT.3) IT=3
NTH=1
IF (IT.GT.2) GO TO 10
CALL CHSIZV(LW,LH)
CALL RITSTV(IS,IR)
CALL SETMIV(ML,MR,MB,MT)
10 CALL GRIDIV(IT,XMIN,XMAX,YMIN,YMAX,WCELL,0,1,1,0,-1,-1,5,3)
CALL PRINTV(80,TITLE,IXT,IYT,1)
CALL RITE2V(IXLX,IYLX,1024,0,1,6,NTH,'X (CM)',N)
20 IF (NLL.EQ.0) GO TO 5
IY1=IYV(YMIN)
IY2=IYV(YMAX)
DO 1 I=1,NLL
IX=IXV(XL(I))
1 CALL LINEV(IX,IY1,IX,IY2)
5 RETURN
END
SUBROUTINE ADJUST(N,X,Y,L)
ADJUSTS THE X AND Y ARRAYS BY DELETING ANY (X,Y) POINTS
WITHIN UK EQUAL TO NR RASTERS DISTANCE OF EACH OTHER.
HOPEFULLY ELIMINATES DARK SPOTS ON THE PlOTS.

L = 0: SOLID LINE PLOTTING: NR = 2.
L = 1: DASHED LINE PLOTTING: NR = 2; INCRV(10,5).
L = 2: DOTTED LINE PLOTTING: NR = 4; INCRV(2,2).

DIMENSION X(1), Y(1)
NR=2
IF (L.EQ.2) NR=4
IF (N.LE.1) RETURN
K=1
IX1=IXV(X(1))
IY1=IYV(Y(1))
DO 1 I=2,N
IX2=IXV(X(I))
IY2=IYV(Y(I))
ID=SQRT(REAL((IX2-IX1)**2+(IY2-IY1)**2))+0.5
IF (ID.LE.NR) GO TO 1
K=K+1
X(K)=X(I)
Y(K)=Y(I)
IX1=IX2
IY1=IY2
1 CONTINUE
N=K
RETURN
END
SUBROUTINE LINE(N,X,Y,K)
C    PLOTS: A LINE (K=0), OR A DASHED LINE (K=1), OR A DOTTED LINE (K=2)
C    THROUGH Y(X) DATA POINTS.
DIMENSION X(I), Y(I)
IX1=IXV(X(1))
IY1=IYV(Y(1))
DO 10 I=2,N
IX2=IXV(X(I))
IY2=IYV(Y(I))
IF (K.NE.0) GO TO 1
C    SOLID LINE PLOT:
    CALL LINEV(IX1,IY1,IX2,IY2)
    GO TO 5
C    DOTTED OR DASHED LINE PLOT:
1  IF (K.EQ.1) CALL INCRV(1),5
  IF (K.EQ.2) CALL INCRV(2,2)
    CALL DOTLV(IX1,IY1,IX2,IY2)
5  IX1=IX2
10  IY1=IY2
RETURN
END