LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although portions of this report are not reproducible, it is being made available in microfiche to facilitate the availability of those parts of the document which are legible.

2

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A08; Microfiche A01

ORNL/CSD/TM-182

ORNL/CSD/TM--182

DE82 018570

DEVELOPMENT OF A NEW TWO-DIMENSIONAL CARTESIAN GEOMETRY NODAL MULTIGROUP DISCRETE-ORDINATES METHOD

Ronald Earl Pevey

Date Published: July 1982

Report Prepared by
University of Tennessee
Department of Nuclear Engineering
Knoxville, Tennessee
under
Subcontract Number 7685

for

COMPUTER SCIENCES
at
Oak Ridge National Laboratory
Post Office Box X
Oak Ridge, Tennessee 37830

Union Carbide Corporation-Nuclear Division
operating the
Oak Ridge Gaseous Diffusion Plant • Oak Ridge National Laboratory
Oak Ridge Y-12 Plant • Paducah Gaseous Diffusion Plant
under Contract No. W-7405-eng-26
for the
Department of Energy

TABLE OF CONTENTS

CHAPTER		PAGE
LIST OF TABLES		v
LIST OF FIGURES		vi
ACKNOWLEDGMENTS		ix
ABSTRACT		xi
1.	INTRODUCTION	1
	A. Project Objectives	2
2.	BACKGROUND	4
	A. Boltzmann Equation	4
	B. Multigroup Formalism	5
	C. Angular Treatment	6
	D. Spatial Considerations	9
	1. Fine-Mesh Spatial Schemes	12
	2. Coarse-Mesh Schemes	13
	3. Nodal Methods	13
3.	REVIEW OF THE LITERATURE	15
4.	DERIVATION OF THE METHODS	20
	A. General Description	20
	1. Flat Leakage, Coupled 1-D Flux Shapes	20
	2. Calculated Leakage Shape, Separable 1-D Within-Node Fluxes	21
	3. Calculated Leakage Shape, 2-D Within-Node Flux Shape	21
	B. Detailed Derivations of the Three Nodal Methods	22
	1. Method 1: Flat Leakage, Coupled 1-D Flux Shape	22
	2. Method 2: Calculated Leakage Shape, Separable 1-D Flux Shape	24
	3. Method 3: Calculated Leakage Shape, 2-D Flux Shape	29
	C. Iterative Solution for the Coefficients	30
	D. Coarse-Mesh Acceleration	31
5.	SAMPLE PROBLEMS	34
	A. Introduction	34
	B. Sample Problem 1: Shielding Benchmark Problem	34
	C. Sample Problem 2: Simple Reactor Problem	47
	D. Sample Problem 3: BWR Lattice Eigenvalue Problem	52

TABLE OF CONTENTS (continued)

CHAPTER			PAGE
	E.	Sample Problem 4: Two-Group Volumetric Source Borehole Shielding Problem.	59
	F.	Sample Problem 5: Five-Group Volumetric Source Borehole Shielding Problem.	67
	G.	Sample Problem 6: Realistic Boundary Source Shielding Problem.	73
6.	CONCLUSIONS		81
7.	RECOMMENDATIONS FOR FUTURE WORK		83
	A.	Optimization of the Cartesian x-y Nodal Method.	83
	B.	Extension to Other Geometries and to Time Dependence. . .	85
		1. Curvilinear Geometry.	85
		2. Three-Dimensional Geometry.	87
		3. Time Dependence	88
LIST OF REFERENCES			89
APPENDICES.			91
	A.	DEVELOPMENT OF THE EXPONENTIAL EXPANSION METHOD IN ONE DIMENSION	93
	B.	DEVELOPMENT OF ANISOTROPIC SCATTERING TREATMENT	97
	C.	COMPARISON OF EXPONENTIAL EXPANSIONS VERSUS LEGENDRE POLYNOMIALS IN REPRESENTING Exp(-ax),0<a<5.	99
	D.	COMPARISON OF OPERATION COUNT FOR THE INNER ITERATION OF A MODAL EXPANSION CHANNEL METHOD FOR EXPONENTIAL VERSUS ORTHONORMAL BASIS FUNCTION SETS.	101
	E.	PROJECTION OF EXPONENTIAL FUNCTIONS ONTO THE EXPONENTIAL BASIS FUNCTION SETS	104
	F.	USER'S MANUAL FOR EXTREME	112
	G.	SAMPLE INPUT AND OUTPUT FOR EXTREME	127
	H.	USE OF A B_m CALCULATION TO CHOOSE MODES FOR EXTREME . . .	153

LIST OF TABLES

TABLE		PAGE
1.	Data for Sample Problem 1	37
2.	Listing of S_8 Quadrature	38
3.	Results of DOT4.2 and EXTREME Calculations of Leakage in Sample Problem 1	39
4.	Total System Leakage for Sample Problem 1 for Various Choices of α in Mode Set 0, $-\alpha$, α	43
5.	EXTREME Method 3 Results for Sample Problem 1 Using N=1,3,5 Mode Sets Chosen by B_0 Calculation.	45
6.	Data for Sample Problem 2	49
7.	Results of DOT4.2 and EXTREME Calculations of Eigenvalue for Sample Problem 2.	50
8.	Results for 16 Node Mode of Sample Problem 2 for Various Choices Mode Set 0, $-\alpha$, α in EXTREME.	54
9.	Data for Sample Problem 3	57
10.	Results of DOT4.2 and EXTREME Calculations of Eigenvalue for Sample Problem 3.	58
11.	Data for Sample Problem 4	63
12.	Listing of S_6 Quadrature.	64
13.	Results of DOT4.2 and EXTREME Calculations of Absorption in Detector Region of Sample Problem 4	65
14.	Data for Sample Problem 5	69
15.	Results of DOT4.2 and EXTREME Calculations of Absorption in Detector Region of Sample Problem 5	71
16.	Data for Sample Problem 6	76
17.	Results of DOT4.2 and EXTREME Calculations of Leakage out the Top of Sample Problem 6	77
A-1.	Data for 1-D Slab Problem	95
A-2.	Comparison of the 1-D Method with ANISN For the Sample Shielding Problem	96

LIST OF FIGURES

FIGURE		PAGE
1.	Boundary and Averaged Fluxes of Node i,j for μ, $\eta>0$	10
2.	Regions of Influence for Boundary Fluxes on "Upstream" Edges.	28
3.	Geometry of Sample Problem 1.	35
4.	Comparison of Percentage Error in Leakage for EXTREME versus DOT4.2 for Sample Problem 1	40
5.	Plot of the Flux Shape on Right Edge of Sample Problem 1 as Calculated by DOT4.2 and EXTREME.	41
6.	Plot of Total Leakage in Sample Problem 1 versus Choice of α in Mode Set 0, $-\alpha$, α for the Three Methods in EXTREME	44
7.	Plot of Percentage Error in Sample Problem 1 versus Computing Time for 1, 3, 5 Order Mode Sets	46
8.	Geometry of Sample Problem 2.	48
9.	Comparison of Percentage Error in Eigenvalue for EXTREME versus DOT4.2 for Sample Problem 2.	51
10.	Ratios of the Group 2 Node Averaged Fluxes in Each Region of Sample Problem 2 as Calculated by EXTREME and DOT4.2	53
11.	Comparison of Effect of Mode Choice on Calculation of Sample Problem 2 Eigenvalues using EXTREME.	55
12.	Geometry of Sample Problem 3.	56
13.	Comparison of Percentage Error in Eigenvalue for DOT4.2 versus EXTREME in Sample Problem 3.	60
14.	Ratios of the Group 2 Node Averaged Fluxes in Each Region of Sample Problem 3 as Calculated by EXTREME and DOT4.2 . . .	61
15.	Geometry of Sample Problems 4 and 5	62
16.	Comparison of Percentage Error in Detector Absorption for EXTREME versus DOT4.2 in Sample Problem 4	66
17.	Plot of the Flux along the Longitudinal Axis of Detector in Sample Problem 4 as Calculated by EXTREME and DOT4.2 . . .	68
18.	Comparison of Percentage Error in Detector Absorption for EXTREME versus DOT4.2 in Sample Problem 5	72

19. Plot of the Flux along the Longitudinal Axis of Detector in Sample Problem 5 as Calculated by EXTREME and DOT4.2 . . . 74

20. Geometry of Sample Problem 6. 75

21. Comparison of Percentage Error in Top Leakage for EXTREME versus DOT4.2 in Sample Problem 6 78

22. Plot of the Flux along the Top Edge of Sample Problem 6 as Calculated by EXTREME and DOT4.2 80

C-1. Comparison of Least Square Error Norm for Exponential Expansions versus Algebraic Expansions in Representing Exp(-ax) for 0<A<5. 100

E-1. Approximation of a Continuous Exponential Function Using an Exponential Expansion. 109

E-2. Approximation of a Discontinous Exponential Function Using an Exponential Expansion. 110

G-1. Sample Input and Output from EXTREME. 128

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation for the support and encouragement of Dr. H. Lee Dodds, Jr., who suggested this project and served as the author's major professor, and the University of Tennessee staff members who served on the Graduate Committee. The author is also grateful for the many interesting discussions and suggestions contributed by Dr. R. A. Little, Mr. W. A. Rhoades, and Dr. M. L. Williams of Oak Ridge National Laboratory. Special thanks are extended to Mr. J. O. Johnson and Mr. B. L. Broadhead for many fruitful discussions and for their help with the manuscript.

Appreciation is also expressed to the Computer Sciences and Engineering Physics Divisions of the Oak Ridge National Laboratory for providing partial funding of this work.

ABSTRACT

The purpose of this work is the development and testing of a new family of methods for calculating the spatial dependence of the neutron density in nuclear systems described in two-dimensional Cartesian geometry. The energy and angular dependence of the neutron density is approximated using the multigroup and discrete ordinates techniques, respectively. The resulting FORTRAN computer code is designed to handle an arbitrary number of spatial, energy, and angle subdivisions. Any degree of scattering anisotropy can be handled by the code for either external source or fission systems.

The basic approach is to (1) approximate the spatial variation of the neutron source across each spatial subdivision as an expansion in terms of a user-supplied set of exponential basis functions; (2) solve analytically for the resulting neutron density inside each region; and (3) approximate this density in the basis function space in order to calculate the next iteration flux-dependent source terms. In the general case the calculation is iterative due to neutron sources which depend on the neutron density itself, such as scattering interactions.

The three methods which were developed differ in the detail of the spatial description:

1. The first method expands the two-dimensional intranode neutron flux as two separable one-dimensional expansions in the x- and y-dimensions and represents the edge fluxes as constant;
2. The second method is the same as the first in the interior of each node, but represents the edge fluxes as one-dimensional expansions in the basis function set; and

3. The third method is the same as the second on the edges, but represents the interior flux shape in a full two-dimensional expansion in the x- and y-dependent basis functions.

In order to test the accuracy versus computer time of the three methods, five sample problems were run and the results compared with those of the finite-difference code DOT4.2. Three shielding problems were run: a simple "benchmark" calculation, a bore-hole geometry volumetric source problem solved with two and five energy groups, and a boundary source large void shielding problem. In addition, two eigenvalue problems, a simple benchmark reactor calculation and a boiling water reactor core lattice case, were run and again compared with DOT4.2. In all of these cases, the first method showed similar accuracy/cost characteristics as DOT4.2, with the second and third methods performing significantly better than the finite difference code.

The major conclusion of the study is that the new exponential expansion methods show promise of reducing the cost of accurately calculating the neutron density inside nuclear systems. Future research work suggested by the present study include:

1. Extension to other geometric systems, such as two-dimensional curvilinear and three-dimensional Cartesian geometries,
2. Extension to handle time-dependent problems,
3. Optimization studies on the choice of basis functions, and
4. Development of acceleration schemes tailored to nodal methods.

CHAPTER 1

INTRODUCTION

The central problem of nuclear reactor analysis has been simply stated[1] as the determination of the density of the free neutron population in an extended region of space containing an arbitrary, but known, mixture of materials. Once the neutron population density is determined, a nuclear analyst can use it along with the basic data of neutron physics to determine quantities of practical interest to engineers and scientists: reactor power densities, energy deposition rates in samples, secondary nuclide production rates, etc.

The primary tools of nuclear analysts are generalized computer programs[2,3] designed to solve the basic neutron transport equation for the neutron density as a function of space, energy, angle, and time. Unfortunately, this complicated phase space dependence tends to make the computer codes expensive to run for realistic reactor and shielding problems even on the present generation of large, fast digital computers. The large amount of time and money people spend on such analyses justifies a considerable research effort to attempt to reduce the costs of the various analysis methods without significantly sacrificing accuracy.

The most successful effort to reduce the costs of nuclear analysis has been in the area of diffusion theory methods. In diffusion theory, the angular dependence of the neutron density is linearly approximated and the angle-averaged density is determined. Diffusion theory has proven to be very useful for analyzing systems for which

the angular variation of the neutron density is not very extreme (i.e. the density is nearly isotropic in direction). Unfortunately, for a large class of problems the directional variation of the neutron density is not so smooth (e.g., streaming problems, strong absorption problems), and the angular dependence must be retained. Such angle-dependent methods are generally referred to as transport theory methods to be distinguished from diffusion theory methods.

In the last decade or so some success has been reported in reducing costs of both diffusion theory and transport theory calculation with new treatments of the spatial dependence of the neutron density[4-9]. The basic idea in these treatments is to replace the very fine finite-difference spatial meshes of most analysis methods with a coarser mesh structure in which the spatial dependence inside each mesh segment ("node") is approximated more accurately than in the fine mesh methods. The use of this approach for diffusion theory has been widely tested in the last fifteen years and has proven to reduce the costs of calculations[4,5,6]. However, the application of the idea to transport methods has only recently been attempted[7,8,9]. Although the experience with transport nodal methods has been limited to geometrically simple, isotropic scattering "benchmark" problems, the results indicate that their use may result in substantial computing cost savings.

A. Project Objectives

In summary, the objectives of the present work are to:

1. develop a family of nodal discrete ordinates transport methods for Cartesian (x,y) geometry using a modal spatial expansion on an exponential basis function set,

2. extend the methods for use in multigroup and anisotropic problems, and
3. evaluate the new methods for realistic reactor eigenvalue problems and fixed source shielding problems by comparing them with the widely used conventional finite difference code DOT4.2 and other recently developed nodal transport methods.

CHAPTER 2

BACKGROUND

A. Boltzmann Equation

Generalized neutron transport methods solve the Boltzmann transport equation, which can be derived by conserving neutrons within a differential volume of energy-angle-space-time phase space. This derivation results in an integrodifferential equation of the neutron angular flux density $\psi(\bar{r},\bar{\Omega},E,t)$ at a particular point $\bar{r}$, as a function of direction of travel $\bar{\Omega}$, particle energy E, and time t. In two-dimensional Cartesian geometry the equation is[2]

$$\frac{1}{v(E)}\frac{\partial\psi(x,y,E,\bar{\Omega},t)}{\partial t} + \mu\frac{\partial\psi(x,y,E,\bar{\Omega},t)}{\partial x} + \eta\frac{\partial\psi(x,y,E,\bar{\Omega},t)}{\partial y} \qquad (2.1)$$

$$+ \sigma_t(x,y,E,t)\psi(x,y,E,\bar{\Omega},t) = Q(x,y,E,\bar{\Omega},t) +$$

$$\int d\bar{\Omega}'\int dE'\{\sigma_s(x,y,(E',\bar{\Omega}')\rightarrow(E,\bar{\Omega}),t) + \nu(E')\sigma_f(x,y,E',t)\chi(x,y,E,t)\}$$

$$\cdot\psi(x,y,E',\bar{\Omega}',t),$$

where

$\psi(x,y,E,\bar{\Omega},t) = v(E)\cdot N(x,y,E,\bar{\Omega},t)$,

$v(E)$ = neutron velocity,

$N(x,y,E,\bar{\Omega},t)$ = neutron population density per unit volume per unit steradian per unit energy,

(μ,η,ξ) = direction cosines for angle $\bar{\Omega}$,

$\sigma_t(x,y,E,t)$ = total macroscopic cross section,

$Q(x,y,E,\bar{\Omega},t)$ = external source,

$\sigma_s(x,y,(E',\bar{\Omega}')\rightarrow(E,\bar{\Omega}),t)$ = differential scattering cross section,

$\sigma_f(x,y,E,t)$ = fission cross section,

$\nu(E)$ = average number of neutrons emitted per fission,

and $\chi(x,y,E,t)$ = fission neutron spectrum.

The method is directed at the static (time-independent) Boltzmann equation:

$$\mu\frac{\partial\psi(x,y,E,\bar{\Omega})}{\partial x} + \eta\frac{\partial\psi(x,y,E,\bar{\Omega})}{\partial y} + \sigma_t(x,y,E)\psi(x,y,E,\bar{\Omega}) = \tag{2.2}$$
$$Q(x,y,E,\bar{\Omega}) + \int d\bar{\Omega}'\int dE'\{\sigma_s(x,y,(E',\bar{\Omega}')\rightarrow(E,\bar{\Omega}))$$
$$+ \nu\sigma_f(x,y,E')\chi(x,y,E)\}.\psi(x,y,E',\bar{\Omega}')$$

with the same definitions as above.

In order to solve for the flux on a digital computer, it is necessary to reduce Eq. 2.2 to a set of coupled algebraic equations. For this, the continuous dimensions of energy, angle, and space must be represented in a discrete manner.

B. Multigroup Formalism

The energy variable of the Boltzmann equation is represented in a straightforward manner using the multigroup approach. In this approach, Eq. 2.2 is integrated separately over a finite number of energy bands ("groups"). For a particular group (E_g, E_{g+1}) this gives

$$\mu\frac{\partial\psi_g(x,y,\bar{\Omega})}{\partial x} + \eta\frac{\partial\psi_g(x,y,\bar{\Omega})}{\partial y} + \sigma_{tg}(x,y,\bar{\Omega})\psi_g(x,y,\bar{\Omega}) = \tag{2.3a}$$
$$Q_g(x,y,\bar{\Omega}) + \sum_{g'}\int d\bar{\Omega}'\{\sigma_{sg'\rightarrow g}(x,y,\bar{\Omega}'\rightarrow\bar{\Omega})$$
$$+ \nu_{g'}\sigma_{fg'}(x,y,\bar{\Omega}')\chi_g(x,y)\}.\psi_{g'}(x,y,\bar{\Omega}'), \quad g=1,2,\ldots G;$$

where

$$\psi_g(x,y,\bar{\Omega}) = \int_{\Delta E} \psi(x,y,\bar{\Omega},E)dE, \tag{2.3b}$$

$$\sigma_{tg}(x,y,\bar{\Omega}) = \frac{\int_{\Delta E} \sigma_t(x,y,E)\psi(x,y,\bar{\Omega},E)dE}{\psi_g(x,y,\bar{\Omega})} \tag{2.3c}$$

$$\nu_g\sigma_{fg}(x,y,\bar{\Omega}) = \frac{\int_{\Delta E} \nu(E)\sigma_f(x,y,E)\psi(x,y,\bar{\Omega},E)dE}{\psi_g(x,y,\bar{\Omega})} \tag{2.3d}$$

$$Q_g(x,y,\bar{\Omega}) = \int_{\Delta E} Q(x,y,\bar{\Omega},E)dE, \tag{2.3e}$$

$$\sigma_{sg'\to g}(x,y,\bar{\Omega}'\to\bar{\Omega}) = \int_{\Delta E}\int_{\Delta E'} \sigma_s(x,y,(E',\bar{\Omega}')\to(E,\bar{\Omega})) \cdot\psi(x,y,\bar{\Omega}',E')dE'dE \;/\; \psi_g(x,y,\bar{\Omega}), \text{ and} \tag{2.3f}$$

$$\chi_g(x,y) = \int_{\Delta E} \chi(x,y,E)dE. \tag{2.3g}$$

Of course the above definitions for these "average" parameters are formally correct, but unattainable in practice without a priori knowledge of $\psi(x,y,\bar{\Omega},E)$, which is not generally available. The determination of adequate data with the use of approximations to the shape of $\psi(x,y,\bar{\Omega},E)$ across each energy group is a discipline in itself, which will not be pursued here. For the purposes of this work, the assumption will be that made that the multigroup data above have been determined using one of the available methods[13,14].

C. Angular Treatment[15,16]

The treatment of the angular dependence of Eq. 2.3 involves approximating the angular integral of $\psi(x,y,\bar{\Omega})$ (referred to as the scalar flux, $\phi(x,y)$) as a weighted sum of the values of $\psi(x,y,\bar{\Omega}_m)$ on an angular quadrature set $\bar{\Omega}_m$, $m=1,2,\ldots,M$, i.e.

$$\phi(x,y) = \int_{\Omega} \psi(x,y,\bar{\Omega})d\bar{\Omega} \approx \sum_{m=1}^{M} w_m\psi_m(x,y) \;, \tag{2.4}$$

where $\psi_m(x,y) = \psi(x,y,\bar{\Omega}_m)$, and

M = number of directions in the quadrature,

w_m = weight of angle m (associated with an area on the angular unit sphere).

Usually the angular quadrature is chosen to cover the angular unit sphere nearly evenly; in this case all of the weights, w_m, are approximately equal (i.e., the distribution is close to isotropic). Also, the quadrature angles are usually chosen to be symmetrical about the axes of the problem. It is not uncommon, however, for quadrature sets to be designed especially for a particular problem; in such cases there are more angles "representing" directions which are important for that application (e.g., neutron streaming paths through holes in a shield).

Regardless of how an angular quadrature is chosen, the resulting problem is the same; the values of $\psi_m(x,y)$ along the discrete ordinates $\bar{\Omega}_m$ must be determined. The equation for the $\psi_m(x,y)$ is, of course, Eq. 2.3 evaluated for a particular direction $\bar{\Omega}_m = (\mu_m, \eta_m, \xi_m)$:

$$\mu_m \frac{\partial \psi_{mg}(x,y)}{\partial x} + \eta_m \frac{\partial \psi_{mg}(x,y)}{\partial y} + \sigma_{tg}(x,y)\psi_{mg}(x,y) \tag{2.5}$$

$$= Q_{mg}(x,y) + \sum_{g'} \sum_{m'} \{\sigma_{sg'\to g, m'\to m}(x,y)$$

$$+ \nu_{g'}\sigma_{fg'}(x,y)\chi_g(x,y)\}\cdot\psi_{m'g'}(x,y); \quad g=1,2,\ldots G, \text{ and}$$

$$m=1,2,\ldots M.$$

After approximating the spatial variables in Eq. 2.5, which is discussed in detail in the next section, Eq. 2.5 is solved using an iterative procedure:

$$\mu_m \frac{\partial \psi^n_{mg}(x,y)}{\partial x} + \eta_m \frac{\partial \psi^n_{mg}(x,y)}{\partial y} + \sigma_{tg}(x,y)\psi^n_{mg}(x,y) = S^n_{mg}(x,y) \qquad (2.6a)$$

$$S^n_{mg}(x,y) = T^n_{mg}(x,y) + \sum_{g'=1}^{g} \sum_{m'}^{M} w_{m'} \sigma_{sg'\to g,m'\to m}(x,y)\psi^n_{m'g'}(x,y) \qquad (2.6b)$$

$$T^n_{mg}(x,y) = Q_{mg}(x,y) + \sum_{g'=g+1}^{G} \sum_{m}^{M} w_{m'}\{\sigma_{sg'\to g,m'\to m}(x,y)\cdot\psi^{n-1}_{m'g'}(x,y)\} \qquad (2.6c)$$

$$+ \sum_{g'=1}^{G} \nu_{g'}\sigma_{fg'}(x,y)\chi_g(x,y)\cdot\phi^{n-1}_{g'}(x,y) \quad .$$

In this double iteration strategy, the ψ^n_{mg} are determined by iterating on the within-group scattering using Eqs. 2.6a and 2.6b ("inner iteration"). Then the ψ^n_{mg} are used to determine the source terms, T^{n+1}_{mg}, for the $n+1^{st}$ "outer" iteration, which includes fission and upscatter sources. (The value "n" here counts outer iterations.)

Also, the cross-section scattering data are usually not given in the form $\sigma_{sg'\to g,m'\to m}$, but as scattering Legendre coefficients, $\sigma^k_{sg'\to g}$, where

$$\sigma_{sg'\to g,m'\to m} = \sigma_{sg'\to g}(\bar{\Omega}_{m'}\cdot\bar{\Omega}_m) = \sum_{k}^{K} \sigma^k_{sg'\to g} P_k(\bar{\Omega}_{m'}\cdot\bar{\Omega}_m) \quad .$$

Appendix A shows that using this relation in Eqs. 2.6 results in (omitting the iteration index):

$$\mu_m \frac{\partial \psi_{mg}(x,y)}{\partial x} + \eta_m \frac{\partial \psi_{mg}(x,y)}{\partial y} + \sigma_{tg}(x,y)\psi_{mg}(x,y) = S_{mg}(x,y) \qquad (2.7a)$$

where

$$S_{mg}(x,y) = T_{mg}(x,y) + \sum_{g'=1}^{g} \sum_{k'}^{K} \sigma_{sg'\to g}^{k'} \sum_{r}^{k'} f_{mr}^{k'} \phi_{g'r}^{k'}(x,y) \qquad (2.7b)$$

$$T_{mg}(x,y) = Q_{mg}(x,y) + \sum_{g'=g+1}^{G} \sum_{k'}^{K} \sigma_{sg'\to g}^{k'} \cdot \sum_{r}^{k'} f_{mr}^{k'} \phi_{g'r}^{k'}(x,y)$$

$$+ \sum_{g'=1}^{G} \nu_{g'} \sigma_{fg'}(x,y) \chi_g(x,y) \} \cdot \phi_{g'0}^{0}(x,y)$$

and

$$\phi_{gr}^{k'}(x,y) = \sum_{m'=0}^{M} w_m f_{mr}^{k'} \psi_{mg}(x,y) \qquad (2.8)$$

$$f_{mr}^{k'} = P_{k'}(\eta_m) \quad \text{if } r=0, \text{ and otherwise}$$

$$= \left\{\frac{2(k'-r)!}{(k'+r)!}\right\}^{1/2} P_{k'}(\eta_m)\cos(r\theta),$$

$$\theta = \tan^{-1}(\eta_m/\xi_m) \ .$$

D. Spatial Considerations

All that remains in reducing Eq. 2.3 to a form that can be programmed for a digital computer is to reduce the spatial dependence of Eq. 2.5 as has been done for energy and angle. This can be accomplished by returning to the idea of the multigroup energy reduction and integrating Eq. 2.5 over discrete regions of space. The regions ("nodes") are determined by laying an x-y grid over the problem geometry, as shown in Fig. 1, integrating Eq. 2.5 over each node $i:(x_j,x_{j+1})*(y_j,y_{j+1})$, and dividing by $\Delta x \Delta y$ to obtain:

ORNL-DWG 82-13805

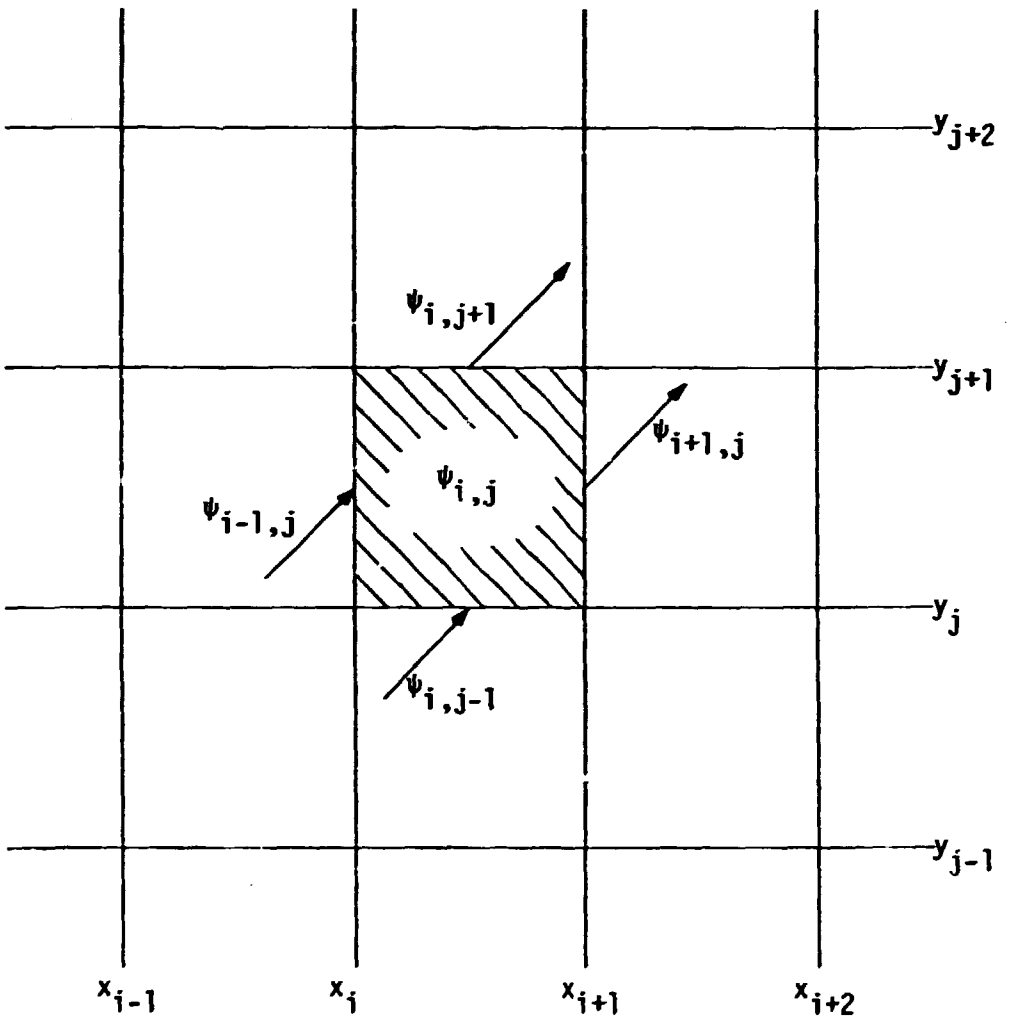


Figure 1. Boundary and Averaged Fluxes of Node i,j for $\mu,\eta>0$

$$\frac{\mu_m}{\Delta x}(\psi_{mg,i+1}-\psi_{mg,i}) + \frac{\eta_m}{\Delta y}(\psi_{mg,j+1}-\psi_{mg,j}) + \sigma_{tg}\psi_{gm} = S_{gm} \qquad (2.9)$$

where

$$\psi_{mg} = \int_{\Delta x}\int_{\Delta y}\psi_{mg}(x,y)dydx/\Delta x\Delta y \ ,$$

$$\psi_{mg,i} = \int_{\Delta y}\psi_{mg}(x_i,y)dy/\Delta x\Delta y \ ,$$

$$\psi_{mg,i+1} = \int_{\Delta y}\psi_{mg}(x_{i+1},y)dy/\Delta x\Delta y \ ,$$

$$\psi_{mg,j} = \int_{\Delta x}\psi_{mg}(x,y_j)dx/\Delta x\Delta y \ ,$$

$$\psi_{mg,j+1} = \int_{\Delta x}\psi_{mg}(x,y_{j+1})dx/\Delta x\Delta y, \text{ and}$$

$$S_{mg} = \int_{\Delta x}\int_{\Delta y}S_{mg}(x,y)dydx/\Delta x\Delta y.$$

It can be seen immediately that the derivatives of Eq. 2.5 have introduced boundary values (by use of the divergence theorem) when integrated; instead of one flux value, Eq. 2.9 has five unknowns: ψ_{mg}, $\psi_{mg,i}$, $\psi_{mg,i+1}$, $\psi_{mg,j}$, and $\psi_{mg,j+1}$.

These equations are solved by a "sweeping" algorithm in which, for each energy group and angle, the spatial nodes are taken one at a time in a particular order. The order is chosen to follow the general direction of neutron travel for that angle; i.e., +x for $\mu>0$, -x for $\mu<0$, +y for $\eta>0$, and -y for $\eta<0$. In this sweeping procedure, when the calculation gets to any node (i,j), the "incoming" boundary terms have already been calculated as "outgoing" fluxes of the "upstream" neighbor node; therefore, two of the five flux values in the node have already been determined. This point is illustrated in Fig. 1 for the case $\mu>0$, $\eta>0$, for which the values $\psi_{mg,i}$ and $\psi_{mg,j}$ are known from previous calculations for nodes (x_{i-1},y_j) and (x_i,y_{j-1}).

Even with the number of unknowns reduced, there are still three unknowns with only one equation to relate them; two more equations are needed. The differences between the many different spatial treatments that have been developed over the years is in the determination of the other two auxiliary relations.

1. Fine-Mesh Spatial Schemes[2,3]

The most commonly used spatial treatments fall into a category called fine-mesh schemes. These methods use mathematical constraints for the auxiliary relations; they are the simplest, fastest (per node) and most flexible methods in use. They are also expensive, in general, because they require many nodes to get accurate flux values.

The constraints which are most often used are of the form

$$\psi_{mg} = a\psi_{mg,i} + b\psi_{mg,i+1} \quad \text{and}$$

$$\psi_{mg} = c\psi_{mg,j} + d\psi_{mg,j+1},$$

which is termed the "weighted difference" formulation. The special case of a=b=c=d=0.5 is referred to as the "diamond difference" approximation; the case a=c=0, b=d=1 is the "step function" approximation. The problem with this approach using pre-determined values of a, b, c, and d is that there is no reason to think that the constraint will be obeyed for finite nodes of a physical system. It is true that any such constraint with a+b=c+d=1 will hold in the limit of infinitely small nodes, but in practice the number of nodes needed for an accurate determination of ψ is much greater than for methods using more accurate auxiliary relations.

2. Coarse-Mesh Schemes[11,12]

Transport theory procedures that use more accurate relations between the flux values without performing intranode flux calculations are referred to as "coarse-mesh" methods. The usual procedure is to

1. Assume a flux shape within each node based on the centered and edge values, e.g.,

$$\psi_{mg}(x,y) = \psi_{mg} \qquad \text{(flat)}$$

or

$$= \psi_{mg} + (\psi_{mg,i+1} - \psi_{mg,i})(\Delta x/2 - x) + (\psi_{mg,j+1} - \psi_{mg,j})(\Delta y/2 - y) \qquad \text{(linear)},$$

2. Form the shaped sources from these fluxes, and

3. Solve for the outgoing fluxes with these sources.

The key word in the previous sentence was "solve"; since the auxiliary relations are based on solutions to Eq. 2.5 rather than constraints which are only true in the limit of zero width nodes, these coarse-mesh methods have been able to obtain more accurate fluxes with much larger nodes than are needed by the fine-mesh methods. The successes of these coarse-mesh methods are rather surprising given the fact that they use no more information than the fine-mesh methods; they succeed by simply assuming flux and source shapes based on the information at hand.

3. Nodal Methods[7-10]

The methods classed as nodal methods go one step beyond the coarse-mesh methods and solve for the within-node flux shapes as sub-problems introducing new unknowns. These within-node calculations are

performed with a different (usually simpler) spatial treatment than the problem as a whole. Although this extra trouble of solving within-node calculations generally causes the nodal methods to require much more computer work per spatial node, the increased accuracy of the auxiliary relation allows for fewer, larger nodes to be used in the calculations.

The transport nodal methods which have been developed have shown a considerable time saving relative to conventional finite difference methods. Unfortunately, their experience has been limited to relatively simple isotropic scattering "benchmark" problems. The development described herein extends the experience with transport nodal methods by calculating "realistic" anisotropic problems as well as benchmark problems.

CHAPTER 3

REVIEW OF THE LITERATURE

Although there has been a considerable body of work in nodal solutions to the diffusion equation[4-6], transport nodal methods have been developed only recently.

In 1979 Wagner[7] reported a method which used two one-dimensional calculations resulting from integrating Eq. 2.5 separately over (x_i, x_{i+1}) and over (y_j, y_{j+1}). The resulting "channel" equations for the x- and y-averaged nodal fluxes were then averaged over the transverse angle to obtain the equations:

$$\mu_m \frac{d\psi^y_{gm}(x)}{dx} + \sigma_{tg}\psi^y_{gm}(x) = S^y_{gm}(x) - L^y(x) \quad \text{and} \tag{3.1}$$

$$\eta_m \frac{d\psi^x_{gm}(y)}{dy} + \sigma_{tg}\psi^x_{gm}(y) = S^x_{gm}(y) - L^x(y) \ , \tag{3.2}$$

where $L^v(u)$ is the u-dependent leakage through the transverse v-direction boundaries.

By assuming isotropic current out of the "sides" of each node, and a constant spatial shape for the transverse leakage, Wagner obtained the coupling conditions:

$$L^y(x) = \frac{1}{\Delta y}\left(J^{+y}(\Delta y) - J^{-y}(\Delta y) + J^{-y}(0) - J^{+y}(0)\right) \quad \text{and} \tag{3.3}$$

$$L^x(y) = \frac{1}{\Delta x}\left(J^{+x}(\Delta x) - J^{-x}(\Delta x) + J^{-x}(0) - J^{+x}(0)\right) \ , \tag{3.4}$$

where the J's denote the partial current out of the boundary faces;

for example,

$$J^{+y}(\Delta y) = \sum_{\eta_m>0}^{M} w_{m'}\eta_m \psi_m^x(\Delta y) \quad .$$

Wagner solved the above equations by using a 1-D "sub-grid" for the within-node "channel" calculations and utilizing his modified block inversion method[17]. The results of this method for several sample isotropic scattering reactor (eigenvalue) problems showed a considerable saving in computing time relative to TWOTRAN[3] to obtain answers accurate to about 1%. Wagner noted in the paper that the most serious drawback of his method was the flat leakage approximation; he tried to incorporate a leakage shape by fitting a quadratic to the leakages of the node and its closest neighbors, but the results of this improvement were inconclusive.

Lawrence and Dorning[8] have developed a method which also uses coupled one-dimensional calculations. By again integrating Eq. 2.5 over Δx and Δy, but keeping the full 2-D angular treatment, they obtained:

$$\mu_m \frac{d\psi_{gm}^y(x)}{dx} + \sigma_{tg}\psi_{gm}^y(x) = S_{gm}^y(x) - \frac{\eta_m}{\Delta y}\left(\psi_{gm}(x,\Delta y)-\psi_{gm}(x,0)\right) \quad , \qquad (3.5)$$

$$\eta_m \frac{d\psi_{gm}^x(y)}{dy} + \sigma_{tg}\psi_{gm}^x(y) = S_{gm}^x(y) - \frac{\mu_m}{\Delta x}\left(\psi_{gm}(\Delta x,y)-\psi_{gm}(0,y)\right) \quad . \qquad (3.6)$$

By assuming a constant dependence of $\psi_x(y)$ and $\psi_y(x)$, i.e.

$$\psi_{gm}(x,y_{ext}) = \psi_{gm}^x(y_{ext}) \quad , \quad y_{ext} = 0, \Delta y \quad \text{and} \qquad (3.7)$$

$$\psi_{gm}(x_{ext},y) = \psi^{y}_{gm}(x_{ext}) \ , \quad x_{ext} = 0, \Delta x \ , \tag{3.8}$$

they obtained the coupled set

$$\mu_m \frac{d\psi^{y}_{gm}(x)}{dx} + \sigma_{tg}\psi^{y}_{gm}(x) = S^{y}_{gm}(x) - \frac{\eta_m}{\Delta y}\left(\psi^{x}_{gm}(\Delta y) - \psi^{x}_{gm}(0)\right) \ , \tag{3.9}$$

$$\eta_m \frac{d\psi^{x}_{gm}(y)}{dy} + \sigma_{tg}\psi^{x}_{gm}(y) = S^{x}_{gm}(y) - \frac{\mu_m}{\Delta x}\left(\psi^{y}_{gm}(\Delta x) - \psi^{y}_{gm}(0)\right) \ . \tag{3.10}$$

They solved the above equations by using Legendre function expansions for the 1-D spatial dependence. They compared their method with the fine-mesh transport code TWOTRAN for several simple isotropic scattering reactor (eigenvalue) and shielding problems. Again the nodal method showed a considerable cost savings relative to the fine-mesh solution. Also, as in Wagner's paper, Lawrence and Dorning point to the flat leakage shape approximation as the most serious weakness of the method.

Walters and O'Dell[9] have also developed a nodal method based on algebraic (x^n) spatial expansions of the flux and source terms. In their approach, nodal methods with various combinations of flat and linear representations of the interior and edge fluxes were compared with the diamond difference procedure in TWOTRAN for Cartesian geometry systems.

For their linear interior flux treatment, for example, the source was approximated by the relation

$$S(x,y) = S_{av} + S_x * 2(x-x_i)/\Delta x + S_y * 2(y-y_i)/\Delta y$$

with (x_i, y_i) = coordinates of the node center. As in Lawrence's approach, Walters solves for the transverse integrated channel fluxes within the node, e.g.

$$\eta\frac{d\psi^0(y)}{dy} + \sigma_t\psi^0(y) = S_{av} + S_y^*2(y-y_j)/\Delta y - \frac{\mu}{\Delta x}[\psi(x_R,y)-\psi(x_L,y)]$$

$$\text{where } \psi^0(y) = \frac{1}{\Delta x}\int_{x_L}^{x_R}\psi(x,y)dx.$$

In addition, however, he solves for the transverse integrated <u>first moment</u>, $\psi^1(y)$:

$$\eta\frac{d\psi^1(y)}{dy} + \sigma_t\psi^1(y) = S_x + (\frac{6y}{\Delta x})^*\{\psi^0(y)-0.5[\psi(x_R,y)+\psi(x_L,y)]\}$$

$$\text{where } \psi^1(y) = \frac{6}{\Delta x^2}*\int_{x_L}^{x_R}(x-x_i)\psi(x,y)dx \ .$$

Evaluation of $\psi^1(y)$ gives the transverse flux <u>slope</u> and thus permits a solution for the coefficients of the linear flux expansion with the four coupled equations for $\psi^0(x)$, $\psi^1(x)$, $\psi^0(y)$, and $\psi^1(y)$.

Of the various combinations of constant and linear functions Walters used for the edge and interior flux expansions, the techniques which used a syzygy for the expansion consistently outperformed the techniques which rely on a flat flux representation.

Larsen and Alcouffe[10] have developed a method which also uses a linear representation of the flux and source inside each spatial node. This method is just one of the many so-called "characteristic methods" which are described by Alcouffe in a review paper[18]. Instead of solving the transverse integrated channel equations, however, Larsen and Alcouffe solve analytically for the flux in the node interior and

on the outgoing edges, which are then represented with linear functions whose coefficients are determined by preserving the zero and first moments of the analytical solution. Their method has been compared with TWOTRAN for both shielding and eigenvalue problems and has shown considerable time savings in computing eigenvalues and system leakages.

Finally, finite element methods were used by Lillie[20] to solve the two-dimensional transport equation in triangular geometry. Although these techniques use flux expansions inside each spatial node, the coefficients for the expansions are determined by minimizing a global even-odd flux parity variational principle rather than by the mesh sweep procedure previously described. The requirements of inverting a large, relatively full banded matrix have thus far kept such methods relatively slow and expensive to run.

CHAPTER 4

DERIVATION OF THE METHODS

A. General Description

The work described in this thesis consisted of the development and testing of three separate nodal 2-D Cartesian spatial methods. The development in one-dimensional slab geometry is given in Appendix B.

1. Flat Leakage, Coupled 1-D Flux Shapes

This method follows Lawrence's structure, in which two coupled one-dimensional "channel" calculations are solved with the transverse leakage considered spatially flat. The only difference between this approach and Lawrence's is that the spatial expansions use an exponential function basis set, $\exp(\alpha_i x)$ instead of the Legendre polynomial set, $P_n(x)$. The use of exponential basis functions can be shown to offer two advantages over the Legendre polynomials:

a. The exponential basis functions can more accurately represent exponential functions. The spatial flux shape deep within homogeneous non-multiplying media can be shown by B_L theory[1] to approach an exponential; thus, the use of exponential basis functions offers an advantage for shielding calculations. This advantage is demonstrated in Appendix C.

b. The use of exponential expansion functions reduces the coupling between source modes and the resulting flux modes. Appendix D demonstrates that the operation count for a given node calculation with N spatial expansion modes is $5N$ compared to $N+2N^2$ when using an orthonormal basis function set.

2. Calculated Leakage Shape, Separable 1-D Within-Node Fluxes

In this refinement to the first method, the spatial shape of the flux on the node boundaries is also represented as expansions in the exponential basis function set. As before, the within-node flux shape is represented by the 1-D "channel" fluxes. The procedure followed consists of three distinct steps:

a. The characteristic problem of Eq. 2.5 is solved within each spatial node for the shape of the flux on the outgoing faces given:
   i. the known flux shape on the incoming faces and
   ii. the source (assumed separable in x and y) within the node obtained from previously calculated 1-D flux shapes;
b. These fluxes on the outgoing faces are approximated by expansions in the exponential basis functions; and
c. The expanded surface flux terms are used in the source terms of Eqs. 2.5 to solve for the within-node separable channel fluxes.

3. Calculated Leakage Shape, 2-D Within-Node Flux Shape

The refinement of the second method is carried one step further; the within-node flux is approximated as a two-dimensional modal expansion in the exponential basis function set. The procedure becomes:

a. The characteristic problem of Eq. 2.5 is solved for the spatial shape everywhere within the node given:
   i. the known flux shape on the incoming faces, and

ii. the 2-D modal expansion of the source within the node.

b. The fluxes on the outgoing faces are approximated by exponential expansions (for use as boundary values for the neighboring nodes).

c. The flux shape within the node is approximated by the 2-D exponential expansions and again used to determine the subsequent source terms.

B. Detailed Derivations of the Three Nodal Methods

1. Method 1: Flat Leakage, Coupled 1-D Flux Shape

Because of the similarity to Lawrence's method, the derivation begins with Eqs. 3.5 and 3.6:

$$\mu_m \frac{d\psi^y_{gm}(x)}{dx} + \sigma_{tg}\psi^y_{gm}(x) = S^y_{gm}(x) - \frac{\eta_m}{\Delta y}(\psi_{gm}(x,\Delta y)-\psi_{gm}(x,0)) \ , \qquad (3.5)$$

$$\eta_m \frac{d\psi^x_{gm}(y)}{dy} + \sigma_{tg}\psi^x_{gm}(y) = S^x_{gm}(y) - \frac{\mu_m}{\Delta x}(\psi_{gm}(\Delta x,y)-\psi_{gm}(0,y)) \ . \qquad (3.6)$$

The solutions to the above equations are, respectively, for $\mu_m, \eta_m > 0$:

$$\psi^y_{gm}(x) = \psi^y_{gm}(0)\exp(\frac{-\sigma_{tg}x}{\mu_m}) + \frac{\exp(\frac{-\sigma_{tg}x}{\mu_m})}{\mu_m}\int_0^x \{S^y_{gm}(x') \qquad (4.1)$$

$$- \frac{\eta_m}{\Delta y}(\psi^x_{gm}(\Delta y)-\psi^x_{gm}(0))\exp(\frac{\sigma_{tg}x'}{\mu_m})\}\,dx' \ , \text{ and}$$

$$\psi^x_{gm}(y) = \psi^x_{gm}(0)\exp\left(\frac{-\sigma_{tg}y}{\eta_m}\right) + \frac{\exp\left(\frac{-\sigma_{tg}y}{\eta_m}\right)}{\eta_m}\int_0^x \{S^x_{gm}(y') - \frac{\mu_m}{\Delta x}(\psi^y_{gm}(\Delta x)-\psi^y_{gm}(0))\exp\left(\frac{\sigma_{tg}y'}{\eta_m}\right)\}\,dy', \quad (4.2)$$

in which the approximations of Eqs. 3.7 and 3.8 have again been used.

Expanding in the 1-D sources $S^y_{gm}(x)$ and $S^x_{gm}(y)$ as exponential expansions; i.e.,

$$S^y_{gm}(x) = \sum_{n=1}^{N} S^{yn}_{gm}\exp(\alpha_n x), \quad (4.3)$$

$$S^x_{gm}(y) = \sum_{n=1}^{N} S^{xn}_{gm}\exp(\beta_n y), \quad (4.4)$$

and substituting into Eqs. 4.1 and 4.2 gives:

$$\psi^y_{gm}(x) = \left[\psi^y_{gm}(0) - \sum_{k=1}^{N}\frac{S^{yk}_{gm}}{\mu_m\alpha_k+\sigma_{tg}} + \frac{\eta_m}{\sigma_{tg}\Delta y}(\psi^x_{gm}(\Delta y)-\psi^x_{gm}(0))\right]\exp\left(\frac{-\sigma_{tg}x}{\mu_m}\right) \quad (4.5)$$

$$+ \sum_{n=1}^{N}\frac{S^{yn}_{gm}}{\mu_m\alpha_n+\sigma_{tg}}\exp(\alpha_n x) - \frac{\eta_m}{\sigma_{tg}\Delta y}(\psi^x_{gm}(\Delta y)-\psi^x_{gm}(0))$$

$$\psi^x_{gm}(y) = \left[\psi^x_{gm}(0) - \sum_{k=1}^{N}\frac{S^{xk}_{gm}}{\eta_m\beta_k+\sigma_{tg}} + \frac{\mu_m}{\sigma_{tg}\Delta x}(\psi^y_{gm}(\Delta x)-\psi^y_{gm}(0))\right]\exp\left(\frac{-\sigma_{tg}y}{\eta_m}\right) \quad (4.6)$$

$$+ \sum_{n=1}^{N}\frac{S^{xn}_{gm}}{\eta_m\beta_n+\sigma_{tg}}\exp(\beta_n y) - \frac{\mu_m}{\sigma_{tg}\Delta x}(\psi^y_{gm}(\Delta x)-\psi^y_{gm}(0)).$$

Using the method outlined in Appendix E to project the functions $\exp(-\sigma_{tg}x/\mu_m)$ and $\exp(-\sigma_{tg}y/\eta_m)$ into the basis function set; i.e.,

$$\exp\left(\frac{-\sigma_{tg}x}{\mu_m}\right) \simeq \sum_{n=1}^{N} E^{xn}_{gm}\exp(\alpha_n x), \text{ and} \quad (4.7)$$

$$\exp\left(\frac{-\sigma_{tg}y}{\eta_m}\right) \simeq \sum_{n=1}^{N} E^{yn}_{gm}\exp(\beta_n y), \quad (4.8)$$

expressions for the flux coefficients ψ^{xn}_{gm} and ψ^{yn}_{gm} are obtained:

$$\psi^{yn}_{gm} = [\psi^{y}_{gm}(0) - \sum_{k=1}^{N} \frac{S^{yk}_{gm}}{\mu_m\alpha_k+\sigma_{tg}} + \frac{\eta_m}{\sigma_{tg}\Delta y}(\psi^{x}_{gm}(\Delta y)-\psi^{x}_{gm}(0))]E^{xn}_{gm} \qquad (4.9)$$

$$+ \frac{S^{yk}_{gm}}{\mu_m\alpha_k+\sigma_{tg}} - \frac{\delta_{k1}\eta_m}{\sigma_{tg}\Delta y}(\psi^{x}_{gm}(\Delta y)-\psi^{x}_{gm}(0)) \text{ , and}$$

$$\psi^{xn}_{gm} = [\psi^{x}_{gm}(0) - \sum_{k=1}^{N} \frac{S^{yk}_{gm}}{\eta_m\beta_k+\sigma_{tg}} + \frac{\mu_m}{\sigma_{tg}\Delta x}(\psi^{y}_{gm}(\Delta x)-\psi^{y}_{gm}(0))]E^{yn}_{gm} \qquad (4.10)$$

$$+ \frac{S^{xk}_{gm}}{\eta_m\beta_n+\sigma_{tg}} - \frac{\delta_{n1}\mu_m}{\sigma_{tg}\Delta x}(\psi^{y}_{gm}(\Delta x)-\psi^{y}_{gm}(0)),$$

where $\alpha_1=\beta_1=0$.

The procedure for determining the flux shape within the node and the flux values on the "outgoing" edges given the source expansion and the flux values on the "incoming" boundaries is to:

1. Evaluate Eq. 4.5 and 4.6 at $x=\Delta x$ and $y=\Delta y$, respectively, and solve the resulting equations simultaneously for the outgoing edge flux values $\psi^{y}_{gm}(\Delta x)$ and $\psi^{x}_{gm}(\Delta y)$;
2. Use Eqs. 4.7 and 4.8 to evaluate the flux expansion coefficients.

2. Method 2: Calculated Leakage Shape, Separable 1-D Flux Shape

The most serious drawback to the previous method is the assumption of the flat transverse leakage shape. The second method remedies this situation by solving for the spatial dependence of the fluxes on the "outgoing" edges given:

1. the spatial flux shape of the source within the node and

2. the spatial flux shape on the "incoming" edges.

Equation 2.5 can be written for a particular node and a particular energy group and angle. For the case $\mu,\eta>0$, the equation is:

$$\mu\frac{\partial\psi(x,y)}{\partial x} + \eta\frac{\partial\psi(x,y)}{\partial y} + \sigma_t\psi(x,y) = S(x,y) \qquad (4.11)$$

with boundary conditions

$$\psi(x,y)\Big|_{x=0} = \psi(0,y),$$

$$\psi(x,y)\Big|_{y=0} = \psi(x,0).$$

Equation 4.11 plus the boundary conditions may be restated as:

$$L\psi(x,y) = S(x,y) \qquad \psi\Big|_{\partial\Omega} = \psi(x,0)U\psi(0,y),$$

where $\psi(0,y)$ and $\psi(x,0)$ are known from previous calculations, and

$$L = \mu\frac{\partial}{\partial x} + \eta\frac{\partial}{\partial y} + \sigma_t .$$

The operator notation is introduced to show that the complexity of the problem can be reduced if a function $\psi_1(x,y)$ which obeys Eq. 4.11 inside the node (but not necessarily the boundary conditions) can be found. If such a $\psi_i(x,y)$ can be determined, it immediately can be seen that Eq. 4.11 can be written as the sum of:

$$L\psi_1(x,y) = S(x,y) \ , \ \psi_1(x,y)\Big|_{\partial\Omega} = \psi_1(x,0)U\psi_1(0,y), \qquad (4.12a)$$

and

$$L\psi_2(x,y) = 0, \psi_2(x,y)\Big|_{\partial\Omega} = [\psi(x,0)-\psi_1(x,0)]U[\psi(0,y)-\psi_1(0,y)]. \qquad (4.12b)$$

Since the source is of the form:

$$S(x,y) = \sum_{m=1}^{N} \sum_{n=1}^{N} S_{mn} \exp(\alpha_m x+\beta_n y),$$

the function $\psi_1(x,y) = \sum_{m=1}^{N} \sum_{n=1}^{N} \frac{S_{mn}}{\mu\alpha_m+\eta\beta_n+\sigma_t} \exp(\alpha_m x+\beta_n y)$ satisfies (4.12a).

Therefore, since the coefficients ψ^m_{x0} and ψ^n_{y0} in the expansions

$$\psi(x,0) = \sum_{m=1}^{N} \psi^m_{x0} \exp(\alpha_m x) \qquad \text{and}$$

$$\psi(0,y) = \sum_{n=1}^{N} \psi^n_{y0} \exp(\beta_n y)$$

are known from previous node calculations, the problem is reduced to solving:

$$\mu\frac{\partial\psi_2(x,y)}{\partial x} + \eta\frac{\partial\psi_2(x,y)}{\partial y} + \sigma_t\psi_2(x,y) = 0 \qquad (4.13)$$

with boundary conditions

$$\psi_2(x,0) = \sum_{m=1}^{M} \left[\psi^m_{x0} - \sum_{n=1}^{N} \frac{S_{mn}}{\mu\alpha_m+\eta\beta_n+\sigma_t}\right] \exp(\alpha_m x) = \sum_{m=1}^{M} \psi^m_{2x} \exp(\alpha_m x)$$

$$\psi_2(0,y) = \sum_{n=1}^{N} \left[\psi^n_{y0} - \sum_{m=1}^{M} \frac{S_{mn}}{\mu\alpha_m+\eta\beta_n+\sigma_t}\right] \exp(\beta_n y) = \sum_{n=1}^{N} \psi^n_{2y} \exp(\beta_n y).$$

The partial differential equation Eq. 4.13 can be written in terms of a parameter, s, as the following system of ordinary differential equations[18]:

$$\frac{dx(s)}{ds} = \mu,$$

$$\frac{dy(s)}{ds} = \eta, \text{ and}$$

$$\frac{d\psi(s)}{ds} = -\sigma_t\psi(s).$$

This system has the solution:

$$\psi_2(s) = \psi_2(0)\exp(-\sigma_t s), \text{ with}$$

$$s = \frac{x-x_0}{\mu} = \frac{y-y_0}{\eta} \quad .$$

There will be two distinct "regions of influence" (shown in Fig. 2 as Region 1 and Region 2) in the node depending on which boundary is intersected by the characteristic lines in that region:

$$\psi_2(x,y) = \psi_2(0,y-\frac{\eta x}{\mu})\exp(\frac{-\sigma_t x}{\mu}) \quad , \quad \text{if} \quad y>\frac{\eta}{\mu}x \tag{4.14}$$

$$= \psi_2(x-\frac{\mu y}{\eta},0)\exp(\frac{-\sigma_t y}{\eta}) \quad , \quad \text{if} \quad y<\frac{\eta}{\mu}x \quad .$$

Recalling that $\psi(x,y) = \psi_1(x,y) + \psi_2(x,y)$, evaluating the resulting functions on the outgoing edges $(\Delta x,y)$ and $(x,\Delta y)$ gives:

$$\psi(\Delta x,y) = \sum_n^N [\sum_m^M \frac{S_{mn}\exp(\alpha_m \Delta x)}{\mu\alpha_m+\eta\beta_n+\sigma_t}] \exp(\beta_n y) \tag{4.15}$$

$$+ \sum_n^N [\psi_{2y}^n \exp(\frac{-\beta_n r\Delta x}{\mu})] \exp\{(\beta_n - \frac{\sigma_t}{\eta})y\} \quad , \text{ if } y>\frac{\eta\Delta x}{\mu}$$

$$+ \sum_m^M [\psi_{2x}^m \exp(\alpha_m \Delta x)] \exp\{(\frac{\alpha_m\mu-\sigma_t}{\eta})y\} \quad , \text{ if } \quad y<\frac{\eta\Delta x}{\mu}$$

$$\psi(x,\Delta y) = \sum_m^M [\sum_n^N \frac{S_{mn}\exp(\beta_n \Delta y)}{\mu\alpha_m+\eta\beta_n+\sigma_t}] \exp(\alpha_m x) \tag{4.16}$$

$$+ \sum_m^M [\psi_{2x}^m \exp(\frac{-\alpha_m\mu\Delta y}{\eta})] \exp\{(\alpha_m - \frac{\sigma_t}{\mu})x\} \quad , \text{ if } x>\frac{\mu\Delta y}{\eta}$$

$$+ \sum_n^N [\psi_{2x}^n \exp(\beta_n \Delta y)] \exp\{(\frac{\beta_n\eta-\sigma_t}{\mu})x\} \quad , \text{ if } \quad x<\frac{\mu\Delta y}{\eta} \quad .$$

ORNL-DWG 82-13806

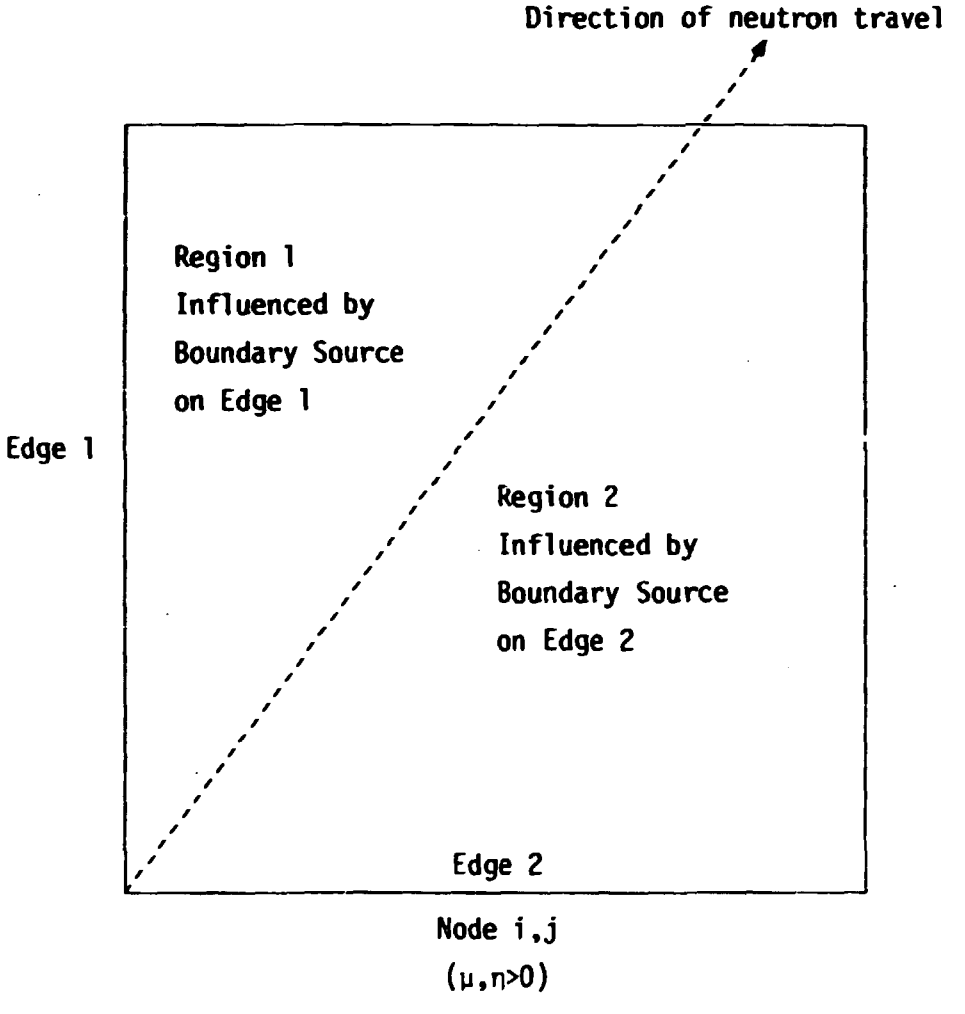


Figure 2. Regions of Influence for Boundary Fluxes on "Upstream" Edges

The procedure outlined in Appendix E can be used to approximate these fluxes in the form

$$\psi(\Delta x,y) = \sum_{n}^{N} \psi^{n}_{\Delta x,y}\exp(\beta_n y) \quad \text{and} \tag{4.17}$$

$$\psi(x,\Delta y) = \sum_{m}^{M} \psi^{n}_{x,\Delta y}\exp(\alpha_m x) \quad . \tag{4.18}$$

With these shaped boundary values, the shaped transverse leakage "sources" can be determined; therefore the channel flux coefficients ψ^{xn}_{gm} and ψ^{yn}_{gm} are given by shaped leakage analogues to Eq. 4.9 and 4.10:

$$\psi^{yn}_{gm} = [\psi^{y}_{gm}(0) - \sum_{n}^{N} \frac{S^{yk}_{gm} + \frac{\eta_m}{\sigma_{tg}\Delta y}(\psi^{k}_{\Delta x,y} - \psi^{k}_{0,y})}{\mu_m\alpha_k + \sigma_{tg}}]E^{yn}_{gm} \tag{4.19}$$

$$+ \frac{S^{yn}_{gm} + \frac{\eta_m}{\sigma_{tg}\Delta y}(\psi^{n}_{\Delta x,y} - \psi^{n}_{0,y})}{\mu_m\alpha_n + \sigma_{tg}} \quad , \text{ and}$$

$$\psi^{xn}_{gm} = [\psi^{x}_{gm}(0) - \sum_{k}^{N} \frac{S^{xk}_{gm} + \frac{\mu_m}{\sigma_{tg}\Delta x}(\psi^{k}_{x,\Delta y} - \psi^{k}_{x,0})}{\eta_m\beta_k + \sigma_{tg}}]E^{xn}_{gm} \tag{4.20}$$

$$+ \frac{S^{xn}_{gm} + \frac{\mu_m}{\sigma_{tg}\Delta x}(\psi^{n}_{x,\Delta y} - \psi^{n}_{x,0})}{\eta_m\beta_n + \sigma_{tg}} \quad .$$

3. Method 3: Calculated Leakage Shape, 2-D Flux Shape

The final improvement eliminates the flux separability assumption by representing the flux as a full 2-D expansion in x and y:

$$\psi(x,y) \approx \sum_{m=1}^{M}\sum_{n=1}^{N} \psi_{mn}\exp(\alpha_m x+\beta_n y) \tag{4.21}$$

$$= \psi_1(x,y) + \psi_2(x,y),$$

where $\psi_1(x,y)$ and $\psi_2(x,y)$ are the two solutions introduced in Section B.2.

As before:

$$\psi_1(x,y) = \sum_{m=1}^{N} \sum_{n=1}^{N} \frac{S_{mn}}{\mu\alpha_m+\eta\beta_n+\sigma_t} \exp(\alpha_m x+\beta_n y) \ .$$

The corresponding expansion of $\psi_2(x,y)$ can be found by projecting $\psi_2(x,y)$ (given in Eq. 4.14) onto the exponential basis function set (see Appendix E):

$$\psi_2(x,y) = \sum_{m=1}^{M} \sum_{n=1}^{N} \psi_2^{mn} \exp(\alpha_m x+\beta_n y) \ . \tag{4.22}$$

In terms of Eq. 4.21, the expansion coefficients are:

$$\psi_{mn} = \frac{S_{mn}}{\mu\alpha_m+\eta\beta_n+\sigma_{tg}} + \psi_2^{mn} \ . \tag{4.23}$$

As in the previous method, the outflow boundary flux expansion given by Eqs. 4.17 and 4.18 is calculated to be used for initial conditions for "downstream" nodes.

C. Iterative Solution for the Coefficients

With the values of the flux expansion coefficients given as a function of the source expansion coefficients for the three versions, the calculational procedure follows the form outlined in Section 2.C. Using the third version of the new method as an example, the equations which are used for angle a and group g are:

$$\psi_{mn}^I = \frac{S_{mn}^I}{\mu\alpha_m+\eta\beta_n+\sigma_t} + \psi_{2,I}^{mn} \tag{4.24}$$

$$S_{mn}^I = T_{mn}^I + \sum_{g'=1}^{g} \sum_{k'=1}^{k} \sigma_{sg'\to g}^{k'} \sum_{r=1}^{k'} f_{ir}^{k'} \phi_{g'rmn}^{k'I} \tag{4.25}$$

$$T^{I}_{mn} = Q_{mn} + \sum_{g'=g+1}^{G} \sum_{k'=1}^{K} \sigma^{k'}_{sg'\to g} \sum_{r=1}^{k'} \sigma^{k'}_{sg'\to g} \phi^{k',I-1}_{g'rmn} \tag{4.26}$$

$$+ \sum_{g'=1}^{G} \nu_{g'} \sigma_{fg'} \chi_g \phi^{0I-1}_{g'0mn}$$

where

$$\phi^{kI}_{grmn} = \sum_{i=1}^{A} w_i f^{k}_{ir} \psi^{iI}_{mng} \quad \text{and A=number of angles} \tag{4.27}$$

with $\psi^{mn}_{2,I}$ and f^{k}_{ir} given by Eq. 4.22 and Eq. 2.8, respectively.

As outlined in Section 2.C, the values of ψ^{I}_{mn} are found by iterating on the within group scattering using Eqs. 4.24 and 4.25, are used to calculate the ϕ^{kI}_{grmn} using Eq. 4.27. After the ϕ^{kI}_{gmn} are found for each group, Eq. 4.26 is used to calculate the outer iteration values of T^{I}_{mn} for each energy group.

D. Coarse-Mesh Acceleration

The iterative procedure described in the previous section is accelerated using the commonly utilized "coarse-mesh acceleration."[20] This procedure can be derived by integrating Eq. 2.6 over angle and space within a given mesh segment ij, resulting in:

$$J^{l}_{g,i+1,j} - J^{r}_{gij} + J^{r}_{g,i-1,j} - J^{l}_{gij} + J^{t}_{gij} \tag{4.28}$$

$$- J^{b}_{g,i,j+1} + J^{t}_{g,i,j-1} - J^{b}_{gij} + \sigma^{ij}_{r} \phi^{ij}_{g} = S^{ij}_{g}$$

where

$$J^{l}_{gij} = \sum_{\substack{m=1 \\ \mu_m<0}}^{M} w_m \mu_m \psi^{m}_{x}(0)/\Delta x_i$$

$$J^r_{gij} = \sum_{\substack{m=1 \\ \mu_m>0}}^{M} w_m \mu_m \phi^m_x(\Delta x)/\Delta x_i$$

$$J^b_{gij} = -\sum_{\substack{m=1 \\ \eta_m<0}}^{M} w_m \eta_m \phi^m_y(0)/\Delta y_j$$

$$J^t_{gij} = \sum_{\substack{m=1 \\ \eta_m>0}}^{M} w_m \eta_m \phi^m_y(\Delta y)/\Delta y_j$$

$$\sigma^{ij}_r = \sigma^{ij}_t - \sigma^{ij}_{g\to g}$$

$$\phi^{ij}_g = \int_{\Delta x_i} \int_{\Delta y_j} \phi^{ij}_g(x,y)\,dy\,dx/\Delta x \Delta y$$

$$S^{ij}_g = \int_{\Delta x} \int_{\Delta y} S^{ij}_g(x,y)\,dx\,dy$$

where $S^{ij}_g(x,y)$ includes external sources, fission sources, and scattering from other energy groups to group g.

Defining a set of constants α^i, i=r,l,t,b, as ratios of partial currents J and averages fluxes ϕ, i.e.

$$\alpha^{gl}_{ij} = \frac{J^l_{gij}}{\phi^{ij}_g} \quad ,$$

equation 4.28 can be written as

$$\alpha^{gr}_{i-1,j}\phi^{i-1,j}_g + \alpha^{gl}_{i+1,j}\phi^{i+1,j}_g + \alpha^{gt}_{i,j-1}\phi^{i,j-1}_g + \alpha^{gb}_{i,j+1}\phi^{i,j+1}_g \qquad (4.29)$$

$$+ (\alpha^{ij}_r - \alpha^{gr}_{ij} - \alpha^{gl}_{ij} - \alpha^{gt}_{ij} - \alpha^{gb}_{ij})\phi^{ij}_g = S^{ij}_g \quad .$$

At the completion of each inner iteration, Eq. 4.29 is solved using the newly calculated values of J and ϕ for a corrected set $\psi_{g.}^{ij}$ The flux coefficients are then corrected using

$$\psi_{mng}^{ij} = \psi_{mng}^{ij} \cdot \frac{\hat{\phi}_g^{ij}}{\phi_g^{ij}} .$$

CHAPTER 5

SAMPLE PROBLEMS

A. Introduction

In order to test the accuracy and speed of the methods that were developed, several reactor eigenvalue and shielding problems were calculated on the IBM 3033 at Oak Ridge National Laboratory:

1. A simple "source in the corner" shielding benchmark problem[21] previously calculated by Lawrence[8],
2. A simple reactor cell eigenvalue problem, obtained from Wagner's work[7], consisting of fuel and absorber regions,
3. A more geometrically complex reactor lattice problem[21] consisting of a 12x12 boiling water reactor fuel bundle,
4. A fixed source two energy group shielding problem which models a borehole source-detector configuration[22],
5. Problem 4 repeated using five energy groups,
6. A more realistic boundary source shielding problem which models a sodium-steel-concrete voided shield[23].

A user's manual for EXTREME is given as Appendix F; a typical sample problem computer output listing is given in Appendix G.

B. Sample Problem 1: Shielding Benchmark Problem

For the first sample problem, Benchmark problem BSS-5[21], shown in Fig. 3, was calculated. This is a two energy group, homogeneous material, isotropic scattering calculation with a spatially constant source in the lower left-hand corner, which was previously calculated

ORNL-DWG 82-13807

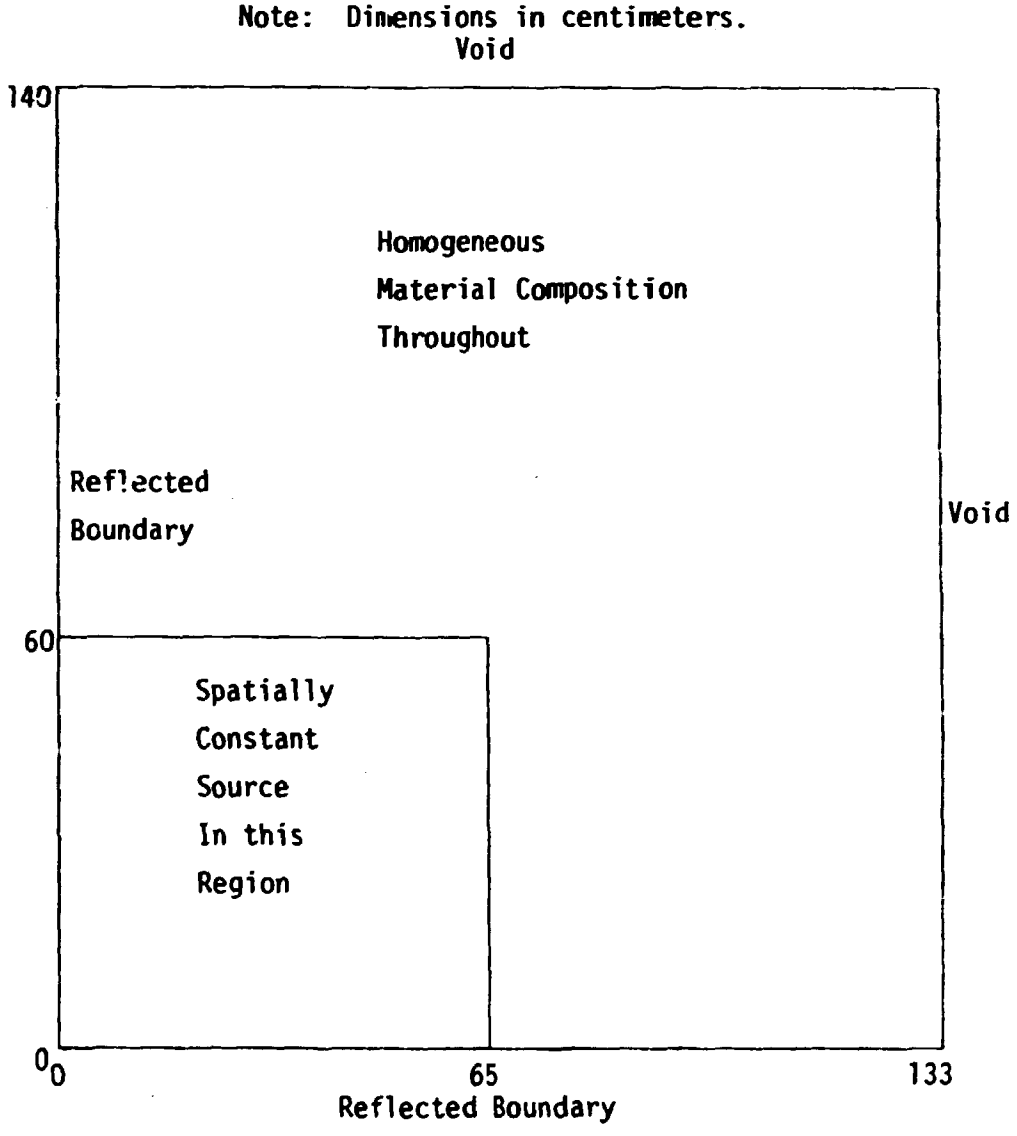


Figure 3. Geometry of Sample Problem 1

by Lawrence[8]. The problem data are given in Table 1. For this study the problem was solved using the three new nodal methods and also with DOT4.2[2], a conventional finite difference code developed at Oak Ridge National Laboratory. The calculations were performed with the S_8 angular quadrature given in Table 2; the spatial mesh was varied from 4 to 64 nodes for the new methods and from 4 to 9216 for DOT4.2. The flux convergence criterion was set at 10^{-4}, the value used for all six sample problems; this value was chosen because this was near the limit of accuracy for the single precision DOT4.2 code. (The new nodal methods use double precision real variables.)

The results are given in Table 3 and shown in Fig. 4, which compare the error in the total calculated leakage for the various methods. The "correct" answer of 5.827×10^{-3}, which was obtained from an extrapolation of the Benchmark solutions given in Reference 22, was used as the standard of comparison. The new nodal methods used three spatial modes in the expansions; the modes themselves were chosen from a material buckling (B_m) spectrum calculation described in Appendix H. The comparisons of accuracy versus cost seem to indicate that all of the versions of the new development calculated the system leakage (to within 1% error) from 40 to 100 times faster DOT4.2 on this problem, and that the flat leakage version is the best of all methods.

This result, obtained also by Lawrence[8] for his flat leakage method in 25.1 seconds on the CYBER 175 for a convergence criterion of 10^{-5}, is surprising in light of Fig. 5, which shows the flux shape for the first energy group along the right edge for the most accurate solutions of the four methods. Specifically, the flux shapes from the

Table 1. Data for Sample Problem 1

Energy Group	σ_t	$\sigma_s^{g \to g}$	$\sigma_s^{g-1 \to g}$	Source Density
1	0.921039-1	0.6947-2	0	0.6546-2
2	0.10088	0.485-2	0.23434-1	0.17701-1

Note: $0.921039\text{-}1 = 0.921039 \times 10^{-1}$

Table 2. Listing of S_8 Quadrature

Angle	μ	η	Weight
1	0.272000	0.962300	0
2	0.192327	0.962300	0.0291971
3	0.608542	0.793522	0
4	0.577350	0.793522	0.0233138
5	0.192327	0.793522	0.0233138
6	0.816496	0.577350	0
7	0.793522	0.577350	0.0233138
8	0.577350	0.577350	0.0225258
9	0.192167	0.577350	0.0233138
10	0.981330	0.192327	0
11	0.962299	0.192327	0.0291971
12	0.793522	0.192327	0.0233138
13	0.577350	0.192327	0.0233138
14	0.192327	0.192327	0.0291971

Table 3. Results of DOT4.2 and EXTREME Calculations of Leakage in Sample Problem 1

Method	Number of Nodes	Number of Modes	Leakage x 10^4	% Error	CPU Time (Seconds)
DOT	256	-	4.55507	21.83	5.42
DOT	1024	-	5.40426	7.25	22.07
DOT	4096	-	5.71621	1.90	79.03
DOT	9216	-	5.78403	0.74	173.11
EXTREME Method 1	4	3	5.79674	0.52	0.75
Method 1	16	3	5.81270	0.25	2.85
Method 1	64	3	5.82718	<0.01	6.55
EXTREME Method 2	4	3	9.84535	68.96	1.66
Method 2	16	3	5.86870	0.72	4.51
Method 2	64	3	5.82930	0.04	17.29
EXTREME Method 3	4	3	9.80770	68.31	2.01
Method 3	16	3	5.86390	0.63	4.51
Method 3	64	3	5.8278	0.01	15.77

The Standard equals 5.827 x 10^{-4} from the extrapolated results in Reference 21.

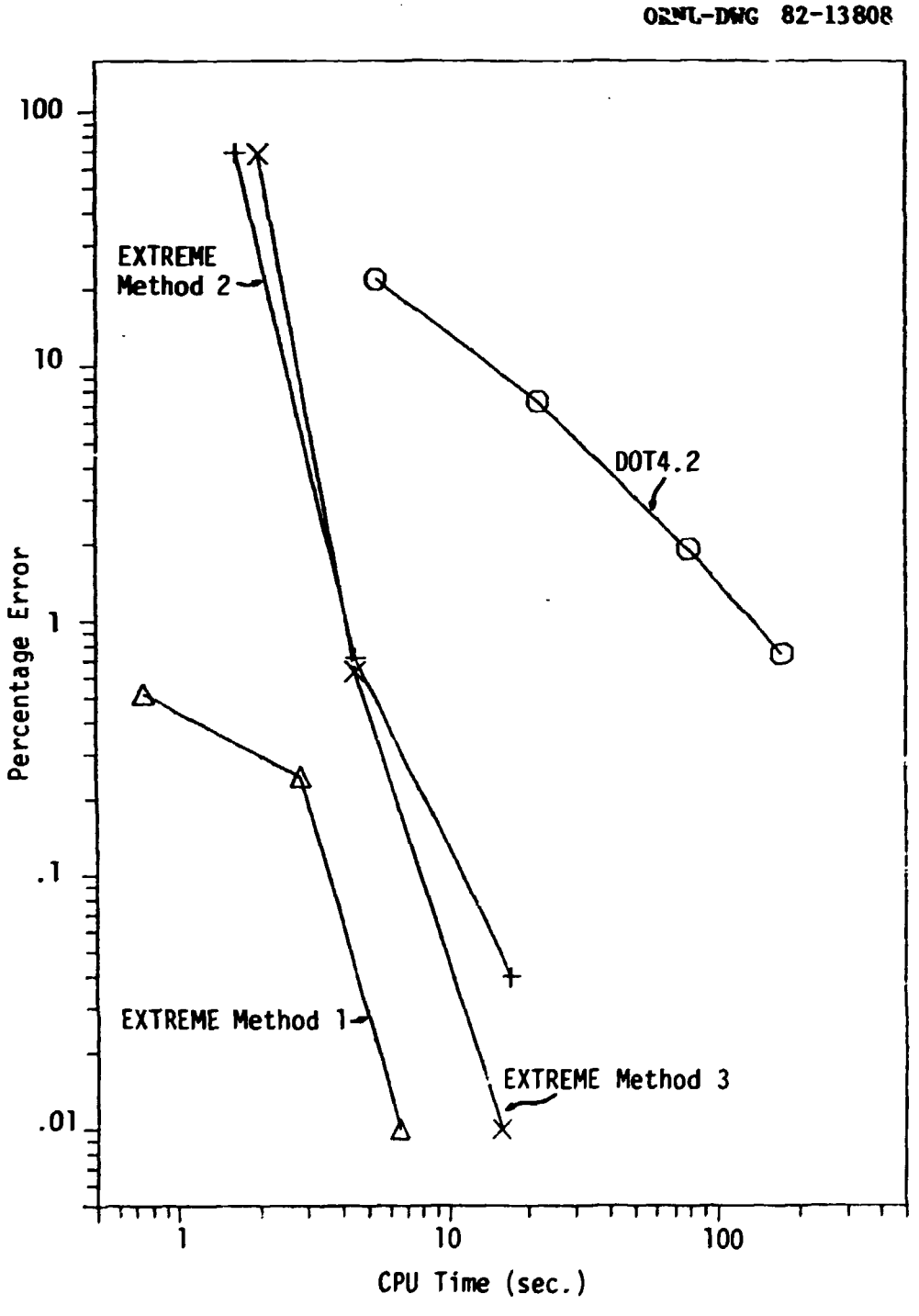


Figure 4. Comparison of Percentage Error in Leakage for EXTREME versus DOT4.2 for Sample Problem 1

ORNL-DWG 82-13809

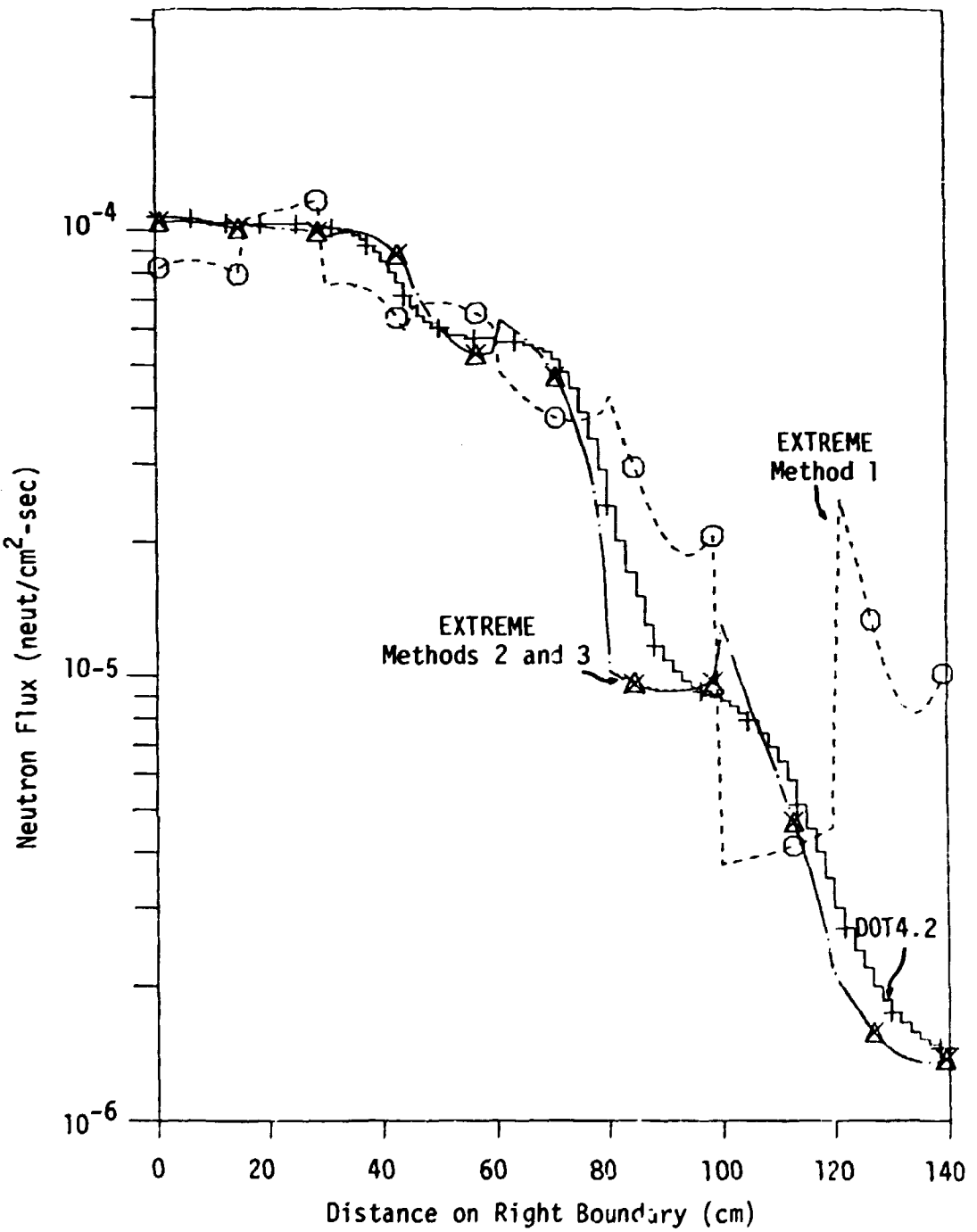


Figure 5. Plot of the Flux Shape on Right Edge of Sample Problem 1 as Calculated by DOT4.2 and EXTREME

DOT4.2 and shaped leakage version calculations agree fairly well while the flat leakage version gives an entirely different shape. This ability of the flat leakage version to deliver an accurate total leakage despite a poor spatial flux shape was also reported by Walters[9].

For this problem the B_m method of choosing the EXTREME modes was tested by resubmitting the 16-node problem for the three exponential methods with various values for the non-zero modes. Table 4 and Fig. 6 show the results using the function set 1, $\exp(-\alpha x/\Delta x)$, $\exp(\alpha x/\Delta x)$ for various values of α. It is interesting to note that the total leakage values calculated by each method as a function of α has a minimum near α=4.5, the mode value chosen a priori by the B_m calculation. For the shaped leakage methods (Methods 2 and 3), this minimum value seems to be the mode that yields the most accurate solution for the flat leakage method, however, the minimum dips below the correct answer. It can be seen from these results that (1) the accuracy of the EXTREME code for this problem is very sensitive to the choice of modes and (2) the mode chosen by the B_m calculation for this problem was close to the optimum choice.

Also, the effect of various orders of expansion was tested by running a series of calculations with the third method using 1, 3, and 5 modes in the expansion and the non-zero modes recommended by the B_m calculation. (The use of paired modes offers an advantage in EXTREME because of the symmetry; also, including a zero mode is necessary to specify spatially flat external sources. Therefore, it is recommended that the mode set chosen by the user be of the form α=0, $\pm\alpha_1$, $\pm\alpha_2$, etc.) The results are given in Table 5 and Fig. 7;

Table 4. Total System Leakage for Sample Problem 1 for Various Choices of α in Mode Set 0, $-\alpha$, α

α	Method 1	(% Error)	Method 2	(% Error)	Method 3	(% Error)
1	5.9074	(1.38)	6.4279	(10.31)	6.3914	(9.69)
2	5.8738	(0.79)	6.2356	(7.01)	6.2044	(6.48)
3	5.8474	(0.35)	6.0233	(3.37)	6.0001	(2.97)
4	5.8227	(-.07)	5.8841	(0.98)	5.8701	(0.74)
4.2	5.8193	(-.13)	5.8704	(0.74)	5.8583	(0.54)
4.4	5.8165	(-.18)	5.8619	(0.60)	5.8517	(0.42)
4.6	5.8145	(-.21)	5.8588	(0.55)	5.8505	(0.40)
4.8	5.8132	(-.24)	5.8611	(0.59)	5.8546	(0.47)
5	5.8127	(-.24)	5.8687	(0.72)	5.8640	(0.64)
5.2	5.8129	(-.24)	5.8816	(0.94)	5.8786	(0.87)
5.4	5.8139	(-.22)	5.8995	(1.24)	5.8983	(1.22)
5.6	5.8156	(-.20)	5.9224	(1.64)	5.9229	(1.65)
5.8	5.8181	(-.15)	5.9500	(2.11)	5.9521	(2.15)
6	5.8213	(-.10)	5.9821	(2.67)	5.9858	(2.73)
7	5.8468	(0.34)	6.2018	(6.43)	6.2128	(6.72)
8	5.8853	(1.00)	6.4970	(11.50)	6.5144	(11.80)
9	5.9323	(1.81)	6.8399	(17.38)	6.8627	(17.78)

ORNL-DWG 82-13810

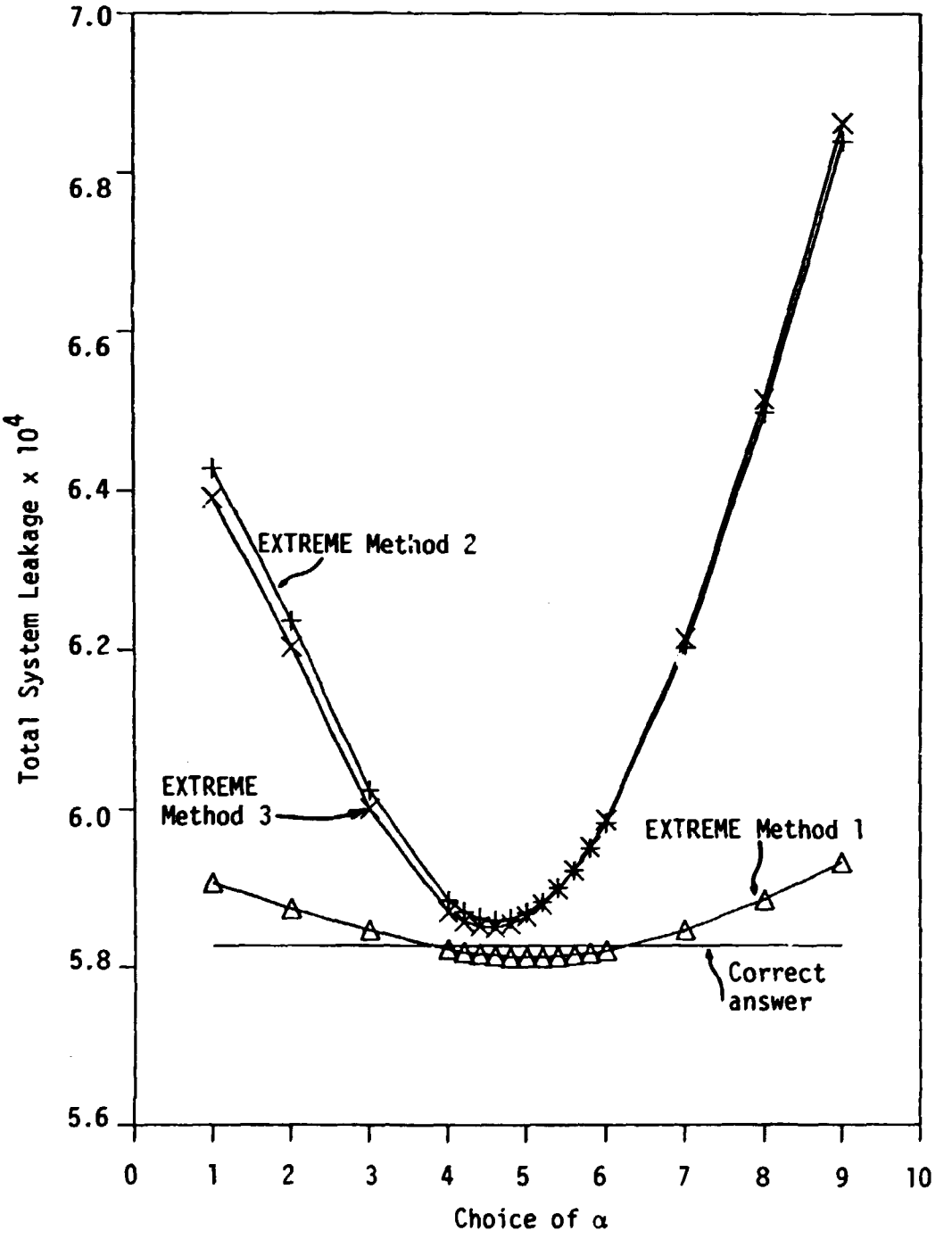


Figure 6. Plot of Total Leakage in Sample Problem 1 versus Choice of α in Mode Set 0, -α, α for the Three Methods in EXTREME

Table 5. EXTREME Method 3 Results for Sample Problem 1 Using N=1,3,5 Mode Sets Chosen by B_0 Calculation

No.of Modes	No. of Nodes	System Leakagex10^4	% Error	CPU time(sec.)
1	4	121.33	1982.2	0.62
1	16	23.11	396.60	1.42
1	64	9.9602	70.93	5.79
3	4	9.8077	68.31	2.21
3	16	5.8640	0.63	5.16
3	64	5.8279	0.02	19.08
5	4	8.4613	45.21	7.07
5	16	5.8279	0.02	15.38
5	64	5.8274	0.01	54.09

ORNL-DWG 82-13811

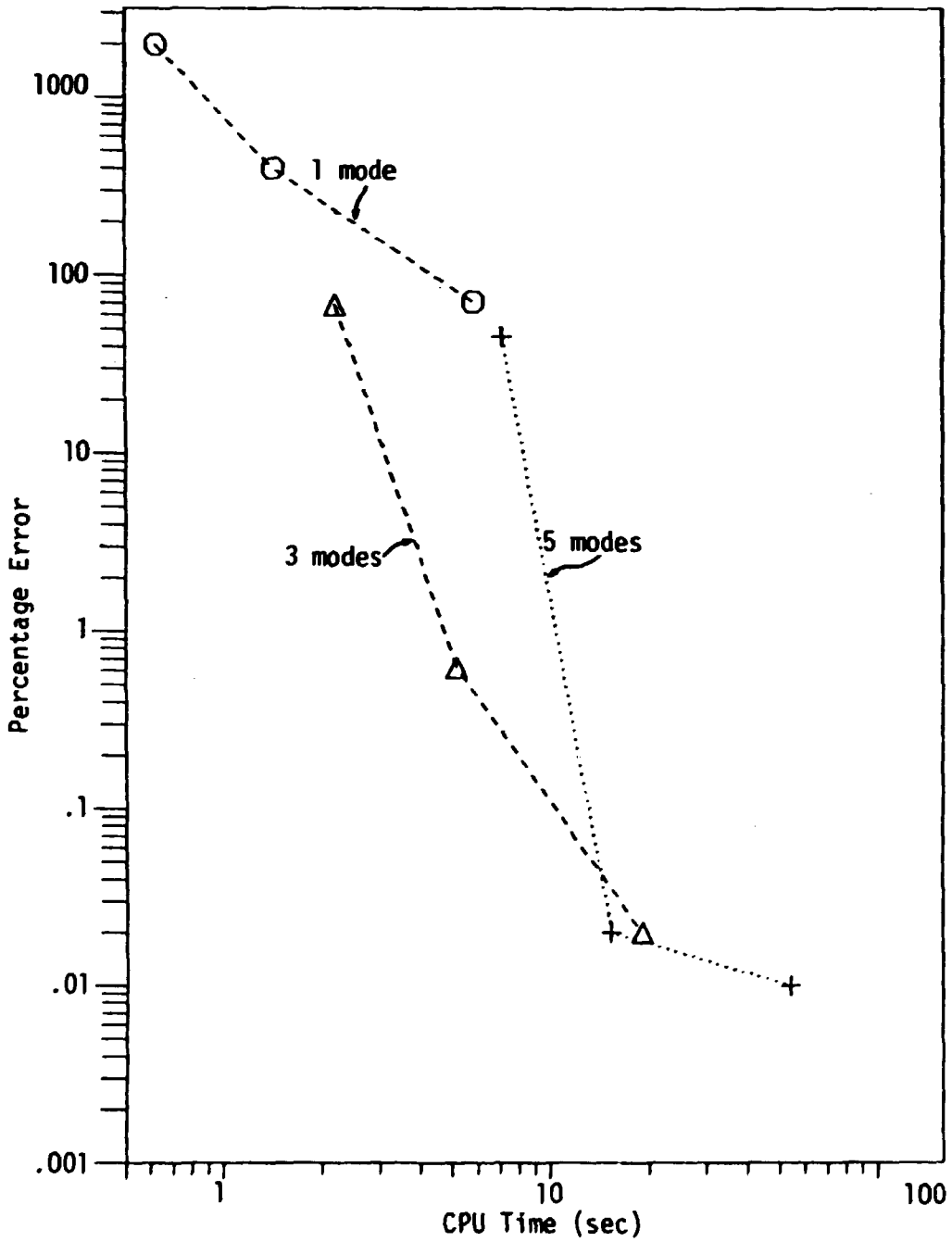


Figure 7. Plot of Percentage Error in Sample Problem versus Computing Time for 1, 3, and 5 Order Mode Sets

they indicate that the flat (one mode) approximation is not very accurate and that the use of three modes seems to converge to accurate (about 1%) answers sooner than the five-mode calculations, although the five-mode convergence rate is at least as fast as for three modes. Based on this study the remaining sample EXTREME calculations have been restricted to the three modes 0, $-\alpha$, and α with α chosen by the B_m calculation.

C. Sample Problem 2: Simple Reactor Problem

Although the primary usage of transport theory methods is for deep-penetration shielding and streaming problems, they are also used for multiplying systems, i.e., reactor design and criticality studies. For the first reactor eigenvalue problem, a cell problem (shown in Fig. 8) defined by Wagner[7] was calculated. The problem data are given in Table 6. Configuration consists of fuel regions surrounding a fuel plus absorber region in the lower left-hand corner. The reflective boundary conditions on all sides simulate an infinite array of these cells.

The eigenvalues determined by S_8 calculations of DOT4.2 and EXTREME are listed in Table 7, with the correct answer taken from Wagner's TWOTRAN calculation. The eigenvalue errors are plotted versus computer time in Fig. 9. As before, the shaped leakage versions of EXTREME spatially converge (to about 0.01% error) about 10 times faster than DOT4.2. The advantage of this rapid convergence, however, is offset by the fact that the EXTREME and DOT4.2 codes seem to be converging to different answers. However, the DOT4.2 and EXTREME results are within 1% even for the coarsest grid. Wagner's nodal

ORNL-DWG 82-13812

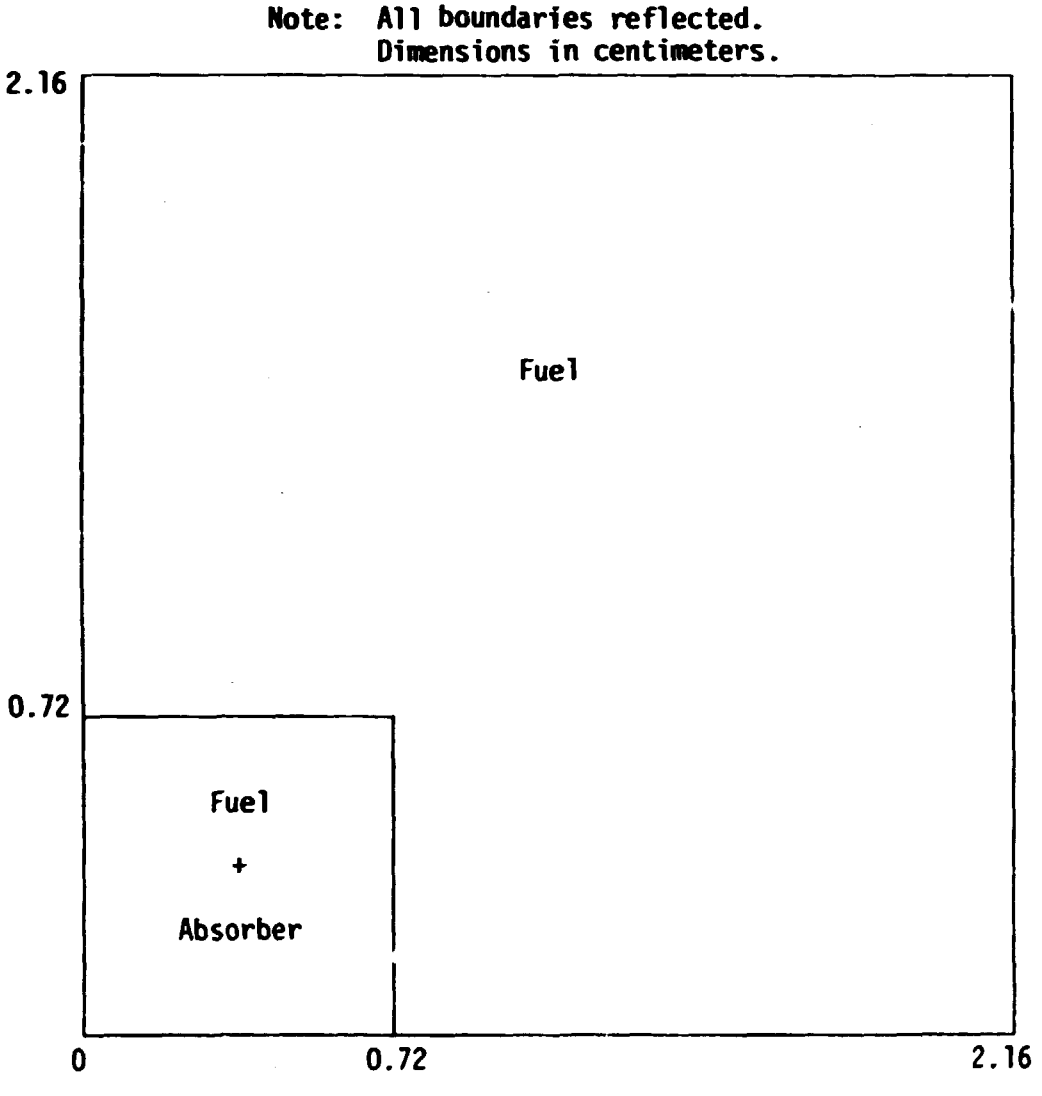


Figure 8. Geometry of Sample Problem 2

Table 6. Data for Sample Problem 2

Material	Energy Group	σ_a	σ_f	σ_t	$\sigma_s^{g \to g}$	$\sigma_s^{g-1 \to g}$
Fuel	1	0.13-1	0.5-2	0.22	0.19	0.
	2	0.5	0.25-1	0.1+1	0.5	0.17-1
Fuel + Absorber	1	0.1	0.6-2	0.22	0.193	0
	2	0.7-1	0.1	0.8	0.73	0.17-1

Table 7. Results of DOT4.2 and EXTREME Calculations of Eigenvalue for Sample Problem 2

Method	Number of Nodes	Number of Modes	Eigenvalue	% Error	CPU Time (Seconds)
DOT	4	-	0.770318	-0.317	0.94
DOT	16	-	0.771523	-0.161	2.26
DOT	64	-	0.772373	-0.051	4.85
DOT	256	-	0.772586	-0.024	14.14
DOT	1024	-	0.772654	-0.015	58.23
EXTREME Method 1	4	3	0.775374	0.337	1.75
Method 1	16	3	0.773703	0.121	5.52
Method 1	64	3	0.773578	0.104	26.86
EXTREME Method 2	4	3	0.772564	-0.027	3.35
Method 2	16	3	0.772837	0.009	12.40
EXTREME Method 3	4	3	0.772816	0.006	3.54
Method 3	16	3	0.772885	0.015	11.47
Method 3	64	3	0.772903	0.017	63.59

Standard is 0.77277 from Reference 7

ORNL-DWG 82-13813

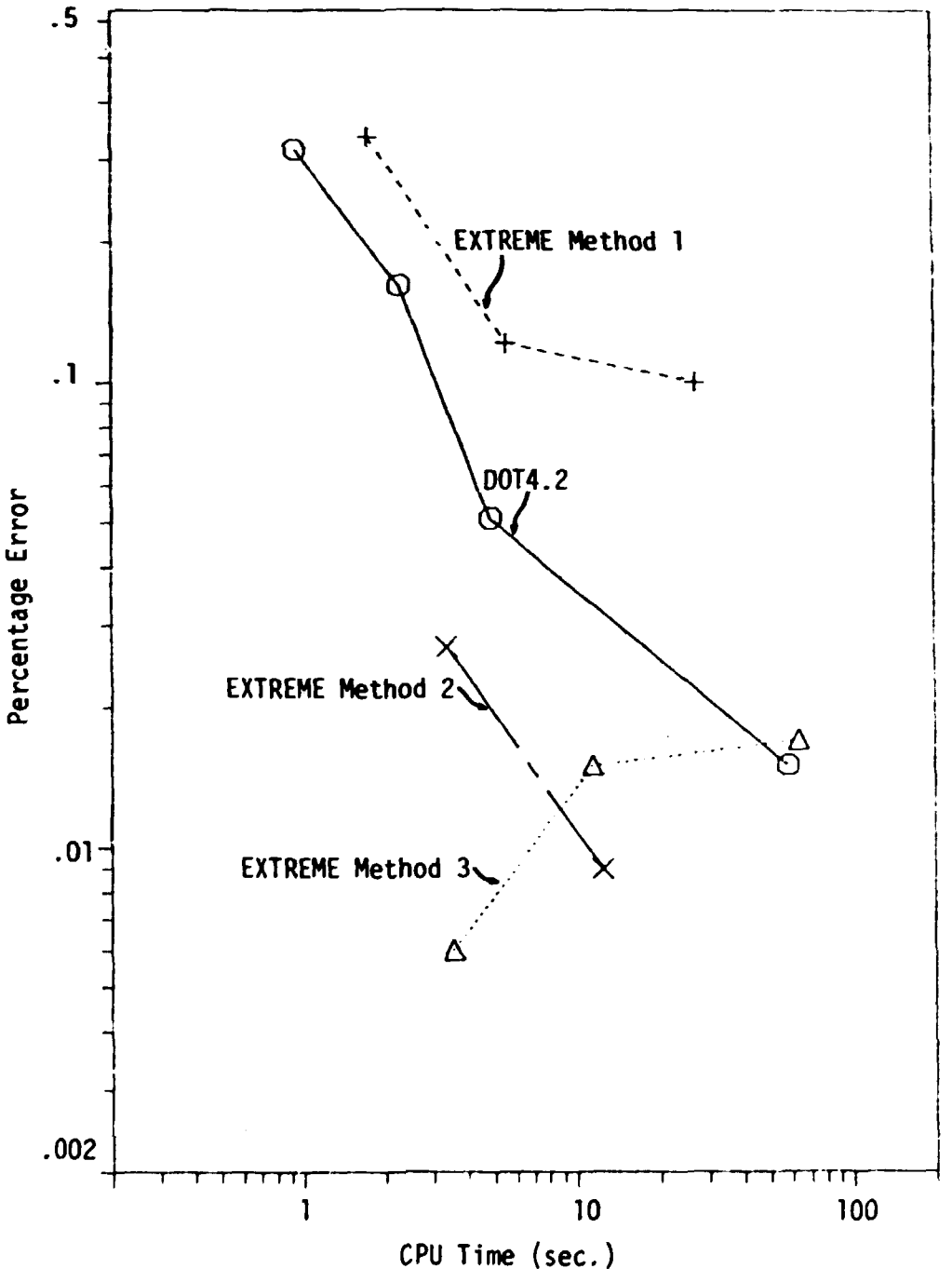


Figure 9. Comparison of Percentage Error in Eigenvalue for EXTREME versus DOT4.2 in Sample Problem 2

method calculated this eigenvalue within 0.1% in less than a second on the CYBER 176. Figure 10 gives a map of the ratios of the group 2 flux in each region as calculated by the three versions of EXTREME using 64 nodes to the flux calculated by the most accurate DOT4.2 case.

For materials with $k_\infty > 1$, B_L theory predicts a cosine spatial shape rather than an exponential shape; therefore, one of the basic advantages of the exponential basis functions is lost. Thus the B_m calculation should be no help in choosing modes for eigenvalue problems. In order to test the effect of different choices of the non-zero mode for the problem, the 16-node EXTREME model was run for various values of α in the mode set 0, $-\alpha$, α. The results are given in Table 8 and the percentage errors are plotted in Fig. 11. These results do not indicate any particular sensitivity to the choice of the non-zero mode, although the fact that the errors are of the same order as the convergence criterion may be clouding this test.

D. Sample Problem 3: BWR Lattice Eigenvalue Problem

For a "realistic" reactor cell problem, the benchmark problem[21] shown in Fig. 12, was calculated. The data for this problem are given in Table 9. This is a rather severe test for the nodal code EXTREME due to the fact that the geometry cannot be represented in fewer than 144 rectangular regions.

As before, the DOT4.2 code was run on several mesh structures; the three versions of EXTREME, however, calculated the eigenvalue for only the basic 144-node mesh. The reason for this, as can be seen from the results given in Table 10, is that the errors for the EXTREME

Note: The three values are the node averaged fluxes from Methods 1, 2, and 3, respectively, divided by those from DOT4.2.

1.0033 1.0001 1.0002	1.0005 1.0004 1.0004
0.9975 1.0000 0.9999	1.0033 1.0001 1.0002

Figure 10. Ratios of the Group 2 Node Averaged Fluxes in Each Region of Sample Problem 2 as Calculated by EXTREME and DOT4.2

Table 8. Results for 16 Node Model of Sample Problem 2 for Various Choices Mode Set 0, $-\alpha$, α in EXTREME

α	Method 1 (% Error)	Method 2 (% Error)	Method 3 (% Error)
0.5	0.773993 (0.158)	0.772721 (-0.006)	0.772660 (-0.014)
1	0.773703 (0.121)	0.772837 (0.009)	0.772885 (0.015)
2	0.773604 (0.108)	0.772723 (-0.006)	0.772785 (0.002)
3	0.773583 (0.105)	0.772737 (-0.004)	0.772785 (0.002)
4	0.773525 (0.098)	0.772705 (-0.008)	0.772745 (-0.003)

ORNL-DWG 82-13815

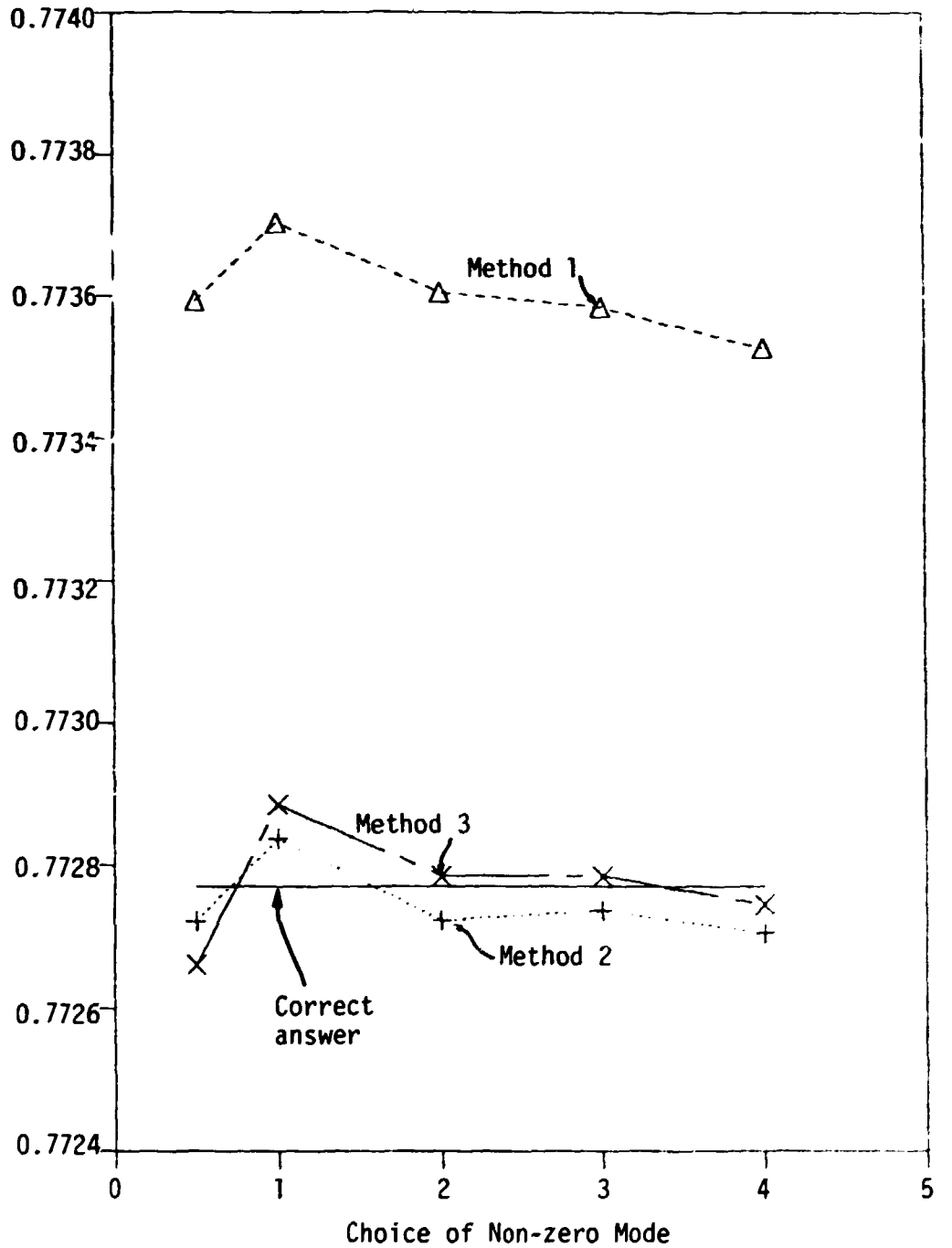
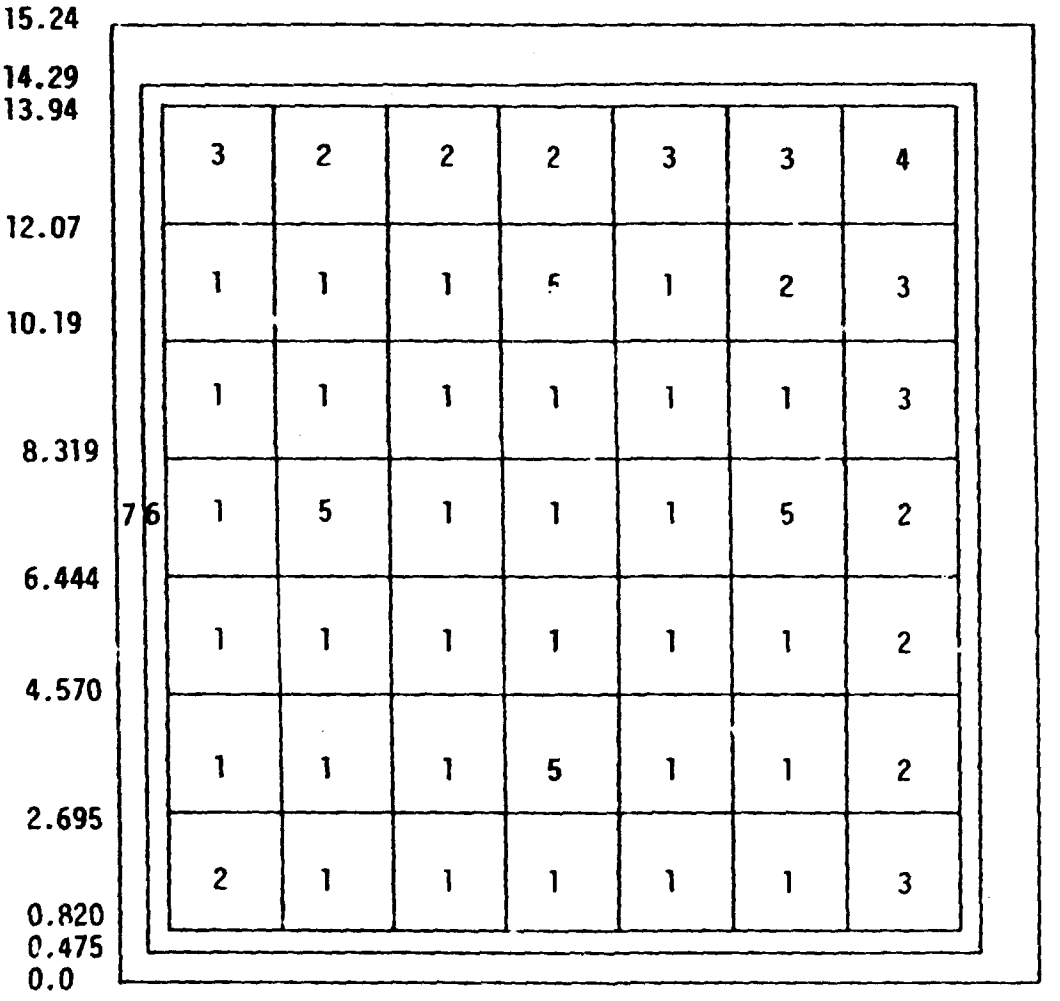


Figure 11. Comparison of Effect of Mode Choice on Calculation of Sample Problem 2 Eigenvalues using EXTREME

ORNL-DWG 82-13816



Note: Dimensions in centimeters.
Material numbers defined in Table 9.

Figure 12. Geometry of Sample Problem 3

Table 9. Data for Sample Problem 3

Material	Energy Group	σ_a	σ_f	σ_t	$\sigma_s^{g\to g}$	$\sigma_s^{g-1\to g}$
1	1	0.8983-2	0.5925-2	0.2531	0.233427	0
	2	0.5892-1	0.9817-1	0.5732	0.514280	0.1069-1
2	1	0.8726-2	0.5242-2	0.2536	0.233924	0
	2	0.5174-1	0.8228-1	0.5767	0.524960	0.1095-1
3	1	0.8587-2	0.4820-2	0.2535	0.233793	0
	2	0.4717-1	0.7200-1	0.5797	0.532530	0.1112-1
4	1	0.8480-2	0.4337-2	0.2533	0.236900	0
	2	0.4140-1	0.5900-1	0.5837	0.542300	0.1113-1
5	1	0.9593-2	0.5605-2	0.2506	0.230847	0
	2	0.1626	0.2424-1	0.5853	0.422700	0.1016-1
6	1	0.1043-2	0	0.2172	0.207062	0
	2	0.4394-2	0	0.4748	0.470406	0.9095-2
7	1	0.1983-3	0	0.2476	0.210582	0
	2	0.7796-2	0	0.1123+1	0.1152+1	0.3682-1

Table 10. Results of DOT4.2 and EXTREME Calculations of Eigenvalue for Sample Problem 3

Method	Number of Nodes	Number of Modes	Eigenvalue	% Error	CPU Time (Seconds)
DOT	144	-	1.08487	-0.221	7.84
DOT	576	-	1.08676	-0.047	89.4
DOT	2304	-	1.08707	-0.018	472.8
EXTREME Method 1	144	3	1.08741	0.013	83.34
EXTREME Method 2	144	3	1.08719	-0.007	192.36
EXTREME Method 3	144	3	1.08728	0.001	178.18

Standard is 1.08727 from Reference 21

calculations on this coarse mesh are already of the same order of magnitude as the problem convergence criterion of 0.01%. (The most accurate TWOTRAN answer of Reference 21 was used as the standard.) The percentage errors in the eigenvalues are plotted in Figure 13. Figure 14 gives a map of the ratios of the group 2 flux in each region as calculated by the most accurate EXTREME case to the flux calculated by the most accurate DOT4.2 case.

For this problem, the shaped leakage versions of EXTREME calculated the eigenvalue with at least five times less error than DOT4.2 for three minutes of computer time. But, again, all of the methods were able to calculate the eigenvalue within 0.1% fairly quickly.

E. Sample Problem 4: Two-group Volumetric Source Borehole Shielding Problem

For the second shielding problem, a realistic two-group, isotropic scattering model of a borehole source-detector configuration shown in Figure 15 was calculated with DOT4.2 and the three versions of EXTREME. The problem data are given in Table 11: the S_6 quadrature listed in Table 12 was used. Although such problems are most often modelled in cylindrical geometry, the dominant characteristic of this problem, reflection from a large "rock" region competing with streaming through "mud" filler, is present in the x-y model.

The comparison, shown in Table 13 and plotted in Fig. 16, is based on the absorption in the gas region, which is proportional to the neutron detector response. For this case the correct answer was taken from a $1/h^2$ extrapolation of the most accurate DOT4.2 results, h

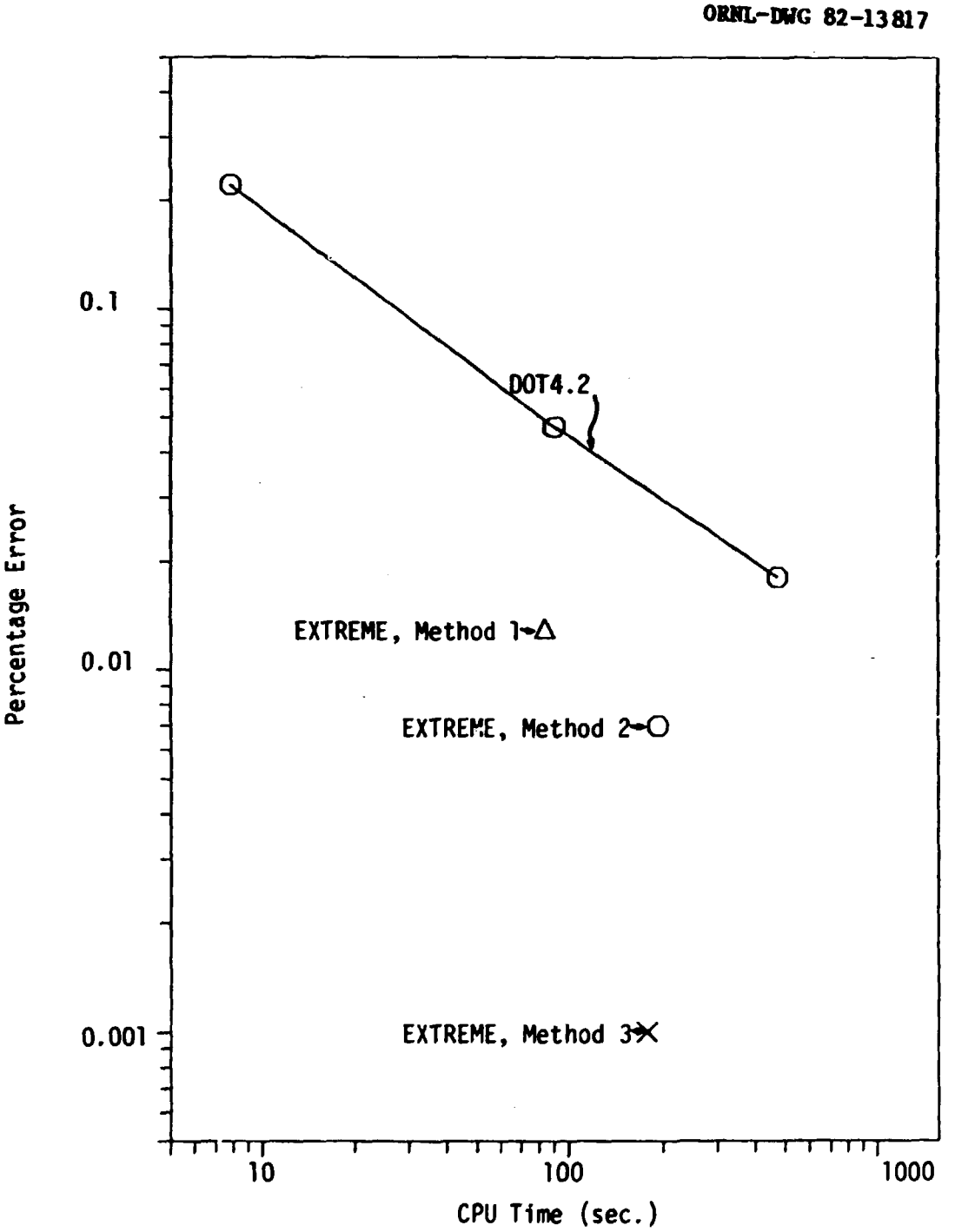


Figure 13. Comparison of Percentage Error in Eigenvalue for DOT4.2 versus EXTREME in Sample Problem 3

ORNL-DWG 82-13818

Note: The three values are the node averaged fluxes from Methods 1, 2, and 3, respectively, divided by those from DOT4.2. (Symmetric about the diagonal.)

1.0006 0.9998 0.9999	1.0003 1.0005 1.0002	1.0019 1.0002 1.0008	0.9981 1.0011 1.0004	1.0023 1.0010 1.0008	1.0008 1.0002 1.0002	1.0008 1.0002 1.0001
1.0004 0.9999 1.0000	1.0036 1.0009 1.0008	0.9978 1.0011 1.0008	0.9988 0.9999 0.9999	0.9983 1.0008 1.0011	1.0032 1.0002 1.0005	
1.0022 1.0008 1.0006	0.9980 1.0006 1.0008	1.0042 1.0042 1.0020	0.9990 1.0011 1.0009	1.0035 1.0024 1.0021		
0.9976 1.0004 1.0002	0.9994 0.9997 1.0002	0.9992 1.0009 1.0008	1.0108 1.0009 1.0005			
1.0019 1.0010 1.0006	0.9980 0.9992 1.0007	1.0044 1.0023 1.0018				
1.0002 1.0001 1.0000	1.0035 1.0005 1.0005					
1.0005 1.0003 1.0003						

Figure 14. Ratios of the Group 2 Node Averaged Fluxes in Each Region of Sample Problem 3 as Calculated by EXTREME and DOT4.2

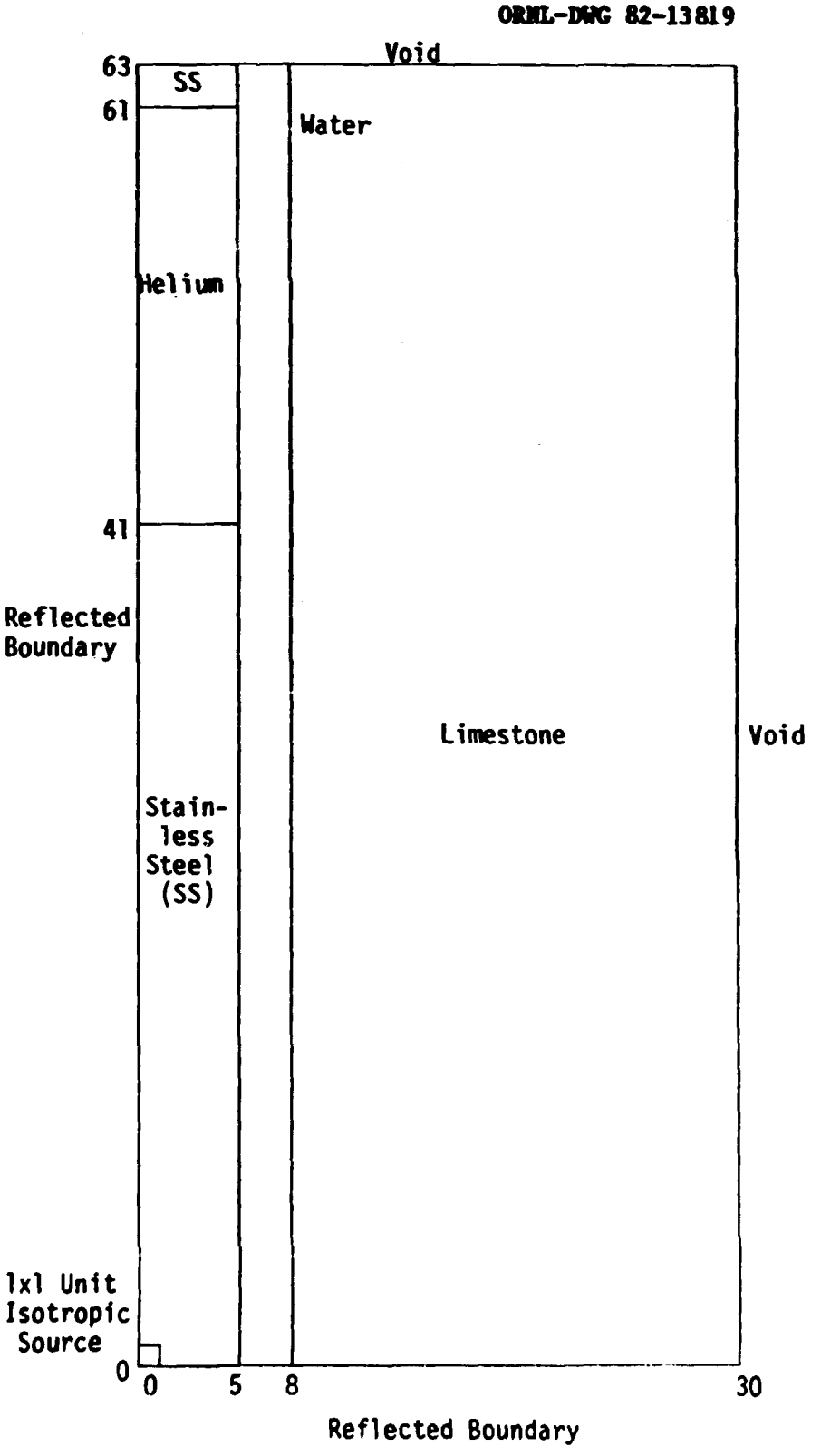


Figure 15. Geometry of Sample Problems 4 and 5

Table 11. Data for Sample Problem 4

Material	Energy Group	σ_a	σ_t	σ_s g→g	σ_s g-1→g
Water	1	0.447196-3	0.694676	0.634883	0
	2	0.186552-2	0.149414+1	0.124404-1	0.59107-1
Helium	1	0.202081-2	0.248635-2	0.449312-3	0
	2	0.119939-1	0.120208	0.242964-3	0.622984-5
Limestone	1	0.271963-2	0.330263	0.314419	0
	2	0.884484-3	0.542416	0.486617	0.130742-1
Steel	1	0.330796-2	0.499122	0.49446	0
	2	0.219522-1	0.910098	0.882483	0.136134-2

Note: Unit isotropic source in group 1

Table 12. Listing of S_6 Quadrature

Angle	μ	η	Weight
1	.36515	.93095	0
2	.25820	.93095	0.041667
3	.73030	.68313	0
4	.68313	.68313	0.041667
5	.25820	.68313	0.041667
6	.96609	.25820	0
7	.93095	.25820	0.041667
8	.68313	.25820	0.041667
9	.25820	.25820	0.041667

Table 13. Results of DOT4.2 and EXTREME Calculations of Absorption in Detector Region of Sample Problem 4

Method	Number of Nodes	Modes Used	Absorption$\times 10^4$	% Error	CPU Time (Seconds)
DOT	1860	-	3.25391	-1.28	44.26
DOT	3960	-	3.35682	1.84	129.4
DOT	7440	-	3.32847	0.98	422.95
	∞	-	3.2962		
EXTREME Method 1	70	3	2.82521	-14.29	23.97
Method 1	140	3	3.07345	6.76	35.82
Method 1	507	3	3.22124	2.31	95.21
EXTREME Method 2	70	3	3.38252	2.62	67.86
Method 2	140	3	3.33113	1.45	109.75
EXTREME Method 3	70	3	3.30255	0.19	61.47
Method 3	140	3	3.30264	0.95	95.16
Method 3	507	3	3.29267	-0.11	241.

Standard is 3.2962×10^{-4} from $1/h^2$ extrapolation of the DOT results

ORNL-DWG 82-13820

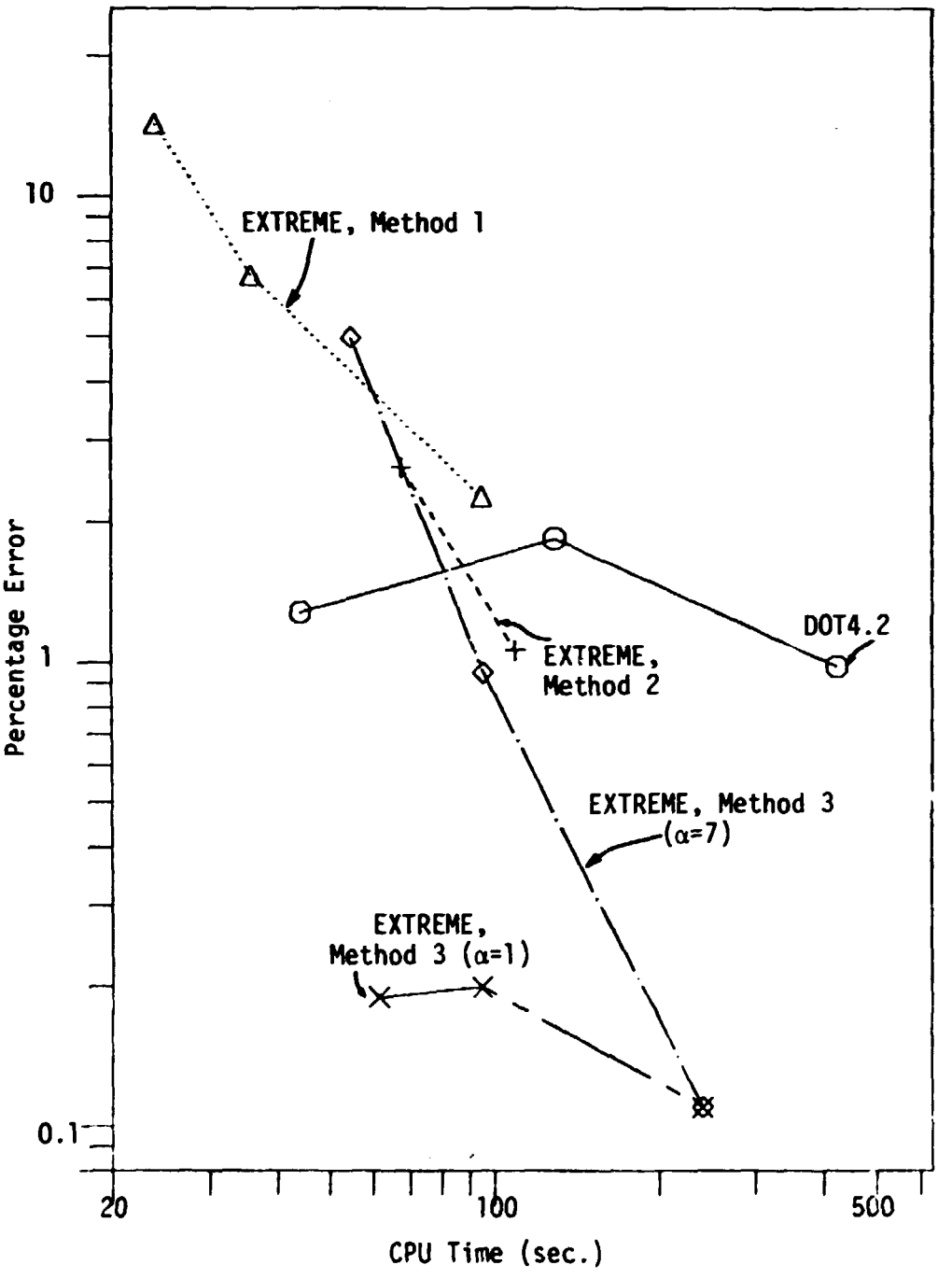


Figure 16. Comparison of Percentage Error in Detector Absorption for EXTREME versus DOT4.2 in Sample Problem 4

being the average mesh spacing. As for the previous problem, the EXTREME calculations using method 3 shows a considerable advantage for calculations more accurate than about 1%; the first two EXTREME versions and DOT4.2 converge slowly as the mesh spacing becomes finer, whereas the full expansion shaped leakage version settles down faster. In fact, the third version calculates the detector absorption almost ten times more accurately than DOT4.2 for the same computing time. As a test of the sensitivity of the third version results to the choice of modes, the third version calculations were repeated with a starting mode choice of seven rather than the value of one suggested by the B_m calculation. The results are also shown on Fig. 14. Again, the third version results are more accurate than DOT4.2, even with an incorrect mode choice. Fig. 17 gives a plot of the second group flux shapes from each of the four methods along the longitudinal axis of the detector for the most accurate runs of each version. All four methods generate similar exponentially decaying flux shapes. As before, the shaped leakage results agree more closely with the DOT4.2 results.

F. Sample Problem 5: Five Group Volumetric Source Borehole Shielding Problem

The third problem was repeated using the five group data shown in Table 14. The results are shown in Table 15 and Figure 18, again using a $1/h^2$ extrapolation of the DOT4.2 results as a standard. As in the previous problem, the DOT4.2 series of calculations did not converge uniformly. However, as before, the third version of the new

ORNL-DWG 82-13821

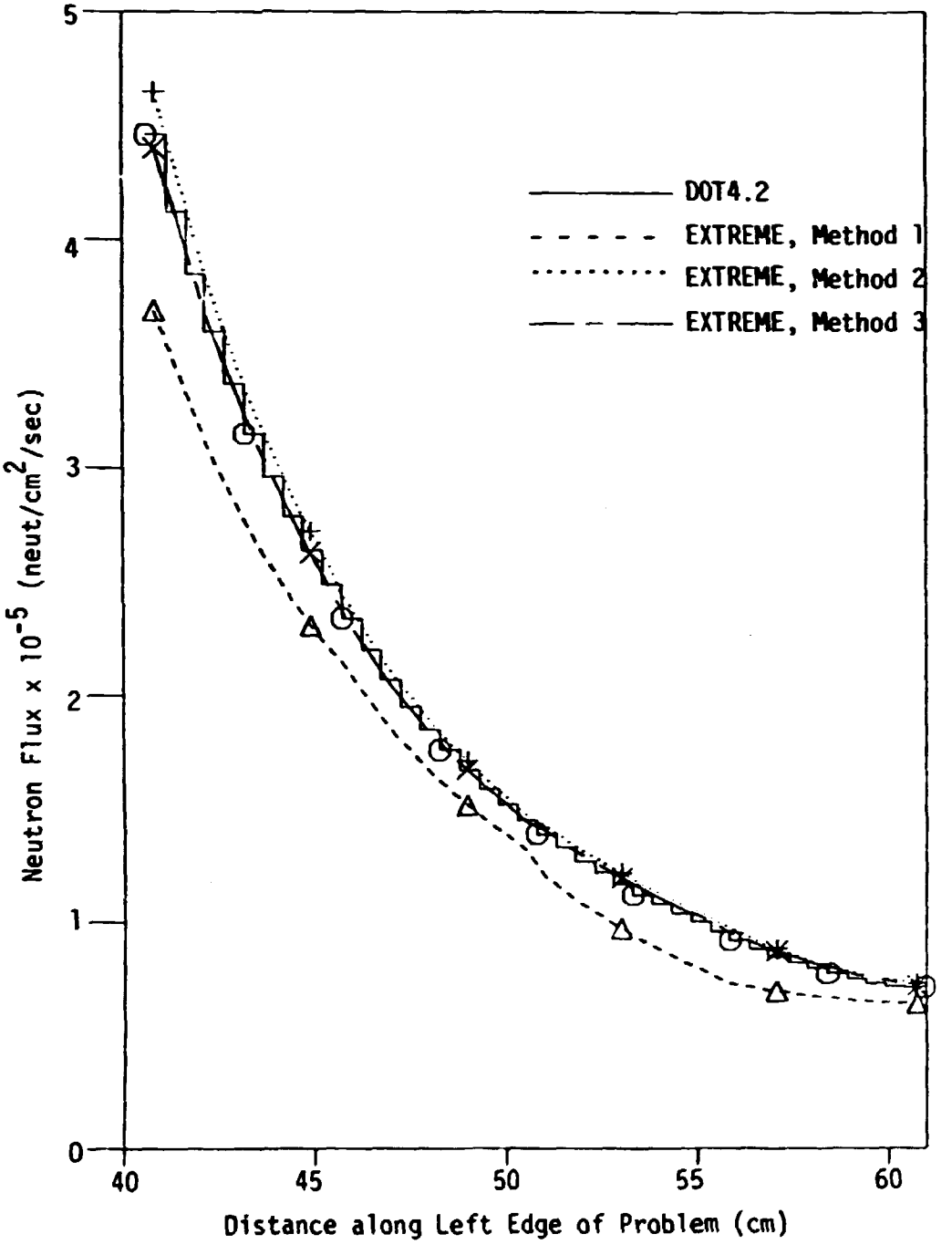


Figure 17. Plot of the Flux along the Longitudinal Axis of Detector in Sample Problem 4 as Calculated by EXTREME and DOT4.2

Table 14. Data for Sample Problem 5

	Energy Group				
	1	2	3	4	5
Water					
σ_a	0.204221-2	0.232815-5	0.370946-5	0.89033-4	0.186552-2
σ_t	0.166371	0.319627	0.706306	0.13691+1	0.149414+1
$\sigma_s^{g\to g}$	0.827352-1	0.184114	0.41781	0.118938+1	0.124404+1
$\sigma_s^{g-1\to g}$	0	0.666382-1	0.104402	0.288236	0.178904
$\sigma_s^{g-2\to g}$	0	0	0.1226-1	0.22089-1	0.255367-3
$\sigma_s^{g-3\to g}$	0	0	0	0.269274-2	0.200773-4
$\sigma_s^{g-4\to g}$	0	0	0	0	0.243368-5
Steel					
σ_a	0.554473-2	0.596984-3	0.490902-3	0.58045-2	0.219522-1
σ_t	0.30321	0.263967	0.327932	0.920694	0.910098
$\sigma_s^{g\to g}$	0.198691	0.248275	0.32223	0.910767	0.882483
$\sigma_s^{g-1\to g}$	0	0.807918-1	0.139007-1	0.521149-2	0.412322-2
$\sigma_s^{g-2\to g}$	0	0	0.165872-1	0.11944-2	0.111508-7
$\sigma_s^{g-3\to g}$	0	0	0	0.163365-2	0.203146-7
$\sigma_s^{g-4\to g}$	0	0	0	0	0.917718-6
Helium					
σ_a	0.111353-3	0.223702-3	0.326044-3	0.568677-2	0.119939
σ_t	0.635097-3	0.812467-3	0.82901-3	0.598435-2	0.120208
$\sigma_s^{g\to g}$	0.343178-3	0.411409-3	0.367448-3	0.278721-3	0.242964-3
$\sigma_s^{g-1\to g}$	0	0.180565-3	0.177358-3	0.135808-3	0.188716-4

Table 14. (continued)

	Concrete				
	Energy Group				
	1	2	3	4	5
σ_a	0.120437-1	0.705338-3	0.546143-4	0.121401-3	0.884484-3
σ_t	0.163225	0.215332	0.392019	0.509037	0.542416
$\sigma_s^{g\to g}$	0.114487	0.178828	0.321353	0.469195	0.486617
$\sigma_s^{g-1\to g}$	0	0.325458-1	0.313744-1	0.705603-1	0.395757-1
$\sigma_s^{g-2\to g}$	0	0	0.353284-2	0.441935-2	0.510733-4
$\sigma_s^{g-3\to g}$	0	0	0	0.604909-3	0.401545-5
$\sigma_s^{g-4\to g}$	0	0	0	0	0.497439-6
Source Density	0.6261	0.2832	0.745-1	0.14-1	0

Table 15. Results of DOT4.2 and EXTREME Calculations of Absorption in Detector Region of Sample Problem 5

Method	Number of Nodes	Modes Used	Absorption$\times 10^4$	% Error	CPU Time (Minutes)
DOT	1860	-	4.5058	-2.17	2.09
DOT	3960	-	4.6554	1.08	4.37
DOT	7440	-	4.6322	0.57	8.90
	∞	-	4.6057		
EXTREME Method 1	25	3	2.8642	-37.8	0.52
Method 1	72	3	4.0178	-12.8	1.06
EXTREME Method 2	25	3	3.7246	-19.1	1.31
Method 2	72	3	4.7423	2.97	2.19
EXTREME Method 3	25	3	3.9082	-14.0	1.23
Method 3	72	3	4.6641	1.26	1.97
Method 3	200	3	4.6481	0.92	4.88

Standard is 4.6057×10^{-4} from $1/h^2$ extrapolation of the DOT results

ORNL-DWG 82-13822

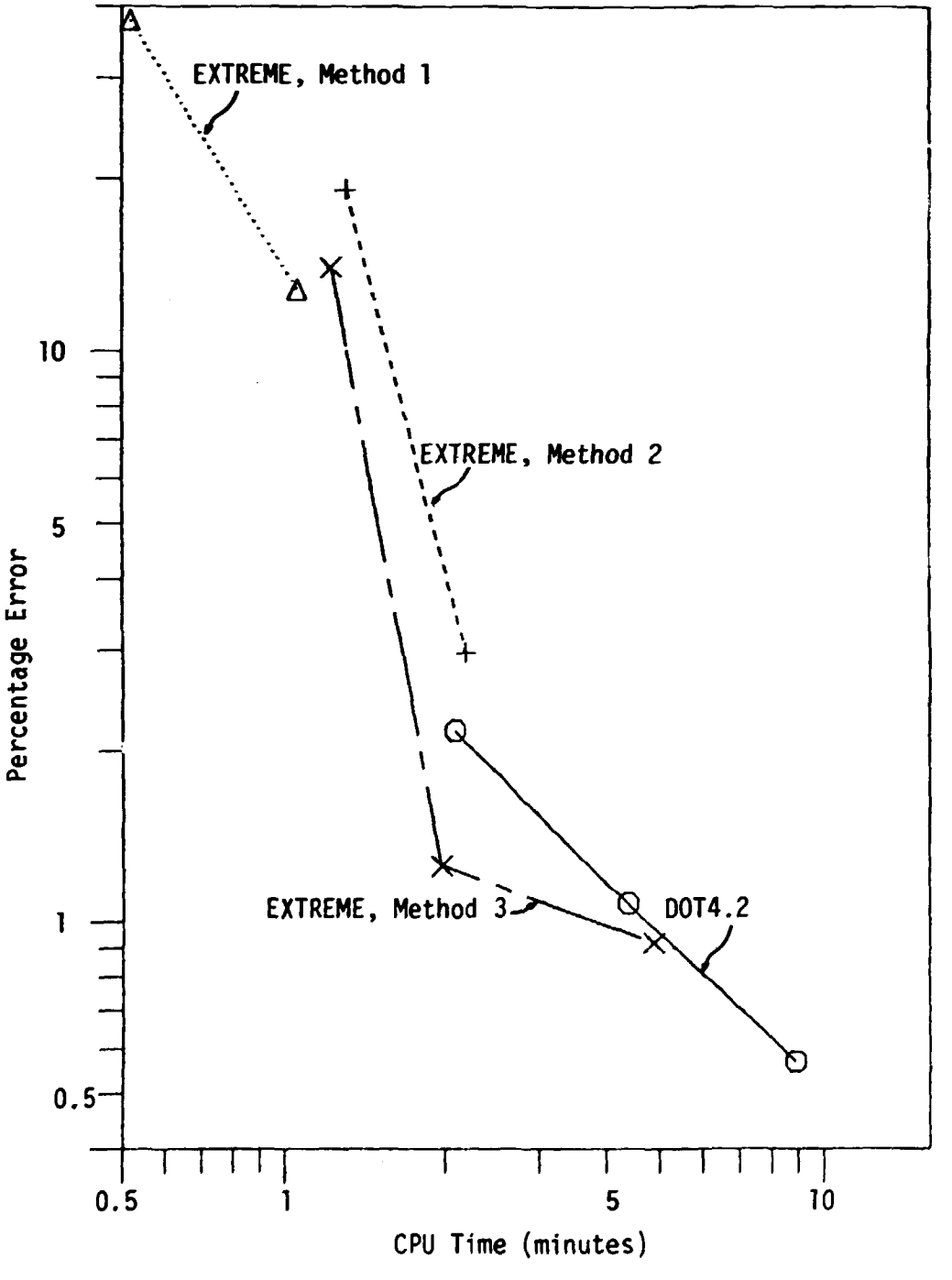


Figure 18. Comparison of Percentage Error in Detector Absorption for EXTREME versus DOT4.2 in Sample Problem 5

method shows a calculational advantage over the other methods, although the advantage is slight for this problem. Again, the DOT4.2 result does not seem to be converging to the same result as the EXTREME runs. The flux along the detector centerline for energy group 5 is shown for the four methods in Fig. 19. Again, the DOT4.2 and the shaped leakage results are similar. The flat leakage shape, however, is quite different from DOT4.2.

G. Sample Problem 6: Realistic Boundary Source Shielding Problem

The final sample problem was a calculation of a two-group, anisotropic scattering Cartesian geometry version of a previously calculated r-z scattering problem[20] (See Fig. 20 for geometry and Table 16 for problem data). The source for this shielding problem was specified along the bottom edge as boundary flux sources; the same S_8 quadrature used in Sample Problem 1 was used again. In addition to the challenge of deep-penetration shielding, this problem contained void regions, which are often difficult for discrete ordinate methods to calculate. In fact, the first two EXTREME versions failed in the void region due to division by zero (σ_{tg}) in Eqs. 4.10 and 4.19, respectively.

The leakage out of the top edge of the system was compared from calculations using DOT4.2 and the third version of EXTREME, using the extrapolated DOT4.2 results as the correct answer. The results, which are shown in Table 17 and illustrated in Fig. 21, indicate that the full expansion version of the new method performed somewhat better than the finite difference code, calculating the leakage with an error

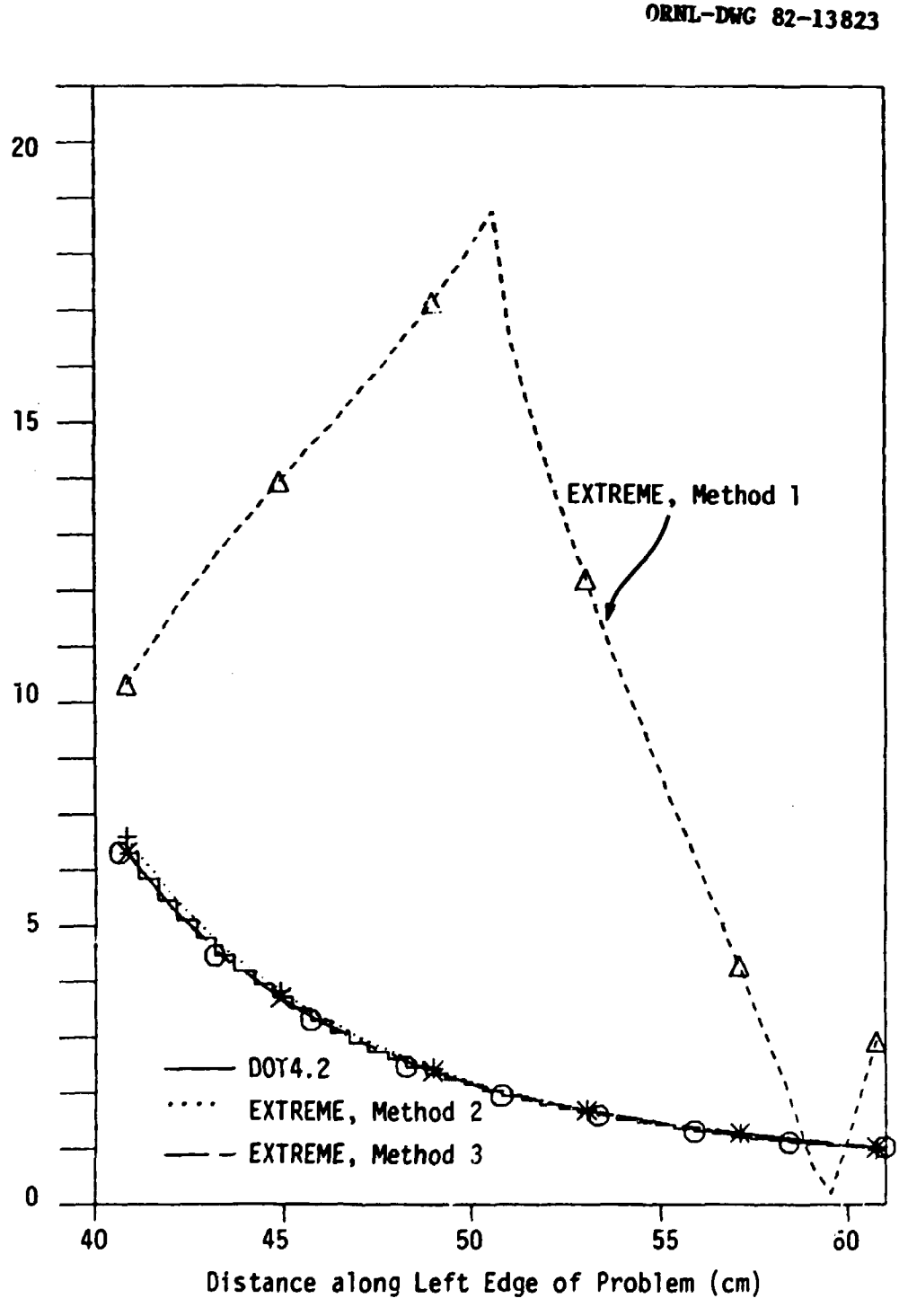


Figure 19. Plot of the Flux along the Longitudinal Axis of Detector in Sample Problem 5 as Calculated by EXTREME and DOT4.2

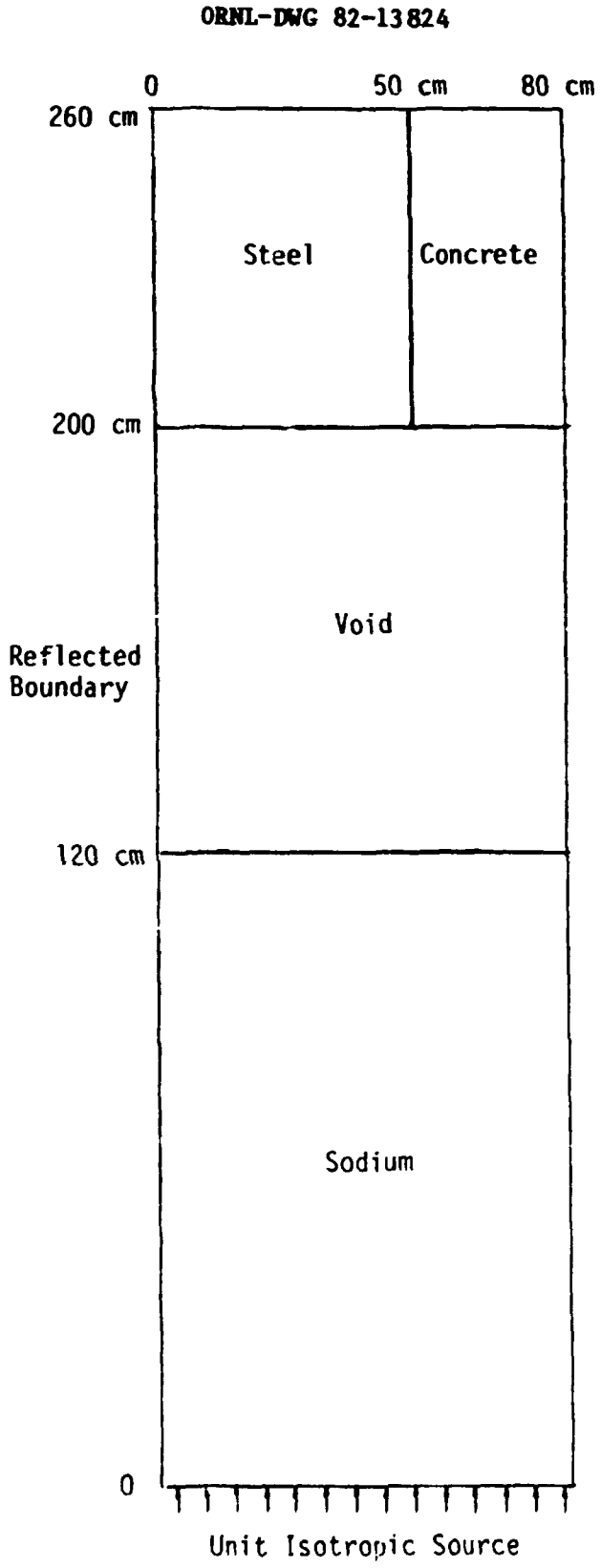


Figure 20. Geometry of Sample Problem 6

Table 16. Data for Sample Problem 6

Material	Energy Group	P_n	σ_t	$\sigma_s^{g\to g}$	$\sigma_s^{g-1\to g}$	$\sigma_s^{g-2\to g}$
Sodium	1	0	0.43137584-1	0.26473090-1	0	0
		1	0	0.37911385-1	0	0
		2	0	0.35527978-1	0	0
		3	0	0.27925890-1	0	0
	2	0	0.88249624-1	0.65755069-1	0.13400991-1	0
		1	0	0.86740136-1	-0.96387594-4	0
		2	0	0.51053967-1	-0.44981577-3	0
		3	0	0.17441791-1	-0.83661382-3	0
	3	0	0.89246333-1	0.73297560-1	0.22483591-1	0.21863093-2
		1	0	0.35939686-1	-0.34310389-2	0
		2	0	0.69684274-2	-0.51349252-2	0
		3	0	0.13753897-3	-0.62880255-2	0
Steel	1	0	0.30138278	0.19000387	0	0
		1	0	0.34245878	0	0
		2	0	0.43545473	0	0
		3	0	0.41211671	0	0
	2	0	0.24521184	0.20927387	0.75049460-1	0
		1	0	0.18235999	-0.48843361-2	0
		2	0	0.20093083	0.37485985-2	0
		3	0	0.70490658-1	0.10296197-2	0
	3	0	0.27366883	0.24890482	0.30904386-1	0.2691919-1
		1	0	0.12005287	-0.20314790-1	0
		2	0	0.73446751-1	0.76499656-2	0
		3	0	0.17678391-1	-0.79512261-2	0
Concrete	1	0	0.15216088	0.10326391	0	0
		1	0	0.15365785	0	0
		2	0	0.1699329	0	0
		3	0	0.1394459	0	0
	2	0	0.24736381	0.16833687	0.39397683-1	0
		1	0	0.14726287	-0.79861768	0
		2	0	0.15240687	0.22006292-1	0
		3	0	0.37392784-1	-0.64892247-2	0
	3	0	0.37129378	0.24303782	0.72304130-1	0.31531390-2
		1	0	0.10104793	0.8108/486-3	0.19144095-2
		2	0	0.69319725-1	0.13692595-1	-0.29986990-2
		3	0	0.60740877-1	-0.71794577-2	-0.52304268-2

Table 17. Results of DOT4.2 and EXTREME Calculations of Leakage out the Top of Sample Problem 6

Method	Number of Nodes	Modes Used	Leakage$\times 10^4$	% Error	CPU Time (Minutes)
DOT	1300	-	2.9908	-8.34	1.34
DOT	3328	-	3.2053	-1.76	3.30
DOT	5200	-	3.2506	-0.37	5.09
	∞	-	3.2627		
EXTREME					
Method 3	8	3	3.2367	-0.80	0.46
Method 3	32	3	3.2991	1.10	1.26
Method 3	128	3	3.2659	-0.10	2.38

Standard is 3.2627×10^{-4} from extrapolation of the DOT results.

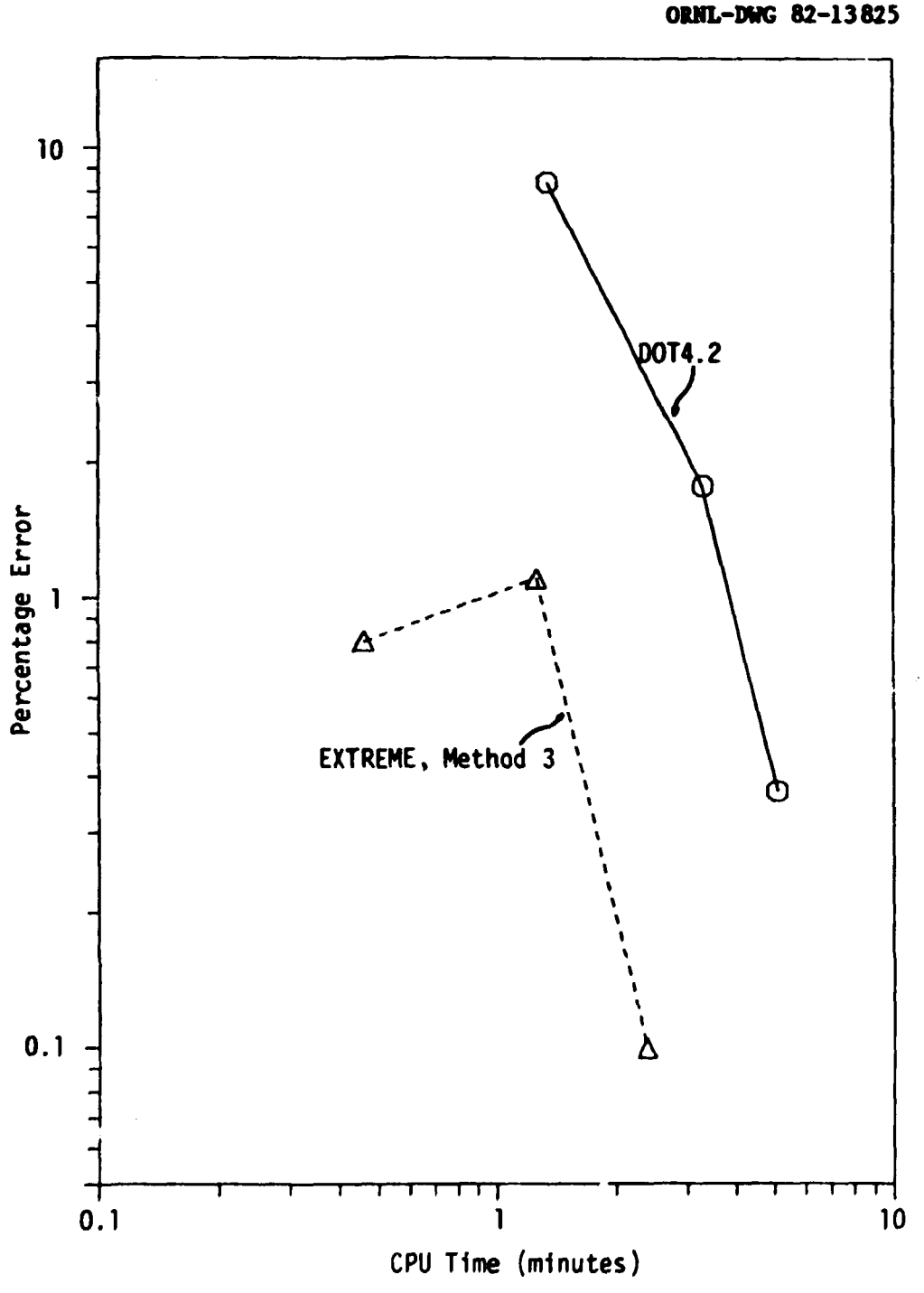


Figure 21. Comparison of Percentage Error in Top Leakage for EXTREME versus DOT4.2 in Sample Problem 6

less than 1% three to five times faster than DOT4.2. Figure 22 shows the flux shape along the top edge of the system as calculated by the two methods. Again, the shapes agree very closely. (The difference in magnitude can be explained by the fact that the DOT4.2 values are the average fluxes for the top mesh row, whereas the EXTREME values are actually on the top edge.)

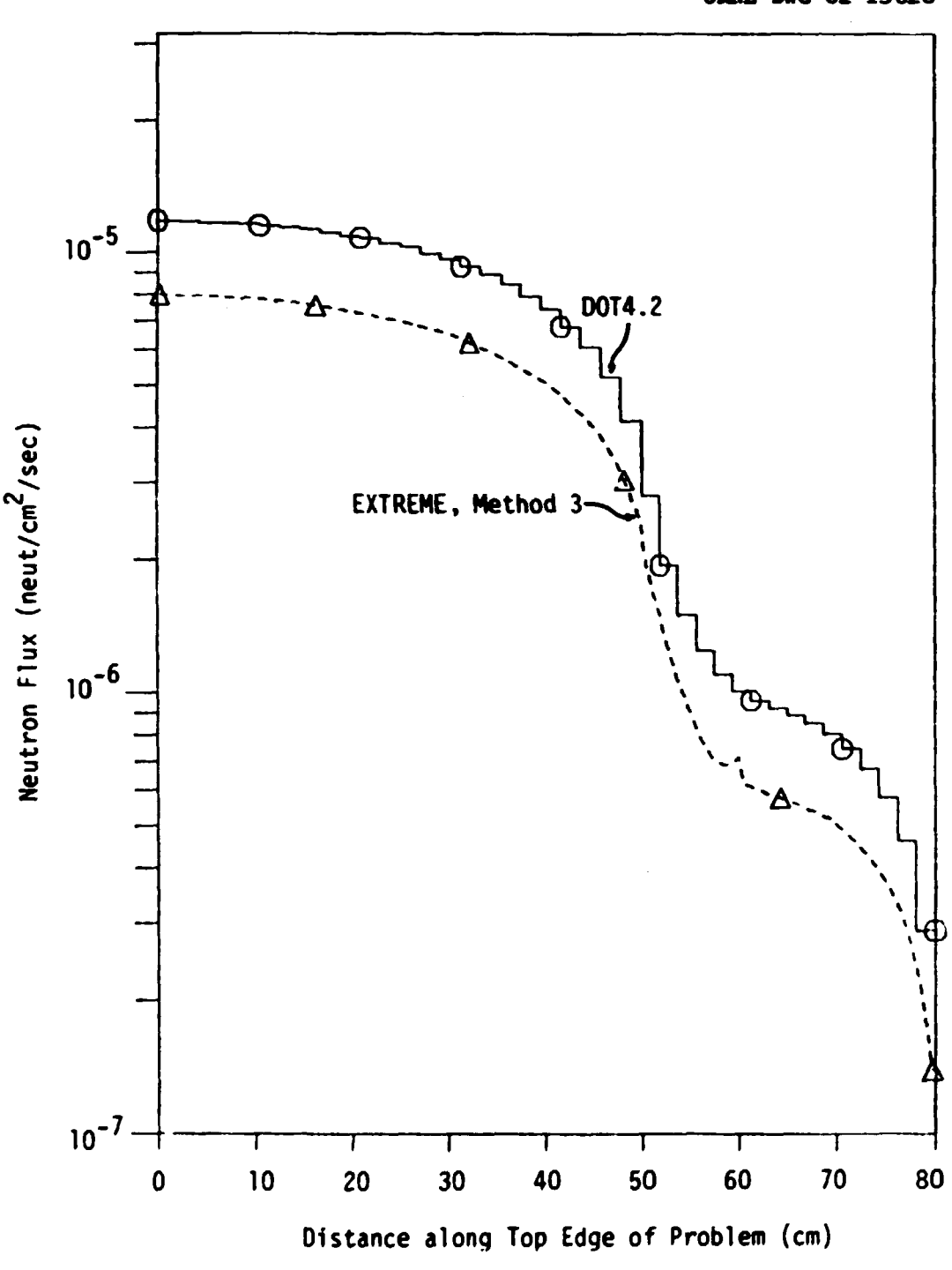


Figure 22. Plot of the Flux along the Top Edge of Sample Problem 6 as Calculated by EXTREME and DOT4.2

CHAPTER 6

CONCLUSIONS

This research project has consisted of the development and evaluation of a new modal expansion treatment of the spatial variation of the neutron flux within the framework of a two-dimensional Cartesian geometry, multigroup, discrete ordinates computer code. The code EXTREME, which implements this new spatial treatment, was compared with the code DOT4.2, which uses a finite difference spatial treatment and is in general use throughout the world. The sample problems chosen for comparison spanned a range of calculations commonly encountered in the nuclear analysis field--reactor eigenvalue cell and lattice problems and also fixed source and boundary source shielding problems.

The full expansion, shaped leakage version of EXTREME (Method 3) was consistently able to deliver very accurate solutions of both types of problems using up to 40 times less computer time than DOT4.2 for the same accuracy (e.g., within 1% error). The results for the separable interior flux, shaped leakage version (Method 2) were nearly as good as the third method for the eigenvalue problems (Sample Problems 2 and 3) and the simple shielding problem (Sample Problem 1), but did not calculate the realistic shielding problems well. The flat leakage version (Method 1) showed no advantage over DOT4.2 except for the total leakage (but not the edge flux shape) of the simple shielding problem. The results for all of the problems considered indicate that the use of exponential modal interior and edge spatial

expansions (Method 3) is a more efficient method of calculating very accurate spatial neutron flux variations than the traditional use of fine-mesh spatial schemes. It is recommended, of course, that the method be tested further by calculating an even wider range of problems, including detailed streaming calculations and very finely resolved energy spectrum calculations involving many energy groups. Additional recommendations for future work are presented in the next chapter.

CHAPTER 7

RECOMMENDATIONS FOR FUTURE WORK

This study has demonstrated the effectiveness of the use of exponential modal expansions in a prototype two-dimensional Cartesian multigroup discrete ordinates computer code. The possibly fruitful areas of research that are suggested by the results of this study fall into two major categories: (1) optimization of the new method and (2) extension to other geometries and to time-dependent problems.

A. Optimization of the Cartesian x-y Nodal Method

As was stated in Chapter 1, the FORTRAN code which implemented the exponential spatial expansion discrete ordinates treatment is only a prototype, not a production code. The current version of EXTREME could most likely be improved along several lines within the present scope of Cartesian geometry static neutron flux calculations:

1. Most obvious is the fact that the programming itself could be improved through the use of more efficient algorithms, ASSEMBLER language streamlining of the most often used subroutines, etc.

2. A second pressing need is for improved data management of the fairly large number of precalculated expansion coefficients, in order for the code to handle larger problems.

3. The many particular options which have evolved over the years of practical experience with codes like DOT4.2 could be added to EXTREME. These include the use of variable mesh spacing, different quadratures in different spatial regions, and incorporation of the many improvements in coarse mesh rebalance techniques.

4. Several types of improvements are suggested by the unique structure of the exponential expansion methods.

One of these is the possibility of allowing each repeating node type to have its own set of expansion modes rather than specifying a single mode set for the problem. Implementing this idea would involve the use of transfer matrices to convert expansion coefficients from each individual mode set to those of all neighbors which share a boundary with the node in question. This would add to the storage requirements and computing time of EXTREME, but would allow the spatial expansions to be more closely tailored to the problem.

Once the use of many mode sets is introduced, the actual choice of the mode sets, which was shown in Sample Problems 1 and 4 to strongly influence the accuracy of shielding calculations, would become even more crucial. A second possible improvement, therefore, is the implementation of iterative updates to the mode sets. In such an approach, the calculation could begin with a preselected set of modes and proceed to a relaxed convergence criterion (possibly 1%). At this point, the calculated flux variation across the spatial nodes could be used to calculate more appropriate choices of the modes. This technique would also probably use more computer time than the present version of EXTREME because the calculation would have to, in effect, start over with the new mode sets. (Of course, the partially converged results in the original expansions could be used to generate a flux guess in the new mode sets.)

A third possible improvement suggested by the structure of the nodal treatment is the development of acceleration techniques

appropriate to nodal methods. For example, although it is true that the coarse mesh rebalance method does not work very well for large variations of the flux, conventional finite difference methods can effectively accelerate by rebalancing on the fine mesh. However, this cannot be done by nodal methods in which the fine spatial variation is given in a modal expansion rather than a fine mesh. Nodal methods would be accelerated more effectively by a rebalance technique in which the flux corrections on the coarse mesh were also given as a modal expansions. Unfortunately, no such method is currently available.

B. Extension to Other Geometries and to Time Dependence

Unfortunately for the current development, not all nuclear analysis problems fit in the framework of static two-dimensional Cartesian geometry calculations. It would be very useful to be able to calculate other problem configurations with exponential spatial expansions.

1. Curvilinear Geometry

Calculations of the neutron flux in geometry systems other than Cartesian introduce additional terms in the basic equation due to the fact that the directional parameters μ, η, and ξ change as a neutron travels between collision sites. This extra angular transfer term can be seen in the cylindrical (r,z) geometry analogue to Eq. 4.11:

$$\mu \frac{\partial \psi(r,z)}{\partial r} - \frac{\xi}{r} \frac{\partial \psi(r,z)}{\partial \Omega} + \eta \frac{\partial \psi(r,z)}{\partial z} + \sigma_t \psi(r,z) = S(r,z), \qquad (7.1)$$

where for this system:

$$\mu = \bar{\Omega} \cdot \hat{i} ,$$

$$\eta = \bar{\Omega} \cdot \hat{k} ,$$

θ is the azimuthal angle of $\bar{\Omega}$ about the z-axis,

and $\hat{i}$ and $\hat{k}$ are the unit vectors in the r and z directions, respectively.

In the DOT4.2 code, this extra angular complexity is handled by rewriting Eq. 7.1 in conservative form:

$$\mu\frac{\partial r\psi}{\partial r} + \eta\frac{\partial r\psi}{\partial z} - \frac{\xi}{r}\frac{\partial r\psi}{\partial \theta} + \sigma_t r\psi = rS, \tag{7.2}$$

integrating over θ for a particular discrete ordinate segment of the unit angular sphere, and applying a constraint similar to the spatial constraints to relate the angular average to the angular endpoint values. (This procedure causes neighboring angular segments to be coupled in the same way as neighboring spatial segments.) Before the spatial variables are dealt with, the use of a step approximation:

$$\psi_{M+1}(r,z) = \psi_m(r,z)$$

where m represents the discrete ordinate direction and M+1 denotes the angular boundary between angles m and m+1, results in the equation:

$$\mu\frac{\partial(r\psi_m)}{\partial r} + \eta\frac{\partial(r\psi_m)}{\partial z} + \left(\sigma_t - \frac{\xi}{\Delta\theta}\right)(r\psi_m) = rS - \frac{\xi}{\Delta\theta}\psi_{m-1} . \tag{7.3}$$

When the spatial treatment is added to this structure, the term is, in effect, treated as a change in the total cross-section.

Using the same approach in the exponential expansion nodal methods would pose a problem because the constantly changing value of

σ_t would cause the precalculated expansion coefficients E_{gm}^{xn} to change also. This problem could be avoided by using the previous iteration value of $\psi_{m+1}(r,z)$, i.e.,

$$\mu\frac{\partial(r\psi_m^{n+1})}{\partial r} + \eta\frac{\partial(r\psi_m^{n+1})}{\partial z} + r\sigma_t\psi_m^{n+1} = rS^{n+1} - \frac{\xi}{\Delta\theta}(\psi_m^n - \psi_{m-1}^n) , \qquad (7.4)$$

in which n is used as an inner iteration counter. Switching the angular transfer terms from the left side to the right side of the equation would most likely effect the rate of convergence, however, so the stability of the resulting iterative procedure would need to be studied.

In principle, however, the same approach could be used for the angular transfer terms of the other orthogonal curvilinear geometry systems (e.g., cylindrical r-0, spherical, toroidal, etc.).

2 Three-Dimensional Geometry

The exponential expansion methods described in Chapter 4 could also be applied to three-dimensional Cartesian geometry configurations. The shaped leakage approaches, however, would become correspondingly more complicated, as it would be necessary to generate and store transfer matrices from two-dimensional surfaces to other surfaces and possibly to the three-dimensional interior of the node. The number of combinations of flat, separable, or full expansion representations of the edge and interior fluxes would make such a study a rich research project.

The simplest case, the flat leakage, separable one-dimensional interior flux approximation, could be developed fairly easily from the current versions of EXTREME.

Also, the curvilinear considerations discussed in the previous section could be applied to the three-dimensional case if they work for the two-dimensional case, allowing geometric systems such as cylindrical (r-θ-z) or spherical (r-θ-φ).

3. Time Dependence

The introduction of time as the third (or fourth) dimension poses the same mathematical problem, in principle, as the addition of another spatial dimension. For example, for the full expansion interior flux treatment (Method 3), the flux can be found as:

$$\psi(x,y,t) = \psi_1(x,y,t) + \psi_2(x,y,t)$$

where

$$\psi_1(x,y,t,) = \sum_{m=1}^{N} \sum_{n=1}^{N} \sum_{p=1}^{T} \frac{S_{mnp}}{\mu\alpha_m+\eta\beta_n+\frac{\gamma_p}{v}+\sigma_t} \exp(\alpha_m x+\beta_n y+\gamma_p t)$$

and

$$\psi_2(s) = \psi_2(0)\ \exp(-\sigma_t s), \text{ with}$$

$$s = \frac{x-x_0}{\mu} = \frac{y-y_0}{\eta} = v(t-t_0)$$

v = neutron speed

T = number of time expansion modes.

The transfer matrices would be more complicated than in the static case, but for many cases of interest (e.g., time-dependent shielding) the cross-section data would be independent of time. Also, it is interesting to note that for computations in which delayed neutrons are important, the time-dependence of the resulting "source" at time t is generally exponential in time, which would allow the method the advantages discussed in Appendix C.

LIST OF REFERENCES

1. Henry, A. F., Nuclear-Reactor Analysis, MIT Press, Cambridge, Mass. (1975).

2. Rhoades, W. A.,Simpson, D. B., Childs, R. L. and Engle, W. W., Jr., "The DOT-IV Two-Dimensional Discrete Ordinates Transport Code with Space-Dependent Mesh and Quadrature," ORNL/TM-6529, Oak Ridge National Laboratory (Jan. 1979).

3. Lathrop, K. D. and Brinkley, F. W., "TWOTRAN II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport," Los Alamos Scientific Laboratory Report LA-4848-MS (July 1973).

4. Finnemann, H. and Wagner, M. R., "The Nodal Expansion Method, A New Computational Technique for the Solution of Multidimensional Neutron Diffusion Problems," IAEA Specialist's Meeting on Methods of Neutron Transport Theory in Reactor Calculations, Bologna, Nov. 1975.

5. Maeder, C., "A Nodal Diffusion Method with Legendre Polynomials," Proceedings of the Conference on Advances in Reactor Physics, Gatlinburg, TN, (April, 1978).

6. Shober, R. A. and Henry, A. F., "Transactions American Nuclear Society, Vol. 24, p. 192 (1976).

7. Wagner, M. R., "A Nodal Discrete-Ordinates Method for the Numerical Solution of the Multidimensional Transport Equation," Proceedings of the American Nuclear Society Topical Meeting on Computational Methods in Nuclear Engineering, Williamsburg, VA, Vol. 2, pp. 4-117 to 4-133 (April, 1979).

8. Lawrence, R. D. and Dorning, J. J., "A Discrete Nodal Integral Transport Theory for Multidimensional Reactor Physics and Shielding Calculations," Proceedings of the American Nuclear Society Topical Meeting, Sun Valley, Idaho, Sept., 1980.

9. Walters, W. F. and O'Dell R. D., "Nodal Methods for Discrete-Ordinates Transport Problems in (x,y) Geometry," Proceedings of the International Topical Mtg. Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, Munich, Vol. 1, pp. 115-129 (April, 1981).

10. Larsen, E. W. and Alcouffe, R. E., "The Linear Characteristic Method for Spatially Discretizing the Discrete-Ordinates Equations in (X,Y)-Geometry," Proceedings of the International Topical Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, Munchen, Vol. 1, pp. 99-113 (1981).

11. Filippone, W. L., "Particle Transport Calculations with the Method of Streaming Rays," accepted for publication, Nuclear Science and Engineering (1981).

12. Askew, J. R., "A Characteristic Formulation of the Neutron Transport Equation in Complicated Geometries," AEEEW-M 1108 (1972).

13. Greene, N. M., et al, "AMPX: A Modular Code System for Generating Coupled Multigroup Neutron-Gamma Libraries from ENDF/B," ORNL/TM-3706 (March 1976).

14. Weisbin, C. R., et al, "MINX: A Multigroup Interpretation of Nuclear X-Sections from ENDF/B," Los Alamos Scientific Laboratory Report LA-6486-MS (Sept. 1976).

15. Carlson, B. G., "Solution of the Transport Equation by S_n Approximations," Los Alamos Scientific Laboratory Report LA-1599 (Oct. 1953).

16. Greenspan, H., Kelber, C. N., and Okrent, D., Computing Methods in Reactor Physics, Gordon and Breach, New York (1968).

17. Wagner, M. R., "The Application of the Block Inversion Technique to Multidimensional Reactor Calculations," Proceedings of the Conference on the Effective Use of Computers in the Nuclear Industry, April 21-23, 1969, Knoxville, TN.

18. Alcouffe, R. E. and Larsen, E. W., "A Review of Characteristic Methods Used to Solve the Linear Transport Equation," Proceedings of the International Topical Meeting on Advances in Mathematical Methods for the Solution of Nuclear Engineering Problems, Munchen, Vol. 1, pp.3-16 (1981).

19. John, F., Partial Differential Equations, Springer-Verlag, New York (1978).

20. Lillie, R. A. and Robinson, J. C., "A Linear Triangle Finite Element Formulation for Multigroup Neutron Transport Analysis with Anisotropic Scattering," Oak Ridge National Laboratory Report ORNL/TM-5281 (May 1976).

21. "Argonne Code Center: Benchmark Problem Book," ANL-7416, Suppl.1, Argonne National Laboratory (Dec. 1972).

22. Ullo, J., Schlumberger-Doll Research Laboratory, personal communication (Sept. 1981).

23. Rhoades, W. A., Oak Ridge National Laboratory, personal communication (November 1981).

APPENDICES

APPENDIX A

DEVELOPMENT OF THE EXPONENTIAL EXPANSION METHOD IN ONE DIMENSION

The exponential expansion method can be derived for the Boltzmann transport equation for one-dimensional slab geometry, discrete angle i, and particular energy group g,

$$\mu_i \frac{d\psi_i(x)}{dx} + \sigma_i^g \psi_i^g(x) = \sum_{j\neq i} w_j \sigma_{s,g}^{j\to i} \psi_j^g(x) + Q_i^g(x), \tag{A.1}$$

where $\sigma_i^g = \sigma_t^g - \sigma_{s,g}^{i\to i}$.

Equation A.1 has the solution

$$\psi_i^g(x) = \psi_i^g(x_0)\exp[\sigma_i^g(x_0-x)/\mu_i] + \frac{\exp(-\sigma_i^g x/\mu_i)}{\mu_i} \cdot \int_{x_0}^{x} dx' \Big[\sum_{j\neq i} w_j \sigma_{s,g}^{j\to i} \psi_j^g(x') + Q_i^g(x') \Big] \exp(\sigma_i^g x'/\mu_i) \tag{A.2}$$

with known value $\psi_i^g(x_0)$.

Approximating $\psi_i^g(x)$ and $Q_i^g(x)$ in terms of the exponential expansion set $\exp[\alpha_n(x-x_1)]$, n=1,2,3,...,N, gives

$$\psi_i^g(x) = \sum_{n=1}^{N} \psi_{i,g}^n \exp[\alpha_n(x-x_1)], \tag{A.3}$$

where

$$\psi_{i,g}^n = \Big[\psi_i^g(x_0) - \sum_{m=1}^{N} \frac{\sum_{j\neq i} w_j \sigma_s^{j\to i} \psi_{j,g}^m + Q_{i,g}^m}{\mu_i\alpha_m + \sigma_i^g}\Big] \cdot \exp[\sigma_i^g(x_0-x_1)/\mu_i] E_i^m + \frac{\sum_{j\neq i} w_j \sigma_s^{j\to i} \psi_{j,g}^m + Q_{i,g}^m}{\mu_i\alpha_m + \sigma_i^g} . \tag{A.4}$$

This expression provides a relationship between the source and angular flux expansion coefficients, allowing the system flux coefficients to be solved iteratively. The integers between -(N-1)/2 and (N-1)/2 have been successfully used for the values of $\alpha_1, \alpha_2, \ldots, \alpha_N$.

The method was used to solve a deep-penetration shielding problem using the data in Table A-1. The results were compared with the results using the ANISN one-dimensional discrete ordinates code.[23] The problem employed two energy groups, S_4 quadrature, and a convergence criterion of 10^{-4}. The weighted difference option was used on the ANISN runs. The current was calculated by both ANISN and 1-D EXTREME exiting out of a 100-cm slab (~40 mean-free-paths) of a concrete-like homogeneous medium from a unit incident current.

Comparisons of the results of EXTREME-1D and ANISN for the sample problem are shown in Table A-2. For an error of ~0.1%, EXTREME-D ran over four times faster than ANISN. These results indicate that the exponential expansion method should be able to solve other deep-penetration shielding problems similar to this one with a considerable time savings relative to ANISN.

Table A-1. Data for 1-D Slab Problem

Energy Group g	σ_t	$\sigma_s^{g \to g}$	$\sigma_s^{g-1 \to g}$
1	0.364750	0.350790	0.013366
2	0.436903	0.426550	0

Table A-2. Comparison of the 1-D Method with ANISN For the Sample Shielding Problem

Code	Space Points	Expansion Modes	$J^{out} \times 10^6$	% Error	CPU Time (seconds)
ANISN	50	-	7.49192	-4.230	0.48
ANISN	100	-	7.73695	-1.100	0.62
ANISN	200	-	7.79783	-0.332	1.02
ANISN	400	-	7.81226	-0.137	1.80
ANISN	800	-	7.81763	-0.068	3.36
ANISN	∞	-	7.823	0	-
EEM	20	3	7.77193	-0.652	0.30
EEM	20	5	7.83397	0.140	0.40
EEM	20	7	7.82411	-0.014	0.50
EEM	20	9	7.82257	-0.005	0.69
EEM	20	13	7.82254	-0.005	0.76

Standard taken from extrapolated ANISN values

APPENDIX B

DEVELOPMENT OF ANISOTROPIC SCATTERING TREATMENT

As mentioned in Chapter 2, the angular dependence of the scattering cross-section is given in terms of a Legendre expansion in angle, i.e.

$$\sigma_{s,g'\to g}(\Omega'\to\Omega) = \sum_{k=0}^{\infty} \sigma_{s,g'\to g}^{k} \; P_k \; (\Omega'\cdot\Omega) \; .$$

Substituting this into the angle-integrated scattering source relation:

$$S_s^g \; (x,y,\Omega) = \frac{1}{4\pi} \sum_{g'}^{G} \int_{\Omega'} \psi_{g'} \; (x,y,\Omega') \; \sigma_{s,g'\to g}(\Omega'\to\Omega) d\Omega'$$

gives:

$$S_s^g(x,y,\eta,\theta) = \frac{1}{4\pi} \sum_{g'}^{G} \int_0^{2\pi} d\theta' \int_{-1}^{1} d\eta' \sum_{k=0}^{\infty} \psi_{g'}(x,y,\eta',\theta') \; \sigma_{s,g'\to g}^{k} P_k(\mu_0) \qquad (B.1)$$

where $d\Omega'$ has been divided into its components $d\eta'$ and $d\theta'$ and $\mu_0 = \mu_0 = \Omega'\cdot\Omega$.

Evaluation of this integral requires that the scattering cross-section be described in terms of the variables η, η', θ, and θ' instead of μ_0. From the addition theorem for Legendre polynomials, this is:

$$P_k(\mu_0) = P_k(\eta) \; P_k(\eta') + 2 \sum_{m=1}^{k} \frac{(k-m)!}{(k+m)!} \; P_k^m \; (\eta) \; P_k^m \; (\eta') \cos m(\theta-\theta') \; .$$

The cosine terms can be expanded to give:

$$P_k(\mu_0) = P_k(\eta) \; P_k(\eta') + 2 \sum_{m=1}^{k} \frac{(k-m)!}{(k+m)!} \; P_k^m \; (\eta) \; P_k^m \; (\eta')$$

$$\cdot \; (\cos m\theta \cos m\theta' + \sin m\theta \sin m\theta') \; .$$

Using this expression, Eq. B.1 becomes:

$$S_s^g(x,y,\eta,\theta) = \sum_{g'}^{G} \int_0^{2\pi} d\theta' \int_{-1}^{1} d\eta' \sum_{k=0}^{\infty} \psi_{g'}(x,y,\eta',\xi') \, \sigma_{sg'\to g}^k \qquad (B.2)$$

$$\cdot \left[P_k(\eta) \, P_k(\eta') + 2 \sum_{m=1}^{k} P_k^m(\eta) \, P_k^m(\eta')(\cos m\theta \cos m\theta' \right.$$

$$\left. + \sin m\theta \sin m\theta')\right] .$$

Due to symmetry of x-y geometry, the flux is an even function of θ'; therefore, all terms with sin $m\theta'$ vanish under integration over θ'. Equation B.2 can be rewritten as:

$$S_s(x,y,\eta,\theta) = \sum_{g'}^{G} \sum_{k=0}^{\infty} \sum_{m=0}^{k} \sigma_{s\, g'\to g}^k \, {}^{m}A_k(\eta_d) \cos m\theta_d \qquad (B.3)$$

$$\cdot \int_{-1}^{1} d\eta' \int_0^{2\pi} d\theta' \, \psi_{g'}(x,y,\eta',\theta') \, A_k^m(\eta_d') \cos m\theta_d'$$

where

$$A_k^m(\eta_d) \cos m\theta_d = P_k(\eta_d) \qquad \text{if } m=0$$

$$= \left[2 \frac{(k-m)!}{(k+m)!}\right]^{1/2} P_k^m(\eta_d) \cos m\theta_d \text{ if } m \neq 0$$

In terms of discrete ordinates, if angle d corresponds to direction (η,θ) and d' to (η',θ') and the angular integration is replaced by a weighted quadrature sum, the approximation to Eq. B.3 is:

$$S_s^d(x,y) = \sum_{g'}^{G} \sum_{k=0}^{\infty} \sum_{m=0}^{k} A_d^{k,m} \, \sigma_{s,g'\to g}^k \, \phi_{g'}^{l,m}(x,y)$$

where

$$A_d^{k,m} = P_k(n,d) \quad \text{if } m=0$$

$$= \left[2 \frac{(k-m)!}{(k+m)!}\right]^{1/2} P_k^m(\eta_d) \cos m\theta_d \quad \text{if } m\neq 0,$$

and

$$\phi_g^{k,m}(x,y) = \sum_{d'} w_{d'} \, A_{d'}^{k,m} \, \psi_g^{d'}(x,y) .$$

APPENDIX C

COMPARISON OF EXPONENTIAL EXPANSIONS VERSUS LEGENDRE POLYNOMIALS IN REPRESENTING EXP(-AX), 0<A<5

To compare the effectiveness of the function set $\exp(\alpha_n x)$ versus the Legendre polynomial set $P_n(x)$ in approximating functions of the form:

$$f(x) = \exp(-ax) \ , \ 0<x<1 \ ,$$

the function was approximated in the two function sets; i.e.,

$$\exp(-ax) \approx \sum_{n=0}^{N-1} L_n^a P_n(x)$$

$$\approx \sum_{n=1}^{N} E_n^a \exp(\alpha_n x) \ , \ \alpha_n = 0,-1,1,-2,2,\ldots \ .$$

The coefficients of the exponential expansion were calculated by using the procedure of Appendix D to minimize the least-squares norm:

$$L^2[\tilde{f}(x)] = \int_0^1 [f(x) - \tilde{f}(x)]^2 dx \ .$$

(The Legendre expansion minimizes the same norm for the x^n basis function set.)

The value of this norm for the two expansions were compared as a function of a for values ranging from 0 to 5. Fig. C-1 shows the comparisons for N=3. As can be seen in this figure, the exponential expansion set can represent this type of function slightly better than the Legendre polynomials for values of greater than 1. Below 1, both expansions can represent the function within about 1%.

ORNL-DWG 82-13827

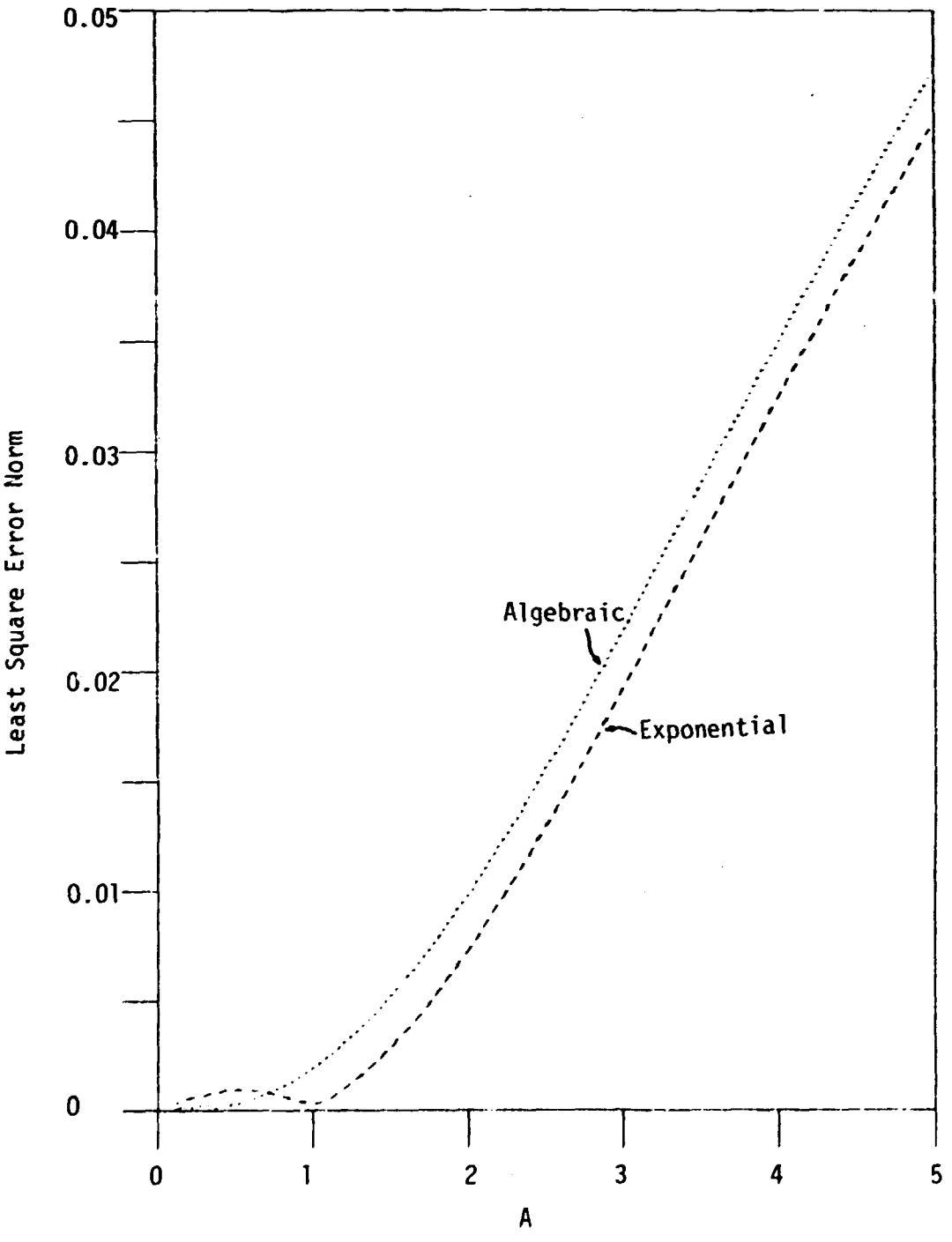


Figure C-1. Comparison of Least Square Error Norm for Exponential Expansions versus Algebraic Expansions in Representing Exp(-ax) for 0<A<5

APPENDIX D

COMPARISON OF OPERATION COUNT FOR THE INNER ITERATION OF A MODAL EXPANSION CHANNEL METHOD FOR EXPONENTIAL VERSUS ORTHONORMAL BASIS FUNCTION SETS

The relationship between the inner iteration source and the neutron flux for a given x-dimension channel calculation is given by a simplified version of Eq. 4.1:

$$\psi(x) = \psi(0)\exp(-ax) + \frac{\exp(-ax)}{\eta}\int_0^x S(x')\exp(ax')dx' \qquad (D.1)$$

where the source S(x) is taken to include all sources external to $\psi(x)$, including transverse leakage and $a = \sigma_t/\mu$. If S(x) is expanded in a basis function set $f_n(x)$, n=1,2,...,N:

$$S(x) \approx \sum_m S_n f_n(x)$$

and this expansion is substituted into Eq. D.1, the equation becomes:

$$\psi(x) = \psi(0)\exp(-ax) + \sum_n S_n\exp(-ax)\int_0^x f_n(x')\exp(ax')dx' \qquad (D.2)$$
$$\approx \sum_m \psi_m f_m(x) \quad .$$

For the case in which $f_n(x)$ is a function set orthonormal over the range $(0,\Delta x)$ with a unit weight function, the coefficients can be found from the relation:

$$\psi_m = \int_0^{\Delta x} f_m(x')\psi(x')dx'$$

$$= \psi(0)\int_0^{\Delta x} f_m(x')\exp(-ax')dx + \sum_n \frac{S_n}{\eta}\int_0^{\Delta x} f_m x'')\exp(-ax)\int^{x''} f_n(x')\exp(ax')dx'dx''$$

$$\psi_m = \psi(0)\ G_m + \sum S_n\ H_{mn} \qquad (D.3)$$

where

$$G_m = \int_0^{\Delta x} f_m(x')\exp(-ax)dx'$$

and

$$H_{mn} = \frac{1}{n}\int_0^{\Delta x} f_m(x'')\exp(-ax'')\int_0^{x''} f_n(x')\exp(ax')dx'dx'' \quad .$$

The operation count for this equation is 1+2N; therefore, the inner iteration count for all N modes would be $N+2N^2$.

For the case in which the $f_n(x)$ are exponential functions, $\exp(\alpha_n x)$, Eq. D.2 becomes:

$$\psi(x) = \psi(0)\exp(-ax) + \sum_n \frac{S_n}{\mu(\alpha_n+a)}[\exp(\alpha_n x) - \exp(-ax)]$$

$$= [\psi(0) - \sum_k \frac{S_k}{\mu(\alpha_k+a)}]\exp(-ax) + \sum_n \frac{S_n}{\mu(\alpha_n+a)}\exp(\alpha_n x)$$

$$= \sum_m \psi_m \exp(\alpha_m x) \quad .$$

If the procedure in Appendix E is used to find the coefficients of the expansion:

$$\exp(-ax) = \sum_m E_m \exp(\alpha_m x),$$

the ψ_m can be found from the relation:

$$\psi_m = [\psi(0) - \sum_k P_k S_k]E_m + P_m S_m \qquad (D.4)$$

where

$$P_m = \frac{1}{n(\alpha_m+a)} \quad .$$

The operation count for finding the ψ_m for all N values is 5N. (Note that the term in brackets in Eq. D.4 is independent of m and need be calculated only once.)

Therefore, use of orthonormal expansions functions requires a factor of 0.2+0.4N times the operations and a factor of 1+N times the storage requirement of the exponential functions.

Although this comparison would indicate that there is an operation count advantage for the orthonormal algebraic set if N=1 or 2, Eq. C.3 can be reformulated as:

$$\psi_m = \psi(0)E_m + \sum_n Q_{mn}\, S_n$$

where

$$Q_{mn} = (\delta_{mn}-E_m)P_m \ ,$$

eliminating the difference in operation count and storage requirements.

APPENDIX E

PROJECTION OF EXPONENTIAL FUNCTIONS ONTO THE EXPONENTIAL BASIS FUNCTION SETS

In order to set up the iterative procedure of determining the flux expansion coefficients of Eqs. 4.7, 4.8, 4.17, 4.18, and 4.22, several types of functions must be expressed in terms of our exponential basis functions. The procedure for each of the cases is the same:

1. Find a set of functions, $g_m(x)$, orthonormal over the range $0<x<1$ which are linear combinations of the basis function sets $\exp(\alpha_n x)$, i.e.

$$g_m(x) = \sum_{n=1}^{m} g_m^n \exp(\alpha_n x);$$

2. Approximate the given function, $f(x)$, as a linear combination of the orthonormal functions:

$$f(x) \simeq \sum_{m=1}^{M} \tilde{f}_m g_m(x)$$

where

$$\tilde{f}_k = \int_0^1 f(x) g_k(x) dx \; ; \quad \text{and}$$

3. Find the coefficients to the related expansion:

$$f(x) = \sum_{n=1}^{M} f_n \exp(\alpha_n x)$$

$$= \sum_{m=1}^{M} \tilde{f}_m g_m(x)$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{n} \tilde{f}_m g_m^n \exp(\alpha_n x) \ .$$

Therefore,

$$f_n = \sum_{m=1}^{n} f_m g_m^n \qquad .$$

The set of orthonormal functions $g_m(x)$ can be found by applying the Gram-Schmidt[19] procedure to the basis function set $\exp(\alpha_n x)$, $n= 1,2,...N$. The Gram-Schmidt procedure allows one to start with a basis function set $e_m(x)$, $m=1,2,...M$ and determine a set of functions:

$$u_n(x) = \sum_{m=1}^{n} u_n^m e_m(x)$$

which are orthonormal over a given range with respect to some inner product $\langle .,. \rangle$, i.e.

$$\langle u_i(x), u_j(x) \rangle = \delta_{ij} \ .$$

The functions $u_n(x)$, $n=1,2,...$ N are found to be:

$$u_n(x) = \langle \tilde{u}_n(x), \tilde{u}_n(x) \rangle^{-2} \cdot \tilde{u}_n(x) \ , \qquad \text{(E.1)}$$

where

$$\tilde{u}_n(x) = e_n(x) - \sum_{m=1}^{n-1} \langle u_m(x), e_n(x) \rangle \cdot u_m(x) \ .$$

Equation E.1 can be evaluated in terms of the basis functions $e_n(x)$. To do this, one begins by simplifying the norm of the equation:

$$\langle \tilde{u}_n(x), \tilde{u}_n(y) \rangle = \langle e_n(x) - \sum_{m=1}^{n-1} \langle u_m(x), e_n(x) \rangle, e_n(x) - \sum_{m=1}^{n-1} \langle u_m(x), e_n(x) \rangle \rangle$$

$$= \langle e_n(x), e_n(x)\rangle - 2\sum_{m=1}^{n-1} \langle u_m(x), e_n(x)\rangle^2$$

$$- \sum_{m=1}^{n-1}\sum_{k=1}^{n-1} \langle u_m(x), e_n(x)\rangle\langle u_k(x), e_n(x)\rangle\langle \tilde{u}_m(x), \tilde{u}_k(x)\rangle$$

$$= \langle e_n(x), e_n(x)\rangle - \sum_{m=1}^{n-1} \langle u_m(x), e_n(x)\rangle^2$$

$$= \langle e_n(x), e_n(x)\rangle - \sum_{m=1}^{n-1} \left[\sum_{k=1}^{m} u_m^k \langle e_k(x), e_n(x)\rangle\right]^2 .$$

Using this result in Eq. E.1 gives:

$$u_n(x) = u_n^n \cdot [e_n(x) - \sum_{m=1}^{n-1} \langle u_m(x), e_n(x)\rangle u_n(x)]$$

$$= u_n^n \cdot [e_n(x) - \sum_{m=1}^{n-1} \{\sum_{k=1}^{m} u_m^k \langle e_k(x), e_n(x)\rangle\} \sum_{l=1}^{m} u_m^l e_l(x)]$$

$$= \sum_{j=1}^{n} u_n^j e_n(x) ,$$

where

$$u_n^n = [\langle e_n(x), e_n(x)\rangle - \sum_{m=1}^{n-1} \{\sum_{k=1}^{m} u_m^k \langle e_k(x), e_n(x)\rangle\}^2]^{-2}$$

$$u_n^j = u_n^n \cdot \sum_{m=1}^{j} u_m^j \sum_{k=1}^{m} u_m^k \langle e_k(x), e_n(x)\rangle \text{ , for } j<n .$$

For our application,

$$e_n(x) = \exp(\alpha_n x),$$

$$u_n(x) = g_n(x) = \sum_{m=1}^{n} g_m \exp(\alpha_n x),$$

$$\langle e_k(x), e_l(x)\rangle = \int_0^1 e_k(x) e_l(x) dx = \int_0^1 \exp[(\alpha_k+\alpha_l)x] dx = h_{kl} ,$$

$$h_{kl} = \frac{\exp(\alpha_k+\alpha_l) - 1}{\alpha_k + \alpha_l} \qquad \text{if } \alpha_k \neq -\alpha_l ,$$

$$= 1 \qquad \text{if } \alpha_k = -\alpha_l ,$$

$$g_1^1 = h_{11}^{-2}, \text{ and}$$

$$g_n^j = [\sum_{m=1}^{j} u_m \sum_{k=1}^{m} u_m h_k] \cdot [h_{nn} - \sum_{m=1}^{n-1} (\sum_{k=1}^{n} h_{kn})^2]^{-2}.$$

The choice of the mode set, α_0, α_1, ... α_N are restricted to the family 0, $\pm\alpha_1$, $\pm\alpha_3$, ... $\pm\alpha(N+1)/2$, N odd.

In the derivation of the method, there are only two types of functions which must be represented in the basis function set. The first is the approximation:

$$f(x) = \exp(ax), x_1 < x < x_u \text{ or}$$

$$= 0, \quad 0 < x < x_1 \text{ or } x_u < x < 1,$$

$$= \sum_n E_n \exp(\alpha_n x) .$$

The coefficients E_n can be found from:

$$f(x) = \sum_{m=1}^{n} [\int_0^1 f(x) g_m(x) dx] g_m(x)$$

$$= \sum_{m=1}^{n} [\sum_{k=1}^{m} g_m^k \int_0^1 f(x) \exp(\alpha_k x) dx] \sum_{n=1}^{m} g_m^n \exp(\alpha_n x)$$

$$= \sum_{n=1}^{N} \sum_{m=1}^{n} (\sum_{k=1}^{m} g_m^k \frac{\exp(\{\alpha_k+a\}x_u) - \exp(\{\alpha_k+a\}x_1)}{\alpha_k+a}) g_m^n \exp(\alpha_n x)$$

Therefore,

$$E_n = \sum_{m=1}^{n} (\sum_{k=1}^{m} g_m^k \frac{\exp(\{\alpha_k+a\}x_u) - \exp(\{\alpha_k+a\})x_1}{\alpha_k+a}) g_m^n$$

The second type of function which must be approximated is:

$$f(x,y) = e^{ax+by}, \quad 0<x<x_u, \quad kx<y<1$$

$$= 0 \quad , \text{otherwise}$$

$$= \sum_{m=1}^{M} \sum_{n=1}^{N} G_{mn} \exp(\alpha_m x+\beta_n y)$$

The coefficients G_{mn} can be found from:

$$F(x,y) = \sum_{m=1}^{M} \sum_{n=1}^{N} \left[\int_0^1 dx \int_0^1 dy\ f(x,y) g_m(x) h_n(y)\right] g_m(x) h_n(y)$$

$$= \sum_{m=1}^{M} \sum_{n=1}^{N} \left[\sum_{k=1}^{m} \sum_{l=1}^{n} g_m^k h_n^l \int_0^1 dx \int_0^1 dy f(x,y) \exp(\alpha_k x+\beta_l y)\right]$$

$$\cdot \sum_{p=1}^{m} \sum_{q=1}^{n} g_m^p h_n^q \exp(\alpha_p x+\beta_q y)$$

$$= \sum_{p=1}^{M} \sum_{q=1}^{N} \sum_{m=p}^{M} \sum_{n=q}^{N} \left(\sum_{k=1}^{m} \sum_{l=1}^{n} \frac{g_m^k h_n^l}{b+\beta_l} \left\{ \frac{e^{b+\beta_l}}{a+\alpha_k} \left(e^{(a+\alpha_k)x_u} - 1\right) \right.\right.$$

$$\left.\left. - \frac{e^{(a+\alpha_k+kb+k\beta_l)x_u} - 1)}{a+\alpha_k+kb+k\beta_l} \right\} \right) g_m^p h_n^q \exp(\alpha_p x+\beta_q y)$$

Therefore,

$$G_{pq} = \sum_{m=p}^{M} \sum_{n=q}^{N} \left(\sum_{k=1}^{m} \sum_{l=1}^{n} \frac{g_m^k h_n^l}{b+\beta_l} \left\{ \frac{e^{b+\beta_l}}{a+\alpha_k} \left(e^{(a+\alpha_k)x_u} - 1\right) \right.\right.$$

$$\left.\left. - \frac{e^{(a+\alpha_k+kb+k\beta_l)x_u} - 1}{a+\alpha_k+kb+k\beta_l} \right\} \right) g_m^p h_n^q$$

where $h_n(y)$ is the orthonormal set based on $\exp(\beta_n y)$, $n=1,2,\ldots,N$.

Two examples of the results of this procedure are shown in Figures E-1 and E-2. In both examples the basis function modes

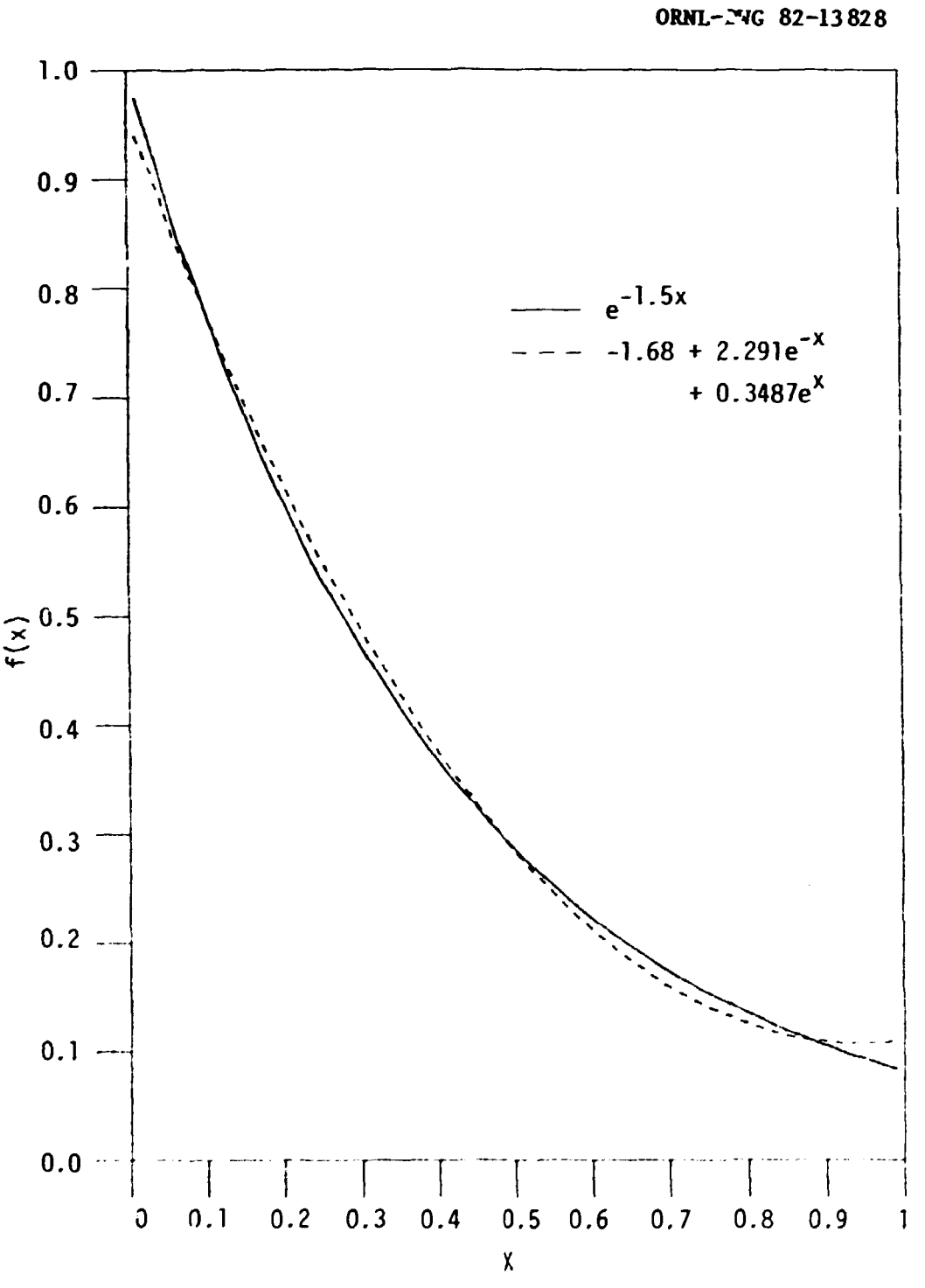


Figure E-1. Approximation of a Continuous Exponential Function Using an Exponential Expansion

ORNL-DWG 82-13829

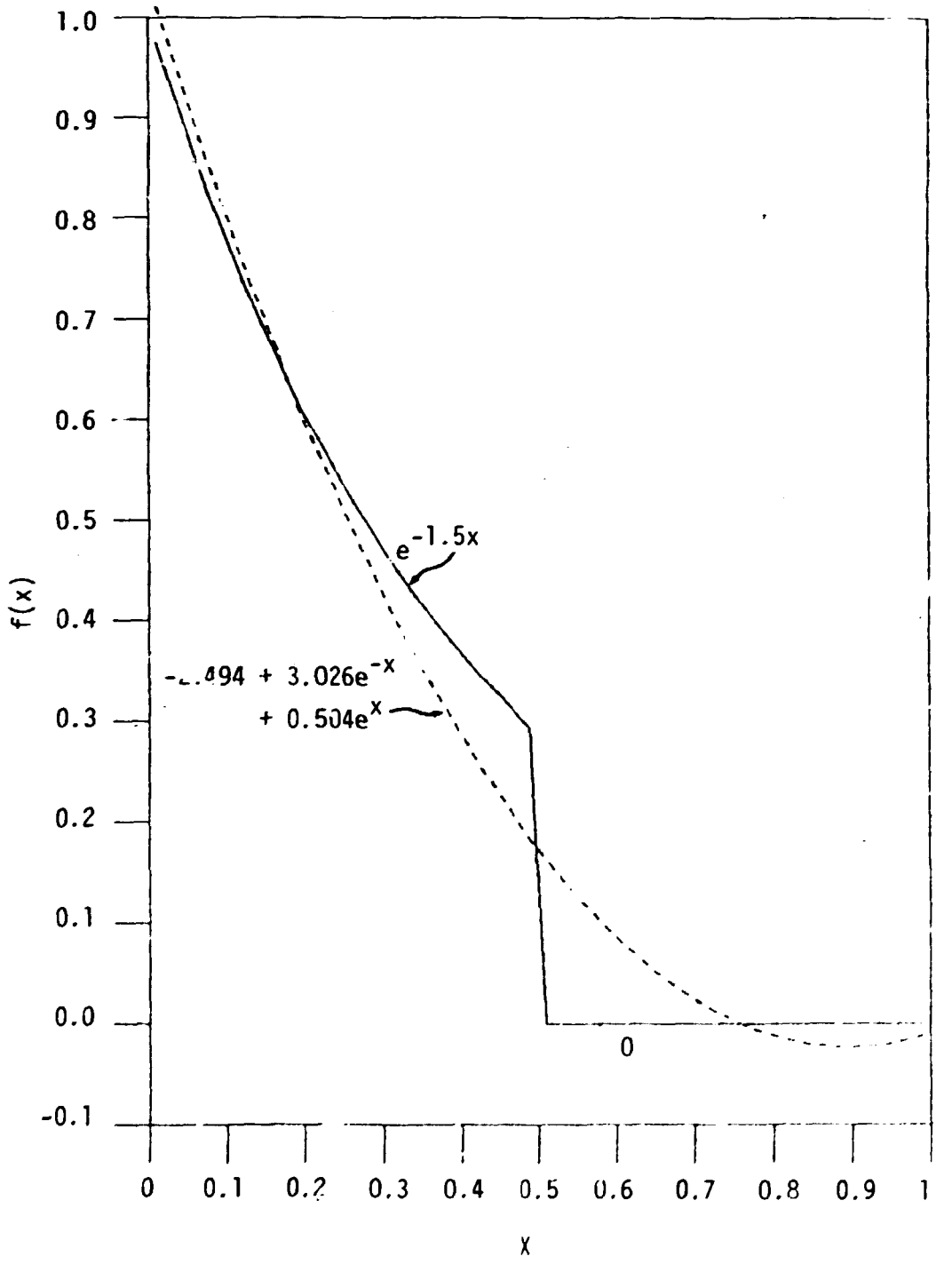


Figure E-2. Approximation of a Discontinuous Exponential Function Using an Exponential Expansion

α_n = 0, -1, 1 are used to represent f(x). In Fig. E-1, the continuous function:

$$f_1(x) = e^{-1.5x},\ 0<x<1$$

is represented. In Fig. E-2, the discontinuous function:

$$f_2(x) = e^{-1.5x},\ 0<x<0.5$$
$$= 0\ ,\ 0.5<x<1$$

is approximated. As expected, the continuous function can be represented more accurately than the discontinuous function.

APPENDIX F

USER'S MANUAL FOR EXTREME

A. Introduction

The primary objective of the development of the code EXTREME was to implement and test the use of a new spatial treatment in the static multigroup discrete ordinates calculation procedure. This version of the code cannot really be classified as a "production" code. Its main shortcomings are in the areas of programming efficiency, solution acceleration, and data management. The first two of these will most likely only be a headache for the user; the last limits the size problem that can be accommodated.

The code can calculate the neutron fluxes with any consistent units of length and time. The directional fluxes are normalized such that, if the flux is isotropic, then the scalar and directional fluxes are equal in measure. This means, of course, that the angular quadrature weights must sum to 1. The specification of anisotropic scattering cross-sections follow the DOT4.2 convention; i.e.,

$$\sigma_s(\vec{\Omega}' \rightarrow \vec{\Omega}) \simeq \sum_k \sigma_{sk} P_k(\vec{\Omega}' \cdot \vec{\Omega}) \quad .$$

Therefore, some standard cross-section files such as ISOTXS and MATXS must be multiplied by 2l+1 for use in EXTREME.

Since the code is only designed to handle Cartesian x-y geometry, there are few restrictions on the quadrature sets. The only requirements are that the weights sum to one and that the necessary reflection angles are present at reflection boundaries; if either of these

is not met, the code will say so and stop. Quadrature directions with zero weights serve no purpose in Cartesian geometry and are not used in the calculation.

B. Functional Flowchart of EXTREME

The following is a functional logic flow chart of EXTREME. The subroutines which implement each step are given in parentheses. Each subroutine mentioned in the outline calls subroutines below it in structure. (The subroutines listed in the Roman numeral headings are called by subroutine CONTRL.)

I. Read problem input (INPUT1, INPUT2).

II. Calculate materials buckling, if desired (BL).

III. Calculate anisotropic scattering weights (PCON).

IV. Calculate necessary constants for first method, if necessary (EXPAND).

- A. Orthonormalize the user-supplied exponential basis function set (ORNORM).
- B. Expand characteristic one-dimensional exponentials in the basis function exponentials (COEF).

V. Calculate necessary constants for second and third methods, if necessary, by solving the two-dimensional characteristic problems within each node type (CFCALC).

- A. Expand the edge fluxes into the basis functions (PEXP).
- B. Expand the interior flux into the basis functions (PEXP2).

VI. Perform iterative calculation (RUN).

- A. Calculate fission source and check outer iteration convergence (OUTER).

B. Calculate scattering source for particular group from all other groups (SOURCE).

C. Perform inner iteration calculation on the inscatter source for three versions (CALC0, CALC1, CALC2).

i. Compute shaped leakage sources, if necessary (LEAK1, LEAK2).

D. Accelerate the inner iteration using coarse-mesh rebalance (ACCEL).

E. Determine whether the inner iteration has converged (CONVER).

VII. Print flux results, balance tables, etc. (PRINT).

C. Input Data Format (This section is taken almost verbatim from Reference 2. It is included herein for completeness.)

With the exception of the title card, all data are read with the FIDO system also used in DOT4.2. The FIDO input method is especially devised to allow the entering or modifying of large data arrays with minimum effort. Special advantage is taken of patterns of repetition or symmetry wherever possible. The FIDO system was patterned after the input method used with the FLOCO coding system at Los Alamos, and was first Applied by Atomics International to the DTF-II code. Since that time, numerous features requested by users have been added, a free-field option has been developed, and the application of FIDO has spread to innumerable codes.

The data are entered in units called "arrays". An array comprises a group of contiguous storage locations which are to be filled

with data at one time. These arrays usually correspond on a one-to-one basis with FORTRAN arrays used in the program. A group of one or more arrays read with a single call to the FIDO package forms a "block", and a special delimiter is required to signify the end of each block. Arrays within a block may be read in any order with respect to each other, but an array belonging to one block must not be shifted to another. The same array can be entered repeatedly within the same block. For example, an array could be filled with "0" using a special option, and then a few scattered locations could be changed by reading in a new set of data for that array. If no entries to the arrays in a block are required but the condition requiring the block is met, the delimiter alone satisfies the input requirement.

Three major types of input are available: fixed-field input, free-field input, and user-field input.

Fixed Field Input--Each card is divided into six 12-column data fields, each of which is divided into three subfields. The three subfields always comprise 2, 1, and 9 columns, respectively.

To begin the first array of a block, an array originator field is placed in any field on a card:

Subfield 1: An integer array identified < 100 specifying the data array to read.

Subfield 2: An array-type indicator:

"$" if the array is integer data,

"*" if the array is single-precision real data, or

"#" if the array is double-precision real data.

Subfield 3: Blank.

Data are then placed in successive fields until the required number of entries has been accounted for.

In entering data, it is convenient to think of an "index" or "pointer" which is under control of the user, and which specifies the position in the array into which the next data entry is to go. The pointer is always positioned at array location #1 by entering the array originator field. The pointer subsequently moves according to the data operator chosen. Blank fields are a special case in that they do not cause any data modification and do not move the pointer.

A data field has the following form:

Subfield 1: The data numerator, an integer < 100. We refer to this entry as N_1 in the following discussion.

Subfield 2: One of the special data operators listed below.

Subfield 3: A nine-character data entry, to be read in F9.0 format. It will be converted to an integer if the array is a "$" array or if a special array operator such as "Q" is being used. Note that an exponent is permissible but not required. If no decimal is supplied, it is assumed to be immediately to the left of the exponent, if any; and otherwise to the right of the last column. This entry is referred to as N_3 in the following discussion.

A list of data operators and their effect on the array being input follows:

"Blank" indicates a single entry of data. The data entry in the third subfield is entered in the location indicated by the pointer, and the pointer is advanced by 1. However, an entirely blank field is ignorred.

"+" or "-" indicates exponentiation. The data entry in the third field is entered and multiplied by 10^{N_1}, where N_1 is the data numerator in the first subfield, given the sign indicated by the data operator itself. The pointer is advanced by 1. In cases where exponent is needed, this option allows the entering of more significant figures than the blank option.

"&" has the same effect as "+" on IBM systems.

"R" indicates that the data entry is to be repeated N_1 times. The pointer is advanced by N_1.

"I" indicates linear interpolation. The data numerator, N_1, indicates the number of interpolated points to be supplied. The data entry in the third subfield is entered, followed by N_1 interpolated entries equally spaced between that value and the data entry found in the third subfield of the next nonblank field. The pointer is advanced by $N_1 + 1$. The field following an "I" field is then processed normally, according to its own data operator. The "I" entry is especially valuable for specifying a spatial mesh. In "$" arrays, interpolated values will be rounded to the nearest integer.

"L" indicates logarithmic interpolation. The effect is the same as that of "I" except that the resulting data are evenly separated in log-space. This is especially convenient for specifying an energy mesh.

"Q" is used to repeat sequences of numbers. The length of the sequence is given by the third subfield, N_3. The sequence of

N_3 entries is to be repeated N_1 times. The pointer is advanced by N_1*N_3. If either N_1 or N_3 is 0, then a sequence of $N_1 + N_3$ is repeated one time only, and the pointer is advanced by $N_1 + N_3$. This feature is especially valuable for geometry specification.

"G" has the same effect as "Q", except that the sign of the sequence is changed each time it is entered.

The "N" option has the same effect as "Q", except that the order of the sequence is reversed each time it is entered. This is valuable for the type of symmetry possessed by quadrature coefficients.

"M" has the same effect as "N", except that the sign of each entry in the sequence is reversed each time the sequence is entered. For example, the entries:

1 2 3 2M2

would be equivalent to:

1 2 3 -3 -2 2 3 .

This option is also useful in entering quadrature coefficients.

"Z" causes $N_1 + N_3$ locations to be set to 0. The pointer is advanced by $N_1 + N_3$.

"C" causes the position of the last array item entered to be printed. This is the position of the pointer, less 1. The pointer is not moved.

"O" causes the point trigger to be turned on. The trigger is originally off. When the trigger is on, each card image is listed as it is read.

"P" causes the print trigger to be turned off.

"S" indicates that the pointer is to skip N_1 positions leaving those array positions unchanged. If the third subfield is nonblank, that data entry is entered following the skip, and the pointer is advanced by $N_1 + 1$.

"A" moves the pointer to the position N_3, specified in the third subfield.

"F" fills the remainder of the array with the datum entered in the third subfield.

"E" skips over the remainder of the array. The array length criterion is always satisfied by an "E", no matter how many entries have been specified. No more entries to an array may be given following an "E", except that data entry may be restarted with an "A."

The reading of data to an array is terminated when a new array origin field is supplied, or when the block is terminated. If an incorrect number of positions has been filled, an error edit is given, and a flag is set which will later abort execution of the problem. FIDO then continues with the next array if an array origin was read. Otherwise, it returns control to the calling program.

A block termination consists of a field having "T" in the second subfield. All entries following "T" on a card are ignored, and control is returned from FIDO to the calling program.

Comment cards can be entered within a block by placing a slash (/) in column 1. Then columns 2-80 will be listed, with column 2 being used for printer carriage control. Such cards have no effect on the data array or pointer.

With free-field input, data are written without fixed restrictions as to field and subfield size and positioning on the card. The options used with fixed-field input are available, although some are slightly restricted in form. In general, fewer data cards are required for a problem, the interpreting print is easier to read, a card listing is more intelligible, the cards are easier to keypunch, and certain common keypunch errors are tolerated without affecting the problem. Data arrays using fixed- and free-field input can be intermingled at will within a given block.

The concept of three subfields per field is still applicable to free-field input, but if no entry for a field is required, no space for it need be left. Only columns 1-72 may be used, as with fixed-field input.

The array originator field can begin in any position. The array identifiers and type indicators are used as in fixed-field input. The type indicator is entered twice, to designate free-field input (i.e., "$$" or "**"). The blank third subfield required in fixed-field input is not required. For example: 31** indicates that array 31, a real-data array, will follow in free-field format.

Data fields may follow the array origin field immediately. The data field entries are identical to the fixed-field entries with the following restrictions:

1. Any number of blanks may separate fields, but at least one blank must follow a third subfield entry if one is used.
2. If both first and second subfield entries are used, no blanks may separate them; i.e., 24S, but not 24 S.

3. Numbers written with exponents must not have imbedded blanks, i.e., 1.0E+4, 1.0E4, 1.0+4, or even 1+4, but not 1.0 E4.
4. In third-subfield data entries, only nine digits, including the decimal but not including the exponent field, can be used, i.e., 123456.89E07, but not 123456.789E07.
5. The Z entry must be of the form: 738Z, not Z738 or 738 Z.
6. The + or - data operators are not needed and are not available.
7. The Q, N, and M entries are restricted: 3Q4, 1N4, or M4, but not 4Q, 4N, or 4M. G is similarly restricted.
8. A field must not span two cards.
9. All items on a card entered after a slash in any column except the first are ignored.

User-field input: If the user follows the array identifier in the array originator field with the character "U" or "V", the input format is to be specified by the user. If "U" is specified, the FORTRAN format to be used must be supplied in columns 1-72 of the next card. The format must be enclosed by the usual parentheses. Then the data for the entire input as to exponents, blanks, etc., apply. If the array data do not fill the last card, the remainder must be left blank.

"V" has the same effect as "U" except that the format read in the last preceding "U" array is used.

D. Input for EXTREME

Title Card (20A4)

BLOCK 1

1$$

1. NMAT = No. of Materials
2. NEG = No. of Energy groups
3. NANG = No. of Angles
4. NX = No. of X intervals for fine mesh
5. NY = No. of Y intervals for fine mesh
6. NZX = No. of X intervals for coarse mesh
7. NZY = No. of Y intervals for coarse mesh
8. NREP = No. of expansion modes [3]
9. INMAX = Maximum no. of inners/outer
10. IMAX = Maximum no. of outers
11. IBL = 0/1/2/3 = None/Print Recom'd modes & stop/Same but continue/Same and use (See Note 1)
12. IEIG = 0/1 = Source/Eigenvalue
13. ISOUR = 0/1/2/3 = No source/Bound srce/Vol srce/Both (See Note 2)
14. IMETH = 0/1/2 = Flat leakage/Calc'd lkge/Same & 2D flux [2]
15. IACC = 0/1 = No accel/Coarse mesh rebal [1]
16. IPRT = 0/1/2 = None/Print flux for all nodes/Print for edge nodes only
17. ISCT = Order of scattering

18. IHT = Position of σ_{total} [3]

19. IHS = Position of self-scatter σ [4]

20. IHL = Length of XS table

21. NXS = 0/NXS = XS in 4##/XS read from Logical unit NXS
Assumed double precision if <0. (See Note 3)

22. NACT = No. of Activities

2##

1. EPS = Convergence criterion [.0001]

2. EST = Estimate of K-eff (for B_L calculation) [1.]

3. MULT = Factor used in rebalance [1.] (See Note 4)

T

BLOCK 2

3$$

1. IBC(1) = Boundary condition for left boundary of system
0/1 = vacuum/reflected

2. IBC(2) = Same for bottom

3. IBC(3) = Same for right

4. IBC(4) = Same for top

4## Cross sections if NXS=0 (NMAT*NEG*IHL)

5## X boundaries of mesh (NX+1)

6## Y boundaries of mesh (NY+1)

7## μ values in angular quadrature (NANG)

8## η values in quadrature (NANG)

9## Angle weights (NANG)

10## χ values--fission spectrum (NEG)

11$$ Material no. for each mesh pt. (NX*NY)

12## External boundary source along y-axis (ISOUR=1 or 3)
(NANG*NY) Enter by angle for each y spacing. The sign of μ will determine whether the value is on the right or left.

13## External boundary source along x-axis (ISOUR=1 or 3)
(NANG*NX) Enter by angle for each x spacing. The sign of η will determine whether the value is on the top or bottom.

14## X boundaries of coarse mesh (NZX+1)

15## Y boundaries of coarse mesh (NZY+1)

16## Volumetric external source (scalar only) (ISOUR=2 or 3)
(NX*NY*NEG) For each group, each y spacing enter the x spacing value.

17## Modes for the problem (NREP)
Must be of form a_1, $-a_2$, a_2, ... {0, -1, 1, ...}

18$$ Material no. for each activity desired (NACT)
(See Note 5).

19$$ Cross section position for each activity (NACT)

20## Flux guess [0.] (NX*NY*NEG)
For each group, for each y spacing, enter value for each x spacing.

T

Notes:

1. The code has the capability of helping the user choose the expansion modes with a B_m calculation. (See Appendix H.) The choices open to the user are:

(0) Ignore this availability and choose the modes in the 17## array (or default to α=0, ±1, ±2, ...);

(1) Let the B_m calculation run, make its recommendations and stop, so that the user can make a mode choice and resubmit the problem;

(2) Let the B_m calculation run, but proceed with the 17## array modes anyway; or

(3) Let the B_m calculation choose the modes.

The B_m calculation has a different recommendation for each mode type and dimension; if the last option is chosen, the values chosen will span all of the recommended mode choices. This option seems to work satisfactorily for large (>3 mean free path) homogeneous regions in the problem, but not so well for thin material regions. Use the default for eigenvalue problems.

2. This option is the heart of the code. Our experience to date indicates that

   a. Method 2 is the best choice for most shielding problems in terms of accuracy versus cost;

   b. Method 0 has a strange and unexplained ability to calculate total leakages for some problems;

   c. For eigenvalue problems Method 2 is also best for delivering very accurate answers (~0.01% errors), but Method 0 can calculate answers to about 1% accuracy much faster; and

   d. Method 1 is included for completeness; it did not surpass Method 2 in any of the sample problems.

3. The standard GIP format for the cross-sections is used in EXTREME as well as DOT.

   The user's choices are:

   (0) The 4## array is used to input the cross-sections. Note that there will be only <u>one</u> 4## array per problem, not one per material type as in DOT;

   (>0) Logical unit NXS will be assumed to hold a single-precision GIP cross-section file;

   (<0) Logical unit NXS will be assumed to contain a double-precision GIP cross-section file. (There is currently no code which will create this kind of file.)

4. This value MULT is the rebalance constant for the ϕ-method coarse-mesh rebalance stabilization scheme[22]. It corresponds to DOT4.2's parameter WSOLCN. This stabilization is necessary for stabilizing the rebalance acceleration for some problems. The default is 1; the normal range for this parameter when needed is about 1 to 50.

5. The code can calculate any number of "activities" or total, volume-integrated reaction rates for user-supplied cross-section positions and materials. The equation is:

$$A_{mn} = \sum_I \sum_J \phi_{IJ} * \sigma_m * \text{Volume}_{IJ} * W(n)$$

   where W(n) = 1 if node I,J is made of material n.

   = 0 otherwise.

APPENDIX G

SAMPLE INPUT AND OUTPUT FOR EXTREME

As an example of the input and output of EXTREME, the 2x8 mesh calculation of Sample Problem 6 for Method 2 is included in the following pages. The first page contains the input to the code for this problem, and the remaining pages constitute the printed output.

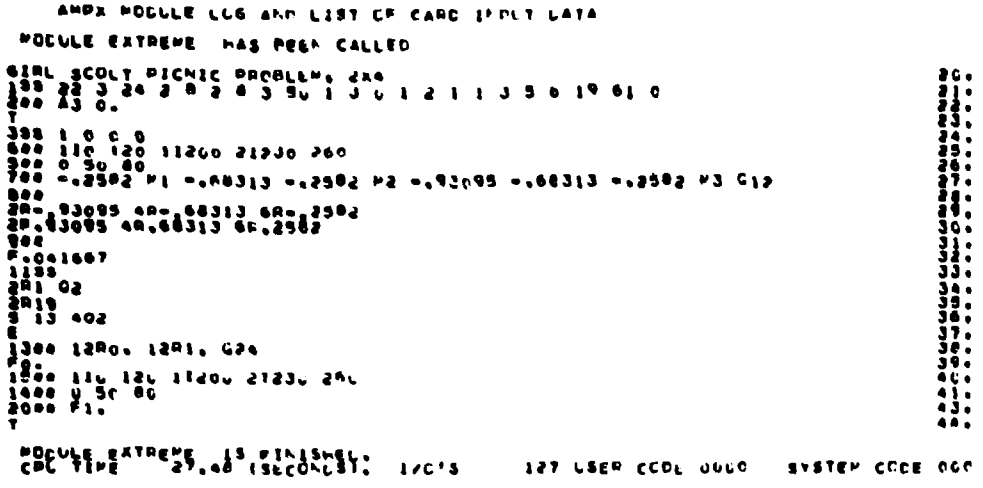


Figure G-1. Sample Input and Output from EXTREME

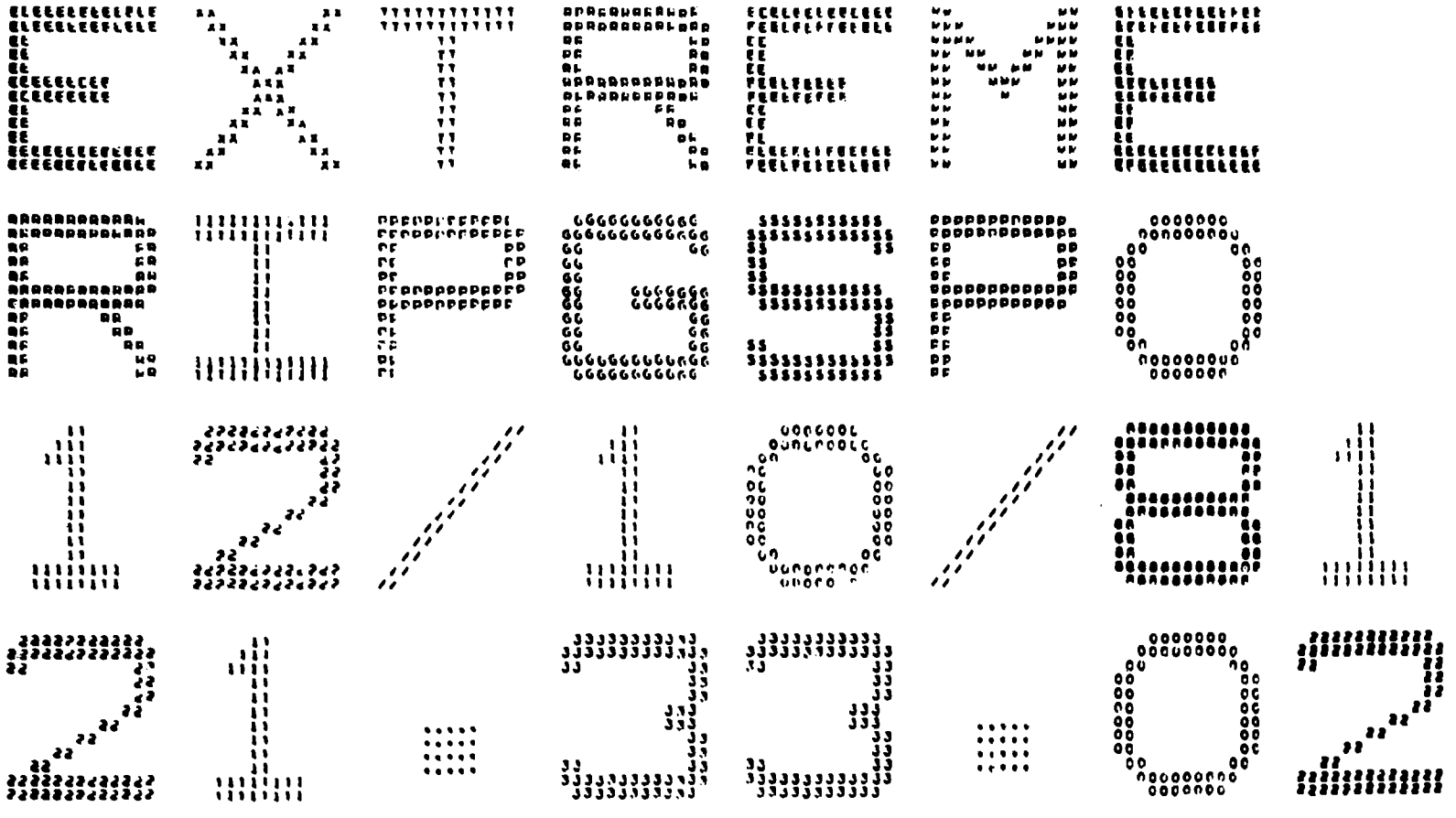


Figure G-1. (continued)

```
GIRL SCOUT PICNIC PROBLEM, 2X6

   1* ARRAY      22 ENTRIES REAL

   2* ARRAY       3 ENTRIES REAL


   0T
100

NMAT   =     22    NO. OF MATERIALS
NEG    =      3    NO. OF ENERGY GROUPS
NANG   =     24    NO. OF ANGLES
NX     =      2    NO. OF X INTERVALS FOR FINE MESH
NY     =      8    NO. OF Y INTERVALS FOR FINE MESH
NXM    =      1    NO. OF X INTERVALS FOR COARSE MESH
NYM    =      4    NO. OF Y INTERVALS FOR COARSE MESH
NMED   =      3    NO. OF EXPANSION MODES
INMAX  =     50    MAX. ALLD NO. OF INNERS/OUTER
IWMAX  =      0    MAXIMUM NO. OF CUTOFFS
IBAL   =      3    0/1/2/3 = NONE/PRINT MESSG, MODES & STOP/SAME PLT CONTINUE/SAME AAC USE
IBAL6  =      0    0/1 = SOURCE/FISSIONABLE
ISCUR  =      1    0/1/2/3 = NO SRCE/READ SRCE/VOL SRCE/PCTR
IMETH  =      2    0/1/2 = FLAT LEAKAGE/CALCD/SAME+DC PLUS
IBAC   =      1    0/1 = NONE/CLASSL MESH REBALANCE
IBMAT  =      1    0/1/2 = NONE/POINT PLUS CORES FOR ALL MODES/SAME FOR EDGE MODES ONLY
ISCT   =      3    ORDER OF SCATTERING
INT    =      8    POSITION OF TOTAL XS
INS    =      6    POSITION OF SELF-SCATTER XS
INF    =     10    LENGTH OF XS TABLE
NHS    =     61    0/1 = XS IF YES/XS READ FROM LL MXS
NACT   =      0    NO. OF ACTIVITIES

200
EPS    =  0.100000E-03   CONVERGENCE CRITERION
SBT    =  0.100000E+01   ESTIMATE OF K-EFF IN BL CALC
ABLY   =  0.0            MULT FACTOR FOR REBALANCE
ISC    =     16

   3* ARRAY       4 ENTRIES REAL

   4* ARRAY       8 ENTRIES REAL

   5* ARRAY       3 ENTRIES REAL

   7* ARRAY      24 ENTRIES REAL

   8* ARRAY      24 ENTRIES REAL

   9* ARRAY      24 ENTRIES REAL

  11* ARRAY      16 ENTRIES REAL

  13* ARRAY     144 ENTRIES REAL

  15* ARRAY       8 ENTRIES REAL

  16* ARRAY       3 ENTRIES REAL
```

Figure G-1. (continued)

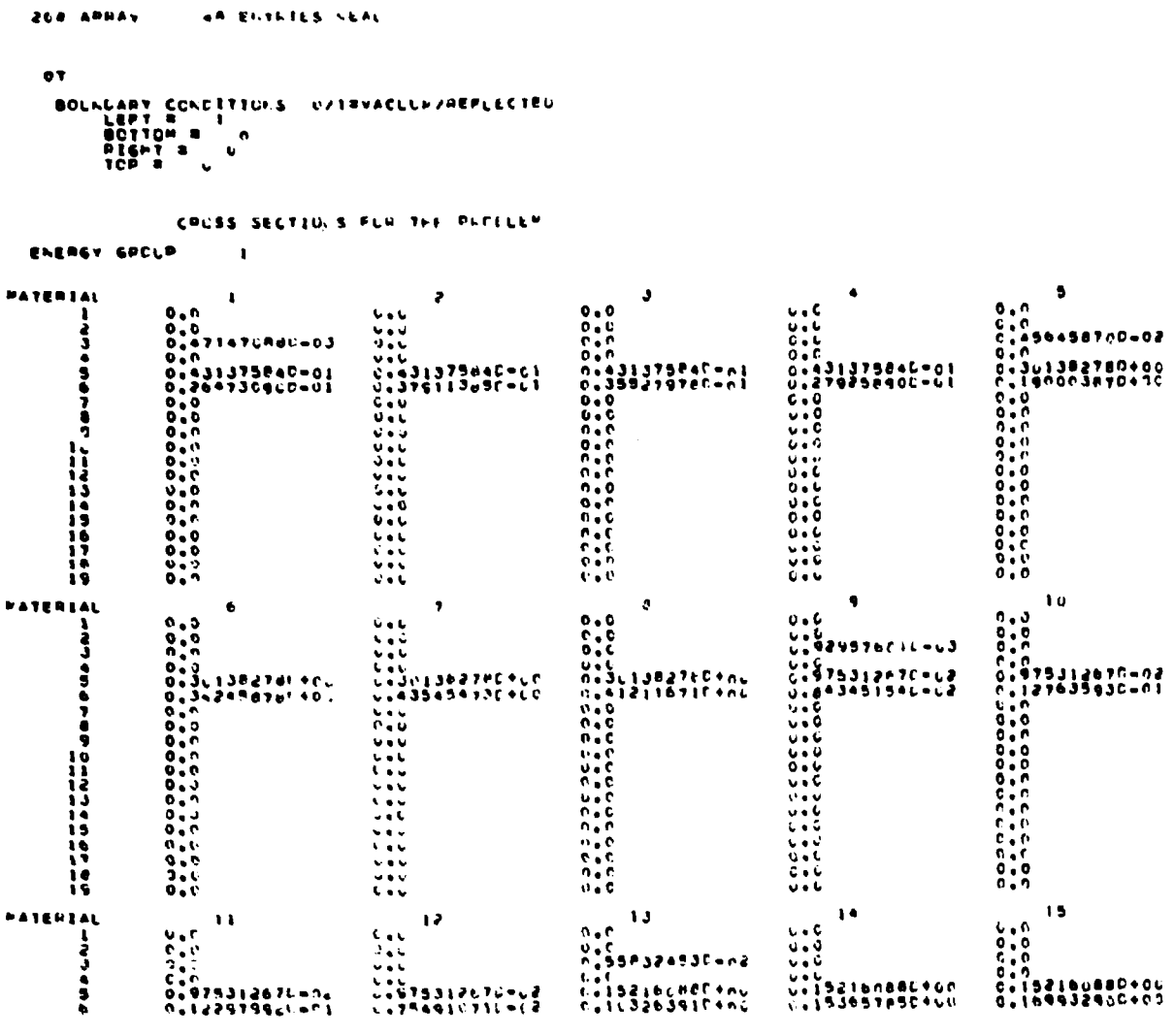


Figure G-1. (continued)

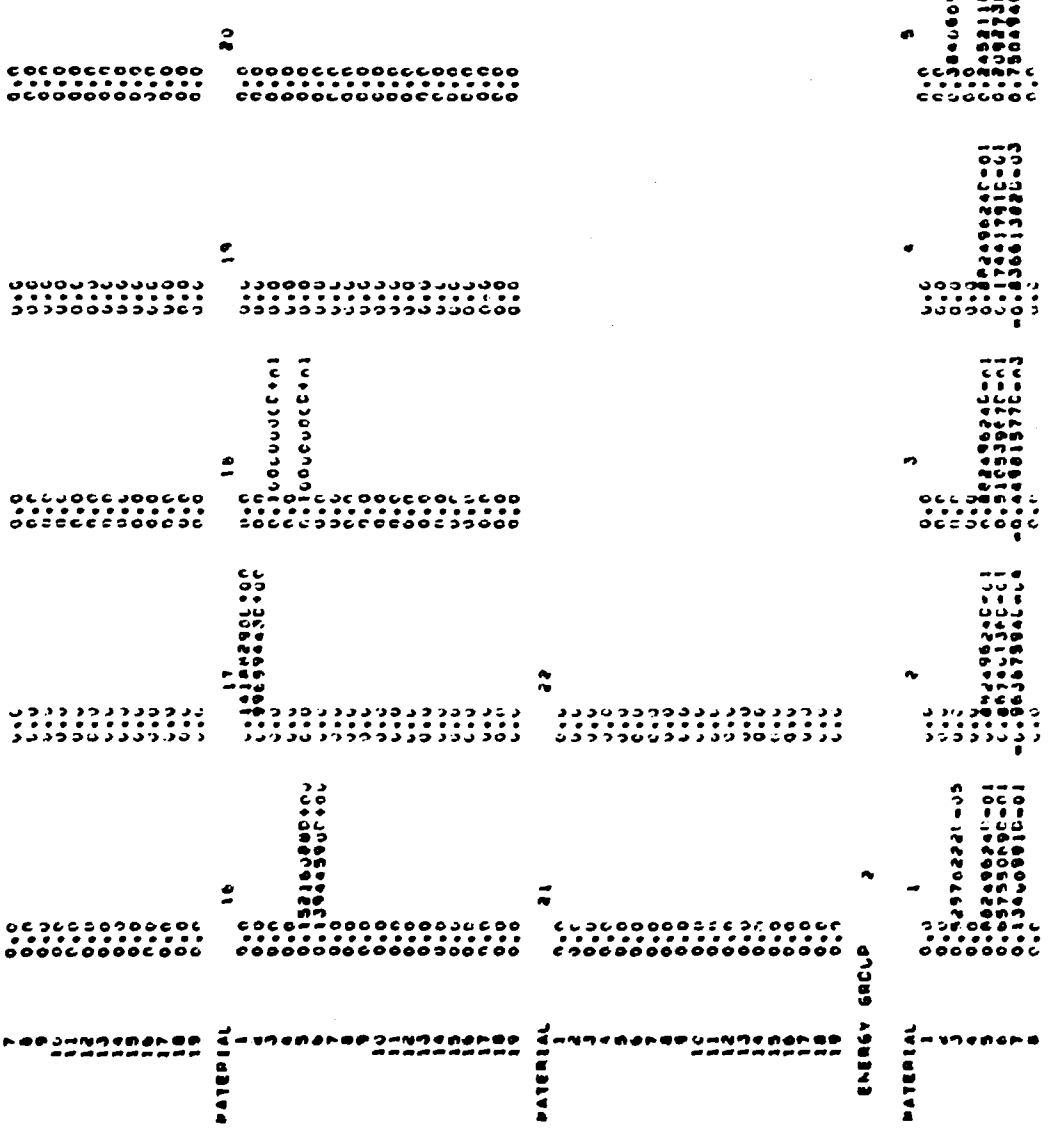


Figure G-1. (continued)

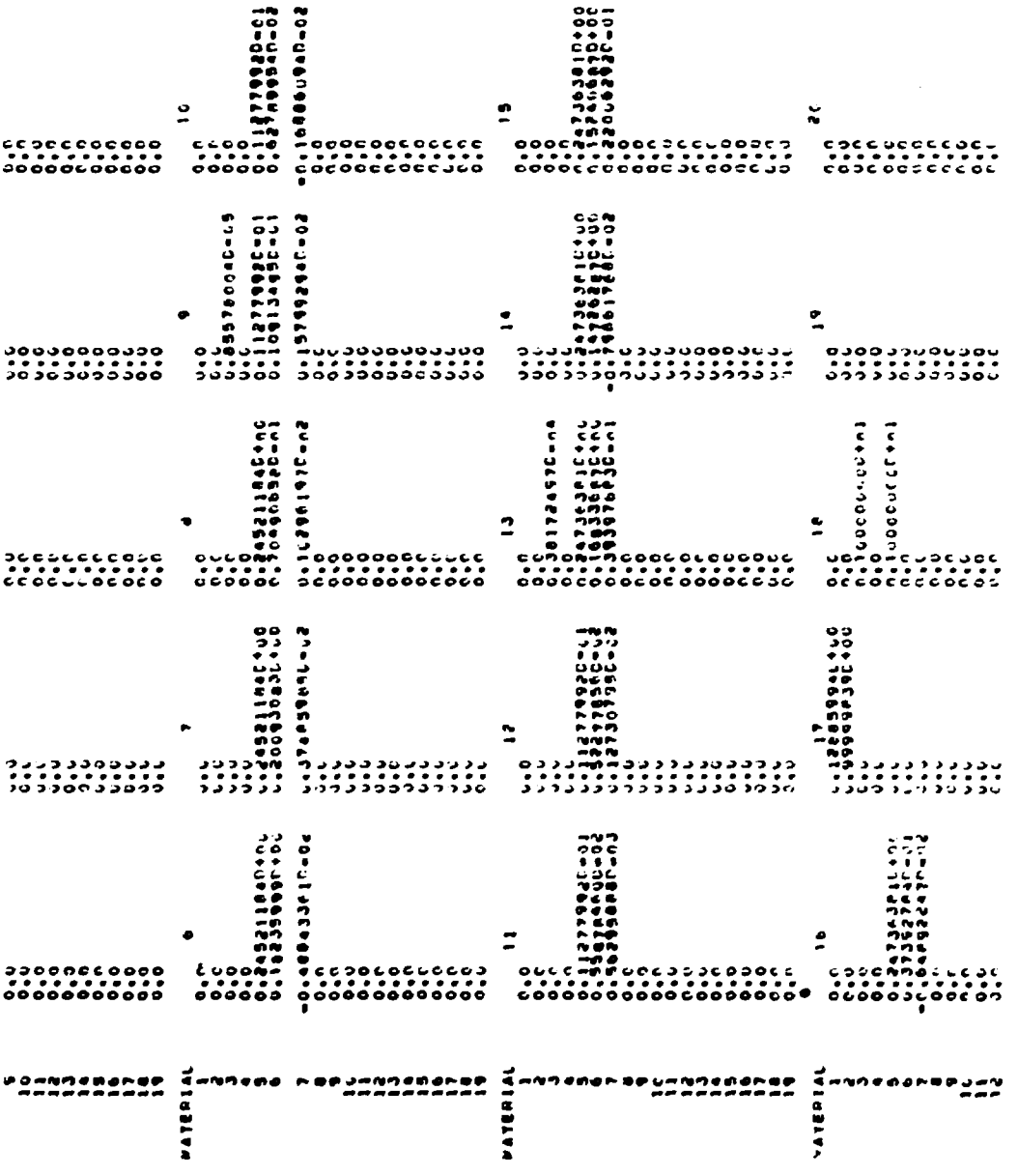


Figure G-1. (continued)

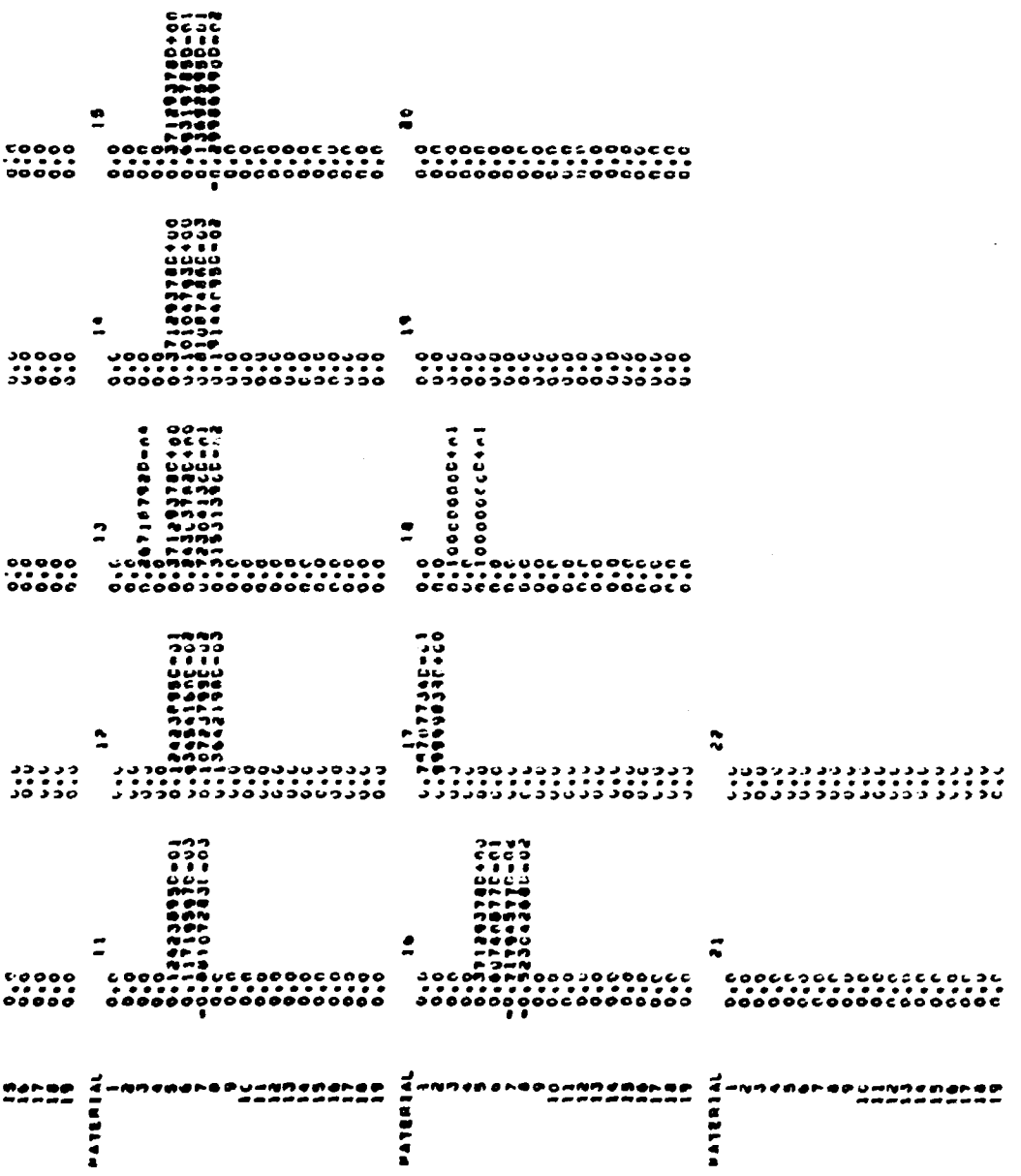


Figure G-1. (continued)

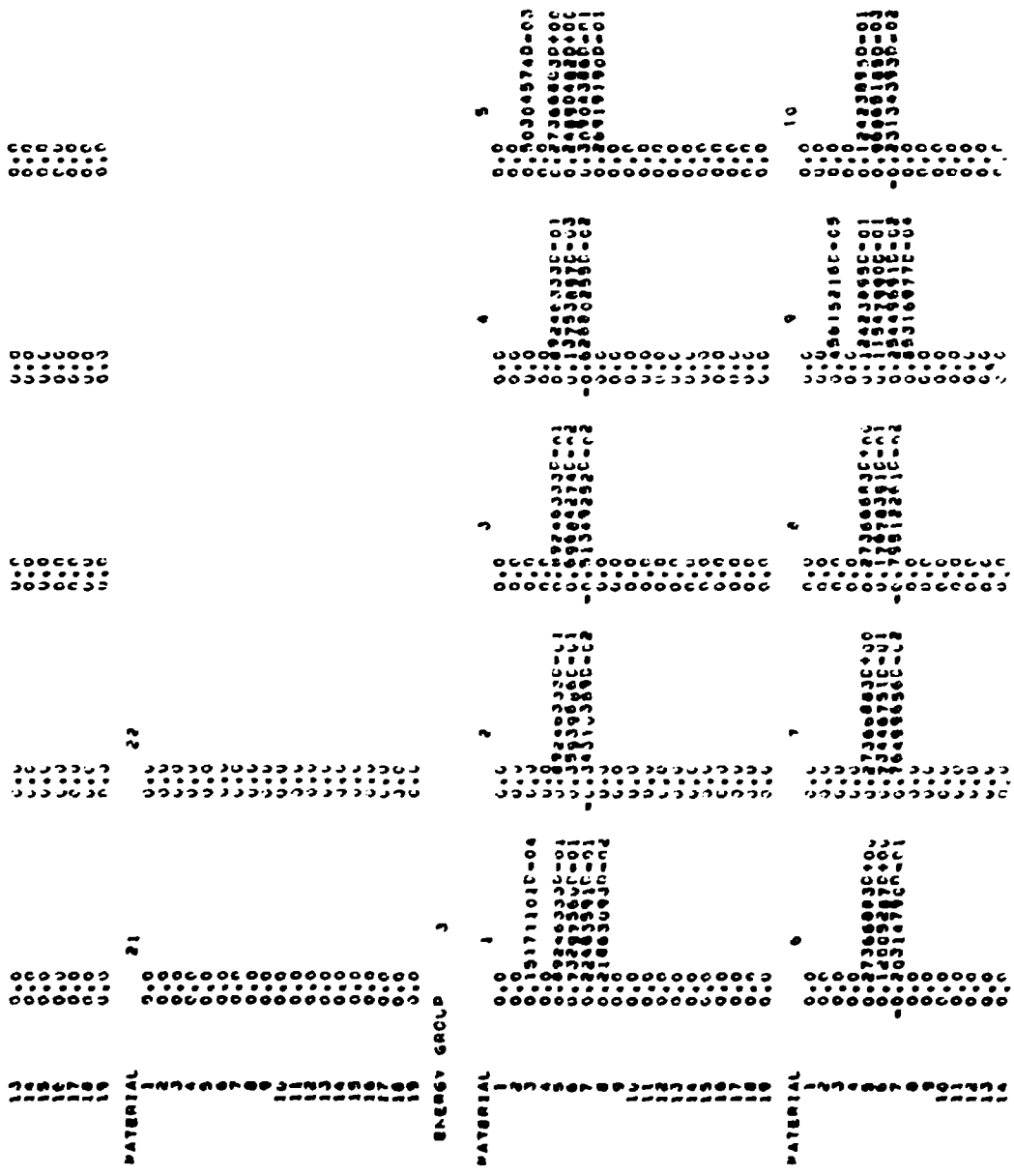
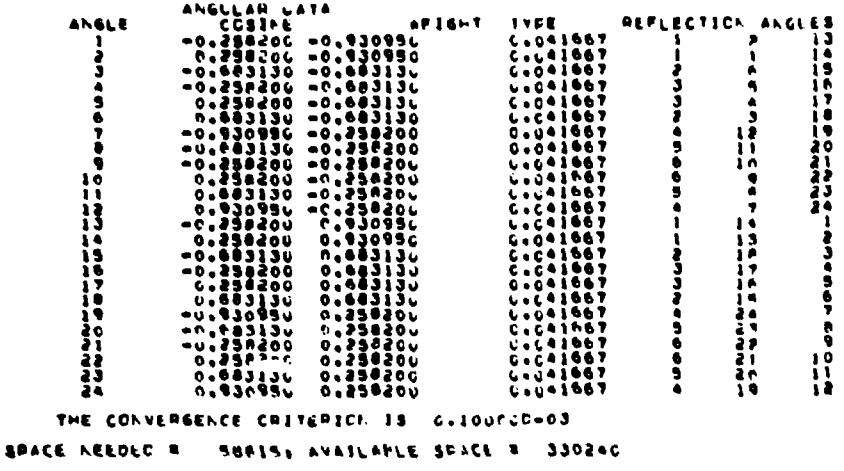


Figure G-1. (continued)

```
X-DIMENSION LEFT BOUNDARIES
   1 0.0              0.5000D+02
   2 0.3500D+02       0.3000D+02
   3 0.8000D+02

Y-DIMENSION LOWER BOUNDARIES
   1 0.0              0.6000D+02
   2 0.6000D+02       0.4000D+02
   3 0.1200D+03       0.8000D+02
   4 0.2000D+03       0.1500D+03
   5 0.3500D+03       0.2500D+03
   6 0.4500D+03       0.1000D+03
   7 0.5500D+03       0.1000D+03
   8 0.6500D+03       0.1000D+03
   9 0.7500D+03
   THE FISSION SPECTRUM
   GROUP        VALUE
         1    0.0
         2    0.0
         3    0.0

  NUMBER OF NODE TYPES =      8

 MAP OF MATERIALS
   8   33
   7   33
   6   33
   5   33
   4   33
   3   88
   2   11
   1   11

       12

 MAP OF NODE TYPES
   8   78
   7   78
   6   78
   5   56
   4   56
   3   34
   2   12
   1   12

       12

 COARSE MESH MAP
   8   56
   7   34
   6   12
   5   90
   4   78
   3   56
   2   34
   1   12

       12
   8    1
   7    1
   6    1
   5    1
   4    1
   3    1
   2    1
   1    1

THE NUMBER OF ANGLE TYPES IS    8
```

Figure G-1. (continued)



ANGULAR DATA

ANGLE	COSINE	WEIGHT	TYPE	REFLECTION ANGLES	
1	-0.288200	-0.930950	0.041667	1	13
2	0.288200	-0.930950	0.041667	1	14
3	-0.883130	-0.883130	0.041667	2	15
4	-0.258200	-0.883130	0.041667	3	16
5	0.258200	-0.883130	0.041667	3	17
6	0.883130	-0.883130	0.041667	2	18
7	-0.930950	-0.258200	0.041667	4	19
8	-0.883130	-0.258200	0.041667	5	20
9	-0.288200	-0.288200	0.041667	6	21
10	0.288200	-0.288200	0.041667	6	22
11	0.883130	-0.258200	0.041667	5	23
12	0.930950	-0.258200	0.041667	4	24
13	-0.258200	0.930950	0.041667	1	1
14	0.258200	0.930950	0.041667	1	2
15	-0.883130	0.883130	0.041667	2	3
16	-0.288200	0.883130	0.041667	3	4
17	0.288200	0.883130	0.041667	3	5
18	0.883130	0.883130	0.041667	2	6
19	-0.930950	0.258200	0.041667	4	7
20	-0.883130	0.258200	0.041667	5	8
21	-0.288200	0.288200	0.041667	6	9
22	0.288200	0.288200	0.041667	6	10
23	0.883130	0.258200	0.041667	5	11
24	0.930950	0.258200	0.041667	4	12

THE CONVERGENCE CRITERION IS 0.100000E-03

SPACE NEEDED = 98815, AVAILABLE SPACE = 330240

Figure G-1. (continued)

```
*****INFINITE MEDIUM CALCULATION FOR THE MATERIALS *****
    ESTIMATED EIGENVALUE IS  0.1000000D+01
MATERIAL    5 FAILED TO CONVERGE AFTER 20 ITERS
FOR NODE TYPE    1 MATERIAL    1,  B**2 =  0.2136409?D-02
        THE X-DIMENSION DISTANCE IS  0.900000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          1.64180
        THE Y-DIMENSION DISTANCE IS  0.400000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          1.97010

FOR NODE TYPE    2 MATERIAL    1,  B**2 =  0.2136409?D-02
        THE X-DIMENSION DISTANCE IS  0.300000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          0.88808
        THE Y-DIMENSION DISTANCE IS  0.400000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          1.97010

FOR NODE TYPE    3 MATERIAL   10,  B**2 =  0.2999999?D-09
        THE X-DIMENSION DISTANCE IS  0.500000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          0.00000
        THE Y-DIMENSION DISTANCE IS  0.400000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          0.00000

FOR NODE TYPE    4 MATERIAL   10,  B**2 =  0.2999999?D-09
        THE X-DIMENSION DISTANCE IS  0.300000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          0.00000
        THE Y-DIMENSION DISTANCE IS  0.800000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          0.00000

FOR NODE TYPE    5 MATERIAL    5,  B**2 =  0.2673861?D-01
        THE X-DIMENSION DISTANCE IS  0.900000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          5.09149
        THE Y-DIMENSION DISTANCE IS  0.150000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          1.52745

FOR NODE TYPE    6 MATERIAL   13,  B**2 =  0.2?321474?D-01
        THE X-DIMENSION DISTANCE IS  0.300000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          3.10933
        THE Y-DIMENSION DISTANCE IS  0.150000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          1.50467

FOR NODE TYPE    7 MATERIAL    5,  B**2 =  0.2?738613D-01
        THE X-DIMENSION DISTANCE IS  0.900000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          5.09149
        THE Y-DIMENSION DISTANCE IS  0.100000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          1.01830

FOR NODE TYPE    8 MATERIAL   13,  B**2 =  0.2?321474?D-01
        THE X-DIMENSION DISTANCE IS  0.300000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          3.10933
        THE Y-DIMENSION DISTANCE IS  0.100000000+02 CM. AND THE RECOMMENDED MODE CHOICE IS          1.03644

ANGLE #    1 COS =   -0.25820  -0.93095, AT =    0.04167, CF =    2.87105
L=K,PN=AD =    1   1    1.00000    1.00000
L=K,PN=AD =    2   1   -0.25820   -0.25820
L=K,PN=AD =    2   2    0.96600   -0.93095
L=K,PN=AD =    3   1   -0.40000   -0.40000
L=K,PN=AD =    3   2   -0.10333    0.41634
L=K,PN=AD =    3   3    2.80000    0.69282
L=K,PN=AD =    4   1    0.34427    0.34427
L=K,PN=AD =    4   2   -0.86600    0.38006
L=K,PN=AD =    4   3   -3.41480   -0.40000
L=K,PN=AD =    4   4   13.92527   -0.49066

ANGLE #    2 COS =    0.25820  -0.93095, AT =    0.04167, CF =    2.87105
L=K,PN=AD =    1   1    1.00000    1.00000
L=K,PN=AD =    2   1    0.25820    0.25820
L=K,PN=AD =    2   2    0.96600   -0.93095
L=K,PN=AD =    3   1   -0.40000   -0.40000
L=K,PN=AD =    3   2    0.74833   -0.41634
L=K,PN=AD =    3   3    2.80000    0.69282
L=K,PN=AD =    4   1   -0.34427   -0.34427
L=K,PN=AD =    4   2   -0.86600    0.38006
L=K,PN=AD =    4   3    3.41480    0.40000
L=K,PN=AD =    4   4   13.92527   -0.49066

ANGLE #    3 COS =   -0.86313  -0.86313, AT =    0.04167, CF =    2.78023
```

Figure G-1. (continued)

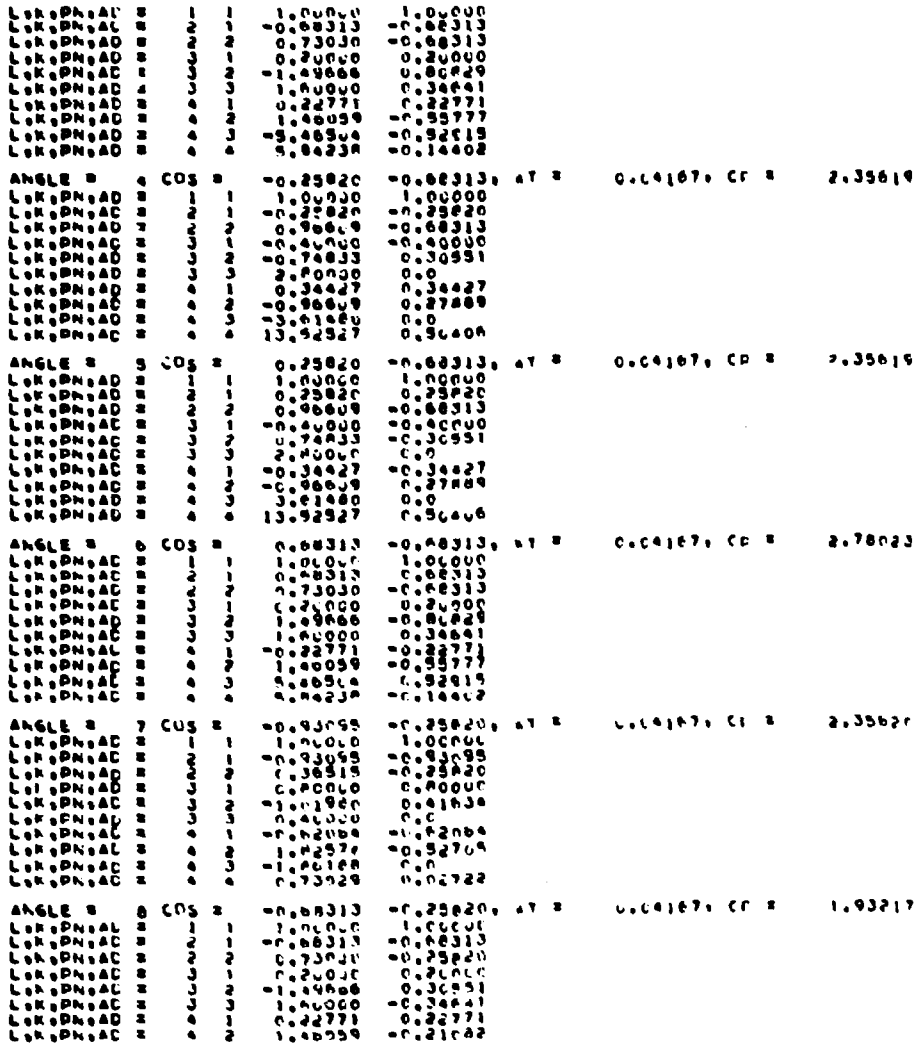


Figure G-1. (continued)

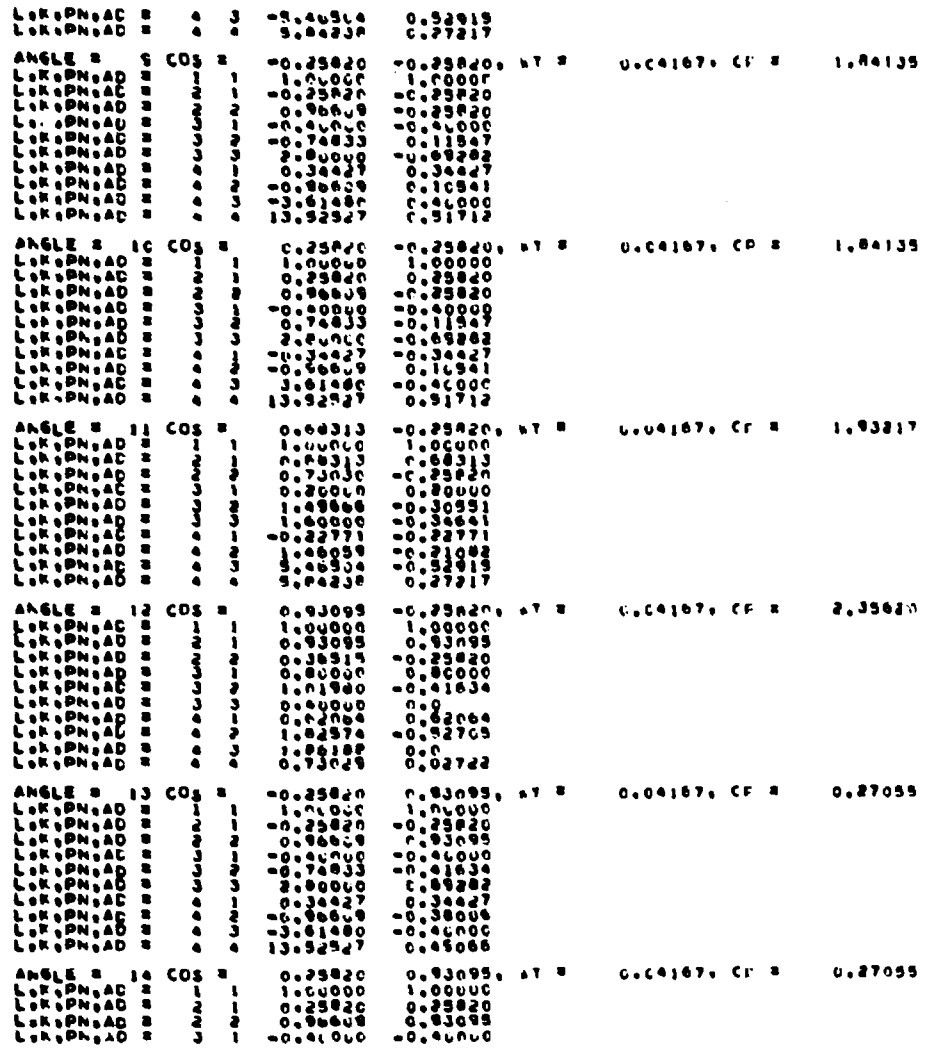


Figure G-1. (continued)

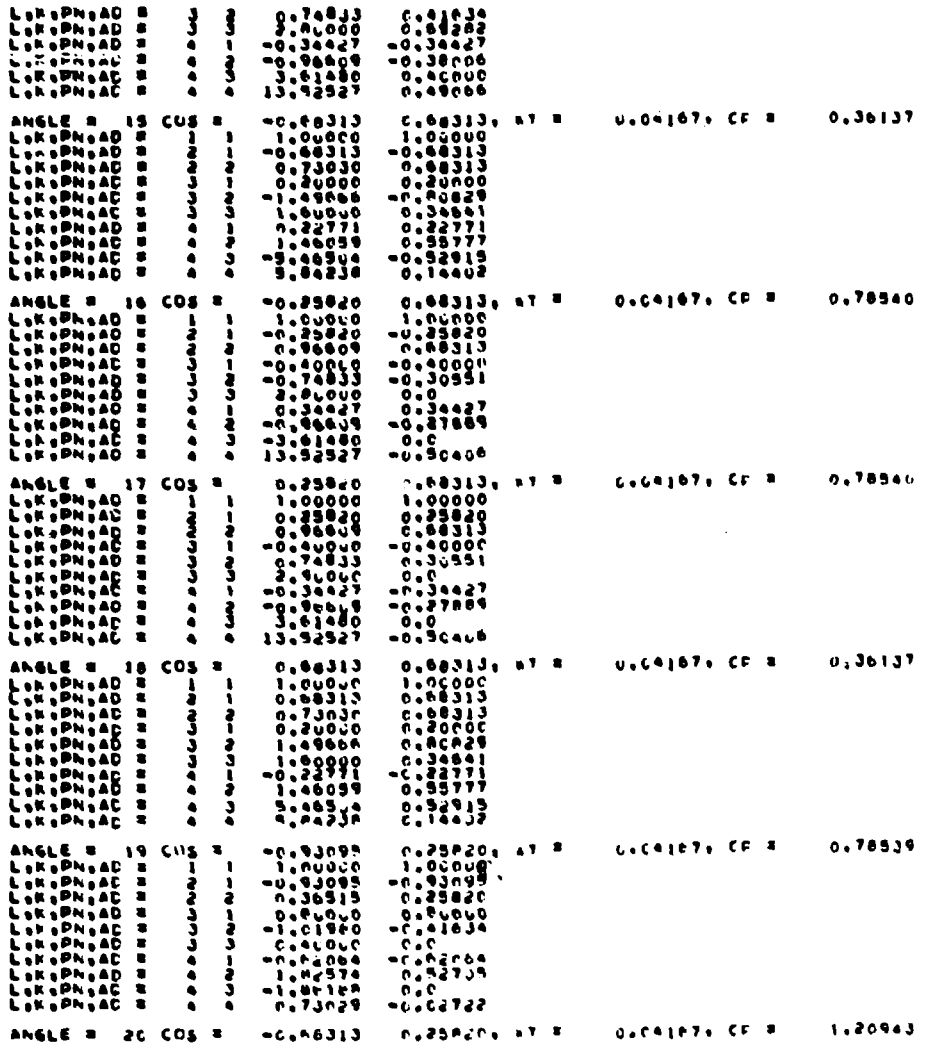


Figure G-1. (continued)

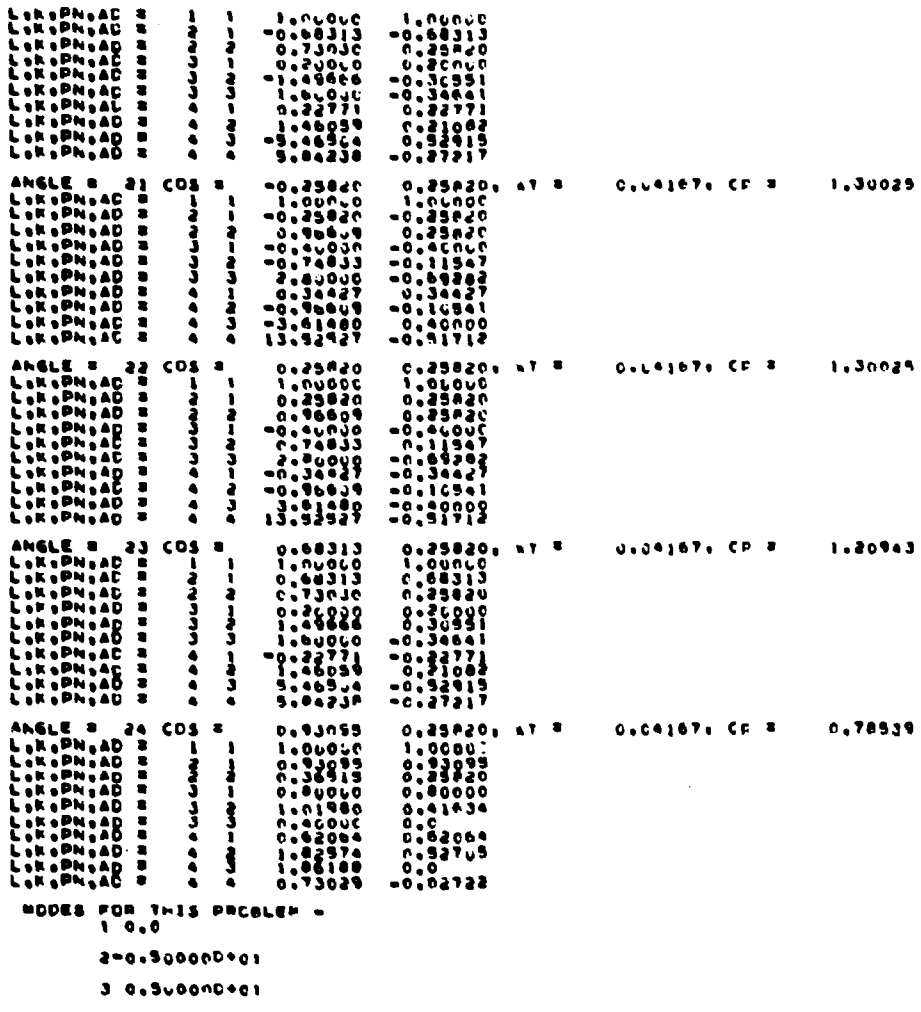


Figure G-1. (continued)

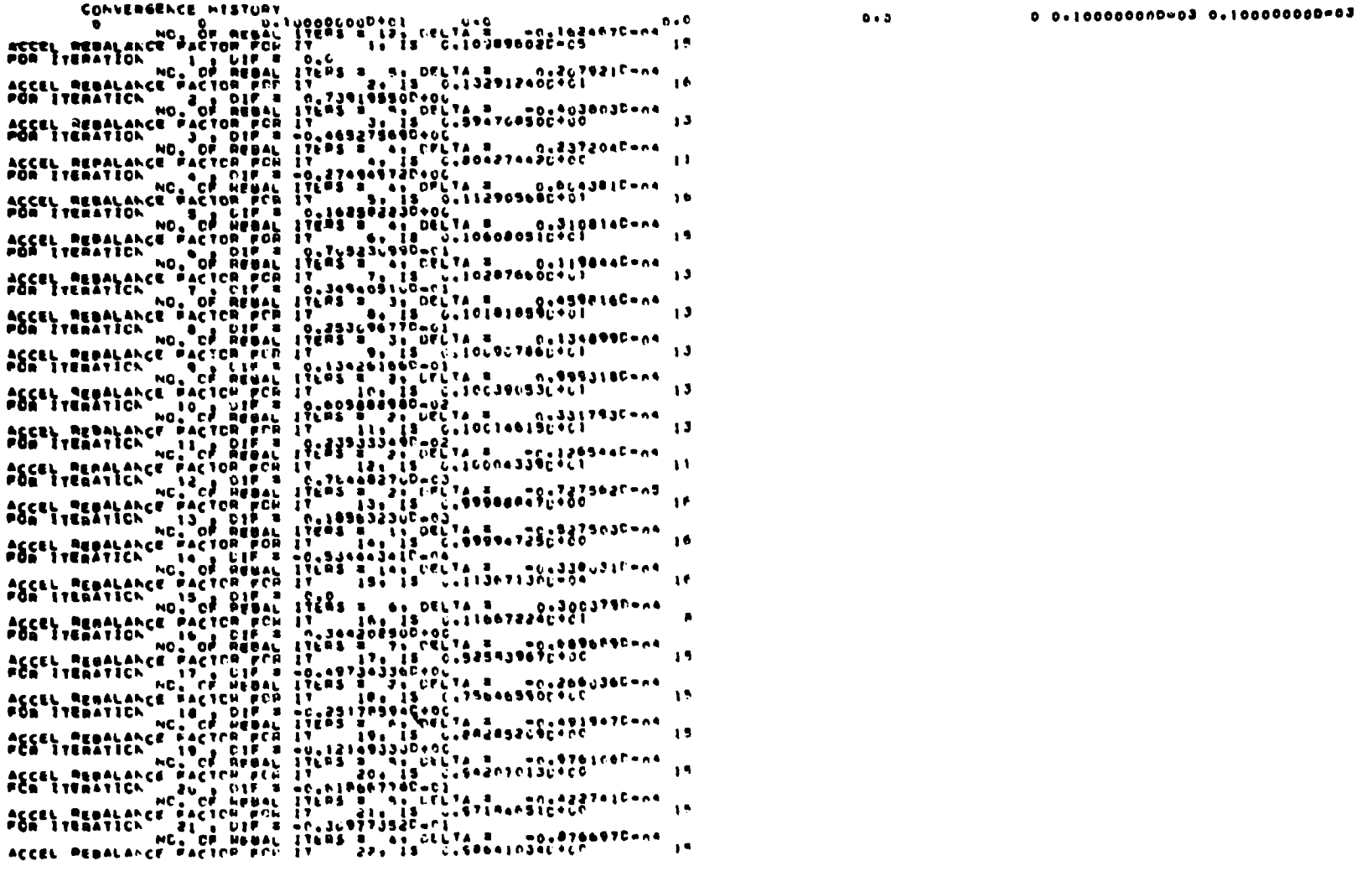


Figure G-1. (continued)

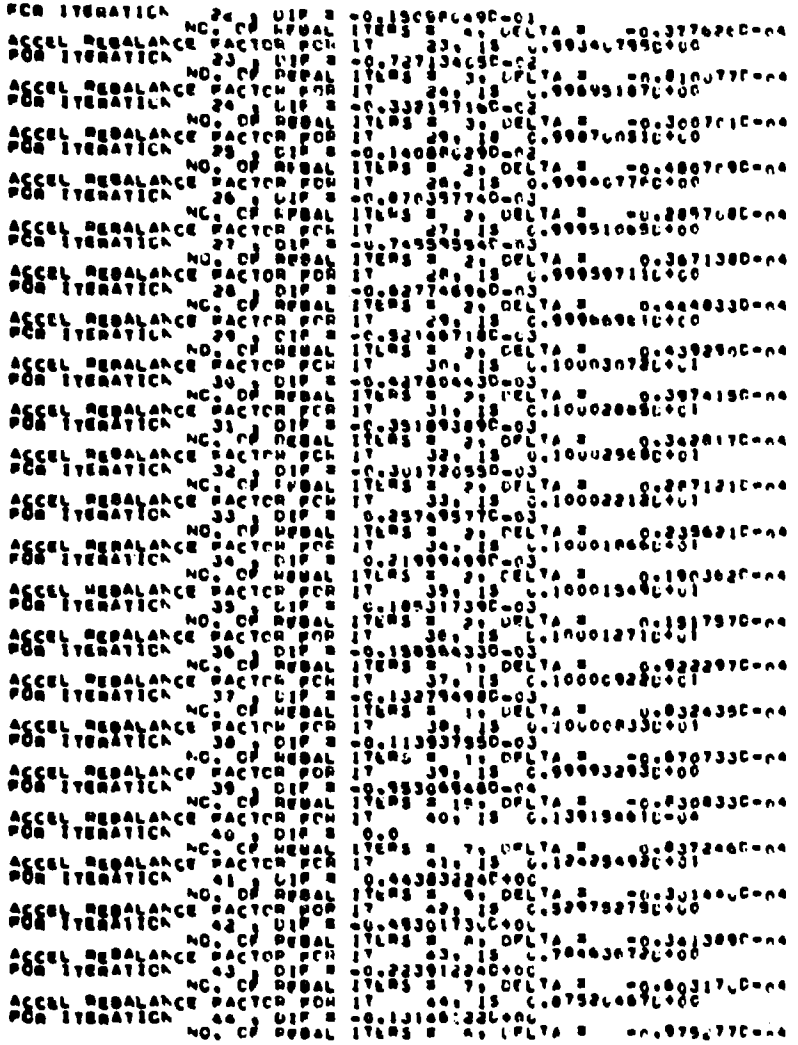


Figure G-1. (continued)

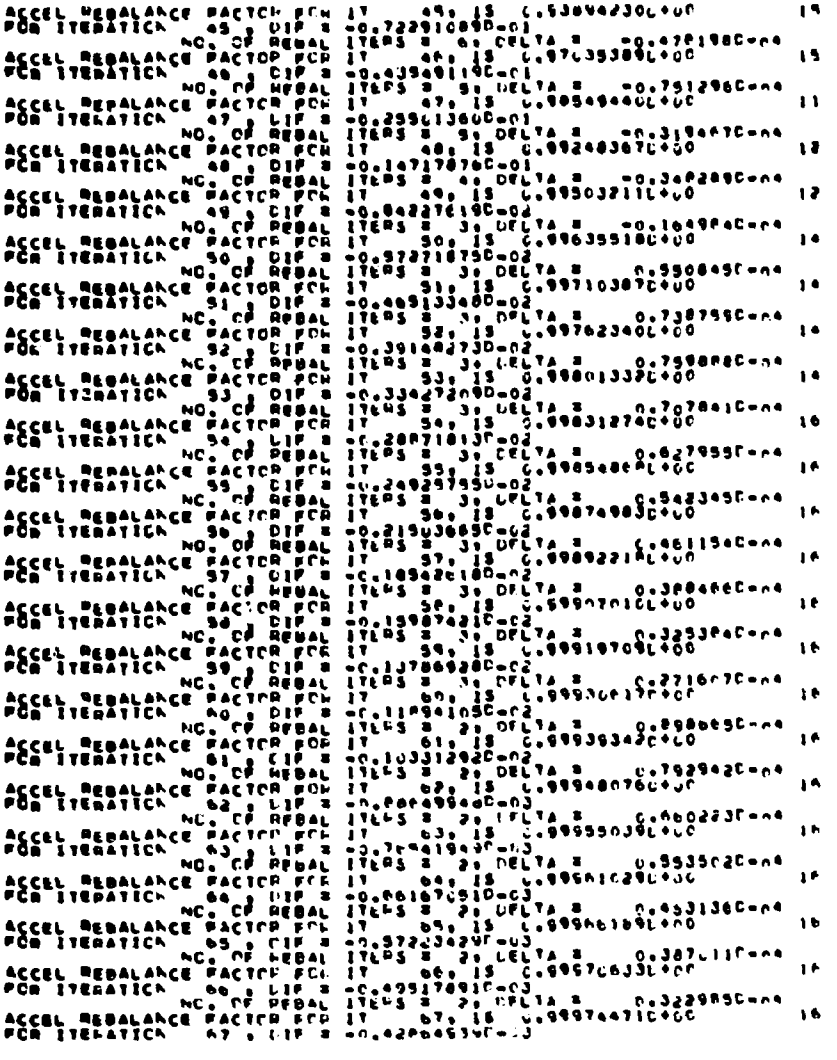
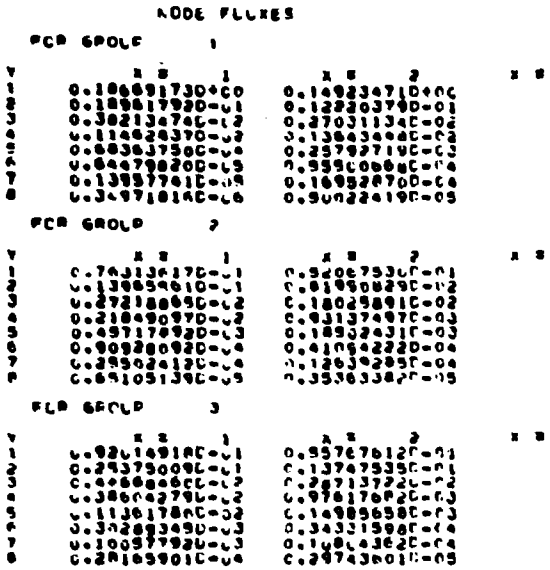


Figure G-1. (continued)



NODE FLUXES

FCR GROUP 1

Y	X = 1	X = 2	X =
1	0.1866917300+00	0.1492334710+00	
2	0.1898617930-01	0.1228203790-01	
3	0.3021347650-02	0.2703113420-02	
4	0.1146263700-02	0.1384336480-02	
5	0.6036137500-03	0.2579271900-03	
6	0.6447988200-04	0.9595060840-04	
7	0.1395777410-04	0.1695287700-04	
8	0.3497718140-05	0.9002224190-05	

FCR GROUP 2

Y	X = 1	X = 2	X =
1	0.7431364170-01	0.5406753000-01	
2	0.1386596610-01	0.8199008290-02	
3	0.2721886950-02	0.1802588910-02	
4	0.2184990970-02	0.9313749700-03	
5	0.4971769200-03	0.1893243310-03	
6	0.9092806920-04	0.4106842720-04	
7	0.2950241250-04	0.1263992850-04	
8	0.8910513900-05	0.3530333420-05	

FLR GROUP 3

Y	X = 1	X = 2	X =
1	0.9261491800-01	0.9576761200-01	
2	0.2437900090-01	0.1374753350-01	
3	0.4488866600-02	0.2871137220-02	
4	0.3860642790-02	0.9761767820-03	
5	0.1136178000-02	0.1498965800-03	
6	0.3428893490-03	0.3433159980-04	
7	0.1009779920-03	0.1408643620-04	
8	0.2816599010-04	0.2974360010-05	

Figure G-1. (continued)

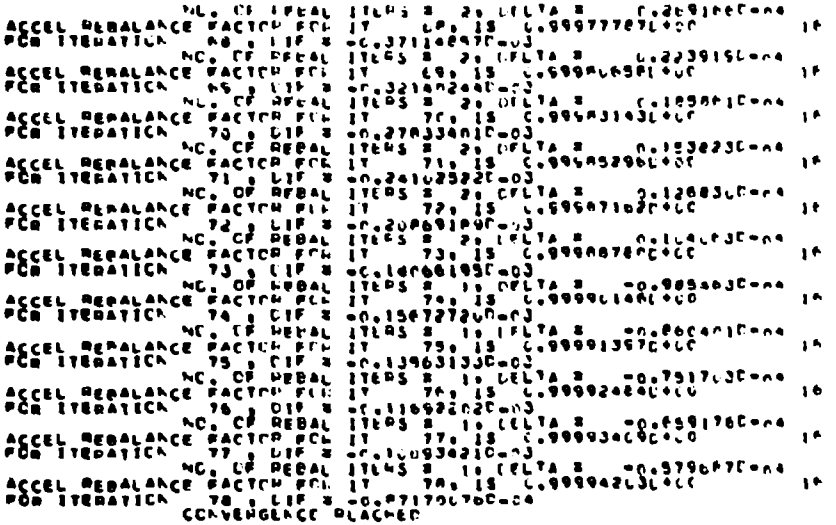


Figure G-1. (continued)

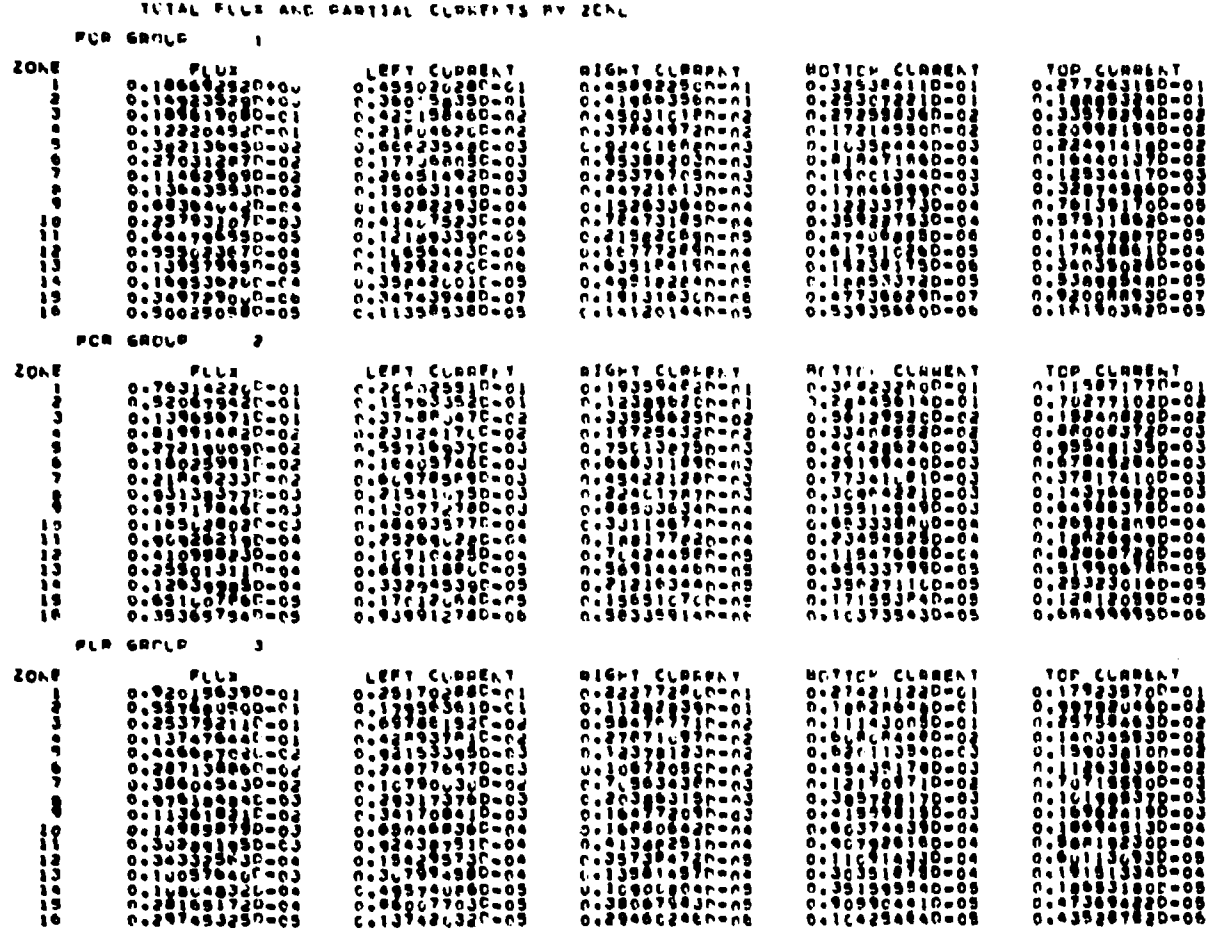


Figure G-1. (continued)

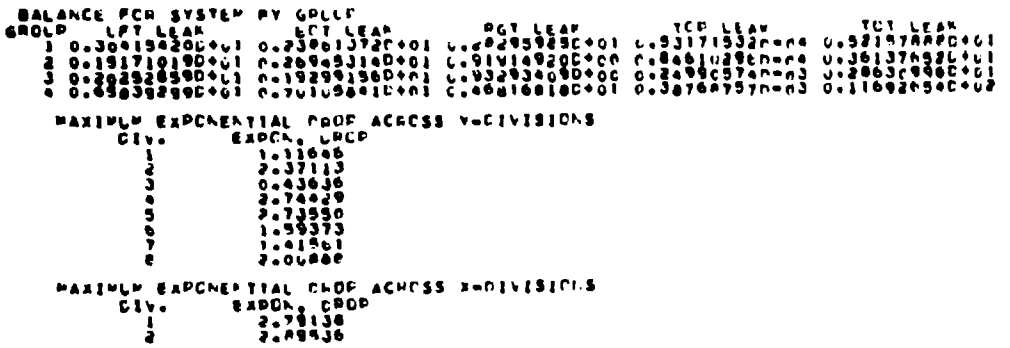


Figure G-1. (continued)

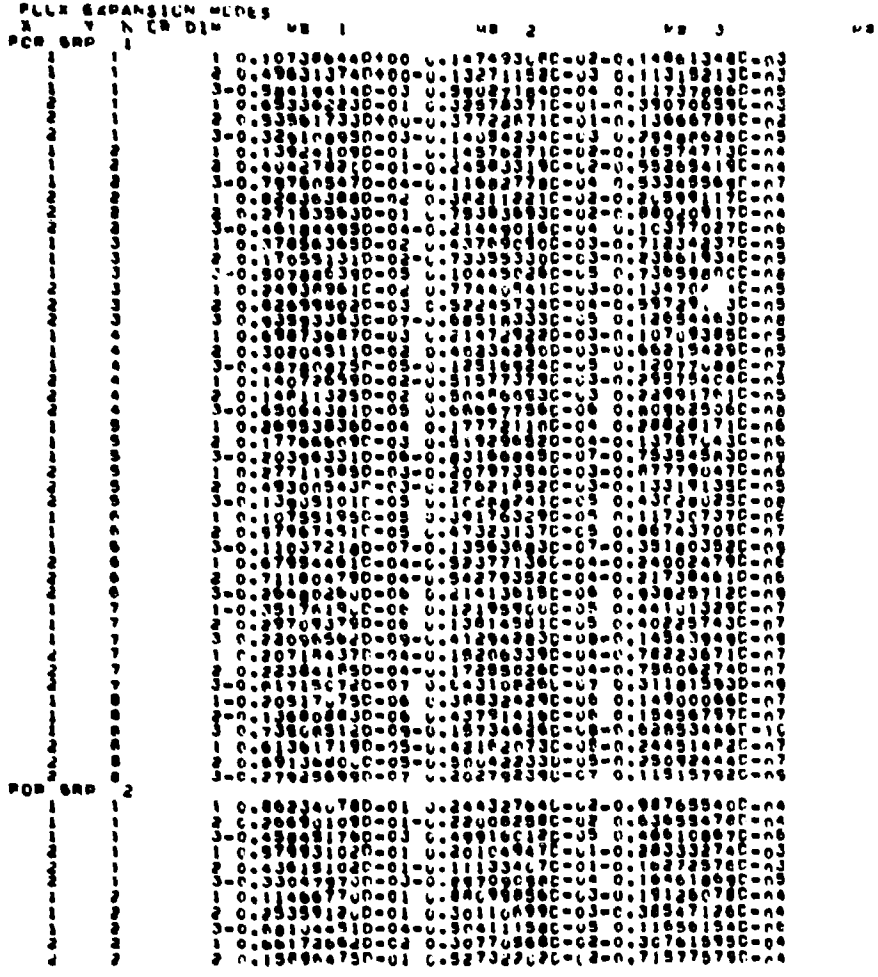


Figure G-1. (continued)

Figure G-1. (continued)

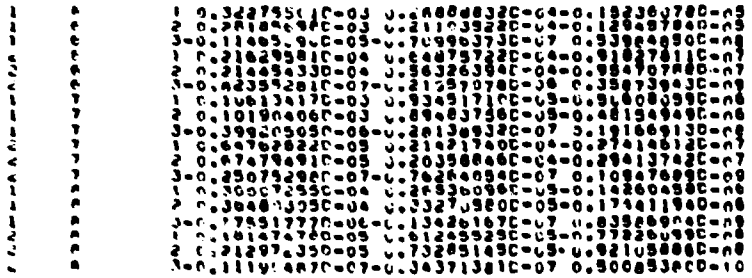


Figure G-1. (continued)

APPENDIX H

USE OF A B_m CALCULATION TO CHOOSE MODES FOR EXTREME

The choice of a mode set for the exponential spatial expansion of the neutron flux inside each spatial node can be made easier if an approximation of this flux shape is available. Such an approximate shape is determined at problem set-up in EXTREME using a material buckling (B_m) calculation for each problem material.

This B_m calculation utilizes the static diffusion equation approximation[1] to the transport equation:

$$-\nabla\cdot D(\bar{r},E)\nabla\psi(\bar{r},E) + \sigma_t(\bar{r},E)\phi(\bar{r},E) = \int_0^\infty dE'[\chi(\bar{r},E)\nu(\bar{r},E)\sigma_f(\bar{r},E) + \sigma_s(\bar{r},E'\rightarrow E)]\phi(\bar{r},E') \qquad (H.1)$$

where

$$D(\bar{r},E) = \frac{1}{3\,\sigma_t(\bar{r},E)}$$

and the other variables are as defined for the transport equation in Eqs. 2.1 and 2.4.

A further approximation used in the B_m calculation (shared by the analogous B_L transport theory approximation) is that deep within a given sourceless material region the neutron sources will come from scattering and fission interactions within the material region itself[1]. Therefore, the neutron energy spectrum can be approximated by the spectrum from a single material calculation. For such a single material calculation, the cross-section data are independent of $\bar{r}$ and Eq. H.1 becomes

$$-D(E)\nabla^2\phi(\bar{r},E) + \sigma_t(E)\phi(\bar{r},E) = \int_0^\infty dE'[\chi(E)\nu(E')\sigma_f(E') + \sigma_s(E'\to E)]\phi(\bar{r},E) \qquad (H.2)$$

If the neutron flux is separated into the product of two terms,

$$\phi(\bar{r},E) = F(\bar{r})S(E)$$

Eq. H.2 becomes

$$-D(E)S(E)\nabla^2F(\bar{r}) + \sigma_t(E)S(E)F(\bar{r}) = F(\bar{r}) \int_0^\infty dE'[\chi(E)\nu(E') + \sigma_s(E'\to E)]S(E').$$

Dividing by F(r)S(E) and rearranging produces

$$\frac{\nabla^2F(\bar{r})}{F(\bar{r})} = \frac{\sigma_t(E)}{D(E)} - \frac{1}{D(E)S(F)} \int^\infty dE'[\chi(E)\nu(E')\sigma_f(E') + \sigma_s(E'\to E)]S(E') \qquad (H.3)$$

$$\equiv B_m^2$$

Equation H.3 states that a function of $\bar{r}$ equals a function of E. For this to be true, both sides of the equation must equal a constant, here denoted as B_m^2, the materials buckling.

The constant B_m^2 is determined by solving the "energy side" of Equation H.3:

$$\sigma_t(E)S(E) - \int_0^\infty dE'[\chi(E)\nu(E')\sigma_f(E') + \sigma_s(E'\to E)] = D(E)B_m^2S(E)$$

In multigroup form, this is

$$\sigma_t^gS_g - \sum_{g'=1}^{G} [\chi_g\nu_{g'}\sigma_f^{g'} + \sigma_s^{g'\to g}]S_{g'} = B_m^2D_gS_g \qquad g=1,2,\ldots G \; ; \qquad (H.4)$$

where

$$S_g = \int_{\Delta E} S(E)d(E) \; ,$$

$$D_g = \int_{\Delta E} D(E)S(F)d(E)/S_g \; ,$$

and the other multigroup constants are the same as in Eq. 2.3a (with S(E) weighting function and no space dependence.)

Equation H.4 can be written in matrix form as

$$\underline{\underline{A}}\ \underline{S} = B_m\ \underline{S}$$

where

$$\underline{S} = [S_1,\ S_2,\ \ldots S_G]^T$$

and A is a GxG matrix with elements

$$A_{ij} = \sigma_t^i \delta_{ij} - \chi_i \nu_j \sigma_f^i - \sigma_s^{j \to i}$$

There will be G values of the eigenvalue B_m^2; the one needed for this study is the smallest one as this will correspond to the eigenmode of $\underline{S}$ which decays slowest and therefore "survives" deeper inside the node.

This eigenvalue is determined by solving for $\underline{S}$ iteratively using

$$\underline{S} = B_m^2\ \underline{\underline{A}}^{-1}\ \underline{S}$$

As $\underline{\underline{A}}^{-1}$ is a full matrix, the actual procedure followed by EXTREME is the iterative process:

$$\underline{S}^{n+1} = \underline{\underline{L}}^{-1}\ \underline{\underline{U}}\ \underline{S}^n$$

$$(B_m^2)^{n+1} = \| \underline{S}^n \| \ / \ \| S^{n+1} \|$$

$$\underline{S}^{n+1} = (B_m^2)^{n+1}\ \underline{S}^{n+1}$$

where $\| \underline{S} \| = \sum_g S_g$ and $S_0 = [1 \quad 1 \quad 1 \quad \ldots 1]^T$

Once the smallest B_m^2 is known, the "space side" of Eq. H.3 can be solved for Cartesian geometry to get:

$$F(x,y) = F_{1e} B_x x + F_{2e} B_y y$$

where $$B_x^2 + B_y^2 = B_m^2 \ .$$

In choosing the modes, EXTREME assumes

$$B_x = B_y = (B_m^2/2)^{1/2}$$

If the resulting value B is real ($B_m^2>0$), it is used with the geometric information Δx and Δy for each node containing this material to obtain "recommended modes". If the value B is imaginary ($B_m^2<0$), the "recommended mode" for the nodes containing the material is set to 1.

Once the recommended modes have been determined for all of the node types, the largest value is used for the problem mode set.