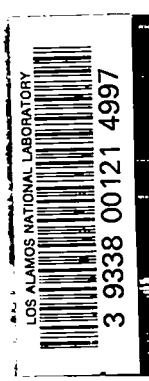
LA-10983-MS

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

Transport Calculations for Nuclear Analyses: Theory and Guidelines for Effective Use of Transport Codes



Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Prepared by Ann Nagy, Group X-6

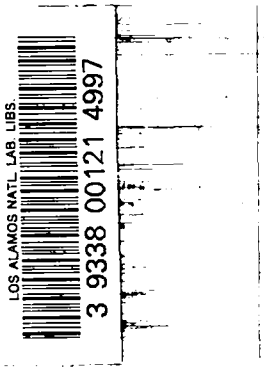
LA-10983-MS

UC-32
Issued: September 1987

Transport Calculations for Nuclear Analyses: Theory and Guidelines for Effective Use of Transport Codes

R. Douglas O'Dell
Raymond E. Alcouffe

667 - 4614



Los Alamos National Laboratory
Los Alamos, New Mexico 87545

CONTENTS

TRANSPORT CALCULATIONS FOR NUCLEAR ANALYSES: THEORY AND GUIDELINES FOR EFFECTIVE USE OF TRANSPORT CODES

by

R. Douglas O'Dell and Raymond E. Alcouffe

ABSTRACT

This report is for the serious user of discrete ordinates transport computer codes for performing nuclear analysis calculations. The first section after the introduction provides a reasonably thorough mathematical description of the analytic Boltzmann transport equation. Next is a section on the numerical discretization of the energy, angle, and space variables in the transport equation, along with an introduction to the source iteration method. The fourth section provides numerical details and features pertinent to discrete ordinates codes. That section details angular quadrature, spatial discretization methods, iteration acceleration methods, and search capabilities. The fifth section presents considerations in choosing a discrete ordinates code for use, and this is followed by a section on typical discrete ordinates codes available throughout the world. The report ends with some guidance for the user.

This report is a revision of the chapter titled "Transport Calculations for Nuclear Reactors," written by the authors for Volume 1 of the three-volume CRC Handbook of Nuclear Reactors Calculations, Y. Ronen, Editor, published by the CRC Press, Boca Raton, Florida (1986).

I. INTRODUCTION

The use of computer codes to perform nuclear analysis calculations costs money. Transport calculations are more expensive than diffusion calculations. Although continuing development in computer hardware and codes (software) have reduced the expense of transport calculations to more practical and acceptable

levels than in the past, transport calculations are still more expensive than diffusion calculations. Moreover, there arises the question of whether a transport calculation must adequately describe the physics in a given problem or whether a diffusion calculation will suffice. To answer this question intelligently and to use a computer code effectively and properly requires understanding the physics modeled by a computer code and the numerical techniques used in the code. The purpose of this report is to provide the nuclear analyst with information about the use of deterministic particle transport codes for nuclear analysis calculations.

This introduction describes briefly the principal differences between diffusion theory and transport theory and some of the recent advances that have reduced the cost of running a transport calculation. Subsequent sections provide a more detailed description of the mathematics, physics, and numerical methods in deterministic transport codes and give the reader guides to the use of such computer codes.

Any precise description of neutral particle transport must involve treatment of the migration of particles in phase space (space, energy, direction, and time) as influenced by interactions with the underlying matter. Below, the description of particle migration is as a single linear equation under the assumptions that the particles interacting with the underlying matter do not affect the matter and that the particles do not interact with themselves. Stated another way, the descriptive equation is linear if interactions between particles and material can be described in terms of cross sections independent of the particle density. The neutral particle transport equation is then developed by performing a mathematical balance on the physical production and losses of particles. Section II of this report describes this development of the transport equation in some detail. The result is, in its full generality, an equation with seven independent variables, three spatial variables, three momentum variables (or equivalently, an energy or speed variable plus two direction-of-motion variables), and one time variable. Although in practical applications the transport equation can be cast in specialized forms involving fewer independent variables, the equation still provides a rather elaborate description of the particle migration process, and its computational solution is a fairly expensive method for performing reactor or other nuclear analyses. It is thus desirable to discuss the utility of the

full transport description and to describe some of the simplified approximations to it.

The most useful approximation for performing neutronic analyses is the diffusion equation. The diffusion equation is computationally less expensive to solve than is the transport equation because diffusion theory reduces the maximum number of independent variables from seven to five--three spatial variables, one momentum variable (energy or speed), and one time variable. Mathematically, the diffusion approximation places some rather severe restrictions on the momentum dependence of the neutron density. Physically, these restrictions consist of two major requirements. First, the neutron migration process must be dominated by scattering interactions, or collisions; that is, the material must be highly scattering and weakly absorbing for neutrons. Second, the neutron migration process must be far removed from system interfaces - that is, material discontinuities - where large gradients in the neutron density may occur. In practice, diffusion theory has been applied extensively to reactor analyses and generally found to perform better than it theoretically has any right to. That is, if proper corrections are made from transport theory, diffusion theory can be used quite acceptably for a large class of reactor analysis problems. These corrections come from a transport-based homogenization theory and are normally made in preparing cross-section data. These transport-based corrections do much to foster the success of diffusion theory-based reactor neutronic analysis.

It is natural, then, to ask "If diffusion theory, successfully corrected for transport effects, is computationally less expensive than transport theory, why be concerned with transport theory to perform nuclear analyses?" To answer this question, one must recall the two basic requirements of diffusion theory which, even with transport-based corrections, still largely exist. Diffusion theory relies on the absence of large gradients in the neutron density (or flux) in any spatial region of the problem. Large gradients imply a highly directional migration of neutrons, and diffusion theory does not include the direction-of-motion variables. Also, for an accurate diffusion coefficient to be defined for the material, diffusion theory requires that the migration process be dominated by scattering collision. Both of these conditions relate directly to the description of the neutron leakage, and in this description diffusion theory suffers the most relative to transport theory. Two examples in reactor analysis where large gradients are known to exist are

1) in thermal reactors near control rods and 2) in fast reactors in the vicinity of internal blankets. In both cases, unless special, somewhat ad hoc, treatments are applied, the diffusion solution will be inaccurate or, at best, uncertain, and transport analysis is needed. Another application requiring transport analysis is in shielding calculations or in any calculation of deep penetration. In such applications, the collision (interaction) processes between particles and material are usually not weakly absorbing and highly scattering, but are commonly just the reverse, and diffusion theory cannot be used with any confidence. Also, particle flow or migration through a shield tends to be directional, and diffusion theory treats this situation poorly. For these reasons, virtually all shielding and analyses of deep penetration are performed with transport theory.

Given the experience with diffusion theory, the preponderance of reactor core analyses will likely continue to be performed with this simpler, computationally cheaper tool. Several developments in the late 1970s and early 1980s, however, reduced the expense of transport calculations to levels where they can be performed much more routinely. One of the developments is the continuing computer hardware improvement in computational speed, in memory size, and in data storage availability. The Class-VI computers developed in the early 1980s (CRAY-1 and Control Data Corporation's Cyber 205), which perform tens of millions of floating point operations per second, have reduced transport calculational times into the seconds and minutes range instead of the minutes and hours range of earlier computers. Further, developments in solution methods for the transport equation have shown that the nuclear designer/analyst can have at his disposal an enriched set of transport analysis tools completely compatible with the diffusion theory tools he has been accustomed to using. This is discussed in our section on iteration convergence of the transport solution process using the very effective diffusion synthetic acceleration (DSA) method. The DSA method employs the diffusion equation to accelerate the convergence of transport iterations by using the intermediate-iteration transport results to correct the diffusion equation. When DSA is carried to completion, the corrected diffusion equation solution is the same as the transport solution within a specified convergence criterion. Note, however, that since the iterative process is carried out in stages (iterations), at each successive stage the diffusion solution becomes a better approximation to the transport solution, so the DSA method can be thought of as

a diffusion improvement method. Thus, a properly designed transport computer code offers the user a diffusion equation solver that can be consistently improved by transport theory so any doubts or uncertainties about transport effects can be systematically removed conveniently. Thus, with faster and larger computer hardware and better, more modern computer codes, the use of transport theory as a routine nuclear analysis tool is likely to increase.

In subsequent sections of this report, we attempt to provide the nuclear designer/analyst with a reasonably complete overview of numerical methods for deterministically solving the transport equation. In Sec. II we present a mathematical description of transport theory, quickly specializing to some convenient geometries for the spatial resolution of the transport divergence operator that appears in the transport equation. Next, in Sec. III, we proceed to numerical descriptions based on the analytical equations, with emphasis on the discretization of the independent variables to produce the discrete ordinates equations. We also include an introduction to the concept of solving the discrete ordinates transport equations by source iteration techniques. Section IV is devoted, in some detail, to many of the numerical procedures that are computationally effective and are in use in today's computer codes. Section V discusses some considerations that are important in choosing a computer code as a transport calculational tool, and this is followed by a short section on some typical discrete ordinates codes that are readily available for use. This report concludes with Sec. VIII, a brief section giving some general guidance to the discrete ordinates code user.

II. MATHEMATICAL DESCRIPTION: THE ANALYTIC EQUATION

The linear Boltzmann transport equation is an integro-partial differential equation embodying the physics of neutral particle transport. This equation and boundary conditions are required for problems of finite geometric extent. The boundary conditions specify the distribution of particles entering the geometric problem through its exterior boundaries. The Boltzmann equation, together with the appropriate boundary conditions (and an initial condition for time-dependent problems), constitutes a mathematically well-posed problem having a unique solution. This solution consists of the complete, that is, deterministic, distribution of particles throughout the space, energy, direction-of-motion, and time (for time-dependent problems) portions of the

problem. The linear Boltzmann equation, together with the boundary conditions (and initial condition, if required) in discretized form, is solved by deterministic transport computer codes.

In this section, we present a brief development of the linear Boltzmann transport equation in its general analytic form. Following this, we show some of the specific forms the equation takes for common geometries - that is, coordinate systems - encountered in practice. We then describe the various boundary conditions that are appropriate for use in these geometries. Next, we develop the spherical harmonics expansion forms for the particle source (production) terms in the Boltzmann equation since these forms are almost universally used in deterministic transport codes. We conclude this section with a presentation of the adjoint form of the Boltzmann equation.

A. The Balance Equation (Linear Boltzmann Transport Equation)

The distribution of particles as a function of the seven variables constituting phase space is obtained, in principle, by solving the linear Boltzmann transport equation. This equation serves to precisely describe particle balance in which the rate of accumulation of particles is equal to the difference between their rates of production and removal. If $N(\vec{r},E,\vec{\Omega},t)d\vec{r}dEd\vec{\Omega}dt$ is the number of particles in volume $d\vec{r}$ about space point $\vec{r}$, with energy in dE about E, moving in direction $d\vec{\Omega}$ about $\vec{\Omega}$, in time interval dt about t, then the Boltzmann transport equation can be written as

$$\frac{\partial N(\vec{r},E,\vec{\Omega},t)}{\partial t} = - v\vec{\Omega} \cdot \vec{\nabla}N(\vec{r},E,\vec{\Omega},t) - vN(\vec{r},E,\vec{\Omega},t)\Sigma_t(\vec{r},E,\vec{\Omega},t)$$

$$+ S(\vec{r},E,\vec{\Omega},t) \quad . \tag{1}$$

This equation is linear if the macroscopic cross sections of the medium, such as Σ_t, are not functions of the particle density. This linearity condition is met for virtually all practical applications.

The term on the left side of Eq. (1) represents the rate of accumulation of particles at the phase space point in question, namely

$$\lim_{\Delta t \to 0} \left[\frac{N(\vec{r},E,\vec{\Omega},t + \Delta t) - N(\vec{r},E,\vec{\Omega},t)}{\Delta t}\right] \equiv \frac{\partial N(\vec{r},E,\vec{\Omega},t)}{\partial t} \quad . \tag{2}$$

The first term on the right side of Eq. (1), $v\vec{\Omega} \cdot \vec{\nabla}N(\vec{r},E,\vec{\Omega},t)$, represents the rate of change of the particle density at spatial position $\vec{r}$ resulting from streaming of the particles with speed v, that is, motion in a straight line without collisions. That this term represents streaming can be seen by considering, without loss of generality, a Cartesian incremental volume $\Delta V = \Delta x \Delta y \Delta z$ as shown in Fig. 1. In particular, consider the face of area $\Delta x \Delta y$ at z. The rate at which particles enter the volume through the face at z is $(v\vec{\Omega}\cdot\vec{k})N(x,y,z,E,\vec{\Omega},t)\Delta x\Delta y$, where $v\vec{\Omega}\cdot\vec{k}$ is simply the z-component of the velocity $v\vec{\Omega}$. Similarly, the rate at which particles leave the incremental volume through the face at $z+\Delta z$ is $(v\vec{\Omega}\cdot\vec{k})N(x,y,z+\Delta z,E,\vec{\Omega},t)\Delta x\Delta y$. The difference between outflowing and inflowing particles is, in the limit of vanishingly small Δz,

$$\lim_{\Delta z \to 0} (v\vec{\Omega}\cdot\vec{k}) \left[\frac{N(x,y,z+\Delta z,E,\vec{\Omega},t) - N(x,y,z,E,\vec{\Omega},t)}{\Delta z}\right]\Delta V$$

$$= (v\vec{\Omega}\cdot\vec{k}) \frac{\partial N(x,y,z,E,\vec{\Omega},t)}{\partial z} \Delta V \quad .$$

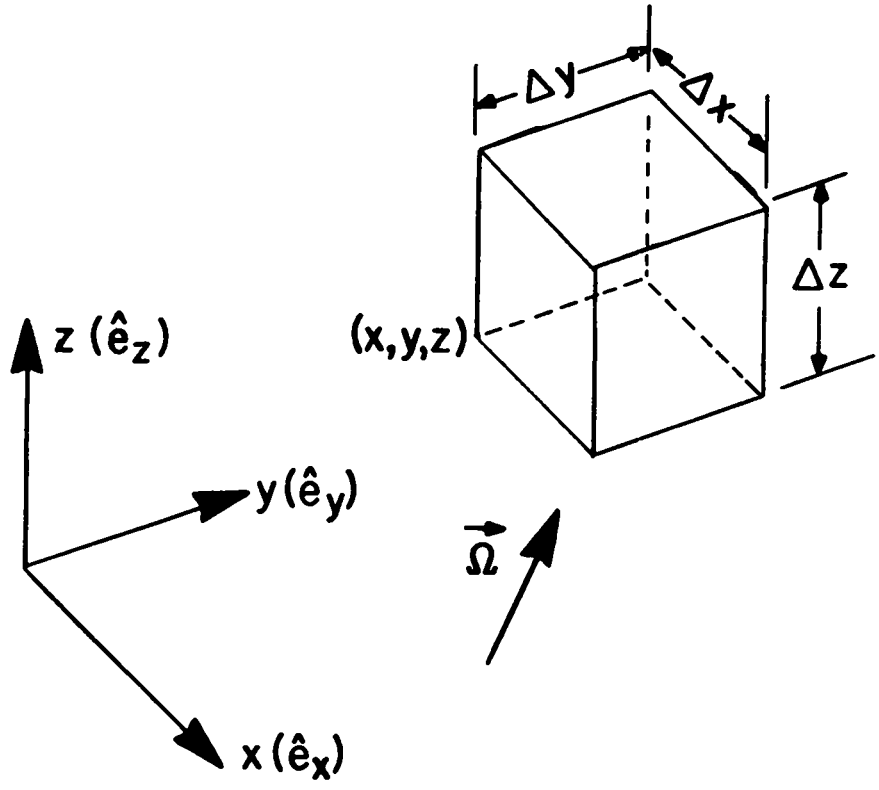


Fig. 1. Incremental volume in Cartesian coordinates.

Similar expressions can be written for flow (or streaming) in the x- and y-directions so that the net rate at which particles are lost from the volume is

$$\left[(v\vec{\Omega}\cdot\vec{i})\,\frac{\partial N}{\partial x} + (v\vec{\Omega}\cdot\vec{j})\,\frac{\partial N}{\partial y} + (v\vec{\Omega}\cdot\vec{k})\,\frac{\partial N}{\partial z}\right]\Delta V \quad ,$$

where the arguments $(x,y,z,E,\vec{\Omega},t)$ have been omitted for simplicity. Thus, using the definition of the divergence operator $\vec{\Omega}\cdot\vec{\nabla}$, the net rate at which particles are lost from an incremental volume because of streaming is $v\vec{\Omega}\cdot\vec{\nabla}N(r,E,\vec{\Omega},t)$ per unit volume.

The second term on the right side of Eq. (1), $vN(\vec{r},E,\vec{\Omega},t)\Sigma_t(\vec{r},E,\vec{\Omega},t)$, accounts for the rate at which particles are lost because of collisions of any kind with the nuclei constituting the medium. Here $\Sigma_t(\vec{r},E,\vec{\Omega},t)$ is the macroscopic total cross section of the medium defined such that Σds is the probability of a collision in a path length, ds.

The third term on the right side of Eq. (1), $S(\vec{r},E,\vec{\Omega},t)$, represents the source rate of particles, that is, the rate at which particles are produced. This source includes contributions from scattering, fission, and inhomogeneous (fixed) sources.

The scattering source accounts for the rate at which particles are produced as a result of particle interactions with nuclei - other than interactions that result in fissioning of the nuclei. The scattering source is denoted $S_s(\vec{r},E,\vec{\Omega},t)$ and is described by the equation

$$S_s(\vec{r},E,\vec{\Omega},t) = \int_{E'} dE' \int_{4\pi} d\vec{\Omega}'\; v(E')N(\vec{r},E',\vec{\Omega}',t)$$

$$\times\, \Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega},t) \quad . \tag{3}$$

where $\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega},t)$ is the macroscopic differential cross section for scattering particles from energy E' and direction $\vec{\Omega}'$ to energy E and direction $\vec{\Omega}$. Here we have denoted the particle speed as v(E') for particles of energy, E', since particle speed is a function of particle energy. The fission contribution to the source is denoted $S_F(\vec{r},E,\vec{\Omega},t)$ and is given by

$$S_F(\vec{r},E,\vec{\Omega},t) = \int_{4\rho\pi} d\vec{\Omega}' \int_{E'} \chi(\vec{r},E'\rightarrow E)\nu\Sigma_f(\vec{r},E',\vec{\Omega}'\rightarrow\vec{\Omega},t)$$

$$\times\ \nu(E')N(\vec{r},E',\vec{\Omega}',t) \quad , \tag{4}$$

where $\chi(\vec{r},E'\rightarrow E)$ represents the probability of particles appearing at energy E as a result of a fission caused by a particle of energy E' at space point $\vec{r}$; $\Sigma_f(\vec{r},E',\vec{\Omega}'\rightarrow\vec{\Omega},t)$ is the macroscopic cross section for fission induced by neutrons with energy E' and direction $\vec{\Omega}'$, with emergent neutrons from the fission having direction $\vec{\Omega}$; and ν is the average number of neutrons emerging from a fission. The above expression for the fission source makes no distinction between prompt and delayed particles. All particles have been assumed to emerge simultaneously from fission. This assumption is valid for most applications since the vast majority of transport calculations are time independent, that is, steady state. Only for fission system dynamics applications should delayed particles be explicitly described.

The inhomogeneous (or fixed) source is denoted $Q(\vec{r},E,\vec{\Omega},t)$. It represents all sources not dependent on the particle density in the medium. Thus, the total source rate of particles in Eq. (1) is

$$S(\vec{r},E,\vec{\Omega},t) = S_s(\vec{r},E,\vec{\Omega},t) + S_F(\vec{r},E,\vec{\Omega},t) + Q(\vec{r},E,\vec{\Omega},t) \quad . \tag{5}$$

Up to this point, the time variable has been included in the Boltzmann transport equation. Since the usual application of transport methods is to time-independent problems, we limit our considerations in this report to the time-dependent equation. Accordingly, the remaining development will be simplified to the time-independent Boltzmann transport equation,

$$\vec{\Omega}\cdot\vec{\nabla}\Phi(\vec{r},E,\vec{\Omega}) + \Sigma_t(\vec{r},E,\vec{\Omega})\Phi(\vec{r},E,\vec{\Omega}) = S(\vec{r},E,\vec{\Omega}) \quad , \tag{6}$$

in which the angular particle flux $\Phi(\vec{r},E,\vec{\Omega}) \equiv v(E)N(\vec{r},E,\vec{\Omega})$. As before, the total source rate of particles $S(\vec{r},E,\vec{\Omega})$ is written

$$S(\vec{r},E,\vec{\Omega}) = S_s(\vec{r},E,\vec{\Omega}) + S_F(\vec{r},E,\vec{\Omega}) + Q(\vec{r},E,\vec{\Omega}) \quad , \tag{7}$$

with

$$S_s(\vec{r},E,\vec{\Omega}) = \int_{4\pi} d\vec{\Omega}' \int_{E'} dE' \; \Phi(\vec{r},E',\vec{\Omega}')\Sigma_s(\vec{r},E'\rightarrow E,\vec{\Omega}'\rightarrow\vec{\Omega}) \quad , \tag{8}$$

$$S_F(\vec{r},E,\vec{\Omega}) = \int_{4\pi} d\vec{\Omega}' \int_{E'} dE' \; [\chi(\vec{r},E'\rightarrow E)\nu\Sigma_f(\vec{r},E',\vec{\Omega}'\rightarrow\vec{\Omega}) \times \Phi(\vec{r},E',\vec{\Omega})] \quad , \tag{9}$$

and with $Q(\vec{r},E,\vec{\Omega})$ denoting the inhomogeneous, or fixed, source rate of particles.

B. Coordinate Systems and Divergence Operator Forms

The streaming term $v\vec{\Omega}\cdot\vec{\nabla}N(\vec{r},E,\vec{\Omega}) = \vec{\Omega}\cdot\vec{\nabla}\Phi(\vec{r},E,\vec{\Omega})$ in Eq. (6) actually represents the rate of change of the particle angular flux Φ along the streaming path, s, in the direction of particle motion $\vec{\Omega}$. In other words, $\vec{\Omega}\cdot\vec{\nabla}\Phi = d\Phi/ds$ where the arguments of Φ, $(\vec{r},E,\vec{\Omega})$ have been omitted for simplicity. A description of the path, s, requires the specification of up to five variables (three spatial variables and two variables to define the direction $\vec{\Omega}$). The choice of these variables is governed by both the geometrical coordinate system to be used and a suitable angular-direction coordinate system. Thus, the particular form of the divergence operator $\vec{\Omega}\cdot\vec{\nabla}$ d/ds or, more specifically, the streaming term $\vec{\Omega}\cdot\vec{\nabla}\Phi$ $d\Phi/ds$, requires specification of suitable coordinate systems.

Shown below are the three common coordinate systems used in deterministic transport codes and the form of the streaming term for each. Note that in each geometrical coordinate system, an angular direction coordinate system is defined in which the direction variable $\vec{\Omega}$ is described in terms of a polar angle (or its cosine) measured from a directional coordinate axis and an azimuthal angle specifying the angle of rotation about that axis. The symbol $\hat{e}$ denotes direction.

1. Rectangular Cartesian Coordinates (x,y,z). The three-dimensional rectangular Cartesian coordinate system is shown in Fig. 2, together with the angular-direction coordinate system used to define the direction $\vec{\Omega}$. In

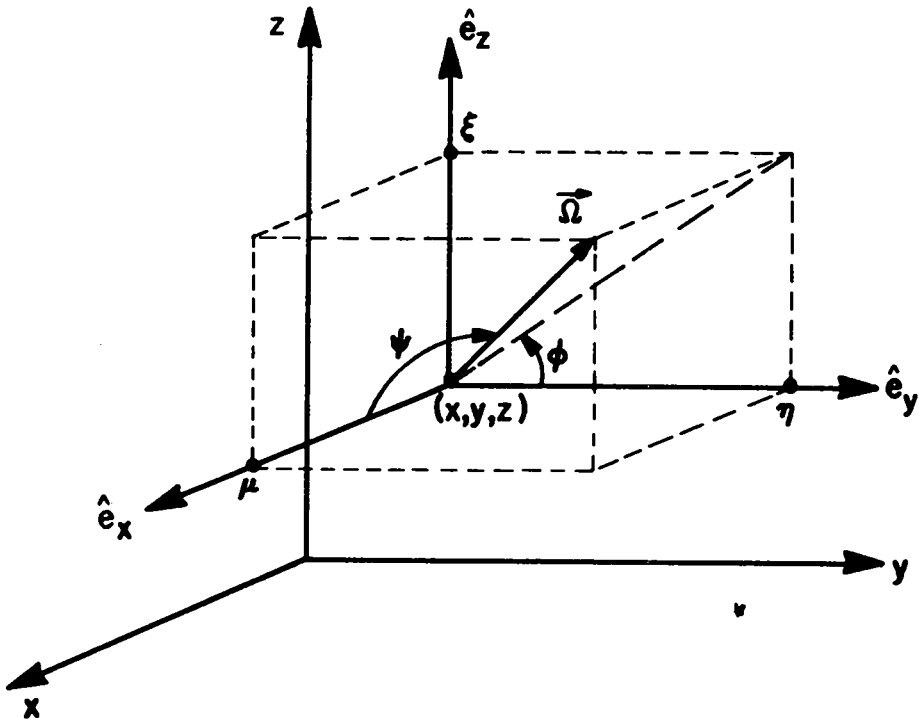


Fig. 2. Rectangular Cartesian coordinates.

this system, a space point is described by its (x,y,z) coordinates, an incremental spatial volume dV = dxdydz, and

$$\Omega \equiv \hat{e}_x \Omega_x + \hat{e}_y \Omega_y + \hat{e}_z \Omega_z \quad , \tag{10}$$

where

$$\Omega_x \equiv \hat{e}_x \cdot \vec{\Omega} = \cos\psi \quad \mu \quad , \tag{11a}$$

$$\Omega_y \equiv \hat{e}_y \cdot \vec{\Omega} = \sin\psi \cos\phi = \sqrt{1 - \mu^2} \cos\phi \equiv \eta \quad , \tag{11b}$$

$$\Omega_z \equiv \hat{e}_z \cdot \vec{\Omega} = \sin\psi \sin\phi = \sqrt{1 - \mu^2} \sin\phi \equiv \xi \quad , \tag{11c}$$

and $d\vec{\Omega} = d\mu d\phi$.

The divergence operator $\vec{\Omega} \cdot \vec{\nabla}$ is written, in general, as

$$\vec{\Omega} \cdot \vec{\nabla} = \frac{d}{ds} = \frac{dx}{ds}\frac{\partial}{\partial x} + \frac{dy}{ds}\frac{\partial}{\partial y} + \frac{dz}{ds}\frac{\partial}{\partial z} + \frac{d\mu}{ds}\frac{\partial}{\partial \mu} + \frac{d\phi}{ds}\frac{\partial}{\partial \phi} \quad .$$

With the rectangular coordinates, $dx/ds = \mu$, $dy/ds = \eta$, $dz/ds = \xi$, and $d\mu/ds = d\phi/ds = 0$. Thus,

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \mu \frac{\partial \Phi}{\partial x} + \eta \frac{\partial \Phi}{\partial y} + \xi \frac{\partial \Phi}{\partial z} \quad , \tag{12}$$

where $\Phi(\vec{r},E,\vec{\Omega}) = \Phi(x,y,z,E,\mu,\phi)$.

For two-dimensional rectangular Cartesian (x,y) geometry, there is no z-dependence of the particle angular flux, and Eq. (12) reduces to

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \mu \frac{\partial \Phi}{\partial x} + \eta \frac{\partial \Phi}{\partial y} \quad , \tag{13}$$

where $\Phi = \Phi(x,y,E,\mu,\phi)$.

For one-dimensional rectangular Cartesian (slab) geometry, there is neither a y- nor a z-dependence of the particle angular flux, and Eq. (12) reduces to

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \mu \frac{\partial \Phi}{\partial x} \quad , \tag{14}$$

where $\Phi = \Phi(x,E,\mu,\phi)$. In many one-dimensional slab applications, azimuthal angular symmetry exists, so the angular flux is a function only of the variables x, E, and μ.

2. General Cylindrical Coordinates (r,θ,z). The general three-dimensional cylindrical coordinate system is shown in Fig. 3, together with an angular-direction coordinates system used to define the particle direction $\vec{\Omega}$. In Fig. 3, the η-ξ plane is tangent to the cylindrical surface at (r,θ). In this system, a spatial point is defined by its (r,θ,z) coordinates, an incremental spatial volume is given by $dV = rdrd\theta dz$, and

$$\vec{\Omega} = \hat{e}_r\Omega_\theta + \hat{e}_\theta\Omega_\theta + \hat{e}_z\Omega_z \quad , \tag{15}$$

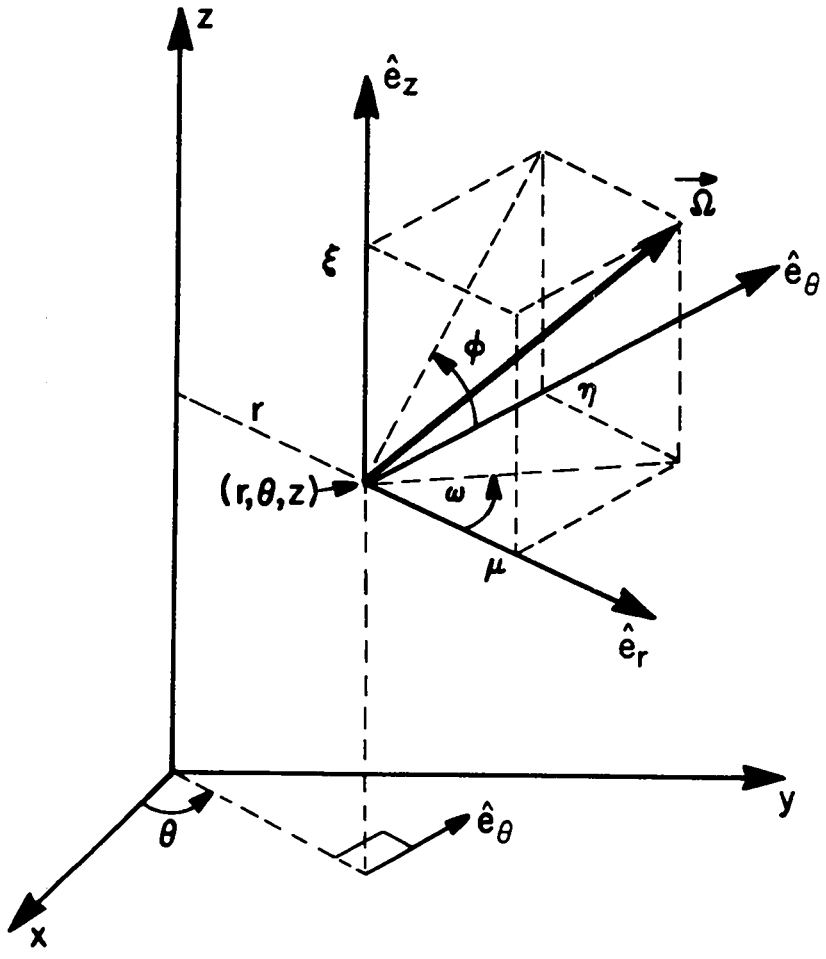


Fig. 3. General cylindrical coordinates.

where

$$\Omega_r \equiv \hat{e}_r \cdot \vec{\Omega} = \sqrt{1 - \xi^2}\ \cos\omega \equiv \mu \quad , \tag{16a}$$

$$\Omega_\theta \equiv \hat{e}_\theta \cdot \vec{\Omega} = \sqrt{1 - \xi^2}\ \sin\omega \equiv \eta \quad , \tag{16b}$$

$$\Omega_z \equiv \hat{e}_z \cdot \vec{\Omega} \equiv \xi \quad , \tag{16c}$$

and $d\vec{\Omega} = d\xi d\omega$.

The divergence operator $\vec{\Omega} \cdot \vec{\nabla}$ is

$$\vec{\Omega} \cdot \vec{\nabla} = \frac{d}{ds} = \frac{dr}{ds}\frac{\partial}{\partial r} + \frac{d\theta}{ds}\frac{\partial}{\partial \theta} + \frac{dz}{ds}\frac{\partial}{\partial z} + \frac{d\xi}{ds}\frac{\partial}{\partial \xi} + \frac{d\omega}{ds}\frac{\partial}{\partial \omega} \quad .$$

With cylindrical coordinates, $dr/ds = \mu$, $d\theta/ds = \eta/r$, $dz/ds = \xi$, $d\xi/ds = 0$, and $d\omega/ds = -\eta/r$. With some manipulation,

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \frac{\mu}{r}\frac{\partial(r\Phi)}{\partial r} + \frac{\eta}{r}\frac{\partial \Phi}{\partial \theta} - \frac{1}{r}\frac{\partial(\eta\Phi)}{\partial \omega} + \xi\frac{\partial \Phi}{\partial z} \quad , \tag{17}$$

where $\Phi = \Phi(r,\theta,z,E,\xi,\omega)$.

The selection of the angular direction variables ξ, ω for the specification of $\vec{\Omega}$ is arbitrary, and the variables μ, ϕ could just as well be used. In the latter case,

$$\Omega_r \equiv \hat{e}_r \cdot \vec{\Omega} = \mu \quad , \tag{18a}$$

$$\Omega_\theta \equiv \hat{e}_\theta \cdot \vec{\Omega} = \sqrt{1 - \mu^2}\cos\phi = \eta \quad , \tag{18b}$$

$$\Omega_z \equiv \hat{e}_z \cdot \vec{\Omega} = \sqrt{1 - \mu^2}\sin\phi = \xi \quad , \text{ and} \tag{18c}$$

$$d\vec{\Omega} = d\mu d\phi.$$

Equation (17) remains unchanged.

In two space dimensions, there are two rather widely used cylindrical geometries, both subsets of the general (r,θ,z) geometry. These are the finite, or (r,z), cylindrical geometry and the planar, or (r,θ), cylindrical geometry.

In (r,z) two-dimensional cylindrical geometry, a space point is defined by the spatial coordinates (r,z), an incremental volume dV is given by $2\pi r dr dz$, and Eq. (17) reduces to

$$\vec{\Omega} \cdot \nabla\Phi = \frac{\mu}{r}\frac{\partial(r\Phi)}{\partial r} - \frac{1}{r}\frac{\partial(\eta\Phi)}{\partial \omega} + \xi\frac{\partial \Phi}{\partial z} \quad , \tag{19}$$

with $\Phi = \Phi(r,z,E,\xi,\omega)$ or $\Phi(r,z,E,\mu,\phi)$. The spatial variable θ does not appear.

In (r,θ) two-dimensional cylindrical geometry, a space point is defined by the spatial coordinates (r,θ), an incremental volume dV is given by $rdrd\theta$, the angular particle flux is described by $\Phi(r,\theta,E,\xi,\omega)$ or $\Phi(r,\theta,E,\mu,\phi)$, and Eq. (17) reduces to

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \frac{\mu}{r}\frac{\partial(r\Phi)}{\partial r} + \frac{\eta}{r}\frac{\partial\Phi}{\partial\theta} - \frac{1}{r}\frac{\partial(\eta\Phi)}{\partial\omega} \quad . \tag{20}$$

In one-dimensional cylindrical geometry, a space point is defined solely by r, its radial position. An incremental volume dV is given by $2\pi rdr$, and Eq. (17) reduces to

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \frac{\mu}{r}\frac{\partial(r\Phi)}{\partial r} - \frac{1}{r}\frac{\partial(\eta\Phi)}{\partial\omega} \quad . \tag{21}$$

The angular particle flux is $\Phi(r,E,\xi,\omega)$ or, equivalently, $\Phi(r,E,\mu,\phi)$.

3. One-Dimensional Spherical Coordinates. In spherical coordinates, the only geometry for which deterministic transport has received much attention is the one (space) dimensional sphere. Although a two-dimensional spherical geometry computer code has been developed,[1] its usage has been quite specialized and limited. Accordingly, this section will be limited to one-dimensional spherical geometry.

The coordinate system for spherical geometry is shown in Fig. 4, together with the angular direction coordinate system used to define the direction $\vec{\Omega}$. In Fig. 4, the η-ξ plane is tangent to the spherical surface of radius r. For one-dimensional spheres, a space "point" is defined simply by its radius, r, and the incremental volume dV associated with this point is the spherical shell of volume $4\pi r^2 dr$. The angular variable $\vec{\Omega}$ is defined solely by $\mu \quad \hat{e}_r \cdot \vec{\Omega}$ with no dependence on the azimuthal angle ϕ. Thus, the angular particle flux Φ is described by the arguments (r,E,μ) and

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \frac{d\Phi}{ds} = \frac{\partial\Phi}{\partial r}\frac{dr}{ds} + \frac{\partial\Phi}{\partial\mu}\frac{d\mu}{ds} \quad .$$

For this geometry, $dr/ds = \mu$ and $d\mu/ds = (1 - \mu^2)/r$ so that, with some rearrangement,

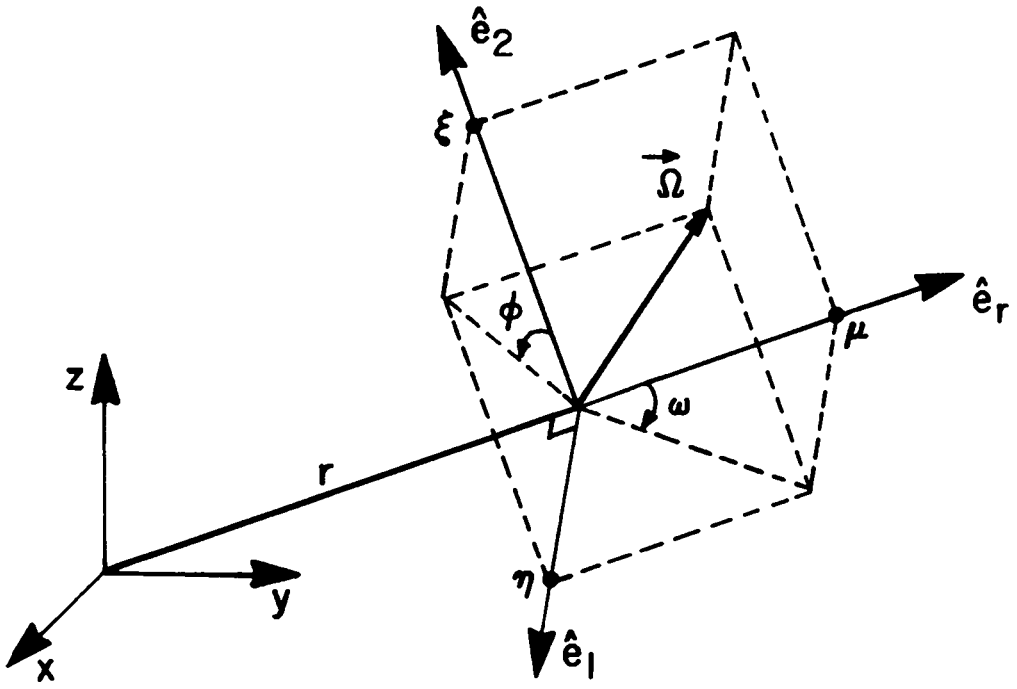


Fig. 4. Spherical geometry coordinates.

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \frac{\mu}{r^2}\frac{\partial(r^2\Phi)}{\partial r} + \frac{1}{r}\frac{\partial[(1-\mu^2)\Phi]}{\partial \mu} \quad , \tag{22}$$

where the argument of Φ has been omitted for simplicity.

4. Angular Redistribution in Curvilinear Geometries. In curvilinear geometries such as those described in Figs. 3 and 4, a particle transport phenomenon occurs that does not occur in rectangular Cartesian geometries. This phenomenon is known as angular redistribution and is defined in cylindrical geometries by the term

$$\frac{1}{r}\frac{\partial(\eta\Phi)}{\partial \omega}$$

in Eqs. (17), (19), (20), (21), and in spherical geometry by the term

$$\frac{1}{r}\frac{\partial[(1-\mu^2)\Phi]}{\partial \mu}$$

in Eq. (22). To understand the physical meaning of angular redistribution, recognize that in both cylindrical and spherical geometries the angular variable μ is proportional to the cosine of the angle ω shown in Figs. 3 and 4. (In cylindrical geometries, the angular variable η is proportional to $\sin\omega$.) In both geometries, the angle ω is measured from the radius vector, r. As a particle moves without collision in a straight line from the point r_1 to the point r_2, the angle ω changes from ω_1 to ω_2 (Fig. 5). Angular redistribution is simply the change in the directional variable μ (and η in cylindrical geometry) as particles move from one radial position to another. Several observations can be made regarding angular redistribution.

First, there can be no net gain or loss of particles because of angular redistribution, that is, if one integrates over all angles, the net redistribution gain or loss term must vanish.

Second, there is no way in which a particle, moving without collision and with $\omega \neq 0$, can acquire a direction $\omega = \pi(\cos\omega = -1)$ by angular redistribution. Similarly, there is no way for a particle moving without collision and with $\cos\omega \neq 1$ to become anything but a particle with $\cos\omega$ closer to unity because of angular redistribution. Restating this second observation, angular redistribution always serves to increase the value of directional variable μ (or $\cos\omega$) provided $\omega \neq n\pi$, $n = 0, 1, 2$.

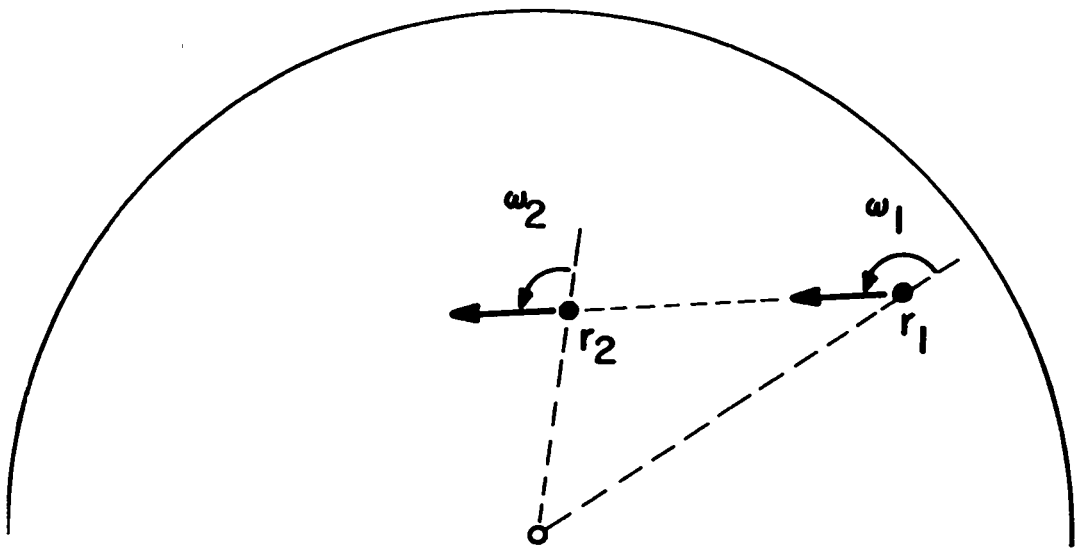


Fig. 5. Illustration of angular redistribution of particles in curvilinear coordinates.

Third, there is no angular redistribution for particles with $\cos\omega = \pm 1$, that is, for particles moving with directions along a radius vector of a cylinder or sphere.

Finally, angular redistribution involves only the directional variables μ and η, as shown in Figs. 3 and 4. The value of the directional variable ξ is unchanged with angular redistribution.

C. Boundary Conditions

The Boltzmann transport equation is normally used to describe transport of particles in a finite region of space in which cross sections are known functions of particle energy and position. To effect the solution to the transport equation corresponding to the physical system being modeled, it is necessary to specify the appropriate conditions on the particle density, or flux, at the external boundaries of the region. Below are described the boundary conditions most commonly used in deterministic transport calculations.

1. Vacuum Boundary. If no particles enter the region of solution from external sources and if a particle, once it exits the region across its external boundary, cannot return to the region, then the boundary is called a free surface or vacuum boundary. Let $\vec{n}$ denote the outward-directed unit normal vector at the boundary surface at spatial position $\vec{r}_0$. Then, at a vacuum boundary, any particle having $\vec{n} \cdot \vec{\Omega} > 0$ will be crossing the boundary in an outward direction and any particle having $\vec{n} \cdot \vec{\Omega} < 0$ will be crossing in an inward direction. The vacuum boundary condition, then, is

$$\Phi(\vec{r}_0, E, \vec{\Omega}) = 0 \quad , \quad \text{if } \vec{n} \cdot \vec{\Omega} < 0 \quad .$$

This boundary condition is the one most commonly applied at the external surfaces of the region of solution.

In reality, of course, the vacuum boundary is an idealization. Particles leaving a system will always have a finite probability of returning to the system. Nevertheless, the vacuum boundary condition is quite acceptable if either the probability of particle return is negligible or the boundary surface is so far removed from the volume of interest that an approximate boundary condition is sufficient.

2. Reflecting Boundary. The reflecting boundary occurs at a plane of symmetry in the system being analyzed. At a reflecting boundary, the value of

the angular flux for incoming directions is set equal to the value of the outgoing flux in the direction corresponding to specular (mirror-like) reflection. For example, in a rectangular Cartesian geometry (see Fig. 2), if a y-z plane is a reflecting surface, then at the surface the incoming particle flux with direction cosines μ, η, ξ is set equal to the outgoing particle flux with direction cosines $-\mu$, η, ξ. Although rigorously correct only for planes of symmetry, the reflecting boundary is frequently applied at the radial center line of cylinders and at the origin of spheres. The theoretically correct boundary conditions for these two cases are described below. In practice, the use of reflecting boundary conditions at the radial origin of cylinders and spheres usually yields satisfactory results.

In cylindrical "cell" calculations, the reflecting boundary condition is also frequently used at the outer radial surface of the cell. Such calculations are used to analyze a typical cylinder in an extended lattice of cylinders, in which case the cell is usually a fuel rod surrounded by an annulus of moderator or coolant. Use of the reflecting boundary condition at the cell surface is satisfactory only if the moderating annulus is reasonably thick (about one thermal neutron mean free path or greater).

3. Spherical Origin Boundary Condition. In one-dimensional spherical systems, a boundary condition is required at the center of the sphere. The theoretically correct condition is that the angular flux be isotropic at the center. The value of the angular flux at the center of the sphere can be found by first solving the Boltzmann transport equation for $\mu = -1$, that is, for a straight-in directed particle at the origin. Then, $\Phi(r=0,E,\vec{\Omega}) = \Phi(r=0,E,\mu=-1)$ for all other $\vec{\Omega}(\mu)$.

4. Cylindrical Origin Boundary Conditions. In cylindrical geometries, a boundary condition is required at the radial origin. The theoretically correct condition is that for a fixed value of the polar angle (or its cosine, ξ) as shown in Fig. 3, the flux is azimuthally isotropic along the cylinder's radial center line. The value of the angular flux at $r = 0$ can be found by first solving the Boltzmann transport equation for $\eta = 0$, $\mu = -\sqrt{1 - \xi^2}$. Then, letting $\vec{\Omega} = \vec{\Omega}(\xi,\eta)$, $\Phi(r=0,E,\xi,\eta) = \Phi(r,=0,E,\xi,\eta=0)$.

5. Periodic Boundary Condition. The periodic boundary condition sets the values of the incoming angular fluxes at a boundary equal in detail to the values of the outgoing angular fluxes on the opposite boundary. The periodic

condition is used on the boundaries of an asymmetric "unit" cell, which represents one of an "infinite" array of such cells. The boundary condition can be applied to x-, y-, or z-dimension boundaries in Cartesian coordinates, or to the z-dimension boundaries of (r,z) or (r,θ,z) cylindrical geometries. It must be used as the θ-dimension boundary condition where the θ-dimension is used to represent a 360° circular mesh in, for example, (r,θ) cylindrical geometry.

6. White Boundary Condition. With the white boundary condition, the values of the incoming angular boundary fluxes are set equal to a constant value. In other words, the incoming angular flux is made isotropic. The constant value used for the incoming angular flux is the average of the outgoing angular fluxes such that the net flow of particles across the boundary is zero. For example, in one-dimensional cylindrical geometry with a white boundary condition at radius R,

$$\Phi(R,E,\mu,\phi) = \frac{\int_0^{2\pi}\int_0^{1} \mu'\Phi(R,E,\mu',\phi')\, d\mu' d\phi}{\int_0^{2\pi}\int_0^{1} \mu' d\mu' d\phi'}$$

for $\mu\varepsilon[-1,0]$.

This condition was designed to be meaningful as an exterior boundary condition for cylindrical "cell" calculations,[2,3] but with limited success. A variant of the white boundary condition, known as the cylindrical boundary condition,[4] adjusts the incoming angular flux to be azimuthally isotropic for constant values of the polar angle (or its cosine, ξ). This cylindrical boundary condition shows evidence of producing good results for cylindrical cell calculations.

7. Albedo Boundary Condition. The albedo, or grey, boundary condition is similar to the white boundary condition described above except that the ratio of the incoming (isotropic) particle current to the outgoing current is a constant, $\alpha < 1$. For example, in one-dimensional plane geometry with a grey boundary condition at X = R,

$$\Phi(R,E,\mu) = \alpha \frac{\int_0^1 \mu'\Phi(R,E,\mu')\, d\mu'}{\int_0^1 \mu' d\mu'}$$

for $\mu\varepsilon[-1,0]$.

When $\alpha = 1$, the albedo condition is identical to the white boundary condition. The albedo condition is sometimes used to approximate the effects of external materials for which a detailed calculation is not required. Another variant of the albedo condition permits the application of the albedo fraction, α, to the reflective boundary condition so the angular incoming flux is equal to α times the outgoing angular flux in the direction corresponding to specular reflection. For example, in spherical geometry with an albedo-reflective boundary condition at radius R,

$$\Phi(R,E,\mu) = \alpha\Phi(R,E,-\mu)$$

for $\mu\varepsilon[-1,0]$.

D. Spherical Harmonics Expansion of the Source Terms

Spherical harmonics series expansions are commonly used in representing the angular dependence of the source terms in the Boltzmann transport equation. In this section, the use of these expansions is examined for each of the contributions to the total source of particles, namely scattering, fission, and inhomogeneous sources.

1. Scattering Source Expansion. The differential scattering cross section $\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega})$ represents the probability that a particle of energy E' and direction $\vec{\Omega}'$ will emerge from the scattering collision as a particle of energy E and direction $\vec{\Omega}$. For most materials, it is satisfactory to replace the angular argument $\vec{\Omega}' \to \vec{\Omega}$ with $\mu_0 \equiv \vec{\Omega}'\cdot\vec{\Omega}$. Then we say that the material is isotropic, that is, there is no preferred direction in the material itself. In other words, the scattering angle or its cosine, μ_0, is usually sufficient for describing the angular dependence of the scattering cross section. Thus,

$$\Sigma_s(\vec{r},E'\to E,\vec{\Omega}'\to\vec{\Omega}) = \Sigma_s(\vec{r},E'\to E,\mu_0) \quad . \tag{23}$$

Since the angular dependence of the differential scattering cross section is now described in terms of μ_0, it is usually advantageous in reactor physics to represent the cross section by a finite Legendre polynomial expansion

$$\Sigma_s(\vec{r},E'\to E,\mu_0) = \sum_{\ell=0}^{L} \left(\frac{2\ell + 1}{4\pi}\right) \Sigma_s^{\ell} (\vec{r},E'\to E)\, P_\ell(\mu_0)^{\ell} , \tag{24}$$

where

$$\Sigma_s^{\ell} (\vec{r},E'\to E \quad 2\pi \int_{-1}^{1} d\mu_0\, \Sigma_s (r,E'\to E,\mu_0)\, P_\ell(\mu_0) \quad . \tag{25}$$

The series truncation index, L, should be large enough to provide adequate representation of the differential cross section. If the scattering is isotropic - that is, independent of the scattering angle μ_0 - then clearly L = 0. If the scattering is linearly varying in μ_0 (linearly anisotropic scattering), then L = 1. For higher degrees of anisotropy in the differential scattering cross section, larger values of L should be selected. Practical considerations, however, usually limit the value of L to 3, or perhaps 5, even though such a limit may not provide an entirely satisfactory approximation to the true angular variation in the differential cross section.

The first few Legendre polynomials are

$$P_0(x) = 1$$

$$P_1(x) = x$$

$$P_2(x) = \frac{1}{2} (3x^2 - 1)$$

$$P_3(x) = \frac{1}{2} (5x^3 - 3x)$$

$$P_4(x) = \frac{1}{8} (35x^4 - 30x^2 + 3)$$

$$P_5(x) = \frac{1}{8} (63x^5 - 70x^3 + 15x) \quad .$$

Since the particle direction of motion $\vec{\Omega}$ is defined by the variables μ and ϕ indicated in Figs. 2-4,

$$\mu_0 = \vec{\Omega} \cdot \vec{\Omega}' = \mu\mu' + (1 - \mu^2)^{1/2}(1 - \mu'^2)^{1/2} \cos(\phi - \phi') \quad ,$$

and the addition theorem for spherical harmonics can be used to define $P_\ell(\mu_0)$ in terms of μ, μ', ϕ, and ϕ' by

$$P_\ell(\mu_0) = \frac{4\pi}{2\ell + 1} \sum_{m=-\ell}^{\ell} Y^*_{\ell m}(\mu',\phi') Y_{\ell m}(\mu,\phi) \quad . \tag{26}$$

The $Y_{\ell m}$ are the spherical harmonics with the definition

$$Y_{\ell m}(\mu,\phi) = \frac{2\ell + 1}{4\pi} \frac{(\ell - m)!}{(\ell + m)!} P^m_\ell(\mu)\, e^{im\phi} \quad , \tag{27}$$

where

$$P^m_\ell(\mu) = (-1)^m(1- \mu^2)^{m/2} \frac{d^m P_\ell(\mu)}{d\mu^m}$$

is the associated Legendre function of the first kind, and

$$Y_{\ell,-m}(\mu,\phi) = (-1)^m Y^*_{\ell,m}(\mu,\phi) \quad .$$

To represent the scattering source of Eq. (8), we also expand the angular flux in terms of the spherical harmonics as

$$\Phi(\vec{r},E',\vec{\Omega}') = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} \Phi^m_\ell(\vec{r},E')\, Y_{\ell m}(\mu',\phi') \quad , \tag{28}$$

where

$$\Phi^m_\ell(\vec{r},E') = \int_{-1}^{1} d\mu \int_0^{2\pi} d\phi\, \Phi(\vec{r},E',\vec{\Omega})\, Y^*_{\ell m}(\mu,\phi) \quad . \tag{29}$$

Substituting Eqs. (24), (26), and (28) into Eq. (8), and evaluating, yields

$$S_s(\vec{r},E,\vec{\Omega}) = \int_0^\infty dE' \sum_{\ell=0}^{L} \Sigma_s^\ell (\vec{r},E'\to E) \sum_{m=-\ell}^{\ell} Y_{\ell m}(\mu,\phi)\Phi_\ell^m(\vec{r},E') \quad . \tag{30}$$

This gives the scattering source in the general case in terms of a spherical harmonic expansion. For some special geometric situations, we can take advantage of symmetry to simplify this expression so not all the terms are needed. For example, in one-dimensional slab and spherical geometries with azimuthal (ϕ) angular symmetry, the angular flux in Eq. (29) is a function of μ only. Thus,

$$\phi_\ell^m(\vec{r},E') = \begin{cases} \sqrt{(2\ell+1)\pi} \int_{-1}^{1} d\mu\ \phi(r,E',\mu)\ P_\ell(\mu), & \text{for } m = 0 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad , & \text{for } m \neq 0 \ , \end{cases} \tag{31}$$

where we have used the fact that

$$Y_{\ell 0}(\mu,\phi) = \sqrt{\frac{2\ell + 1}{4\pi}}\ P_\ell(\mu) \quad .$$

Therefore, for one-dimensional spheres and slabs, Eq. (3) reduces to

$$S_s(\vec{r},E,\vec{\Omega}) = \int_0^\infty dE' \sum_{\ell=0}^{L} \frac{2\ell + 1}{2}\ \Sigma_s^\ell (r,E'\to E)\ P_\ell(\mu)\ \Phi_\ell^0(\vec{r},E') \quad . \tag{32}$$

With two-dimensional symmetry, the flux in Eq. (29) as a function of ϕ is symmetric about $\phi = \pi$ [assuming the appropriate definition of ϕ - for example, ϕ for (r,z) geometry measured about the $\hat{e}_\theta$ axis, as shown in Fig. 3]. In this case, Eq. (29) becomes

$$\phi_{\ell}^{0} = 2 \int_{-1}^{1} d\mu \int_{0}^{\pi} d\phi \; \Phi(\vec{r},E',\vec{\Omega}) \; Y^{*}_{\ell,0}(\mu,0) \quad ,$$

$$\phi_{\ell}^{m} = \begin{cases} \int_{-1}^{1} d\mu \int_{0}^{\pi} \phi(\vec{r},E',\vec{\Omega})[Y_{\ell m}(\mu,0) + \overset{*}{Y}_{\ell m}(\mu,0)], & \text{for } m > 0 \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad , & \text{for } m < 0 \quad , \end{cases} \tag{33}$$

and this effectively reduces by half the number of moments that need be computed for the scattering source. For one-dimensional cylindrical geometry, a $\pi/2$ symmetry exists in ϕ in the angular flux; therefore, reduction in the number of moments is reduced further. Thus, in Eq. (33) all moments with $\ell + m$ odd are zero for the one-dimensional cylinders.

We can summarize all these special cases by writing the following form for the scattering source:

$$S_s(\vec{r},E,\vec{\Omega}) = \int_{0}^{\infty} dE' \sum_{n=1}^{NM} (2\ell + 1) \; \Sigma_s^{\ell} (\vec{r},E' \rightarrow E) \; R_n(\vec{\Omega}) \; \tilde{\Phi}_n(\vec{r},E') \quad , \tag{34}$$

where

NM = the total number of moments,

$R_n(\vec{\Omega})$ = the appropriate spherical harmonic functions indexed by the simple index n, and

$\tilde{\Phi}_n(\vec{r},E')$ = the flux moment corresponding to the spherical harmonic indexed by n.

Equation (34) is a conveniently programmable form of Eq. (30). In Appendix A, we show how to evaluate the R_n and $\tilde{\Phi}_n$ for both general and specific cases. In Table I is the number of required spherical harmonic terms for each order of scattering. In Table II are the results of our derivation in Appendix A, giving the expansion terms explicitly for scattering orders up to L = 3. Note that the flux moment corresponding to n = 1 in Eq. (34), $\tilde{\Phi}_1$, is simply the scalar flux, as is ϕ_0^0 using Eq. (29). We will normally denote the scalar flux simply as Φ_0.

TABLE I

NUMBER OF SPHERICAL HARMONICS, NM, AS A FUNCTION OF LEGENDRE EXPANSION ORDER, L

L	One-Dimensional Standard Planes and Spheres	One-Dimensional Cylinders	Two-Dimensional Geometries	Two-Angle Planes and Three-Dimensional Geometries
0	1	1	1	1
1	2	2	3	4
2	3	4	6	9
3	4	6	10	16
4	5	9	15	25
5	6	12	21	36
L	L+1	$(L+2)^2/4$, L even $(L+1)(L+3)/4$, L odd	$(L+1)(L+2)/2$	$(L+1)^2$

2. Fission Source. Fission is normally treated as an isotropic process. Accordingly, when the fission source term of Eq. (9) is expanded in a Legendre series representation, similar to the scattering source, only the first term in the expansion is retained and $S_F(\vec{r},E,\vec{\Omega})$ is written

$$S_F(\vec{r},E,\vec{\Omega}) = \int_{E'} dE' \; \chi(r,E'\to E)\nu\Sigma_f(\vec{r},E') \; \Phi_0(\vec{r},E') \quad , \tag{35}$$

where $\Phi_0(\vec{r},E)$ is the scalar flux as given by Eq. (29) for ℓ, m = 0. In Eq. (35), the fission fraction, $\chi(\vec{r},E'\to E)$, is the probability that a fission induced by a particle with energy E' will produce a particle with energy E. It is frequently assumed that this fission fraction is independent of the energy

TABLE II

SPHERICAL HARMONICS, $R_n(\vec{\Omega})$, FOR DIFFERENT GEOMETRIES

N	One-Dimensional Standard Planes and Spheres P_5[a]	One-Dimensional Cylinders P_4	Two-Dimensional Geometries P_3	Two-Angle Planes and Three-Dimensional Geometries P_3
1	$P_0(\mu)$	$P_0(\mu)$	$P_0(\mu)$	$P_0(\mu)$
2	$P_1(\mu)$	$P_1^1(\mu)\cos\phi$	$P_1(\mu)$	$P_1(\mu)$
3	$P_2(\mu)$	$P_2(\mu)$	$P_1^1(\mu)\cos\phi$	$P_1^1(\mu)\cos\phi$
4	$P_3(\mu)$	$\frac{\sqrt{3}}{6} P_2^2(\mu)\cos2\phi$	$P_2^2(\mu)$	$P_1^1(\mu)\sin\phi$
5	$P_4(\mu)$	$\frac{\sqrt{6}}{6} P_3^1(\mu)\cos\phi$	$\frac{\sqrt{3}}{3} P_2^1(\mu)\cos\phi$	$P_2(\mu)$
6	$P_5(\mu)$	$\frac{\sqrt{10}}{6} P_3^3(\mu)\cos3\phi$	$\frac{\sqrt{3}}{6} P_2^2(\mu)\cos2\phi$	$\frac{\sqrt{3}}{3} P_2^1(\mu)\cos\phi$
7		$P_4(\mu)$	$P_3(\mu)$	$\frac{\sqrt{3}}{3} P_2^1(\mu)\sin\phi$
8		$\frac{\sqrt{5}}{30} P_4^2(\mu)\cos2\phi$	$\frac{\sqrt{6}}{6} P_3^1(\mu)\cos\phi$	$\frac{\sqrt{3}}{6} P_2^2(\mu)\cos2\phi$
9		$\frac{\sqrt{35}}{840} P_4^4(\mu)\cos4\phi$	$\frac{\sqrt{15}}{30} P_3^2(\mu)\cos2\phi$	$\frac{\sqrt{3}}{6} P_2^2(\mu)\sin2\phi$
10			$\frac{\sqrt{10}}{60} P_3^3(\mu)\cos3\phi$	$P_3(\mu)$
11				$\frac{\sqrt{6}}{6} P_3^1(\mu)\cos\phi$
12				$\frac{\sqrt{6}}{6} P_3^1(\mu)\sin\phi$
13				$\frac{\sqrt{15}}{30} P_3^2(\mu)\cos2\phi$
14				$\frac{\sqrt{15}}{30} P_3^2(\mu)\sin2\phi$
15				$\frac{\sqrt{10}}{60} P_3^3(\mu)\cos3\phi$
16				$\frac{\sqrt{10}}{60} P_3^3(\mu)\sin3\phi$

[a] P_L denotes L^{th} order Legendre expansion.

of the particle that induces the fission so that $\chi(\vec{r},E'\rightarrow E)$ is written simply as $\chi(\vec{r},E)$, the probability that a particle produced by fission will emerge with energy, E.

3. Inhomogeneous Source Expansion. In a manner similar to that used for the scattering source, the inhomogeneous (or fixed) source $Q(\vec{r},E,\vec{\Omega})$ can be represented as a finite expansion using the spherical harmonics $R_n(\vec{\Omega})$ defined in Table II. That is, we make the expansion

$$Q(\vec{r},E,\vec{\Omega}) = \sum_{\ell=0}^{L} \sum_{m=-\ell}^{\ell} Q_\ell^m(\vec{r},E)\ Y_{\ell m}(\mu,\phi) \quad ,$$

with

$$Q_\ell^m(\vec{r},E) = \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi\ Q(\vec{r},E,\vec{\Omega})\ Y_{\ell m}^*(\mu,\phi) \quad . \tag{36}$$

This is a general spherical harmonics expression for the inhomogeneous source; if we perform the same manipulations as in Appendix A, we can write a programmable version of this in the form

$$Q(\vec{r},E,\vec{\Omega}) = \sum_{n=1}^{NMQ} (2\ell + 1)\ R_n(\vec{\Omega})\ \tilde{Q}_n(\vec{r},E) \quad , \tag{37}$$

where $R_n(\vec{\Omega})$ are the same functions as in Appendix A, and NMQ is the total number of spherical harmonics (and fixed source moments) required for a given inhomogeneous source Legendre expansion order LQ as shown in Table I. The index ℓ is the subscript of the Legendre function $P_\ell^m(\mu)$ appearing in the appropriate $R_n(\vec{\Omega})$, $0 \leq \ell \leq LQ$. The $R_n(\vec{\Omega})$ are thus spherical harmonics appropriate to the geometry being used, and the $\tilde{Q}_n(\vec{r},E)$ are the fixed source angular moments corresponding to the $R_n(\vec{\Omega})$. The $R_n(\vec{\Omega})$ for typical Legendre expansion orders are listed in Table II.

Note that the Legendre expansion used for the scattering source is independent of the Legendre expansion order used for the inhomogeneous source.

For example, one might use a P_3 scattering order but only a P_0 (isotropic) expansion for the inhomogeneous source.

E. The Adjoint Equation

Virtually all deterministic transport codes can solve either the forward, or regular, transport equation described previously or the adjoint transport equation. The adjoint solutions, namely the adjoint fluxes, have the special physical significance of the "importance" of particles within the system being solved. The solutions to the adjoint transport equation are used in perturbation theory and variational calculations pertaining to nuclear systems.[5] The adjoint time-independent transport equation corresponding to Eq. (6) is

$$-\vec{\Omega} \cdot \vec{\nabla}\Phi^{+}(\vec{r},E,\vec{\Omega}) + \Sigma_t(\vec{r},E,\vec{\Omega})\Phi^{+}(\vec{r},E,\vec{\Omega}) = S^{+}(\vec{r},E,\vec{\Omega}) \quad , \tag{38}$$

where the superscript + denotes the adjoint functions. In Eq. (38), the adjoint source, $S^{+}(\vec{r},E,\vec{\Omega})$, can be expressed similarly to the regular source in Eqs. (7)-(9), namely

$$S^{+}(\vec{r},E,\vec{\Omega}) = S_s^{+}(\vec{r},E,\vec{\Omega}) + S_F^{+}(\vec{r},E,\vec{\Omega}) + Q^{+}(\vec{r},E,\vec{\Omega}) \quad , \tag{39}$$

with

$$S_s^{+}(\vec{r},E,\vec{\Omega}) = \int d\vec{\Omega}' \int dE' \; \Phi^{+}(\vec{r},E',\vec{\Omega})\Sigma_s(\vec{r},E\to E',\vec{\Omega}\to\vec{\Omega}') \quad , \tag{40}$$

$$S_F^{+}(\vec{r},E,\vec{\Omega}) = \int d\vec{\Omega}' \int dE' \; \nu\Sigma_f(\vec{r},E,\vec{\Omega}\to\vec{\Omega}') \times \chi(\vec{r},E\to E')\Phi^{+}(\vec{r},E',\vec{\Omega}') \quad , \tag{41}$$

and $Q^{+}(\vec{r},E,\vec{\Omega})$ representing the adjoint inhomogeneous "source."

With the spherical harmonics expansion technique described previously, the adjoint sources can be written

$$S_s^{+}(\vec{r},E,\vec{\Omega}) = \int_{E'} dE' \sum_{n=1}^{NM} (2\ell + 1) \; \Sigma_s^{\ell}(\vec{r},E\to E')R_n(\vec{\Omega})\tilde{\Phi}_n^{+}(\vec{r},E') \quad , \tag{42}$$

$$S_F^+(\vec{r},E,\vec{\Omega}) = \nu\Sigma_f(\vec{r},E) \int_{E'} dE' \; \chi(\vec{r},E\to E')\Phi_0^{+0}(\vec{r},E') \quad , \tag{43}$$

$$Q^+(\vec{r},E,\vec{\Omega}) = \sum_{n=1}^{NMQ} (2\ell + 1)R_n(\vec{\Omega})\tilde{Q}_n^+(\vec{r},E) \quad . \tag{44}$$

In these three equations, the spherical harmonics, $R_n(\vec{\Omega})$, the adjoint angular flux moments, $\tilde{\Phi}_n^+(\vec{r},E)$, and the adjoint fixed source angular moments, $\tilde{Q}_n^+(\vec{r},E)$, are defined in the manner described in Sec. D.

The complete specification of the adjoint problem requires specification of boundary conditions for the adjoint particle flux. The most common boundary condition is the vacuum, or free surface, boundary condition. For the regular particle flux at a vacuum boundary, $\vec{r}_0$, $\Phi(\vec{r}_0,E,\vec{\Omega}) = 0$ for all incoming directions, that is, for $\vec{\Omega} \cdot \hat{n} < 0$. The vacuum boundary condition for the adjoint particle flux is $\Phi^+(\vec{r}_0,E,\vec{\Omega}) = 0$ for all outgoing directions, that is, for $\vec{\Omega} \cdot \hat{n} > 0$.

III. NUMERICAL DESCRIPTION

The numerical description of the transport of neutral particles involves discretization of the independent variables of the transport equation. In the following, we start from the linear Boltzmann equation as developed in the previous section, and we consider each of the independent variables in turn. The general goal is to write the transport description in a numerical form that can be efficiently solved using modern computing techniques and machines. In this section, we present methods that, in common experience, lead to the most efficient and most easily programmable computational algorithms.

To give a clear exposition of the methods used for discretization purposes and to explicitly define the phase space cells, it is necessary to particularize the geometries. For nuclear reactor analysis, methods based upon orthogonal geometries are the most common. In this exposition we will restrict ourselves to geometries that can be described in one, two, or three dimensions

using rectangular Cartesian or cylindrical coordinates or in one dimension using spherical coordinates as described in the preceding section.* The basic reason that numerical methods are normally based upon orthogonal geometries is that the spatial region is readily represented in a rectangular domain or its three-dimensional generalization. As such, computationally simple prescriptions for neutral particles traveling through the rectangular domain can be established, and this permits efficient programming in codes based upon such methods.

In the following, we treat the discretization of each of the independent variables in turn. We start with the energy variable, then consider the angular variables, and end with the spatial variables. Our last consideration in this section involves the solution of the resultant set of numerical equations by a source iteration technique.

A. The Energy Variable - The Multigroup Method

Note in Eq. (6) that the energy variable appears only on the source side of the equation under an integral over the entire energy range. On the left side, it appears merely as a parameter. Therefore, the most commonly used discretization method is the multigroup method, in which the energy domain is partitioned into G intervals of width $E_{g-\frac{1}{2}} - E_{g+\frac{1}{2}} \equiv \Delta E_g$, $g = 1, 2, \cdots, G$. By convention, increasing g represents decreasing energy so that $E_{\frac{1}{2}} > E_{\frac{3}{2}} > \cdots > E_{g-\frac{1}{2}} > E_{g+\frac{1}{2}} > \cdots > E_{G+\frac{1}{2}}$. If we integrate Eq. (6) over ΔE_g, using Eqs. (7) and (8) and assuming for the moment that scattering and fission processes are represented by a transfer cross section, $\Sigma(\vec{r},E'\to E,\vec{\Omega}\cdot\vec{\Omega}')$, we obtain

$$\vec{\Omega}\cdot\vec{\nabla}\Phi_g(\vec{r},\vec{\Omega}) + \Sigma_{t,g}(\vec{r},\vec{\Omega})\Phi_g(\vec{r},\vec{\Omega})$$

$$= \sum_{g'=1}^{G} \int_{4\pi} d\vec{\Omega}'\, \Sigma_{g'\to g}(\vec{r},\vec{\Omega}'\cdot\vec{\Omega})\Phi_{g'}(\vec{r},\vec{\Omega}') + Q_g(\vec{r},\vec{\Omega}) \quad , \tag{45}$$

$$g = 1, \cdots, G \quad ,$$

---

*Some nonorthogonal triangular mesh methods are based upon the orthogonal (x,y) or (r,z) geometries, and many of the following considerations apply to them as well.

where

$$\Phi_g(\vec{r},\vec{\Omega}) \equiv \int_{\Delta E_g} \Phi(\vec{r},E,\vec{\Omega})dE \quad , \tag{46}$$

$$\Sigma_{t,g}(\vec{r},\vec{\Omega}) \equiv \frac{\int_{\Delta E_g} dE\ \Phi(\vec{r},E,\vec{\Omega})\Sigma_t(\vec{r},E)}{\Phi_g(\vec{r},\vec{\Omega})} \quad , \tag{47}$$

$$\Sigma_{g'\to g}(\vec{r},\vec{\Omega}'\to\vec{\Omega}) \equiv \frac{\int_{\Delta E_g} dE \int_{\Delta E_{g'}} dE'\ \Sigma(\vec{r},E'\to E,\vec{\Omega}'\cdot\vec{\Omega})\Phi(\vec{r},E',\vec{\Omega}')}{\Phi_{g'}(\vec{r},\vec{\Omega})} \quad , \tag{48}$$

and

$$Q_g(\vec{r},\vec{\Omega}) \int_{\Delta E_g} dE\ Q(\vec{r},E,\vec{\Omega}) \quad . \tag{49}$$

Here the group flux $\Phi_g(\vec{r},\vec{\Omega})$ defined by Eq. (46) is no longer a distribution in energy or an average in energy but is the total flux of particles in the energy interval ΔE_g. Therefore, energy integrals can be replaced with the simple sums. The definitions of multigroup cross sections, for example, those in Eq. (47), are formal definitions since they require knowing the particle flux $\Phi(\vec{r},E,\vec{\Omega})$ before they can be determined. Since the particle flux energy distribution is unknown, suitable methods for employing weighting functions that accurately approximate the spectral dependence of the flux are required. Successful application of the multigroup method for energy discretization in transport computer codes depends on accurate determination of the requisite multigroup cross sections, and a great deal of effort has been devoted to this area.

We observe that, numerically, by use of the multigroup treatment the original transport equation has been reduced to a set of G equations coupled through the source term. This suggests a conventional iterative method of solution; that is, one solves Eq. (45) a group at a time, assuming that the source is known. The source is then updated, and the process is repeated. We can represent this procedure as follows:

$$L_g \phi_g^{k+1} = \sum_{g'=1}^{G} \Sigma_{g' \to g} (\vec{r}, \vec{\Omega} \cdot \vec{\Omega}') \phi_{g'}^{k} (\vec{r}, \vec{\Omega}') + Q_g(\vec{r}, \vec{\Omega}) \quad , \tag{50}$$

where $k = 0, 1, \cdots$ is an iteration index.

Thus, an initial guess, ϕ_g^0, $g = 1, \cdots, G$, is made and the equations solved for ϕ_g^1, $g = 1, \cdots, G$. With this new value of ϕ, the right side is re-evaluated, allowing a new solution ϕ_g^2 to be computed. This simple procedure is repeated until convergence is obtained.*

To further explain the iteration process, it is convenient to modify Eq. (50) by expanding the transfer term into its fission and scattering components. We will also separate the scattering source term into "upscatter" and "downscatter" portions where upscattering denotes scattering from lower energies to higher energies and conversely for downscattering. We then have

$$\begin{aligned} L_g \phi_g^{k+1}(\vec{r}, \vec{\Omega}) = {} & \chi_g(\vec{r}) \sum_{g'=1}^{G} \int_{4\pi} d\vec{\Omega}' [\nu \Sigma_f(\vec{r})]_{g'} \phi_{g'}^{k}(\vec{r}, \vec{\Omega}) \\ & + \sum_{g'=1}^{g} \int_{4\pi} d\Omega' \, \Sigma_{s, g' \to g} (\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \phi_g^{k+1}(\vec{r}, \vec{\Omega}') \\ & + \sum_{g'=g+1}^{G} \int_{4\pi} d\vec{\Omega}' \, \Sigma_{s, g' \to g}(\vec{r}, \vec{\Omega}' \cdot \vec{\Omega}) \phi_{g'}^{k}(\vec{r}, \vec{\Omega}') \\ & + Q_g(\vec{r}, \vec{\Omega}) \quad . \end{aligned} \tag{51}$$

Thus, the downscatter processes are written at iteration k + 1 and the fission and upscatter processes are written at iteration k because we start the solution of this system of equations at the highest energy and proceed down the groups. In this way, the downscatter source is known and utilized. We have written the fission source in a separable form assuming that the fission spectrum $\chi_g(\vec{r})$ does not depend upon the incident neutron energy. If it does,

---

*We explain the convergence procedure in detail in Sec. III.D.

we would have a matrix $\chi_{gg'}(\vec{r})$, which would then appear under the summation sign. (Most production computer codes do not allow for this, however.) The iterative procedure described by Eq. (51) is known as the "source iteration" method. It is almost universally used in production codes for time-independent calculations. The solution of Eq. (51) beginning with energy group 1 and proceeding successively through energy group G constitutes what is referred to as an "outer" iteration. Note that in the absence of fission and upscatter the source iteration method requires only a single pass, or outer iteration, to effect the exact, fully converged, solution.

To consider the classical k_{eff} eigenvalue problem using Eq. (51), we set the inhomogeneous source, $Q_g(\vec{r},\vec{\Omega})$, to zero and replace $\chi_g(\vec{r})$ with $\chi_g(\vec{r})/k_{eff}^k$, where k_{eff}^k is the estimate of the value of k_{eff} at the k-th iteration. This k_{eff} eigenvalue is most conveniently solved by a "power iteration" technique while the outer iteration procedure is being effected. That is, after the k-th outer iteration, we define a parameter λ^{k+1} by

$$\lambda^{k+1} = \frac{\Sigma_{g'=1} \int d\vec{r} \int_{4\pi} d\vec{\Omega}' [\nu\Sigma_f(\vec{r})]_{g'} \Phi_{g'}^{k+1}(r,\vec{\Omega}')}{\Sigma_{g'=1}^{G} \int d\vec{r} \int_{4\pi} d\vec{\Omega}' [\nu\Sigma_f(\vec{r})]_{g'} \Phi_{g'}^{k}(\vec{r},\vec{\Omega})} \quad , \tag{52}$$

and then

$$k_{eff}^{k+1} = \lambda^{k+1} k_{eff}^k \quad . \tag{53}$$

This procedure has been shown[6] to be unconditionally convergent for reactor analysis-type problems. This means that for any problem

$$\lim_{k\to\infty} \lambda^{k+1} = 1 \quad ,$$

and

$$\lim_{k\to\infty} k_{eff}^k = k_{eff} \quad .$$

In computer code applications, convergence is defined when λ^{k+1} differs from unity by less than some user-defined convergence criterion, say 10^{-5}.

Thus, in summary, the multigroup discretization of the energy variable leads to a natural iteration strategy in the solution computation of the transport equation and puts the burden of accuracy on selection of the multigroup cross sections.

B. Discretization of the Angular Variable

In this section, we briefly describe two discretization techniques for the angular variable: the spherical harmonics method and the discrete ordinates method. Discretization of the angular variable requires care because, in general, it appears both in the discretization of the streaming operator $(\vec{\Omega} \cdot \vec{\nabla})$ and under an angular integral in the source side of the transport equation.

1. Spherical Harmonics Method. To describe the spherical harmonics method[7] for treating the angular variable, we use the multigroup form of the Boltzmann transport equation with the source terms expanded in spherical harmonics, as described in Sec. II.D. Without loss of generality, we use the nonfissioning, inhomogeneous source equation. Using the generalized form of Eqs. (34) and (37) for the scattering and inhomogeneous source terms, the transport equation is written

$$\vec{\Omega} \cdot \vec{\nabla}\Phi_g(\vec{r},\vec{\Omega}) + \Sigma_{t,g}(\vec{r})\Phi_g(\vec{r},\vec{\Omega})$$

$$= \sum_{g'=1}^{G} \sum_{n=1}^{NM} (2\ell + 1)\, \Sigma_{s,g'\to g}^{\ell}(\vec{r}) R_n(\vec{\Omega}) \tilde{\Phi}_{n,g'}(\vec{r}) \qquad (54)$$

$$+ \sum_{n=1}^{NM} (2\ell + 1)\, R_n(\vec{\Omega}) \tilde{Q}_{n,g}(\vec{r}) \quad ,$$

where NM, the number of spherical harmonics used in the expansions, is assumed to be the same for both the scattering and inhomogeneous sources. In Eq. (54), as in Sec. II.D, we have defined the spherical harmonic angular flux moments $\tilde{\Phi}_{n,g}(\vec{r})$ and angular inhomogeneous source moments $\tilde{Q}_{n,g}(\vec{r})$, respectively. Thus, in Eq. (54), the source side of the equation is expressed in terms of spherical harmonic flux and inhomogeneous source moments, whereas the left side angular dependence is as yet unapproximated. The spherical harmonics method consists

of also approximating the angular flux on the left side in terms of spherical harmonics, namely

$$\Phi_g(\vec{r},\vec{\Omega}) = \sum_{n=1}^{NM} (2\ell + 1)R_n(\vec{\Omega})\tilde{\Phi}_{n,g}(\vec{r}) \quad , \tag{55}$$

where, as described in Sec. II.D, NM is the total number of spherical harmonics required for a given expansion order, L, as shown in Table I.

Inserting Eq. (55) into Eq. (54) and collecting terms yields

$$\sum_{n=1}^{NM} (2\ell + 1)R_n(\vec{\Omega})[\vec{\Omega} \cdot \vec{\nabla}\tilde{\Phi}_{n,g}(\vec{r}) + \Sigma_{t,g}(\vec{r})\tilde{\Phi}_{n,g}(\vec{r}) - \sum_{g'=1}^{G} \Sigma_{s,g'\to g}(\vec{r})\tilde{\Phi}_{n,g}(\vec{r}) - \tilde{Q}_{n,g}(\vec{r})] = 0 \quad . \tag{56}$$

To reduce this to a set of NM equations in the flux moments $\tilde{\Phi}_{n,g}(\vec{r})$, we multiply through by $R_m(\vec{\Omega})$ and integrate over all $\vec{\Omega}$. Using the orthogonality of the spherical harmonics $R_m(\vec{\Omega})$, the result is

$$\sum_{n=1}^{NM} [(2\ell + 1) \int_{4\pi} d\vec{\Omega}\, R_n(\vec{\Omega})R_m(\vec{\Omega})\vec{\Omega} \cdot \vec{\nabla}\tilde{\Phi}_{n,g}(\vec{r})] + 4\pi [\Sigma_{t,g}(\vec{r})\tilde{\Phi}_{m,g}(\vec{r}) - \sum_{g'=1}^{G} \Sigma_{s,g'\to g}(\vec{r}) - \tilde{Q}_{n,g}(\vec{r})] = 0 \tag{57}$$

for $1 \leq m \leq NM$. Note that we have been unable to use the orthogonality of the $R_n(\vec{\Omega})$ in the streaming term (first term) of Eq. (57) since the streaming term involves

$$\int_{4\pi} d\vec{\Omega}\, \vec{\Omega}R_n(\vec{\Omega})R_m(\vec{\Omega}) \quad .$$

To write the streaming term in terms of the flux moments, we must invoke identities for $\vec{\Omega}R_n(\vec{\Omega})$ specific to the different geometries of interest. For our purposes, in three dimensions, the streaming term evaluation shows a coupling of seven flux moments, whereas for two dimensions five moments are coupled, and even for ordinary one-dimensional slabs and spheres, three moments are coupled.[7,8] For modern machine architecture, such an extensive coupling leads to impractical computational algorithms for the fluxes.

Therefore, there are few production-oriented codes that employ spherical harmonics as an angular discretization method. In the authors' opinion, the chief value of the spherical harmonics method is its role as a successful method for treating "ray' effects arising from discrete ordinates, as discussed below, and the fact that it is a generalization of the extensively used diffusion approximation.

The diffusion approximation is derived from Eq. (57) by truncating Eq. (55) at L = 1, which we write as

$$\Phi_g(\vec{r},\vec{\Omega}) = \Phi_{0g}(\vec{r}) + 3\vec{\Omega} \cdot \vec{J}_g(\vec{r}) \quad , \tag{58}$$

where the scalar flux,

$$\Phi_{0g}(\vec{r}) = \int_{4\pi} d\vec{\Omega}\ \Phi_g(\vec{r},\vec{\Omega}) \quad , \tag{59}$$

and the current term,

$$\vec{J}_g(\vec{r}) = \int_{4\pi} d\vec{\Omega}\ \vec{\Omega}\Phi_g(\vec{r},\vec{\Omega}) \quad . \tag{60}$$

An evaluation of Eq. (57) with $\Phi_g(\vec{r},\vec{\Omega})$ given by Eq. (58) yields the following set of equations:

$$\nabla \cdot \vec{J}_g(\vec{r}) + \Sigma_{t,g}(\vec{r})\Phi_{0g}(\vec{r}) = \sum_{g'=1}^{G} \Sigma_{s,g'\to g}(\vec{r})\Phi_{0g'}(\vec{r}) + Q_{0g}(\vec{r}) \quad . \tag{61a}$$

$$\frac{1}{3}\nabla\Phi_{0g}(\vec{r}) + \Sigma_{t,g}(\vec{r})\vec{J}_g(\vec{r}) = \sum_{g'=1}^{G}\Sigma_{s,g'\to g}(\vec{r})\vec{J}_{g'}(\vec{r}) + \vec{Q}_{1g}(\vec{r}) \quad . \tag{61b}$$

To proceed, we make further approximation. That is,

$$\Sigma^1_{s,g'\to g} \quad \Sigma^1_{s,g}\,\delta_{g'g} \quad , \tag{62}$$

where

$$\Sigma^1_{s,g} = \sum_{g'=1}^{G}\Sigma^1_{s,g\to g'} \quad ,$$

and $\delta_{g'g}$ is the Dirac delta function. With this we can solve Eq. (61b) for $\vec{J}$ to obtain

$$\vec{J}_g(\vec{r}) = -D_g(\vec{r})\nabla\Phi_{0g}(\vec{r}) + \frac{1}{\Sigma_{r,g}(\vec{r})}\vec{Q}_{1g}(\vec{r}) \quad , \tag{63}$$

where

$$D_g(\vec{r}) = \frac{1}{3\Sigma_{tr,g}(\vec{r})} \tag{64}$$

and

$$\Sigma_{tr,g}(\vec{r}) = \Sigma_{t,g}(\vec{r}) - \Sigma^1_{s,g}(\vec{r}) \quad . \tag{65}$$

Combining Eq. (63) with Eq. (61a), we obtain the diffusion equation

$$-\nabla\cdot D_g(\vec{r})\nabla\Phi_{0g}(\vec{r}) + \Sigma_{t,g}\Phi_{0g}(\vec{r}) = \sum_{g'=1}^{G}\Sigma^0_{s,g'\to g}(\vec{r})\Phi_{0g'}(\vec{r}) + Q_{0g}(\vec{r}) - \nabla\left[\frac{1}{\Sigma_{tr,g}(\vec{r})}\vec{Q}_{1g}(\vec{r})\right] \quad . \tag{66}$$

In its usual form for fission problems, the self-scattering term is moved to the left side of the equation and combined with the total cross section to define a "removal" term,

$$\Sigma_{R,g}(\vec{r}) = \Sigma_{t,t}(\vec{r}) - \Sigma_{s,g\to g}(\vec{r}) \quad . \tag{67}$$

In addition, it is usually assumed that $\vec{Q}_{1g}(\vec{r}) = 0$. From Eqs. (58) and (62), one sees the mathematical approximations needed to derive diffusion theory from transport theory. Physically, it can be shown that diffusion theory is an accurate approximation to transport theory when the physical processes are scattering or fission dominated (small capture cross section); when the medium is large ($\Sigma_t L \geq 10$, where L is a characteristic length); and when boundary or interface effects are unimportant. We note that even when all these conditions are not met, the diffusion approximation can still be invaluable in obtaining inexpensive solutions to the transport equation.*

2. Method of Discrete Ordinates. We have noted that except when diffusion theory is accurate, the spherical harmonics discretization of the transport equation is impractical for arriving at an efficient computational algorithm for modern computers. The angular discretization method that is most useful and is incorporated in most production transport codes is that based upon the method of discrete ordinates.[9] In this method, a set of discrete directions for $\vec{\Omega}$ is chosen, and the transport equation is evaluated for these directions by suitable averaging processes. The choice of these ordinates is not arbitrary; it seeks to satisfy the following conditions:

1) physical symmetries are preserved upon discretization;
2) the spherical harmonic moments are well approximated to provide accurate representation for the sources; and
3) derivatives, with respect to the angular coordinates (in curved geometries) resulting from the streaming operator, are simply approximated.

---

*We make additional observations in Sec. IV.D on the usefulness of the diffusion equation and its relationship to transport calculations.

In geometries of more than one dimension, not all of the above conditions can be met exactly with a single selection of a discrete ordinates set. Thus, compromises are made, such as relaxing the complete physical symmetry requirement so more spherical harmonics moments can be accurately calculated or so the angular derivative term remains a simple expression with minimum coupling. These considerations are discussed more thoroughly in Sec. IV.A. For our purposes now, we assume that we have an appropriately chosen discrete set of directions $\vec{\Omega}_m$ with components μ_m, η_m, and ξ_m, in the direction of the unit vector $\vec{\Omega}_m$ as shown in Figs. 2, 3, and 4. Each discrete direction $\vec{\Omega}_m$ can be visualized as a point on the surface of a unit sphere with which a surface area w_m is associated. These w_m are called the weights. The combination of a set of discrete direction cosines together with their respective weights is referred to as a "quadrature" set. Then, by integrating Eq. (54) over $\vec{\Omega}_m$, we obtain the discrete ordinates form:

$$
\begin{aligned}
&[\vec{\Omega} \cdot \vec{\nabla}\Phi_g(\vec{r},\vec{\Omega})]_m + \Sigma_{t,g}(\vec{r})\Phi_{m,g}(\vec{r}) \\
&\quad = \sum_{g'=1}^{G}\sum_{n=1}^{NM}(2\ell+1)\,\Sigma_{s,g'\to g}^{\ell}(\vec{r})R_n(\vec{\Omega}_m)\tilde{\Phi}_{n,g}(\vec{r}) \\
&\quad\quad + \sum_{n=1}^{NM}(2\ell+1)R_n(\vec{\Omega}_m)\tilde{Q}_{n,g}(\vec{r}) \quad , \quad m = 1, 2, \cdots, M \quad .
\end{aligned}
\tag{68}
$$

In Eq. (68) we have used the spherical harmonics expansion forms of the scattering and inhomogeneous sources as described in Sec. II.D with the spherical harmonics evaluated at the discrete direction $\vec{\Omega}_m$. We have also defined the average angular flux $\Phi_{m,g}(\vec{r})$ as

$$
\Phi_{m,g}(\vec{r}) \equiv \frac{1}{w_m}\int_{\vec{\Omega}_m} d\vec{\Omega}\,\Phi_g(\vec{r},\vec{\Omega}) \quad , \tag{69}
$$

with the weight defined as*

$$w_m \equiv \int_{\vec{\Omega}_m} d\vec{\Omega} \quad . \tag{70}$$

In addition, we have used the notation for the streaming term,

$$[\vec{\Omega} \cdot \vec{\nabla}\Phi_g(\vec{r},\vec{\Omega})]_m \equiv \int_{\vec{\Omega}_m} d\vec{\Omega}\, [\vec{\Omega} \cdot \nabla\Phi_g(\vec{r},\vec{\Omega})] \quad . \tag{71}$$

This streaming term has been left in general form since one needs to specify the spatial geometry before the term can be evaluated. We illustrate the considerations necessary to evaluate the streaming operator by considering a specific geometry, (r,z) cylindrical.

For two-dimensional (r,z) cylindrical coordinates, the streaming operator is given by Eq. (19) (repeated below),

$$\vec{\Omega} \cdot \vec{\nabla}\Phi = \frac{\mu}{r}\frac{\partial(r\Phi)}{\partial r} - \frac{1}{r}\frac{\partial(\eta\Phi)}{\partial \omega} + \xi\frac{\partial\Phi}{\partial z} \quad . \tag{19}$$

The independent variables are defined in Sec. II. With this, Eq. (68) becomes, upon multiplying through by r,

$$\mu\left\{\frac{\partial[r\Phi_g(\vec{r},\vec{\Omega})]}{\partial r} - \frac{\partial[\eta\Phi_g(\vec{r},\vec{\Omega})]}{\partial \omega} + \xi r\frac{\partial\Phi_g(\vec{r},\vec{\Omega})}{\partial z}\right\}_m + r\Sigma_{t,g}(\vec{r})\Phi_{m,g}(\vec{r}) = rS_{m,g}(\vec{r}) \quad , \tag{72}$$

where we have expressed the source on the right side as simply $S_{m,g}(\vec{r})$. We now consider, in detail, the discrete ordinates representation of the streaming terms in Eq. (72), recalling Eq. (71).

---

*See Sec. IV.A for a complete discussion on the choice of weights.

The first streaming term is readily evaluated as

$$\int_{\vec{\Omega}_m} d\vec{\Omega}\ \mu \frac{\partial[r\Phi_g(\vec{r},\vec{\Omega})]}{\partial r} \simeq w_m\mu_m \frac{\partial[r\Phi_{m,g}(\vec{r})]}{\partial r} \quad , \tag{73a}$$

and likewise the third term is written as

$$\int_{\vec{\Omega}_m} d\vec{\Omega}\ \xi r \frac{\partial\Phi_g(\vec{r},\vec{\Omega})}{\partial z} \simeq w_m\mu_m \frac{\partial[r\Phi_{m,g}(\vec{r})]}{\partial z} \quad . \tag{73b}$$

Thus, one assumes that for terms involving spatial derivatives, the angular flux can be taken as constant within each angular interval. This assumption is made to minimize the angular coupling between the discrete directions.

The second streaming term represents the angular redistribution term characteristic of curvilinear geometries discussed in Sec. II.B. Since angular coupling is required to properly represent the discrete ordinates form of the angular redistribution term, it is inaccurate to assume that the angular flux is angularly constant for evaluating in this term. Accordingly, we allow $\Phi_g(\vec{r},\vec{\Omega}) = \Phi_g(\vec{r},\xi,\omega)$ to vary linearly over $\vec{\Omega}_m$ and define, with subscripts m+1/2 and m-1/2, the angular flux at the edges of an angular cell. Further, we note from our discussion in Sec. II.B that angular redistribution involves changes only in the angle ω shown in Figs. 3 and 5; it is thus convenient to choose discrete ordinates points on lines of constant ξ_m on the unit sphere so that on any given line, or ξ level, a change in angle involves changing only the angle ω ($0 < \omega < 2\pi$).* With these conventions, then, we write the discrete ordinates form of the angular redistribution term as the general linear form,

$$\int_{\vec{\Omega}_m} d\vec{\Omega} \frac{\partial[\eta\Phi_g(\vec{r},\vec{\Omega})]}{\partial\omega} = \alpha_{m+\frac{1}{2}}\Phi_{m+\frac{1}{2},g}(\vec{r}) - \alpha_{m-\frac{1}{2}}\Phi_{m-\frac{1}{2},g}(\vec{r}) \quad . \tag{73c}$$

---

*This is an example of condition 3) enumerated on page 39.

In Eq. (73c), the $\alpha_{m\pm\frac{1}{2}}$ are, as yet, undefined angular coupling coefficients that are spatially and energy independent. To define these coupling coefficients, we note that the analytic angular redistribution term when integrated over ω, with ξ held constant, yields

$$\int_0^{2\pi} d\omega \frac{\partial[\eta\Phi(\vec{r},\vec{\Omega})]}{\partial\omega} = 0$$

since $\eta = 0$ both when $\omega = 0$ and when $\omega = 2\pi$. By analogy, then, if we "integrate" the discretized form of Eq. (73c) over all points on a given ξ level, we require that

$$\sum_{m=1}^{ML} [\alpha_{m+\frac{1}{2}}\Phi_{m+\frac{1}{2},g}(\vec{r}) - \alpha_{m-\frac{1}{2}}\Phi_{m-\frac{1}{2},g}(\vec{r})]$$

$$= \alpha_{ML+\frac{1}{2}}\Phi_{ML+\frac{1}{2},g}(\vec{r}) - \alpha_{\frac{1}{2}}\Phi_{\frac{1}{2},g}(\vec{r}) = 0 \quad , \tag{74}$$

where ML denotes the number of discrete points on a given ξ level. Equation (74) is satisfied by requiring the first and last α on each ξ level to be zero, that is

$$\alpha_{\frac{1}{2}} = \alpha_{ML+\frac{1}{2}} = 0 \quad . \tag{75}$$

To determine the remaining values of the α coupling coefficients, we observe that in "everywhere constant" particle flux, $\Phi(\vec{r},\vec{\Omega}) = C$ a constant, in which $\vec{\Omega} \cdot \vec{\nabla}\Phi = 0$; so using Eqs. (73a)-(73c),

$$w_m\mu_m C + \alpha_{m+\frac{1}{2}}C - \alpha_{m-\frac{1}{2}}C = 0 \quad ,$$

or

$$\alpha_{m+\frac{1}{2}} = \alpha_{m-\frac{1}{2}} - w_m\mu_m \quad . \tag{76}$$

This recursion relation, together with the starting condition of Eq. (75), completely determines the values of the coupling coefficients on each ξ level

for a given set of discrete ordinate directions. Inserting Eqs. (73a)-(73c) into Eq. (72) yields the cylindrical (r,z) geometry discrete ordinates form

$$w_m\mu_m \frac{\partial[r\Phi_{m,g}(\vec{r})]}{\partial r} + \alpha_{m+\frac{1}{2}}\Phi_{m+\frac{1}{2},g}(\vec{r}) - \alpha_{m-\frac{1}{2}}\Phi_{m-\frac{1}{2},g}(\vec{r})$$

$$+ w_m\xi_m \frac{\partial[r\Phi_{m,g}(\vec{r})]}{\partial z} + w_m r\Sigma_{t,g}(\vec{r})\Phi_{m,g}(\vec{r}) \qquad (77)$$

$$= w_m r S_{m,g}(\vec{r}) \quad .$$

No angular redistribution term occurs in Cartesian (x,y,z) geometry; the angularly discretized form of the multigroup equation is simply

$$w_m\mu_m \frac{\partial\Phi_{m,g}(\vec{r})}{\partial x} + w_m\eta_m \frac{\partial\Phi_{m,g}(\vec{r})}{\partial y} + w_m\xi_m \frac{\partial\Phi_{m,g}(\vec{r})}{\partial z}$$

$$+ w_m\Sigma_{t,g}(\vec{r})\Phi_{m,g}(\vec{r}) = w_m S_{m,g}(\vec{r}) \quad , \qquad (78)$$

where $\vec{r}$ is represented by the Cartesian coordinates x, y, and z.

We can now make several observations about angular discretization by the discrete ordinates method described above.

The procedure for generating a discrete ordinates representation of the streaming operator is to specify the geometry and to work from a conservative form of the streaming operator in that geometry. Conservative, in this context, means that if the streaming operator is integrated over a spatial region, the resulting quantities can be interpreted directly as the net leakage of particles through the surfaces of the region. This conservative form is best because its use greatly enhances accuracy, in an integral sense, of methods based upon it. We present a rather extreme example of this property in the "ray effect" section below.

Next, discrete directions and weights are selected, and the transport operator is evaluated for these discrete directions. With such a selection of discrete directions and weights, the streaming operator in the transport equation is approximated in a manner that yields minimal angular coupling, simply expressed, and is thus amenable to efficient calculation using digital computers.

Another feature normally associated with the basic discrete ordinates method is that the source is expressed in terms of spherical harmonics of the angular flux. The reason is to save computer storage when evaluating the source for neutron transport, which is, in general, angularly dependent. The angular dependence of this source in most reactor applications is due to angular dependence of the scattering interactions. In Sec. II, for isotropic media, this angular dependence is conveniently and accurately represented by a Legendre expansion in the scattering angle as measured in the laboratory coordinate system (the same system that particle transport is measured in). For eigenvalue calculations of reactor systems, this expansion can usually be truncated at L = 0 or L = 1, still giving an accurate representation. For deep penetration or shielding applications, it may be necessary to represent scattering up to L = 5 or L = 7. In Table I, page 26, we present the number of flux moments required to compute the source up to a given order. In Table III, we show for comparison the number of discrete directions in a typical discrete ordinates, or S_N, quadrature. Because of accuracy considerations, the highest Legendre expansion order is one less than the S_N order used, for example, S_8 and P_7. We see that for high-order scattering, the maximum number of moments required approaches the number of angles for the corresponding S_N set. However, in practical applications the scattering order, L, is no larger than 5, or perhaps 7, and for, say, an S_{12} or S_{16} quadrature the number of Legendre moments is considerably less than the number of discrete angles. In the final analysis, the determining factor to using a Legendre expansion of the source is the flexibility of the resultant code. This flexibility allows one to minimize theamount of data needed to compute the angle-dependent source as the user's accuracy requirements dictate.

Thus, having made the choice of using a spherical harmonic representation of the source, we then see the necessity of requirement 2) (page 39) on the choice of the discrete ordinates. That is, once we have picked a truncated spherical harmonics representation of the source, we seek to minimize any additional error by restricting the discrete ordinates set to one that integrates (by quadrature) the spherical harmonics exactly up to the specified order. How this is done is explained in Sec. IV.A.

TABLE III

NUMBER OF TYPICAL DISCRETE ORDINATES (S_N) DIRECTIONS

Spatial	Directions					
	S_2	S_4	S_6	S_8	S_{12}	S_{16}
2-D	4	12	24	40	84	144
3-D	8	24	48	80	168	288

Having chosen discrete ordinates as the method to discretize the angular variable, we offer the following observations upon its accuracy in two (and by inference three) dimensions. In problems dominated by scattering, such as reactor eigenvalue problems, the discrete ordinates method is generally more accurate in integral quantities than is the corresponding spherical harmonics method.[10] These integral quantities include the eigenvalue, region-integrated reaction rates, and leakages. A basic reason for this is that a conservative form of the equation has been used, and the accuracy of the solution to scattering problems depends upon such conservation. A second reason is that discrete ordinates methods are more compatible with the natural boundary conditions of the transport equation than are the spherical harmonics methods.

On the other hand, in low scattering, high absorbing problems or vacuum regions far from the driving source, such as are encountered in many shielding problems, the accuracy of discrete ordinates methods suffers from what are called ray effects.[11] An extreme example can be visualized as a point source radiating into a pure absorber. The exact scalar flux solution of the problem is spherically symmetric. However, a discrete ordinates solution will exhibit maxima and minima corresponding to the presence or absence of ordinates in the directions sampled. In Fig. 6 we show an S_{16}, two-dimensional, (r,z) geometry calculation of the scalar flux at the surface of a purely absorbing sphere that is five mean free paths in radius with a point source at the origin. The figure dramatically shows that ray effects dominate the solution at this distance from the source. Despite the nonphysical peaks in the flux, the average flux over the surface is still fairly accurate (4.59×10^{-5} as compared with 4.34×10^{-5} for the one-dimensional sphere solution shown). These peaks are caused by ray effects; thus, large pointwise errors can be evident in the

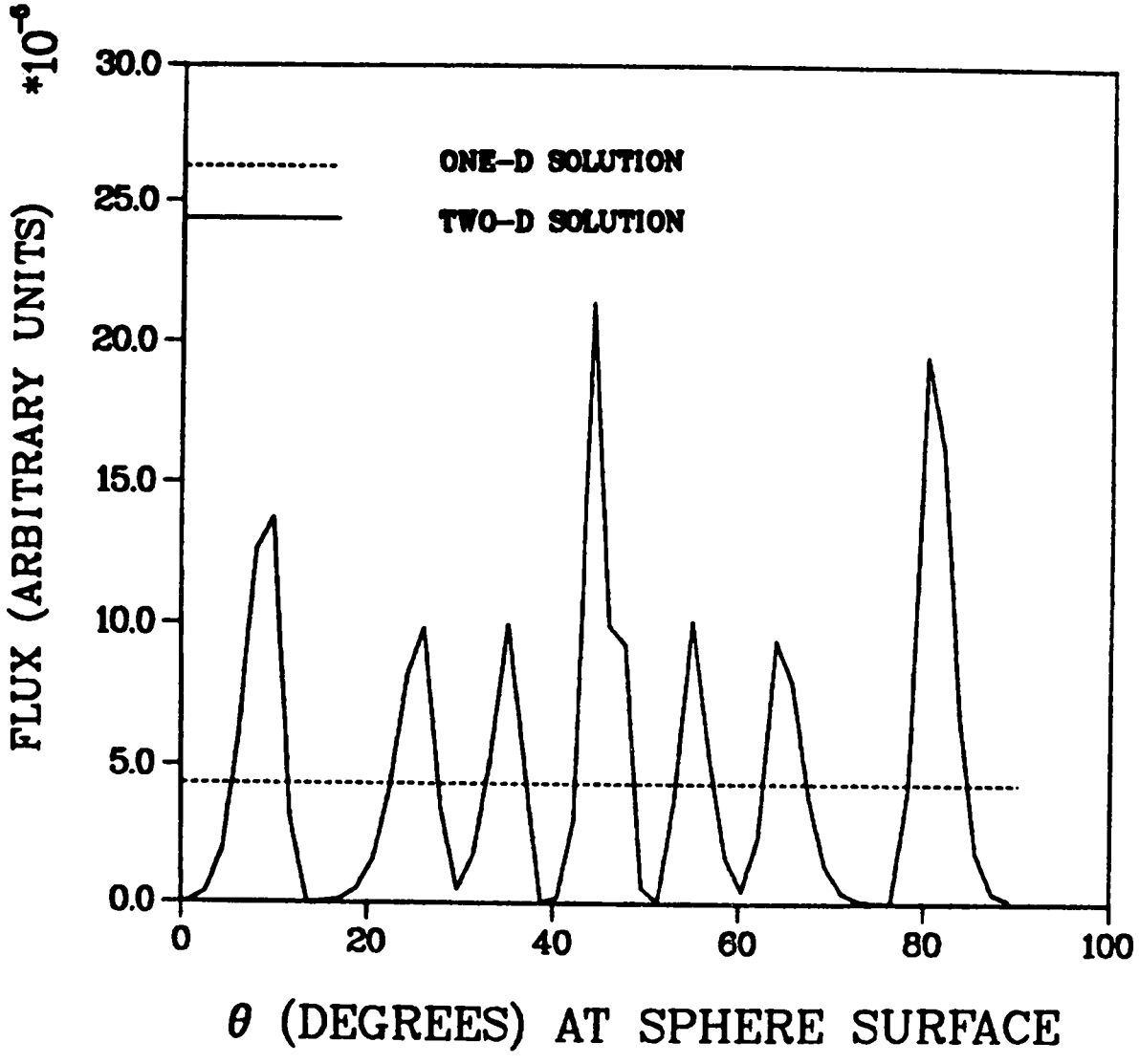


Fig. 6. Flux at the surface of an absorbing sphere (an example of the ray effect).

flux. Use of the conservative form of the transport leakage operator causes the accuracy in the average flux. Anybody interested in knowing accurately the flux pointwise has two alternatives. The first is to add more and more ordinates (go to a higher S_N order), or second, to invoke one of the remedies that mitigate or eliminate the ray effects.[10,11] These latter remedies have been shown to be computationally very expensive if they are effective; they seek to add a source to the discrete ordinates equations that converts the solution to the spherical harmonic solution. The first remedy is straightforward but can be expensive because computing time is directly

proportional to the number of ordinates. A variation on adding more ordinates is to add the ordinates where they are most needed. For shielding problems, this generally means adding ordinates in the direction of particle travel. Such "biased" quadratures are further described in Sec. IV.A. It is generally thought that the ray effect problem is still not satisfactorily solved and awaits further research and development.

C. Spatial Discretization

The final step in discretizing the transport equation involves the spatial variables. For illustrative purposes, we continue our consideration of (r,z) cylindrical geometry. Discrete spatial mesh cells are generated by partitioning the r dimension into IT intervals such that

$$r_{i-\frac{1}{2}} < r < r_{i+\frac{1}{2}} \quad , \quad i = 1, 2, \cdots, IT$$

and the z dimension into JT intervals such that

$$z_{j-\frac{1}{2}} < z < z_{j+\frac{1}{2}} \quad , \quad j = 1, 2, \cdots, JT \quad .$$

The i,j mesh cell, with this nomenclature, is shown in Fig. 7. We also define the width of the i,j cell as

$$\Delta r_i \quad r_{i+\frac{1}{2}} - r_{i-\frac{1}{2}} \quad . \tag{79a}$$

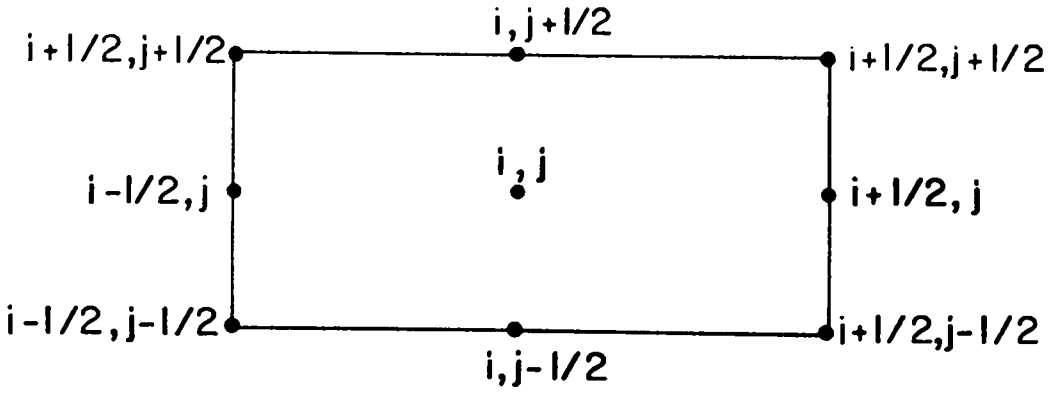


Fig. 7. Typical two-dimensional spatial discretization mesh cell.

and the height, Δz_j, as

$$\Delta z_j \equiv z_{j+\frac{1}{2}} - z_{j-\frac{1}{2}} \quad . \tag{79b}$$

To generate the fully discretized, conservative form of the multigroup, discrete ordinates transport equation, we multiply Eq. (77) by $2\pi drdz$ and integrate over the i,j mesh cell. Considering these operations term by term, we first have

$$\int_{\Delta z_j}\int_{\Delta r_i} 2\pi w_m \mu_m \frac{\partial[r\Phi_{m,g}(r,z)]}{\partial r}\, drdz$$

$$= w_m \mu_m [A_{i+\frac{1}{2},j}\Phi_{i+\frac{1}{2},j,m,g} - A_{i-\frac{1}{2},j}\Phi_{i-\frac{1}{2},j,m,g}] \quad , \tag{80}$$

where we have defined the cell-edge average angular flux

$$\Phi_{i\pm\frac{1}{2},j,m,g} = \frac{1}{\Delta z_j}\int_{\Delta z_j} \Phi_{m,g}(r_{i\pm\frac{1}{2}},z)dz \quad , \tag{81}$$

and the i-direction cell surface area

$$A_{i\pm\frac{1}{2},j} = 2\pi r_{i\pm\frac{1}{2}}\Delta z_j \quad . \tag{82}$$

The angular redistribution term of Eq. (77) becomes, when integrated over the i,j cell,

$$\int_{\Delta z_j}\int_{\Delta r_i} [\alpha_{m+\frac{1}{2}}\Phi_{m+\frac{1}{2},g}(r,z) - \alpha_{m-\frac{1}{2}}\Phi_{m-\frac{1}{2},g}(r,z)]2\pi drdz$$

$$= V_{i,j}\left(\frac{1}{r}\right)_i [\alpha_{m+\frac{1}{2}}\Phi_{i,j,m+\frac{1}{2},g} - \alpha_{m-\frac{1}{2}}\Phi_{i,j,m-\frac{1}{2},g}] \quad . \tag{83}$$

Here we have defined the cell-averaged flux as

$$\Phi_{i,j,m\pm\frac{1}{2},g} = \frac{1}{V_{i,j}}\int_{\Delta z_j}\int_{\Delta r_i} \Phi_{m\pm\frac{1}{2},g}(r,z)2\pi r drdz \quad , \tag{84}$$

and have made the common approximation that

$$\int_{\Delta z_j}\int_{\Delta r_i} \Phi_{m+\frac{1}{2},g}(r,z)2\pi dr dz = \left(\frac{1}{r}\right)_i \Phi_{i,j,m+\frac{1}{2},g} V_{i,j} \quad .$$

Also, the cell volume $V_{i,j}$ is

$$V_{i,j} = \int_{\Delta z_j}\int_{\Delta r_i} 2\pi r dr dz = \pi(r^2_{i+\frac{1}{2}} - r^2_{i-\frac{1}{2}})\Delta z_j \quad . \tag{85}$$

Note that the quantity $(1/r)_i$ in Eq. (83) is, as yet, unspecified. Before it is determined, we consider the last streaming term in Eq. (77), which when integrated over the i,j cell is written

$$\begin{aligned} & w_m \xi_m \int_{\Delta r_i} 2\pi r \int_{\Delta z_j} \frac{\partial \Phi_{m,g}(r,z)}{\partial z} dz dr \\ & \equiv w_m \xi_m \pi[r^2_{i+\frac{1}{2}} - r^2_{i-\frac{1}{2}}][\Phi_{i,j+\frac{1}{2},m,g} - \Phi_{i,j-\frac{1}{2},m,g}] \quad . \end{aligned} \tag{86}$$

In Eq. (86), we have defined the j-direction cell surface average flux

$$\Phi_{i,j\pm\frac{1}{2},m,g} = \frac{\int_{\Delta r_i} \Phi_{m,g}(r,z_{j\pm\frac{1}{2}})2\pi r dr}{\pi\left(r^2_{i+\frac{1}{2}} - r^2_{i-\frac{1}{2}}\right)} \quad , \tag{87}$$

and we note that the j-direction cell surface area is $\pi[r^2_{i+\frac{1}{2}} - r^2_{i-\frac{1}{2}}]$. With the three terms given in Eqs. (80), (83), and (87), the fully discretized streaming terms for the i,j mesh cell are specified. To determine the quantity $(1/r)_i$ in Eq. (83), we form the discretized streaming operator by summing Eqs. (80), (83), and (87) and consider the "everywhere constant" angular flux case for which $\vec{\Omega} \cdot \vec{\nabla}\Phi = 0$. By setting all fluxes equal and using Eq. (76), we get

$$\left(\frac{1}{r}\right)_i = \frac{A_{i+\frac{1}{2},j} - A_{i-\frac{1}{2},j}}{V_{i,j}} \quad . \tag{88}$$

Returning to the last two terms of Eq. (77) and integrating over the i,j cell, we get

$$\int_{\Delta r_i}\int_{\Delta z_j} w_m \Sigma_{t,g} \Phi_{m,g}(r,z) 2\pi r dr dz = w_m \Sigma_{t,i,j,g} \Phi_{i,j,m,g} V_{i,j} \tag{89}$$

and

$$\int_{\Delta r_i}\int_{\Delta z_j} w_m S_{m,g}(\vec{r}) 2\pi r dr dz \equiv w_m S_{i,j,m,g} V_{i,j} \quad , \tag{90}$$

where in Eq. (89) we have assumed that $\Sigma_{t,g}(\vec{r})$ is constant within a given mesh cell; $\Phi_{i,j,m,g}$ and $S_{i,j,m,g}$ are the cell-average angular flux and source, respectively, defined similarly to Eq. (84).

Thus, for (r,z) cylindrical coordinates, the fully discretized multigroup, discrete ordinates transport equation is written using Eqs. (80), (83), (86), (88), and (90). In fact, for the standard geometries under consideration for nuclear reactor analysis, the fully discretized discrete ordinates transport equation can be written in general form as

$$\begin{aligned}
&w_m \mu_m (A_{i+\frac{1}{2}} \Phi_{i+\frac{1}{2},j,k,m,g} - A_{i-\frac{1}{2}} \Phi_{i-\frac{1}{2},j,k,m,g}) \Delta H_{jk} \\
&\quad + (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}) \\
&\quad \times (\alpha_{m+\frac{1}{2}} \Phi_{i,j,k,m+\frac{1}{2},g} - \alpha_{m-\frac{1}{2}} \Phi_{i,j,k,m-\frac{1}{2},g}) \Delta H_{jk} \\
&\quad + w_m \eta_m \Delta B_{ik} (\Phi_{i,j+\frac{1}{2},k,m,g} - \Phi_{i,j-\frac{1}{2},k,m,g}) \\
&\quad + w_m \xi_m \Delta C_{ij} (\Phi_{i,j,k+\frac{1}{2},m,g} - \Phi_{i,j,k-\frac{1}{2},m,g}) \\
&\quad + w_m \Sigma_{t,i,j,k,g} \Phi_{i,j,k,m,g} V_{i,j,k} = w_m S_{i,j,k,m,g} V_{i,j,k} \quad ,
\end{aligned} \tag{91a}$$

$g = 1, \cdots, G$; $m = 1, \cdots, M$; $i = 1, \cdots, I$; $j = 1, \cdots, J$; $k = 1, \cdots, K$; the specific coefficients for each of the normal geometries are given in Table IV.

TABLE IV

AREA AND VOLUME ELEMENTS

Geometry	$A_{i+1/2}$	ΔH_{jk}	ΔB_{ik}	ΔC_{ij}	V_{ijk}
1-D slab	1	1	0	0	ΔX_i
1-D cylinder	$2\pi r_{i+1/2}$	1	0	0	$\pi(r^2_{i+1/2} - r^2_{i-1/2})$
1-D sphere	$4\pi r^2_{i+1/2}$	1	0	0	$4\pi(r^3_{i+1/2} - r^3_{i-1/2})/3$
2-D slab (x,y)	1	ΔY_j	ΔX_i	0	$\Delta X_i \Delta Y_j$
2-D cylinder (r,θ)	$2\pi r_{i+1/2}$	$\Delta\theta_k$	0	Δr_i	$\pi(r^2_{i+1/2} - r^2_{i-1/2})\Delta\theta_k$
2-D cylinder (r,z)	$2\pi r_{i+1/2}$	ΔZ_j	$\pi(r^2_{i+1/2} - r^2_{i-1/2})$	0	$\pi(r^2_{i+1/2} - r^2_{i-1/2})\Delta Z_j$
3-D slab (x,y,z)	1	$\Delta Y_j \Delta Z_k$	$\Delta X_i \Delta Z_k$	$\Delta X_i \Delta Y_j$	$\Delta X_i \Delta Y_j \Delta Z_k$
3-D cylinder (r,θ,z)	$2\pi r_{i+1/2}$	$\Delta\theta_k \Delta Z_j$	$\pi(r^2_{i+1/2} - r^2_{i-1/2})\Delta\theta_k$	$\Delta r_i \Delta Z_j$	$\pi(r^2_{i+1/2} - r^2_{i-1/2})\Delta Z_j \Delta\theta_k$

NOTE: For 2-D cylinders (r,z), η_m and ξ_m should be interchanged in Eq. (91) for consistency with the notation used throughout this chapter.

In writing Eq. (91a), we have started from the conservative form of the transport operator in its discrete ordinates representation [Eq. (77) for r,z geometry]. We have then integrated over a spatial mesh cell and, thus, have derived a spatially discretized form of the operator. Now, if we sum Eq. (91a) over angle, we obtain

$$
\begin{aligned}
&(A_{i+\frac{1}{2}} J_{i+\frac{1}{2},j,k,g} - A_{i-\frac{1}{2}} J_{i-\frac{1}{2},j,k,g}) \Delta H_{j,k} \\
&\quad + \Delta B_{i,k} (I_{i,j+\frac{1}{2},k,g} - I_{i,j-\frac{1}{2},k,g}) \\
&\quad + \Delta C_{i,j} (K_{i,j,k+\frac{1}{2},g} - K_{i,j,k-\frac{1}{2},g}) \\
&\quad + (\Sigma_T V)_{i,j,k} \Phi_{0,i,j,k,g} = S_{0,i,j,k,g} V_{i,j,k} \quad ,
\end{aligned}
\tag{91b}
$$

where

$$J_{i+\frac{1}{2},j,k,g} = \sum_{m=1}^{M} w_m \mu_m \Phi_{i+\frac{1}{2},j,k,m,g} \quad ,$$

$$I_{i,j+\frac{1}{2},k,g} = \sum_{m=1}^{M} w_m \eta_m \Phi_{i,j+\frac{1}{2},k,m,g} \quad ,$$

and

$$K_{i,j,k+\frac{1}{2},g} = \sum_{m=1}^{M} w_m \xi_m \Phi_{i,j,k+\frac{1}{2},m,g} \quad .$$

In this angle-integrated balance equation, the first three terms are interpreted as leakage out of the i, j, k faces, respectively, of the spatial cell. The fourth is the total reaction rate, and last are the sources into the cell. Thus, we consider Eq. (91a) to be a fundamental equation for all conservative forms of spatial differencing of the transport equation, and we will refer to it frequently. As we mentioned in the section on discrete ordinates, satisfying the balance equation has a large impact upon the accuracy of the transport solution for integral quantities such as system leakages, eigenvalues, etc.

We note that Eq. (91a) is a single equation for a mesh cell, but there are more dependent variables (angular fluxes) than one involved; that is, there are more unknowns than equations. For example, in (r,z) cylindrical geometry, Eq. (91) contains a total of seven fluxes, $\Phi_{i,j,m,g}$, $\Phi_{i\pm\frac{1}{2},j,m,g}$, $\Phi_{i,j\pm\frac{1}{2},m,g}$, and $\Phi_{i,j,m\pm\frac{1}{2},g}$. Some of these are known from boundary conditions and the others are established from auxiliary equations. Establishing these auxiliary equations is the distinguishing feature of the spatial differencing methods presented in Sec. IV.B.

Finally, we observe that the design and performance of source iteration acceleration techniques are influenced by the choice of the spatial discretization method. For example, if we choose to discretize the conservative form of the streaming operator, then iteration acceleration by a rebalancing or renormalizing algorithm that seeks to force some integral balance upon the

solution is a natural thing to do. If one chooses a nonconservative algorithm, then a rebalancing method is not natural. Also, the discrete ordinates representation of the streaming operator suggests a natural direction for sweeping (solving) the mesh. Because this operator is first order,* it is essential for algorithm stability to difference (discretize) the operator in the direction of flow. This also leads to a simple, noniterative class of methods for inverting the matrix representation of the streaming operator. Thus, it is very helpful for computational efficiency to minimize the coupling in the spatial mesh so a clear and efficient mesh sweeping algorithm will result. This is rather simple to do for the orthogonal geometries described above.

As a summary, we list the following as desirable attributes for a spatial discretization method for transport problems:

1) The method should be strictly conservative for purposes of accuracy and source iteration acceleration.
2) The method should be accurate as compared to the analytic solution for reasonable size meshes (<3 mfp).
3) The method should be non-negative; positive boundary data and source should yield a non-negative solution for the angular flux in the cell and at its outgoing boundaries.
4) The method should yield the diffusion solution limit $[\Phi(\vec{r},\vec{\Omega}) = \Phi_0(\vec{r}) + 3\vec{\Omega} \cdot J(\vec{r})]$ independent of spatial mesh size when the physical conditions are appropriate.

The last attribute is important to accuracy and to the diffusion synthetic acceleration method, which is described below in the iteration acceleration section. The non-negative requirement has a severe impact upon simple, finite difference (FD) or finite element (FE) types of methods. This requirement seems to be basically incompatible with the requirement of conservation and the diffusion limit; that is, invariably a linear FD or FE method with conservation and the diffusion limit is positive only for restricted mesh sizes.

---

*There is a second-order form of the discrete ordinates operator, the so-called, even-parity form, which is a diffusion-like operator. Methods based upon this form are not in general use in reactor analysis; therefore, they will not be discussed further here. For more information, see Ref. 12.

It may be desirable, because of the above mentioned incompatibility, to relax the conservation condition in favor of strict positivity in a simple algorithm. For reactor analysis, the deep penetration problem involved in calculating neutron and gamma transport through shields may be such a desirable case. What is required mathematically for acceptable accuracy in these methods is a within-group scattering source that is small. In this case, leakage effects dominate the source contribution to the group flux, and conservation is not a strict requirement for accuracy. This becomes more and more the case in shielding problems as the number of energy groups employed in the analysis increases. This has been pointed out by Sasamoto and Takeuchi[13] and incorporated in the PALLAS codes, as discussed below. Thus, the application can dictate the form of spatial discretization that is desirable and needed for a swift, accurate solution.

The diffusion theory limit of requirement 4) (page 54) leads to some restrictions on spatial discretization methods also. For example, in weighted diamond discretization methods in which it is assumed that

$$\Phi_{i,m} = a\Phi_{i+\frac{1}{2},m} + (1 - a)\Phi_{i-\frac{1}{2},m} \quad , \tag{92}$$

for $a \neq 1/2$, the difference equations that arise do not have the diffusion limit.[14] We discuss this and its importance in Sec. IV.B.

D. Source Iteration

In Sec. III.A, we indicated the iteration procedure normally used to solve the multigroup equation because of the group coupling in the source. We term this the outer iteration. In this section, we treat the normal iteration procedure used to solve the discrete ordinates equations because of the angle coupling in the source. We term this the inner iteration.

The considerations involved in iterating the discrete ordinates, within-group source to some convergence criterion is best explained by referring to the multigroup, discrete ordinates equation, written as

$$[\vec{\Omega} \cdot \vec{\nabla}\Phi_g^{k+\frac{1}{2}}(\vec{r})]_m + \Sigma_{t,g}(\vec{r})\Phi_{m,g}^{k+\frac{1}{2}}(\vec{r})$$

$$= \sum_{n=1}^{NM} (2\ell + 1)\, \Sigma_{s,g\to g}^{\ell}(\vec{r}) R_n(\vec{\Omega}_m)\tilde{\Phi}_{n,g}^{k}(\vec{r}) + S_{m,g}(\vec{r}) \quad , \tag{93}$$

where the superscripts k and $k+\frac{1}{2}$ are iteration indexes, and $S_{m,g}(\vec{r})$ is the source-to-group defined as

$$S_{m,g}(\vec{r}) = \frac{\chi_g(\vec{r})}{k_{eff}} \sum_{g'=1}^{G} [\nu\Sigma_f]_{g'}\Phi_{0,g'}^{0}(\vec{r})$$

$$+ \sum_{\substack{g'=1 \\ g'\neq g}}^{G} \sum_{n=1}^{NM} (2\ell + 1)\, \Sigma_{s,g'\to g}^{\ell}(\vec{r}) R_n(\vec{\Omega}_m)\tilde{\Phi}_{n,g'}(\vec{r}) \quad . \tag{94}$$

We have used the spherical harmonics expansion forms for the source, as described in Sec. II.D. Note that we have separated out the within-group (or self-scatter) source from the scattering-from-other-groups source. The scattering-from-other-groups plus fission source (plus any inhomogeneous source, if present) constitutes the "source to group."

We see from Eq. (93) that within each energy group g, the source, in general, depends upon moments of the group g flux. In our remarks on the solution of the discrete ordinates flux on a spatial mesh, we indicated that we strive to make the streaming operator representation on that mesh as simple as possible so the matrix representation of the left side of Eq. (93) is easily inverted without iteration. Because the right side couples all of the angular fluxes together, however, the equation is most easily solved by iteration, which we have indicated by the superscript k. The iteration involves determining a starting value for $\Phi_{m,g}^{k=0}(\vec{r})$, evaluating the source for each angle, m, inverting the left-hand side of Eq. (93) upon the source for each angular direction to obtain $\Phi_{m,g}^{1/2}(\vec{r})$, evaluating a new value of the source at k = 1 and repeating until convergence. A typical convergence criterion is

$$\max_{\vec{r}} \left| [\Phi_{0,g}^{0}(\vec{r})]^{k+1} - [\Phi_{0,g}^{0}(\vec{r})]^{k} \right| \leq \varepsilon_{in} \quad , \tag{95}$$

where ε_{in} is a given convergence criterion and $\Phi^0_{0,g}(\vec{r})$ is the scalar flux for group g.

The procedure outlined above is the inner iteration or within-group iteration. It was pointed out above by means of Eq. (50) that there is also an outer iteration involved in sweeping through all the groups.

Two questions arise in connection with the properties of this inner iteration method: 1) Does it converge, and 2) if it does converge, what is the convergence rate? It is straightforward to show that for reactor-type problems with positive cross sections the iteration procedure is convergent. For a model problem (infinite medium with spatially constant cross sections), the method converges such that the spectral radius of convergence* for group g of the inner iterations is

$$\rho_g \leq c_g \equiv \frac{\Sigma^0_{s,g\to g}}{\Sigma_{t,g}} \quad , \tag{96}$$

where physically $\Sigma^0_{s,g\to g} \leq \Sigma_{t,g}$; c_g is the scattering ratio and, therefore, $0 \leq c_g \leq 1$. Numerical experience for the general, nonmodel problems encountered in reactor analysis also confirms Eq. (96). The number of iterations, p, required for an order of magnitude reduction in the error is related to the spectral radius as follows:

$$p = \frac{-1}{\log\rho} \quad .$$

Therefore, as ρ approaches unity, the number of iterations required approaches infinity; that is, the problem approaches nonconvergence. Because of this, there is good incentive either to choose the group structure so that c_g for each group is far from unity or to find an efficient method of accelerating the convergence. The latter approach is detailed in Sec. IV.C.

A concurrent observation is that the convergence test of Eq. (95) does not mean that the solution is convergent to ε_{in} of the infinitely converged, true solution. In fact, if convergence is slow, ($c_g \simeq 1$), the convergence test

---

*The spectral radius for the convergence of the outer iteration is discussed in Sec. IV.D.

of Eq. (95), even though satisfied, could well be orders of magnitude away from this true convergence. A fairly good rule of thumb is that, given Eq. (95) is satisfied, then

$$\max_{\vec{r}} \left| [\Phi^0_{0,g}(\vec{r})]^{\infty} - [\Phi^0_{0,g}(\vec{r})]^{k} \right| \lesssim \frac{\varepsilon_{in}}{1 - c_g} \quad . \tag{97}$$

This gives further impetus for developing an effective acceleration convergence method because then the "false convergence" as embodied in Eq. (97) when c_g is close to unity will not occur.

In summary, the transport equation is solved by discretizing the independent variables such that the physical content of the equation is maintained while a completely efficient algorithm for machine computation is provided. These considerations have led us to choose a source iteration method based upon discrete ordinates. We see two levels of source iteration: an inner iteration for the within-group scattering source and an outer iteration to converge the fission and upscatter sources and the associated eigenvalue of the fissioning system. The spatial discretization is based upon a particle conservation algorithm for fission (or scattering) dominated problems. Further, the spatial discretization has been combined with the method of discrete ordinates for the angular variable to lead to a non-iterative method for inverting the streaming operator matrix. In the next section, we present some of the details involved in making selections of a discrete ordinates set, a spatial discretization method, and a source iteration-acceleration method.

IV. NUMERICAL DETAILS AND FEATURES

The previous section presented a general description of numerical procedures used in current deterministic transport codes. In this section, we describe details and features related to transport codes using the method of discrete ordinates.

A. Angular Quadrature for Discrete Ordinates Codes

As previously discussed, in the method of discrete ordinates, angular fluxes representing suitable averages are evaluated at discrete directions $\vec{\Omega}_m$ having components μ_m, η_m, and ξ_m, the direction cosines of the unit vector, $\vec{\Omega}_m$.

Consequently, $\mu_m^2 + \eta_m^2 + \xi_m^2 = 1$. Each discrete direction $\vec{\Omega}_m$ can be visualized as a point on the surface of a unit sphere with which a surface area, w_m, is associated. The w_m denote the weights. (The combination of discrete direction cosines and their respective weights is called a quadrature set.) Clearly, the sum of the weights must equal the area of the unit sphere. Choosing M total discrete directions and measuring angular areas in units of 4π,

$$\sum_{m=1}^{M} w_m = 1 \quad . \tag{98}$$

Considerable work has been devoted to developing suitable quadrature sets for discrete ordinates codes.[15-23] Although characterized by the name "discrete ordinates method" and customarily referred to as simply the S_N method, the selection of a quadrature set to be used within the method is somewhat arbitrary. Accordingly, two S_N calculations, identical in all respects except differing quadrature sets, may yield differing results. For most problems, the differences are small, but the user should be aware of the potential for non-negligible differences. Below, we provide some details about quadrature sets and differences in treating the angular variable within the framework of the S_N method.

1. Types of Quadrature Sets. In the development or selection of a quadrature set one must initially consider not only the number and location of discrete points on the unit sphere of directions but also that the set must satisfy certain mechanical integration requirements. For example, in the Legendre expansions of the source terms in the transport equation, one may be required to calculate the spherical-harmonics angular flux moments

$$\Phi_\ell^k(\vec{r},E) = \frac{1}{4\pi} \int_{-1}^{1} d\mu \int_{0}^{2\pi} d\phi \; \Phi(\vec{r},E,\vec{\Omega}) \; Y^*_{\ell,k}(\mu,\phi) \quad ,$$

as defined in Eq. (29).

Using a mechanical quadrature in the discrete ordinates method, this integral is replaced with

$$\Phi_{\ell}^{k} = \sum_{m=1}^{M} w_m \Phi(\vec{r},E,\vec{\Omega}_m)\ Y^{*}_{\ell,k}(\mu_m,\phi_m) \quad ,$$

and one must be assured that the quadrature sum reproduces the integral. Other conditions and requirements exist as constraints on the quadrature set, but for the present it suffices to consider the integration requirements.

It is instructive to consider, in some detail, the generation of the Gauss-Legendre quadrature set over the interval $-1 \leq \mu \leq 1$. Recall that in ordinary plane geometry and in one-dimensional spherical geometry the angular variable, $\vec{\Omega}$, is defined solely by μ since azimuthal (ϕ) invariance is assumed. Thus, in these geometries the discrete ordinates are simply N discrete values of μ in [-1,1]. Most users are familiar with numerical integration schemes (trapezoidal, Simpson's, etc.) in which N-point integration is equivalent to approximating the integrand by a polynomial of degree N-1. The Gauss integration scheme has the extraordinary property of exactly integrating a polynomial of degree 2N-1 using only N points, called quadrature points. If the interval of integration is [-1,1], the quadrature becomes the Gauss-Legendre, or P_N, quadrature, one frequently encountered in discrete ordinates codes. In some references,[15,16] the Gauss-Legendre quadrature is referred to as the P_{N-1} quadrature. The subscript N-1 refers to the fact that N points define a polynomial of degree N-1. In this paper, the Gauss-Legendre quadrature is denoted as the P_N quadrature where N represents the number of quadrature points in the interval [-1,1]. To derive the Gauss-Legendre quadrature, consider an arbitrary polynomial of degree 2N-1, say $g_{2N-1}(\mu)$, in the μ interval [-1,1]. Now suppose another polynomial of degree N-1 exists, say $G_{N-1}(\mu)$, that satisfies the following conditions in $\mu \in [-1,1]$:

(i) $$G_{N-1}(\mu_m) = g_{2N-1}(\mu_m) \quad , \quad m = 1, 2, \cdots, N \quad .$$

(ii) $$\frac{1}{2}\int_{-1}^{1} G_{N-1}(\mu)\, d\mu = \frac{1}{2}\int_{-1}^{1} g_{2N-1}(\mu)\, d\mu \quad .$$

From condition (i), $g_{2N-1}(\mu)$ can be expressed as

$$g_{2n-1}(\mu) = G_{N-1}(\mu) + f_{2N-1}(\mu) \quad , \tag{99}$$

where $f_{2N-1}(\mu)$ is a polynomial of degree 2N-1 that must vanish at each value of μ_m, m = 1, 2, •••, N. Because of this, $f_{2N-1}(\mu)$ can be written in the form

$$f_{2N-1}(\mu) = (\mu - \mu_1)(\mu - \mu_2) \cdots (\mu - \mu_N)F_{N-1}(\mu) \quad , \tag{100}$$

where $F_{N-1}(\mu)$ is a polynomial of degree N-1. Applying condition (ii) to Eqs. (99) and (100) requires that

$$\frac{1}{2}\int_{-1}^{1} (\mu - \mu_1)(\mu - \mu_2) \cdots (\mu - \mu_N)F_{N-1}(\mu)d\mu = 0 \quad . \tag{101}$$

Since $g_{2N-1}(\mu)$ is arbitrary, so is $F_{N-1}(\mu)$, and Eq. (101) requires that each power of μ in $F_{N-1}(\mu)$ must vanish when integrated; that is,

$$\frac{1}{2}\int_{-1}^{1} (\mu - \mu_1)(\mu - \mu_2) \cdots (\mu - \mu_N)\mu^k d\mu = 0 \quad , \qquad k = 0, 1, \cdots, N-1 \quad . \tag{102}$$

Observing that each μ^k has a polynomial of degree N as its coefficient, Eq. (102) states that this coefficient polynomial is orthogonal to polynomials of lower degree over the interval [-1,1]. The Legendre polynomials satisfy this orthogonality property. Thus, if the interpolation or quadrature points μ_m are the zeros of the Legendre polynomials, $P_N(\mu)$, then conditions (i) and (ii) are satisfied exactly. Now, with N quadrature points in [-1,1], any polynomial of degree N-1 can be integrated exactly by mechanical quadrature, that is, N unique coefficients w_m, called quadrature weights, can be determined such that

$$\frac{1}{2}\int_{-1}^{1} G_{N-1}(\mu)d\mu = \sum_{m=1}^{N} w_m G_{N-1}(\mu_m) \quad . \tag{103a}$$

With conditions (i) and (ii), Eq. (103a) can be written in terms of the polynomial of degree 2N-1, namely

$$\frac{1}{2}\int_{-1}^{1} g_{2N-1}(\mu)d\mu = \sum_{m=1}^{N} w_m g_{2N-1}(\mu_m) \quad , \tag{103b}$$

so the polynomial of degree 2N-1 is exactly integrated by an N-point mechanical quadrature known as the Gaussian quadrature. The weights, w_m, which satisfy Eq. (103b), are called the Gauss quadrature weights; these are determined as follows. Let the polynomial $g_{2N-1}(\mu)$ be written

$$g_{2N-1}(\mu) = a_0 + a_1\mu + \cdots + a_{N-1}\mu^{N-1} + \cdots + a_{2N-1}\mu^{2N-1} \quad , \tag{104}$$

where the a_k are arbitrary constants for k = 0, 1, $\cdots$, 2N-1. Substituting Eq. (104) into Eq. (103b), performing the integration, and matching coefficients of like a_k for the first N values of k yields the following set of N simultaneous equations for the weights, w_m:

$$1 = \sum_{m=1}^{N} w_m \quad ,$$

$$0 = \sum_{m=1}^{N} w_m\mu_m \quad ,$$

$$\frac{1}{3} = \sum_{m=1}^{N} w_m\mu_m^2 \quad ,$$

$$\vdots$$

$$\frac{1}{2N}\left[1 - (-1)^N\right] = \sum_{m=1}^{N} w_m\mu_m^{N-1} \quad .$$

For a given N, with μ_m being the zeros of the Legendre polynomial $P_N(\mu)$, the above equations can be used to determine the Gaussian weights, w_m. For example, if N = 2, there are two quadrature points, the zeros of $P_2(\mu)$ (Ref. 23),

$$\mu_1 = -0.577\ 35 \quad ,$$

$$\mu_2 = \ 0.577\ 35 \quad .$$

The weights w_1 and w_2 are then found from the two simultaneous equations

$$1 = w_1 + w_2 \quad ,$$

$$0 = \mu_1 w_1 + \mu_2 w_2 \quad ,$$

whence,

$$w_1 = w_2 = 0.5 \quad .$$

Note that the integration form,

$$\frac{1}{2} \int_{-1}^{1} g(\mu) d\mu \quad ,$$

has been chosen such that the Gauss weights will sum to unity (and not to 2, as in Ref. 24) as is customarily done in S_N codes. Table V lists the Gauss-Legendre quadrature sets for typical N. For ordinary plane geometry and for one-dimensional spheres, the subscript N in S_N denotes the number of quadrature points, N, to be used.

In ordinary plane geometry where the angular variable $\vec{\Omega}$ is described solely by the variable μ, discontinuities in the angular flux for $\mu = 0$ may occur at spatial interfaces. Since the standard Gauss-Legendre, P_N, quadrature assumes continuous polynomial representations over the full range of $\mu \in [-1,1]$, it cannot properly treat the discontinuities. A quadrature set, called the Gauss-double Legendre set, permits treatment of discontinuities in

TABLE V

GAUSS-LEGENDRE INTEGRATION QUADRATURE SETS[24]

N = 2:	$\mu_1 = -\mu_2 = -0.57735$	$w_1 = w_2 = 0.5$
N = 4:	$\mu_1 = -\mu_4 = -0.86114$	$w_1 = w_2 = 0.17393$
	$\mu_2 = -\mu_3 = -0.33998$	$w_2 = w_3 = 0.32607$
N = 6:	$\mu_1 = -\mu_6 = -0.93247$	$w_1 = w_6 = 0.08566$
	$\mu_2 = -\mu_5 = -0.66121$	$w_2 = w_5 = 0.18038$
	$\mu_3 = -\mu_4 = -0.23861$	$w_3 = w_4 = 0.23396$
N = 8:	$\mu_1 = -\mu_8 = -0.96029$	$w_1 = w_8 = 0.05062$
	$\mu_2 = -\mu_7 = -0.79667$	$w_2 = w_7 = 0.11119$
	$\mu_3 = -\mu_6 = -0.52553$	$w_3 = w_6 = 0.15685$
	$\mu_4 = -\mu_5 = -0.18343$	$w_4 = w_5 = 0.18134$
N = 10:	$\mu_1 = -\mu_{10} = -0.97391$	$w_1 = w_{10} = 0.03334$
	$\mu_2 = -\mu_9 = -0.86506$	$w_2 = w_9 = 0.07473$
	$\mu_3 = -\mu_8 = -0.67941$	$w_3 = w_8 = 0.10954$
	$\mu_4 = -\mu_7 = -0.43340$	$w_4 = w_7 = 0.13463$
	$\mu_5 = -\mu_6 = -0.14887$	$w_5 = w_6 = 0.14776$
N = 12:	$\mu_1 = -\mu_{12} = -0.98156$	$w_1 = w_{12} = 0.02359$
	$\mu_2 = -\mu_{11} = -0.90412$	$w_2 = w_{11} = 0.05347$
	$\mu_3 = -\mu_{10} = -0.76990$	$w_3 = w_{10} = 0.08004$
	$\mu_4 = -\mu_9 = -0.58732$	$w_4 = w_9 = 0.10158$
	$\mu_5 = -\mu_8 = -0.36783$	$w_5 = w_8 = 0.11675$
	$\mu_6 = -\mu_7 = -0.12523$	$w_6 = w_7 = 0.12457$

the angular flux at $\mu = 0$. The Gauss-double Legendre, or DP_N, quadrature is based on work by Yvon[25] and involves application of the Gauss-Legendre method to the two half ranges [-1,0] and [0,1] in μ.

Some references[15,16] use the notation $DP_{(N/2)-1}$ for the Gauss-double Legendre quadrature. The subscript (N/2)-1 refers to the fact that the N/2 points in the half interval [-1,0] or [0,1] define a polynomial of degree (N/2)-1 within the half interval. In this paper, the Gauss-double Legendre quadrature is denoted as the DP_N quadrature, where N represents the total number of quadrature points in the full interval [-1,1]. For the DP_N quadrature, the N values of μ, ordered from most negative to most positive, are defined as

$$\mu_m = \frac{1}{2}(\mu'_m - 1) = -\mu_{N-m+1} \quad , \quad m = 1, 2, \cdots, \frac{N}{2} \quad , \tag{105}$$

where the μ' are the zeros of the Legendre polynomials $P_{N/2}(\mu')$ ordered as in Table V with $\mu'_1 < \mu'_2 < \cdots < \mu'_{N/2}$. The DP_N weights, w_m, are related to the corresponding weights in the $P_{N/2}$ quadrature, w'_m, by

$$w_m = \frac{w'_m}{2} = w_{N-m+1} \quad , \quad m = 1, 2, \cdots, \frac{N}{2} \quad . \tag{106}$$

As an example, consider the DP_4 quadrature. From Eq. (105), the DP_4 quadrature points use the Gauss-Legendre quadrature points for N = 2. From Table V, for N = 2, $\mu'_1 = -0.57735$, $\mu'_2 = +0.57735$, so that Eq. (105) gives

$$\mu_1 = \frac{1}{2}(\mu'_1 - 1) = -\mu_4 = -0.78868 \quad ,$$

$$\mu_2 = \frac{1}{2}(\mu'_2 - 1) = -\mu_3 = -0.21132 \quad .$$

Since the weights, w'_m, for the Gauss-Legendre, N = 2, quadrature are $w'_1 = w'_2 = 0.5$, Eq. (106) gives the DP_4 quadrature weights $w_i = 0.25$, $i = 1, \cdots, 4$. Table VI lists DP_N quadrature sets for N = 4, 6, 8, and 12. The DP_N quadrature is generally very good for ordinary plane geometry if the overall thickness is small, that is, for thin slabs. In thin slabs, the correct angular representation of the leakage flux is very important and is accomplished by the DP_N quadrature, which permits a discontinuous polynomial representation on each μ half range. For thick slabs, however, the P_N quadrature is superior to the DP_N quadrature. This is because particles traveling in the most outward direction ($\mu = \pm 1$) are most likely to leak from the right and left faces of the slab, and the P_N quadrature sets have a direction cosine closer to $\mu = \pm 1$ than do the corresponding DP_N sets; therefore, the P_N sets are more accurate. For spheres, the DP_N quadrature set shows no advantage over the P_N quadrature simply because angular flux discontinuities do not appear at spherical interfaces as they do at planar interfaces.

TABLE VI

DOUBLE LEGENDRE, DP_N, QUADRATURE SETS
N = 4, 6, 8, 12

N = 4:	$\mu_1 = -\mu_4 = -0.78868$	$w_1 = w_2 = w_3 = w_4 = 0.25$
	$\mu_2 = -\mu_3 = -0.21132$	
N = 6:	$\mu_1 = -\mu_6 = -0.88730$	$w_1 = w_6 = 0.13889$
	$\mu_2 = -\mu_5 = -0.50000$	$w_2 = w_5 = 0.22222$
	$\mu_3 = -\mu_4 = -0.11270$	$w_3 = w_4 = 0.13889$
N = 8:	$\mu_1 = -\mu_8 = -0.93057$	$w_1 = w_8 = 0.08696$
	$\mu_2 = -\mu_7 = -0.66999$	$w_2 = w_7 = 0.16304$
	$\mu_3 = -\mu_6 = -0.33001$	$w_3 = w_6 = 0.16304$
	$\mu_4 = -\mu_5 = -0.06943$	$w_4 = w_5 = 0.08696$
N = 12:	$\mu_1 = -\mu_{12} = -0.96623$	$w_1 = w_{12} = 0.04283$
	$\mu_2 = -\mu_{11} = -0.83060$	$w_2 = w_{11} = 0.09019$
	$\mu_3 = -\mu_{10} = -0.61931$	$w_3 = w_{10} = 0.11698$
	$\mu_4 = -\mu_9 = -0.38069$	$w_4 = w_9 = 0.11698$
	$\mu_5 = -\mu_8 = -0.16940$	$w_5 = w_8 = 0.09019$
	$\mu_6 = -\mu_7 = -0.03377$	$w_6 = w_7 = 0.04283$

Thus, even in one-dimensional geometries, there is no optimal or "best" quadrature set. This fact is further compounded when geometries other than ordinary planes and one-dimensional spheres are considered; those cases will be discussed next.

In general, discrete directions of particle motion $\vec{\Omega}_m$, are described by the three direction cosines μ_m, η_m, and ξ_m. Only two of these three are independent, of course, since $\mu_m^2 + \eta_m^2 + \xi_m^2 = 1$. The weights associated with the directions $\vec{\Omega}_m$ sum to unity if the surface area on the unit directional sphere is measured in units of 4π, a convention used in most production S_N codes. That is, Eq. (98) must be satisfied. In curved geometries, one or more of the following conditions must also be satisfied to ensure that no particles are lost because of angular distribution:

$$\sum_{m=1}^{M} w_m \mu_m = 0 \quad , \quad \sum_{m=1}^{M} w_m \eta_m = 0 \quad , \quad \sum_{m=1}^{M} w_m \xi_m = 0 \quad . \tag{107}$$

That this is true is seen by integrating (summing) Eq. (76) over all directions to get

$$\alpha_{M+\frac{1}{2}} - \alpha_{\frac{1}{2}} = -\sum_{m=1}^{M} w_m \mu_m = 0 \quad ,$$

since $\alpha_{M+\frac{1}{2}} = \alpha_{\frac{1}{2}} = 0$ from Eq. (75).

A further condition arises from requiring the S_N method to agree with diffusion theory when the latter is applicable, namely when the angular flux is exactly a linear function of the direction cosines μ, η, and ξ. In this circumstance, the angular particle flux at space point $\vec{r}$ and energy E is

$$\Phi(\vec{r},E,\vec{\Omega}) = \frac{1}{4\pi}\left[a(\vec{r},E) + b(\vec{r},E)\mu + c(\vec{r},E)\eta + d(\vec{r},E)\xi\right] \quad . \tag{108}$$

In rectangular Cartesian geometry (x,y,z) coordinate system, for example, the net or scalar particle currents in the x-, y-, and z-directions are, respectively,

$$\begin{aligned} J_x(\vec{r},E) &= \int_0^{2\pi} d\phi \int_{-1}^{1} d\mu\ \mu\Phi(\vec{r},E,\vec{\Omega}) \quad , \\ J_y(\vec{r},E) &= \int_0^{2\pi} d\phi \int_{-1}^{1} d\eta\ \eta\Phi(\vec{r},E,\vec{\Omega}) \quad , \end{aligned} \tag{109}$$

and

$$J_z(\vec{r},E) = \int_0^{2\pi} d\phi \int_{-1}^{1} d\xi\ \xi\Phi(\vec{r},E,\vec{\Omega}) \quad .$$

Using the diffusion condition, Eq. (108), and performing the integrations yield

$$J_x(\vec{r},E) = \frac{1}{3}\, b(\vec{r},E) \quad ,$$

$$J_y(\vec{r},E) = \frac{1}{3}\, c(\vec{r},E) \quad ,$$

and

$$J_z(\vec{r},E) = \frac{1}{3}\, d(\vec{r},E) \quad .$$

In other words, the diffusion condition requires that

$$\frac{1}{4\pi}\int_0^{2\pi}\int_{-1}^{1} \mu^2 d\mu d\phi = \frac{1}{3} \quad , \tag{110}$$

and similarly for η and ξ. Using the discrete ordinates counterpart to Eq. (110) yields the diffusion condition that quadrature sets must satisfy,

$$\sum_{m=1}^{M} w_m \mu_m^2 = \frac{1}{3} \; , \quad \sum_{m=1}^{M} w_m \eta_m^2 = \frac{1}{3} \; , \quad \sum_{m=1}^{M} w_m \xi_m^2 = \frac{1}{3} \quad . \tag{111}$$

Equations (98), (107), and (111) comprise constraints on the selection of quadrature sets. A final, somewhat obvious, constraint on any quadrature set is that all quadrature weights must be non-negative.

Ensuring that physical symmetries are satisfied imposes further constraints on quadrature sets. In other words, the direction mesh embodied in the quadrature set must be made as computationally invariant as possible with respect to the geometric orientation of the problem model. For example, consider a rectangular parallelepiped in Cartesian (x,y,z) geometry with one face of the parallelepiped designated face A. It is desirable that the same computational results be obtained independent of the orientation of parallelepiped relative to the (x,y,z) or the (μ,η,ξ) coordinate axes, as shown in Fig. 2. In other words, the angular flux at, say, a point on face A should not depend on whether face A is oriented normal to and intersected by the positive x-axis (thus, the positive μ-axis) or is oriented normal to and intersected by the negative x-axis (thus, the negative μ-axis). This requires that a positive μ_m must be the same in magnitude as the corresponding negative μ_m; that is, μ_m must be antisymmetric relative to $\mu = 0$. Similarly, the η_m and ξ_m must be antisymmetric relative to their origins. In addition to invariance with respect to this 180° geometric rotation, there should be invariance with respect to any 90° geometric rotation so that for each μ_m there must be an

identical η_m and an identical ξ_m. Accordingly, in three-dimensional Cartesian geometry, quadrature sets should be chosen so that the μ_m, η_m, and ξ_m sets are the same, with each set symmetric about the origin. Such quadrature sets are said to be fully symmetric, and the quadrature points on the surface of the unit directional sphere lie on latitudes or levels. A typical fully symmetric quadrature arrangement is shown in Fig. 8. Note that along a ξ-level, only μ and η change, and, since $\mu^2 + \eta^2 = 1 - \xi^2$, only one variable is independent. Since two independent angular variables are required for all geometries other than ordinary one-dimensional planes and spheres, this arrangement of quadrature points on levels leads to a decided computational advantage since, by sweeping along, say, ξ levels, a two-dimensional quadrature can be programmed simply with a one-dimensional procedure. In one-dimensional ordinary planes and spheres, the quadrature "points," μ_m, represent simply the μ levels (latitudes) of Fig. 8 with the azimuthal integrations along each μ

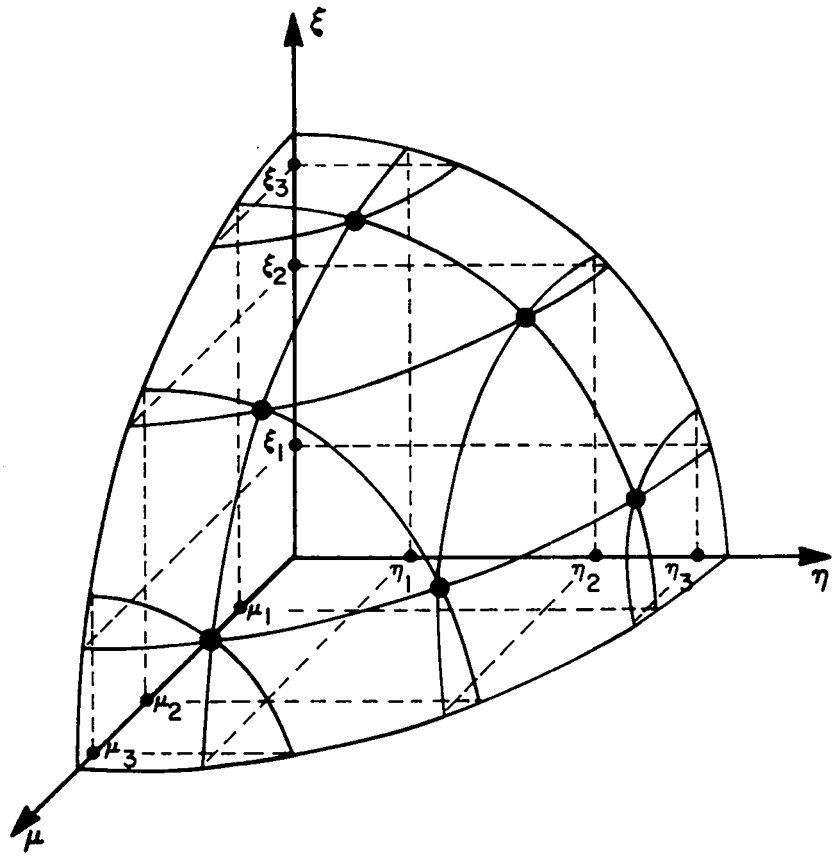


Fig. 8. Fully symmetric S_6 point arrangement.

level performed a priori. For such one-dimensional geometries, there are only N quadrature points (μ levels) on the unit directional sphere, whereas in all other geometries, there are customarily $N(N + 2)$ quadrature points on the unit sphere. This N is the subscript used in the term S_N commonly used to describe the discrete ordinates scheme. It is important to note that, as described here, N denotes a general symmetry-preserving arrangement of quadrature points and not to any specific choice of quadrature values, for although the requirements of Eqs. (98), (107), and (111), together with the condition of full symmetry, are severe constraints, some degrees of freedom remain, and particular additional conditions can be imposed to complete the specification of a desired quadrature set. It should be emphasized here that fully symmetric quadrature sets are not generally required except for full three-dimensional Cartesian geometry. Relaxing of this constraint for other geometries will be discussed below. For the present, however, it is useful to further consider the fully symmetric quadrature arrangement as it is commonly used.

With a fully symmetric S_N quadrature set, there are only N distinct direction cosines (N different levels, or latitudes) even though there are as many as $N(N + 2)$ points on the unit sphere. Further, because of the symmetry requirement, there are only N/2 distinct values of the square of the direction cosines. This is readily seen in Fig. 8 where, for $N = 6$, the values of μ are $\pm\mu_1$, $\pm\mu_2$, and $\pm\mu_3$ and because of the full-symmetry condition, the values of η and ξ are taken from the same set of values as that used for the values of μ. Thus, it is necessary to consider only one octant of the unit sphere (with μ, η, and ξ all positive), as shown in Fig. 8, to fully define the distribution of directions on the full unit sphere. The number of quadrature points per octant is $N(N + 2)/8$. In Fig. 8, the quadrature points are arranged in a triangular pattern over an octant with N/2 different levels on the octant and with $N/2 - i + 1$ quadrature points on the i^{th} level. With the concept of levels, a quadrature point can be assigned three level indices, say, i, j, and k, each with respect to one of the poles of the unit sphere. Then,

$$i + j + k = \frac{N}{2} + 2 \quad , \tag{112}$$

and

$$\mu_i^2 + \mu_j^2 + \mu_k^2 = 1 \quad , \tag{113}$$

where i = 1, 2, •••, N/2 and j = 1, 2, •••, N/2 - i + 1. The correlation between the i and j (hence, the k) subscripts, which represent an ordering of direction cosine levels, and the subscript m, which represents the numbering of points on the unit sphere, is arbitrary and can be made in any desired manner.

For a fully symmetric quadrature set, then, Eqs. (112) and (113) yield the relation

$$\mu_i^2 = \mu_1^2 + \frac{2(i-1)(1-3\mu_1^2)}{N-2} \tag{114}$$

for i = 1, 2, •••, N/2. Equation (114) shows the great constraint that full symmetry places on a quadrature set. The selection of μ_1, which must be taken in the range $0 < \mu_1^2 \leq 1/3$, completely determines the remaining values of μ_i. If μ_1^2 lies close to zero, the cosines tend to be clustered near the ends of the interval [0,1] whereas if μ_1^2 lies close to 1/3, the cosines are clustered near the middle of [0,1]. The freedom of Gaussian quadrature is clearly missing.

Even though for a fully symmetric quadrature set there is only one independent value of the quadrature point direction, μ_1^2, the values of the weights associated with each point must be selected to complete the specification of the quadrature set. The full symmetry condition again places constraints on the number of independent point weights since the weights must also be invariant under geometric rotations. For N = 2, that is, for an S_2 quadrature, there is only one direction and weight on each octant of the unit sphere and all weights are the same to ensure invariance under 90° rotations of the (μ,η,ξ) coordinate systems. For N = 4, the point weights are again all the same. Generally, for $4 \leq N \leq 12$, there are N/2 - 1 independent point weights. For N > 12, the number of independent point weights grows rapidly. With the independent value of μ_1^2 and the (N/2) - 1 independent point weights, there are thus, at most, N/2 free conditions that may be selected to determine the μ_1^2 and the point weights. The diffusion condition of Eq. (111) need not be chosen as one of the conditions, for it can be shown[17] that all fully symmetric quadrature sets satisfy this condition. The conditions most commonly used tend to be various forms of "even-moment" conditions. For example, Table I of Ref. 16 presents an even-moment quadrature set in which

$$\frac{1}{2}\int_{-1}^{1}\mu^{2n}d\mu = \frac{1}{2n+1} = \sum_{m=1}^{M} w_m\mu_m^{2n} \quad , \tag{115}$$

for n = 0, 1, 2, ···, N/2 and M = N(N + 2)/8. Table VII lists this set for N = 2, 4, 6, and 8. Another form of an even-moment quadrature is that used at Oak Ridge National Laboratory,[26] in which μ is selected using the asymptotic prescription of Lee,[17]

TABLE VII

FULLY SYMMETRIC QUADRATURE SETS[16] SATISFYING EVEN-MOMENT CONDITION OF EQ. (115)

	i	μ_i[a]	η_i[a]	w_i[b]
S_2:	1	0.577 350	0.577 350	1.0
S_4:	1	0.350 021	μ_3	0.333 333
	2	μ_1	μ_1	w_1
	3	0.868 890	μ_1	w_1
S_6:	1	0.266 636	μ_6	0.176 126
	2	μ_1	μ_4	0.157 207
	3	μ_1	μ_1	w_1
	4	0.681 508	μ_4	w_2
	5	μ_4	μ_1	w_2
	6	0.926 181	μ_1	w_1
S_8:	1	0.218 218	μ_{10}	0.120 987 7
	2	μ_1	μ_8	0.090 740 7
	3	μ_1	μ_5	w_2
	4	μ_1	μ_1	w_1
	5	0.577 350	μ_8	w_8
	6	μ_5	μ_5	0.092 592 7
	7	μ_5	μ_1	w_2
	8	0.786 796	μ_5	w_2
	9	μ_8	μ_1	w_2
	10	0.951 190	μ_1	w_1

[a] Values provided for principal octant only.

[b] Point weights sum to unity on an octant of the unit sphere.

$$\frac{2}{\pi}\int_0^1 d\mu \int_0^{\pi/2} d\phi \mu^k \eta^\ell = \sum_{m=1}^{M} w_m \mu_m^k \eta_m^\ell \quad , \tag{116}$$

where $\eta(\mu,\phi) = \sqrt{1-\mu^2}\cos\phi$, $M = N(N+2)/8$ and k and ℓ are even integers such that $k > \ell$ and $k + \ell < N$. Table VIII lists this set for $N = 2, 4, 6$, and 8. Numerous other prescriptions exist for selecting μ_1 and the weights for fully symmetric quadrature sets. We note that there is no need to consider satisfying "odd-moment" conditions since these are automatically satisfied by

TABLE VIII

FULLY SYMMETRIC QUADRATURE SETS[26] SATISFYING EVEN-MOMENT CONDITION OF EQ. (116)

	i	μ_i[a]	η_i[a]	w_i[b]
S_2:	1	0.577 350	0.577 350	1.0
S_4:	1	0.333 33	μ_3	0.333 33
	2	μ_1	μ_1	w_1
	3	0.881 92	μ_1	w_1
S_6:	1	0.258 20	μ_6	0.166 67
	2	μ_1	μ_4	w_1
	3	μ_1	μ_1	w_1
	4	0.683 13	μ_4	w_1
	5	μ_4	μ_1	w_1
	6	0.930 95	μ_1	w_1
S_8:	1	0.218 218	μ_{10}	0.120 988
	2	μ_1	μ_8	0.090 741
	3	μ_1	μ_5	w_2
	4	μ_1	μ_1	w_1
	5	0.577 350	μ_8	w_2
	6	μ_5	μ_5	0.0925 92
	7	μ_5	μ_1	w_2
	8	0.786 796	μ_5	w_2
	9	μ_8	μ_1	w_2
	10	0.951 190	μ_1	w_1

[a] Values for principal octant only.

[b] Point weights sum to unity on an octant of the unit sphere.

fully symmetric (hence, odd function) sets. Thus, even with the constraints imposed by full symmetry, degrees of freedom remain, and there appears to be no single best fully symmetric quadrature set. Fortunately, the numerical differences in results obtained from using different quadrature sets of the same S_N order normally tend to be small.

As previously stated, except for three-dimensional Cartesian geometry, fully symmetric quadrature sets are not required, and many of the above constraints can be relaxed to allow additional degrees of freedom in specifying the quadrature sets. Additionally, many geometries do not require treatment of the full unit sphere of directions. As has been described earlier, for example, in one-dimensional spheres and ordinary plane geometry, the quadrature set need be defined only over the μ interval $[-1,1]$, and the quadrature points and weights correspond to μ levels and their weights with no η or ξ levels. Gauss-Legendre or double Legendre quadratures are commonly used for these geometries.

For one-dimensional cylindrical geometry, only two octants of the unit directional sphere need be considered because of the inherent symmetries in η and ξ with such a geometry. In other words, only the ($\mu > 0$, $\eta > 0$, $\xi > 0$) and the ($\mu < 0$, $\eta > 0$, $\xi > 0$) octants need be considered. Full symmetry is not required, and the level point arrangement of Fig. 8 can be relaxed. For example, quadrature points can be arranged on ξ-levels but not on μ- and η-levels. In such geometry, the point arrangement shown in Fig. 9 might be used in place of the fully symmetric arrangement of Fig. 8. For this one-dimensional cylindrical geometry case, then, one requires a quadrature only over the $\eta > 0$, $\xi > 0$ quadrant of the unit sphere; that is, define

$$A = \frac{1}{\pi} \int_0^1 d\xi \int_0^{\pi} d\omega \quad , \tag{117}$$

where ω is the azimuthal angle as shown in Fig. 3 and $\mu = \sqrt{1 - \xi^2}\cos\omega$, $\eta = \sqrt{1 - \xi^2}\sin\omega$. Then

$$A = \frac{1}{\pi} \int_0^1 d\xi \int_{-\sqrt{1-\xi^2}}^{\sqrt{1-\xi^2}} \frac{d\mu}{\sqrt{1 - \xi^2 - \mu^2}} \quad . \tag{118}$$

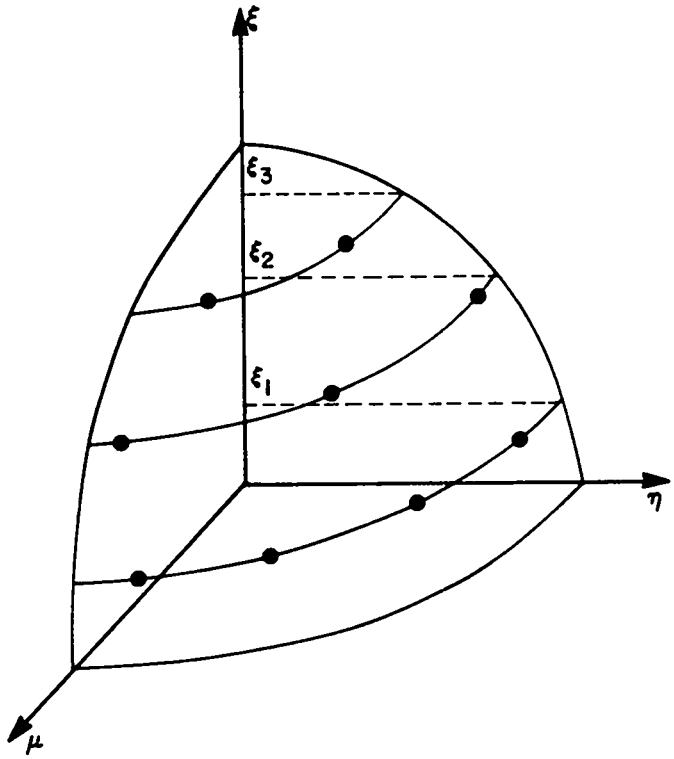


Fig. 9. Nonsymmetric S_6 point arrangement.

Letting $y = \mu/\sqrt{1 - \xi^2}$, Eq. (118) becomes

$$A = \frac{1}{\pi} \int_0^1 d\xi \int_{-1}^{1} \frac{dy}{1 - y^2} \quad , \tag{119}$$

a form that suggests using a Gauss-Chebyshev quadrature for the y, or μ, η, integration on each ξ-level. The Gauss interpolation, or quadrature, points for a given ξ-level are the zeros of the Chebyshev polynomials

$$T_\ell(\cos\omega) \equiv \cos\ell\omega \quad , \tag{120}$$

which satisfy the orthogonality relation

$$\int_{-1}^{1} T_\ell(y)T_k(y)(1 - y^2)^{-\frac{1}{2}}dy = \begin{cases} 0, & \ell \neq k \\ \pi, & \ell = k = 0 \\ \pi/2, & \ell = k \neq 0 \end{cases} \quad , \tag{121}$$

where $y = \cos\omega$. Then, for a T_ℓ quadrature on a given ξ-level, there are ℓ zeros of $\cos\ell\omega$ for $0 \leq \omega \leq \pi$. Denoting these zeros as ω_i, we have

$$\omega_i = \left(\frac{2\ell - 2i + 1}{2\ell}\right)\pi \quad , \quad i = 1, 2, \cdots, \ell \quad .$$

The quadrature points, μ_{ij}, along a given ξ-level for which $\xi = \xi_j$ are

$$\mu_{ij} = \sqrt{1 - \xi_j^2}\,\cos\omega_i \quad , \quad i = 1, 2, \cdots, \ell \quad .$$

The number of μ quadrature points on each ξ-level is arbitrary. For example, the same number of points can be kept on each ξ-level such that for an S_N quadrature, there are N/2 μ values per octant on each ξ-level with, perhaps, N/2 ξ-levels per octant. Alternatively, one could choose a different number of μ-points for each ξ-level - say N/2 points per octant on the first ξ-level, N/2 - 1 points per octant on the second ξ-level, and so forth, with 1 point per octant on ξ-level N/2.

For Chebyshev quadrature, the quadrature weights of all points on a given ξ-level are the same. The location of the ξ-levels is also arbitrary, although Eq. (118) suggests choosing the Gauss-Legendre, P_N, or double Legendre, DP_N, quadrature points for the ξ-levels. If, for example, a P_N quadrature is used, the ξ-values are simply the N/2 P_N quadrature values, and the total weight for all points on a given ξ-level (the level-weight) is simply the Gauss-Legendre weight. With point weights equal on each ξ-level, the point weights on a given ξ-level are simply the level-weight divided by the number of points on that ξ-level. Since for one-dimensional cylinders only one quadrant of the unit sphere need be considered, the point and level weights are commonly normalized to sum to unity over one quadrant. Table IX lists the $P_N(\xi)T_N(\mu)$ quadrature for an equal number of points on each ξ-level for N = 4, 6, and 8. In this quadrature set, $P_N(\xi)$ refers to a Gauss-Legendre quadrature on the N/2 ξ-levels required, and T_N refers to N Gauss-Chebyshev points on each ξ-level. Table X lists the $P_N(\xi)T_N(\mu)$ quadrature for a different number of points on each ξ-level. In these sets, P_N refers to Gauss-Legendre quadrature on the N/2 ξ-levels and T_N refers to N + 2 - 2j Gauss-Chebyshev points on the j^{th} level, j = 1, 2, $\cdots$, N/2.

For two-dimensional (r,z) cylindrical geometry, four octants (one hemisphere) of the unit sphere of directions must be considered, specifically,

TABLE IX

$P_N(\xi)T_N(\mu)$ QUADRATURE SETS - SAME ORDER T_N SET ON EACH ξ-LEVEL[16]

	i	j	$\mu_{i,j}$	$w_{i,j}$[a]	ξ_j
N = 4:	1	2	±0.194 546 4	0.086 963 7	0.861 136 3
	2	2	±0.469 676 5	0.086 963 7	0.861 136 3
	1	1	±0.359 887 9	0.163 036 3	0.339 981 0
	2	1	±0.868 846 1	0.163 036 3	0.339 981 0
N = 6:	1	3	±0.093 498 0	0.028 554 08	0.932 469 5
	2	3	±0.255 441 4	0.028 554 08	0.932 469 5
	3	3	±0.348 939 4	0.028 554 08	0.932 469 5
	1	2	±0.194 166 4	0.060 126 93	0.661 209 4
	2	2	±0.530 472 5	0.060 126 93	0.661 209 4
	3	2	±0.724 638 9	0.060 126 93	0.661 209 4
	1	1	±0.251 342 6	0.077 985 66	0.238 619 2
	2	1	±0.686 680 7	0.077 985 66	0.238 619 2
	3	1	±0.938 023 3	0.077 985 66	0.238 619 2
N = 8:	1	4	±0.054 431 0	0.012 653 57	0.960 289 9
	2	4	±0.155 006 5	0.012 653 57	0.960 289 9
	3	4	±0.231 983 6	0.012 653 57	0.960 289 9
	4	4	±0.273 643 3	0.012 653 57	0.960 289 9
	1	3	±0.117 916 3	0.027 797 63	0.796 666 5
	2	3	±0.335 797 3	0.027 797 63	0.796 666 5
	3	3	±0.502 556 2	0.027 797 63	0.796 666 5
	4	3	±0.592 805 4	0.027 797 63	0.796 666 5
	1	2	±0.165 977 7	0.039 213 33	0.525 532 4
	2	2	±0.472 664 4	0.039 213 33	0.525 532 4
	3	2	±0.707 392 4	0.039 213 33	0.525 532 4
	4	2	±0.834 426 2	0.039 213 33	0.525 532 4
	1	1	±0.191 780 0	0.045 335 47	0.183 434 6
	2	1	±0.546 143 2	0.045 335 47	0.183 434 6
	3	1	±0.817 361 2	0.045 335 47	0.183 434 6
	4	1	±0.964 143 2	0.045 335 47	0.183 434 6

[a]Point weights sum to unity over a quadrant of the unit sphere.

the hemisphere in which $\eta > 0$. Generally, for this geometry quadrature sets are selected from the fully symmetric even-moment sets given in Tables VII or VIII. Alternatively, the $P_N T_N$ sets of Tables IX or X can be used. In any event, the point weights should be normalized to sum to unity over one hemisphere of the unit sphere.

TABLE X

$P_N(\xi)T_N(\mu)$ QUADRATURE SETS - DIFFERENT ORDER T_N SET ON EACH ξ-LEVEL[16]

	i	j	$\mu_{i,j}$	$w_{i,j}$[a]	ξ_j
N = 4:	1	2	±0.359 474 8	0.173 927 4	0.861 136 3
	1	1	±0.359 887 9	0.163 036 3	0.339 981 0
	2	1	±0.868 846 1	0.163 036 3	0.339 981 0
N = 6:	1	3	±0.255 441 4	0.085 662 25	0.932 469 5
	1	2	±0.287 089 6	0.090 190 39	0.661 209 4
	2	2	±0.693 095 7	0.090 190 39	0.661 209 4
	1	1	±0.251 342 6	0.077 985 66	0.238 619 2
	2	1	±0.686 680 7	0.077 985 66	0.238 619 2
	3	1	±0.938 023 3	0.077 985 66	0.238 619 2
N = 8:	1	4	±0.197 285 8	0.050 614 27	0.960 289 9
	1	3	±0.231 301 2	0.055 595 26	0.796 666 5
	2	3	±0.558 410 3	0.055 595 26	0.796 666 5
	1	2	±0.220 196 4	0.052 284 44	0.525 532 4
	2	2	±0.601 587 8	0.052 284 44	0.525 532 4
	3	2	±0.821 784 2	0.052 284 44	0.525 532 4
	1	1	±0.191 780 0	0.045 335 47	0.183 434 6
	2	1	±0.546 143 2	0.045 335 47	0.183 434 6
	3	1	±0.817 361 2	0.045 335 47	0.183 434 6
	4	1	±0.964 143 2	0.045 335 47	0.183 434 6

[a]Point weights sum to unity over a quadrant of the unit sphere.

In two-dimensional (x,y) geometry, the flux is symmetric in ξ, and only the $\xi > 0$ hemisphere of the unit sphere need be considered. Either fully symmetric quadrature sets of $P_N(\xi)T_N(\mu)$ sets are satisfactory for this geometry.

It must be emphasized that there is no optimal quadrature set for all situations. Different geometries and different types of problems for a given geometry lend themselves to differing quadrature types, and for a specific application one quadrature set might be better than another. Generally, however, the even-moment, fully symmetric quadrature sets are used because of their generality and other sets are reserved for special situations in which they are more accurate. A quadrature set that integrates angular moments properly is important if anisotropic scattering is approximated by a spherical harmonics (Legendre polynomial) expansion, so that the polynomials

will be integrated correctly. For example, if the angular flux is isotropic, all the Legendre moments, except the zeroth, must vanish; quadrature sets that properly integrate polynomials guarantee this condition.

No matter which quadrature set is used, problem solutions should be tested for dependence on the order of quadrature. This is not to say that every problem should be calculated several times to test the effects of quadrature sets and/or quadrature order. It often suffices to perform a series of calculations on a problem typical of the class of problems with which the code user is involved. Large reactors with large homogeneous regions are often quite insensitive to quadrature order. On the other hand, small reactors and reactors with local heterogeneties are likely to be quite sensitive to quadrature order. An example of the latter is shown in Table XI, in which the multiplication factor, k_{eff}, is listed as a function of angular quadrature order for both the P_N and DP_N quadrature sets. The problem analyzed for this table is a model of an experimentally critical sphere of 93.71% enriched uranium. The model sphere is uniform and homogeneous with a radius of 8.75 cm and consists of the isotopes U-235 and U-238 with atom densities of 0.045009×10^{24} and 0.003021×10^{24}, respectively. For the analysis, 40 equally spaced mesh intervals were used for spatial discretization. Hansen-Roach[27] 16 energy-group, neutron cross sections were used. The table clearly shows the sensitivity of the calculated k_{eff} to the order of angular quadrature and the lesser sensitivity to type of quadrature. The results indicate that an S_{48} or higher quadrature order is required to achieve a fully converged (with respect to quadrature order) value of k_{eff}. The model problem used for Table XI is a relatively extreme case with regard to its sensitivity to quadrature order. It is a small, high-leakage system (57.05% of the neutrons produced leak from the system) with the angular flux strongly peaked in the outward-flowing directions. This high degree of angular variation - that is, anisotropy - in the angular flux requires a high-order quadrature. Fortunately, most problems do not display this level of sensitivity to quadrature order, and a low-order quadrature, perhaps N = 4, 6, or 8, is commonly satisfactory. Even for the sample problem used for Table XI, a low-order quadrature can be used reliably for parametric studies involving predicted differences caused by changes in the problem. For example, the spherical model problem of Table XI was used to perform calculations to predict the change in k_{eff} that would result if the

TABLE XI

k_{eff} AS A FUNCTION OF ANGULAR QUADRATURE ORDER FOR AN ENRICHED URANIUM SPHERE

Quadrature Type	Angular Quadrature Order, N				
	4	8	16	32	48
P_N	1.006 50	0.999 93	0.998 06	0.997 55	0.997 45
DP_N	1.006 27	0.999 22	0.997 86	0.997 50	0.997 42

uranium enrichment was reduced from 93.71% to 91.71% with no other change in the problem specifications. With Δk defined as k_{eff} (93.71%) - k_{eff} (91.71%), all calculations gave a $\Delta k = 0.0104$, independent of S_N order. That is, even an S_4 calculation, in which the individual values of k_{eff} were nearly 1% in error relative to the corresponding S_{48} values of k_{eff} gave virtually the same value of Δk as did all other S_N orders.

In any event, a person performing discrete ordinates calculations must be aware of and have a feeling for the effects of quadrature order on calculational results.

2. Specialized Quadrature Sets for Specific Applications. In many applications, the space-energy dependent angular flux is anisotropic over a reasonably small portion of the total phase space, and frequently the qualitative nature of this anisotropy is known beforehand. The applications of this foreknowledge can often be used to tailor angular quadrature sets to be most accurate in the phase space domain of flux anisotropy and to be less precise over the remainder of the domain. Without such a use of specialized, or tailored, quadrature sets, one is faced with using a detailed and precise quadrature over the entire phase space. The latter procedure, of course, leads to a much greater computational effort, much of which may be unnecessary.

Since the phase space domain ordinarily consists of the energy (group), spatial, and angular variables (neglecting the time variable), procedures have been developed for tailoring quadratures with respect to each of these variables.

a. Energy Group-Dependent Quadrature Sets. The first and easiest-to-use manner in which quadrature sets can be tailored to specific problems involves

the use of energy group-dependent quadrature. Group-dependent quadrature is appropriate when the angular flux is quite anisotropic in some groups and less so in other groups. One example of this is a spherical geometry problem containing a localized monoenergetic source in a medium which is effectively a pure "absorber" for source particles; that is, most of the interactions between source particles and the medium are captures or scatters-to-other-groups. Then the angular flux for source particles will be highly anisotropic in that it will be nonzero only for directions with direction cosines (μ) near unity away from the localized source. Low-energy group angular fluxes, however, are likely to be much less anisotropic since the source for these particles will be because of scattering from other groups and will, therefore, be distributed throughout the medium. Using group-dependent quadrature, a high-order quadrature can be used for the anisotropic groups, with a lower-order quadrature for the more isotropic groups.

The energy group-dependent quadrature capability is easily implemented in discrete ordinates computer codes with only a minor increase in computer storage requirements, and several current codes provide this feature.[28,29]

The effective use of group-dependent quadrature requires that the user have foreknowledge of the energy-dependent flux anisotropy for the problem being solved. It is also important that the user have knowledge, based on experience, of which quadrature order is adequate for each energy group. In practice, the group-dependent quadrature feature, with its potential for significant reductions in computation time, is not widely used.

b. Space-Dependent Quadrature Sets. The second manner in which quadrature sets can be tailored is to use different quadratures in different spatial regions of the problem being solved. In some spatial regions, the angular flux may be quite anisotropic, for example, near control rods, and a high order quadrature might be necessary. In other regions, the angular flux may be much less anisotropic, and a low-order quadrature is adequate.

Only a few computer codes contain the space-dependent quadrature capability (most notably the DOT series of codes), for example, DOT-IV.[28] The implementation of a space-dependent quadrature capability requires that the code contain a translation algorithm for coupling the angular fluxes across each boundary between regions of differing quadrature order. This coupling necessarily introduces a certain degree of approximation into the solution. Additionally, a computer run time penalty is incurred with space-dependent

quadrature because of the required coupling at interfaces between regions with different quadrature order. However, the use of space-dependent quadrature can be effective when properly applied. The user is advised to acquire a good working knowledge of the use of a space-dependent quadrature, however, before attempting to use this capability on production calculations.

c. Biased Quadrature Sets. The final form of tailored quadrature sets is the biased (asymmetric) quadrature, which is designed for problems in which the angular flux is known to vary rapidly over a reasonably small range of angular directions. For example, consider the spherical geometry problem with a localized source in an effectively pure-absorbing medium for source-energy particles. At distances far removed from the source, the angular flux for source-energy particles is nonzero only for a small range of directions with direction cosines, μ, near unity. Another example involves the presence of a penetration through an absorbing shield in which the emergent angular flux of particles is highly peaked in the direction corresponding to the streaming path through the penetration.

With biased, asymmetric quadrature sets, the quadrature points can be closely clustered in the directional region where the flux is most rapidly varying and can be more loosely spaced over the remaining directions where the angular flux is varying less. With such an asymmetrical arrangement of quadrature points, the freedom of geometric orientation invariance is lost, and the quadrature set is intimately tied to a specific geometric orientation of the problem.

Biased quadrature sets can be formed in several ways. Perhaps the most consistent way was suggested by Cerbone and Lathrop[30] in an analysis of a spherical geometry localized source problem, such as that used as an example above. Since the angular flux is highly forward peaked near $\mu = 1$, that is, in the most outgoing directions, they divided the angular interval $-1 \leq \mu \leq 1$ into two subintervals, $-1 < \mu < 0.95$ and $0.95 \leq \mu \leq 1.0$. Modified directions and weights were obtained from regular Gauss-Legendre sets using the relation for Gaussian quadrature on an arbitrary interval (a,b), namely

$$\tilde{\mu}_m = \left(\frac{b - a}{2}\right) \mu_m + \left(\frac{b + a}{2}\right) \tag{122}$$

and

$$\tilde{w}_m = \left(\frac{b - a}{2}\right) w_m \quad , \tag{123}$$

where the μ_m and w_m are the Gauss-Legendre, P_N, quadrature points and weights, respectively, on the interval (-1,1) (see Table V, page 64), and the $\tilde{\mu}_m$ and $\tilde{w}_m$ are the corresponding, modified quadrature points and weights on the interval (a,b). We note that Eqs. (122) and (123) are those used to generate the Gauss-double-Legendre, DP_N, quadrature relations of Eqs. (105) and (106) defined previously. Cerbone and Lathrop used a modified Gauss-Legendre P_{10} quadrature over (-1,0.95) and a modified Gauss-Legendre P_6 quadrature over (0.95,1.0). This procedure gives a total of 16 quadrature points over the full (-1,1) interval but with 6 points clustered in the (0.95,1.0) interval. If an ordinary Gauss-Legendre S_{16} quadrature had been used, there would still have been 16 total quadrature points in the interval (-1,1) but only one point in the interval (0.95,1.0). An ordinary Gauss-Legendre S_{64} set would also have given 6 quadrature points in (0.95,1.0) but, of course, would have used a total of 64 points over the full interval. Using their asymmetric S_{16} quadrature set, Cerbone and Lathrop found that they could get comparable accuracy in results as an ordinary S_{48} Gauss-Legendre quadrature but in about one-fourth the computing time.

A different and somewhat less formal procedure has been developed at Oak Ridge National Laboratory (ORNL) for generating biased quadrature sets[26] primarily for use in their DOT series of codes. These biased sets are potentially useful in (x,y) and (r,z) geometries. In the following description of these ORNL "standard" biased quadrature sets, the ORNL notation of Ref. 26 is used in which, for (r,z) cylindrical geometry, the η-direction is measured along the z-axis of the cylinder so that their η- and ξ-axes are interchanged relative to those shown in Fig. 3.

The ORNL biased sets available in their quadrature set library are the 100, 166, and 210 direction downward-biased sets and the 100, 166, and 210 direction upward-biased sets. Downward is used to denote directions in the negative η hemisphere. All of these biased quadrature sets are modifications of what are referred to as half-symmetric sets. In these half-symmetric sets the μ and η quadrature values are chosen as the S_{10} Gauss-Legendre (P_{10}) quadrature points shown in Table V. The ξ value for each μ and η value is then

computed from $\xi^2 = 1 - \mu^2 - \eta^2$. Accordingly, the μ and η values are symmetric with one another, but they are not symmetric with the ξ values.

The ORNL 100-direction biased sets contain 65 directions in the biased η-hemisphere of directions and 35 directions in the unbiased hemisphere. The directions in the unbiased hemisphere are taken from the S_{10} half-symmetric set previously described. Of the 35 directions in the unbiased hemisphere, 30 are actual quadrature directions and 5 are starting directions with zero weight. The directions in the biased hemisphere are also chosen from the S_{10} half-symmetric set with the following modification. The η-level for which $|\eta|$ is the maximum and which contains three directions (two symmetric-in-μ directions and one starting direction) is replaced by 11 new η-levels, each containing three directions (two symmetric-in-μ directions and one starting direction). The 11 replacement levels are simply the 11 η-levels from the one-dimensional S_{96} Gauss-Legendre quadrature for which $|\eta|$ is largest. The procedure by which the μ values and their weights on these replacement levels are selected is not specified in Ref. 26. This form of biasing provides clustering of quadrature points near the $|\eta| = 1$ axis, which corresponds to the z-axis in (r,z) geometry or the y-axis in (x,y) geometry. It is designed for problems in which the angular flux is most strongly peaked in directions along the axes.

The ORNL 166-direction and 210-direction biased sets are formed similarly and will not be discussed further here.

The ORNL biased quadrature sets have been used with apparent success in shielding applications. Unfortunately, the procedure used for producing these biased sets appears to be somewhat "ad hoc," and it is not at all clear how other biased quadrature sets for two- or three-dimensional applications could be generated. The following offers a consistent procedure that can be used for completely and consistently producing biased quadrature sets. This procedure applies the approach of Cerbone and Lathrop previously described to the Gauss-Legendre/Gauss-Chebyshev, $P_N T_N$, quadrature formulation discussed in Sec. IV.A.1.

Consider, for example, the (r,z) cylindrical problem in which the angular flux is expected to be rapidly varying for directions with ξ near unity (see Fig. 3) and is expected to be less rapidly varying for other directions. This corresponds to a case in which particle streaming in the positive ξ-direction is expected. The angular interval $-1 \leq \xi \leq 1$ can be divided into two subintervals $-1 < \xi < \xi_0$ and $\xi_0 \leq \xi < 1$, where ξ_0 is some value close to unity;

for example, $\xi_0 = 0.95$. More than two subintervals can be selected, if desired. Modified ξ-level directions and ξ-level weights are then obtained from regular Gauss-Legendre sets using the relations of Eqs. (122) and (123) to achieve a suitable clustering of ξ-levels near the $\xi = 1$, or positive z, axis. The distribution of (μ,η) points on each modified (biased) ξ-level can then be chosen from a suitable Gauss-Chebyshev quadrature (see Tables IX and X), with the advantage of having equal point weights on each ξ-level. Thus, all directions and weights are completely and easily determined.

The use of biased quadrature sets can be quite effective in providing accurate descriptions of localized angular flux anisotropics without severe penalties in computational time or computer storage. As stated repeatedly, however, the proper and successful application of biased quadrature sets to transport problems requires an experienced user. Further, of course, any biased quadrature set should be thoroughly validated before being used.

3. Starting Directions in Quadrature Sets. In curvilinear geometries where angular redistribution occurs, it is necessary to invoke an angular differencing scheme. The diamond difference approximation in angle is almost universally used for angular discretization, as discussed in Sec. IV.B. With such a discretization, it is necessary to provide an initializing boundary condition for the angular fluxes on the "outer boundary" of angular direction space. Zero-weighted starting directions are frequently used for this initializing boundary condition. Starting directions are those inward directions for which there is no angular redistribution. In one-dimensional spheres, there is only one such direction, namely the straight-in direction for which $\mu = -1$; that is, angle $\omega = 180°$ in Fig. 4. For this starting direction $(1 - \mu^2)$ is zero, and the angular redistribution term in Eq. (22) vanishes. As a result, the angular flux in the starting direction can be computed directly, as described in the next section. For cylindrical geometries, there is one starting direction for each ξ-level (see Fig. 3) corresponding to directions for which $\eta = 0$; that is, angle $\omega = 180°$, in which case the particles are inwardly directed toward the cylindrical axis. For each of these starting directions, the angular flux can be computed directly.

These special starting directions appear in quadrature sets for computer codes that require them. The starting directions are assigned a quadrature weight of zero so that they do not contribute to any angular integration by means of quadrature. For example, in the preceding discussion on biased

quadrature sets formulated for use in the DOT series of codes, the directions in a particular quadrature set include both normal quadrature directions and zero-weighted starting directions. Quadrature sets for codes that use starting directions therefore contain more data than are in quadrature sets that do not contain starting directions. It is possible to eliminate the need for starting directions by using a different initializing procedure. One example is the use of a step function approximation for the most inward direction on each ξ-level. This procedure has been used in the TWOTRAN code[31] and is described below. Studies have shown that the use of this step-starting procedure results in virtually no loss in computational accuracy when compared with computations in which starting directions are used.[32]

B. Spatial Discretization Methods

In the numerical description section, we presented some general, desirable features of spatial discretization methods for the discrete ordinates equations. In this section, we describe the commonly used spatial discretization methods and compare their attributes with these desirable features. We also cover some methods not yet incorporated into production codes but which have some desirable properties and are likely to be included in production codes in the near future. The methods we have chosen are the diamond and weighted diamond, linear discontinuous, linear nodal, and short characteristics methods. The first three methods are based upon particle conservation; the last is not a conservative method but is strictly positive.

1. Preliminaries. To display the spatial discretization methods, it is necessary to specialize to a definite mesh-cell structure and to a particular discrete ordinates coupling scheme. In Fig. 10, we show the typical mesh-cell structure that will serve as the reference for all the two-dimensional, orthogonal geometry, spatial discretization methods that follow. We also depict an angular direction and the assumed known boundary values on the exterior of the system "seen" by that direction (indicated by heavy lines in Fig. 10). The common assumptions are made for the spatial discretization methods.

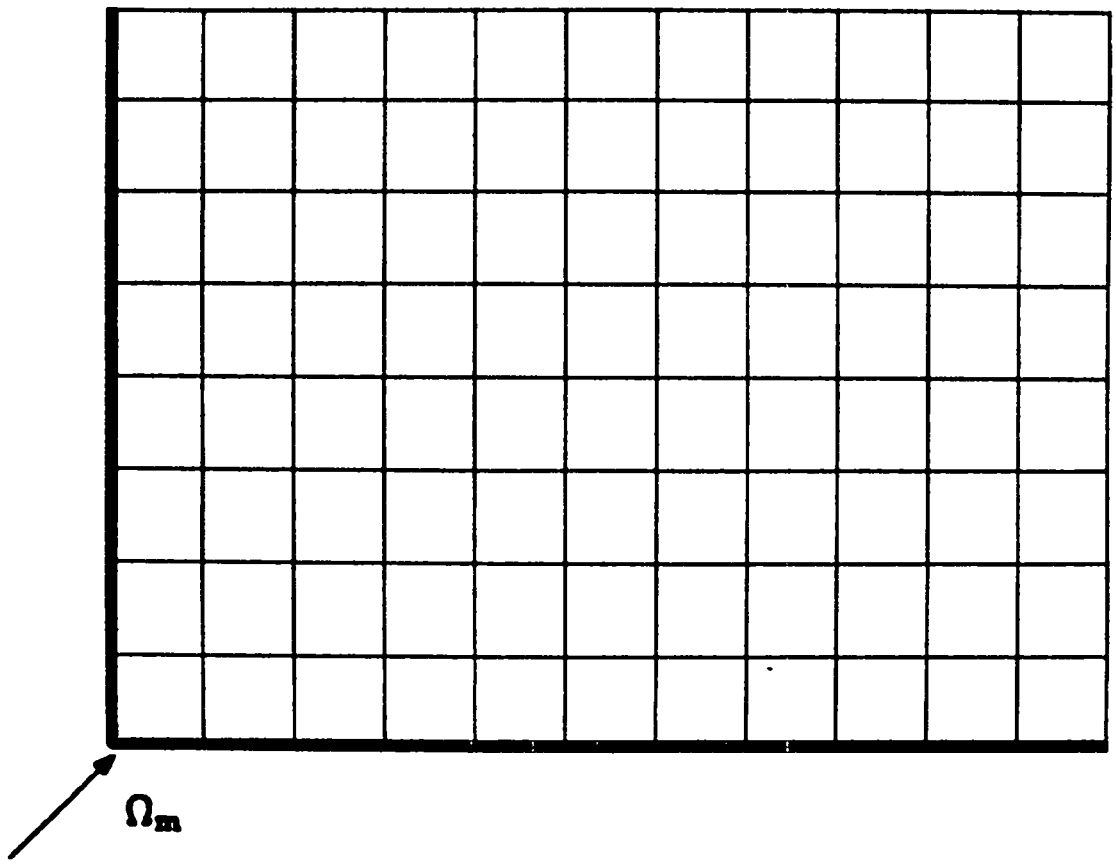


Fig. 10. Typical mesh for spatial discretization.

(i) A regular rectangular grid defines mesh cells.

(ii) Cell-to-cell communication is through cell edges or cell vertices.

(iii) The grid of cells is "swept," starting from the boundary data given on the exterior edges or vertices and marching in a specified order from cell to cell.

(iv) For conservative differencing methods, the basic unknowns are the cell-averaged angular fluxes and the cell edge fluxes.

(v) The angular redistribution term is treated the same in all the conservative methods.

To implement the last point, we take special care in curvilinear geometries to treat the angular directions such that the spatial direction solution methods can be performed directly (noniteratively). The main idea is

to recast the equations and to eliminate one of the unknown fluxes in the angular direction by a suitable approximation relating the cell-averaged flux to the cell-edge flux in the angular direction.

We begin by writing the discretized (r,z) cylindrical geometry equation for energy group g in the form of Eq. (91a):

$$\mu_m(A_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2},j,m} - A_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2},j,m})\Delta z_j + (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})$$

$$\times (\alpha_{m+\frac{1}{2}}\Phi_{i,j,m+\frac{1}{2}} - \alpha_{m-\frac{1}{2}}\Phi_{i,j,m-\frac{1}{2}}) \frac{\Delta z_j}{w_m} \qquad (124)$$

$$+ \xi_m \Delta B_i(\Phi_{i,j+\frac{1}{2},m} - \Phi_{i,j-\frac{1}{2},m}) + (\Sigma_t V)_{i,j}\Phi_{i,j,m} = (S_m V)_{i,j} \quad ,$$

$$i = 1, \cdots, I \quad ; \quad j = 1, \cdots, J \quad ; \quad m = 1, \cdots, M \quad ;$$

where

$$\Delta B_i = \pi(r^2_{i+\frac{1}{2}} - r^2_{i-\frac{1}{2}}) \quad ,$$

and we have dropped the energy group subscript, g, for simplicity. The first approximation, which is common to all the following methods, is to make the diamond approximation in the angular direction; that is, we assume

$$\Phi_{i,j,m} = \frac{1}{2}(\Phi_{i,j,m+\frac{1}{2}} + \Phi_{i,j,m-\frac{1}{2}}) \quad . \qquad (125)$$

The cell-averaged flux is thus related to the angle-edge flux by a simple linear relationship. Combining Eqs. (124) and (125) yields the following equation:

$$\mu_m(A_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2},j,m} - A_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2},j,m})\Delta z_j$$

$$+ (\beta_m - \mu_m)(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Phi_{i,j,m}\Delta z_j$$

$$+ \xi_m \Delta B_i(\Phi_{i,j+\frac{1}{2},m} - \Phi_{i,j-\frac{1}{2},m}) + (\Sigma_t V)_{i,j}\Phi_{i,j,m} \qquad (126)$$

$$= (S_m V)_{i,j} + \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Phi_{i,j,m-\frac{1}{2}}\Delta z_j \quad ,$$

where

$$\beta_m = \frac{\alpha_{m+\frac{1}{2}} + \alpha_{m-\frac{1}{2}}}{w_m} \quad . \tag{127}$$

As can be seen from the form of Eq. (126), the diamond-in-angle approximation leads to a simple set of equations coupled in m. That is, given a starting value $\Phi_{i,j,\frac{1}{2}}$ for all i,j, Eq. (126) is solved, in some approximation, for $\Phi_{i,j,1}$. Then, from Eq. (125), the $\Phi_{i,j,\frac{3}{2}}$ term is evaluated for all i and j, and this process is repeated for all angles. Recall that in two and three dimensions, the angles are on bands or levels on the unit sphere. Thus, there is a "starting direction" at each ξ level for which the flux is assumed known. We then sweep through Eq. (126) for each angle on the ξ level, using Eq. (125) to obtain the requisite angular boundary fluxes for the right side of Eq. (127). This procedure is called the space-angle sweep.

The starting angular fluxes are obtained in one of two ways. In the first, a special starting direction equation is written. To derive this equation, we return to Eq. (19) and note that at $\omega = 180°$, $\eta = 0$ (see Fig. 3); thus, the angular derivative term vanishes for this particular direction that passes through the axis of symmetry of the system. The streaming operator in this case is simply

$$\mu \frac{\partial \psi}{\partial r} + \xi \frac{\partial \psi}{\partial z} \quad , \text{ for } \omega = 180° \quad . \tag{128}$$

This is just the two-dimensional slab streaming operator; hence, the starting direction equation for $\mu_{\frac{1}{2}}$ on each level is written as

$$\begin{aligned} &\mu_{\frac{1}{2}}(\Phi_{i+\frac{1}{2},j,\frac{1}{2}} - \Phi_{i-\frac{1}{2},j,\frac{1}{2}})\Delta z_j \\ &\quad + \xi_{\frac{1}{2}}(\Phi_{i,j+\frac{1}{2},\frac{1}{2}} - \Phi_{i,j-\frac{1}{2},\frac{1}{2}})\Delta r_i \\ &\quad + (\Sigma_t V)_{i,j}\Phi_{i,j,\frac{1}{2}} = (S_{\frac{1}{2}}V)_{i,j} \quad , \text{ for all } i,j \quad . \end{aligned} \tag{129}$$

The second method is called the step-start method. In this case, we assume

$$\Phi_{i,j,-\frac{3}{2}} = \Phi_{i,j,1} \quad , \tag{130}$$

for each starting interval in angle (on each ξ level). Thus, for the first angular interval, the discretized transport equation is

$$\begin{aligned} &\mu_1(A_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2},j,1} - A_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2},j,1})\Delta z_j \\ &\quad + \mu_1(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Phi_{i,j,1}\Delta z_j \\ &\quad + \xi_1 \Delta B_i(\Phi_{i,j+\frac{1}{2},1} - \Phi_{i,j-\frac{1}{2},1}) + (\Sigma_t V)_{i,j}\Phi_{i,j,1} = (S_1 V)_{i,j} \quad , \end{aligned} \tag{131}$$

where we have used $\alpha_{\frac{3}{2}} = -\mu_1 w_1$ from Eqs. (75) and (76).

Equation (131) is solved for $\Phi_{i,j,1}$. Then, $\Phi_{i,j,-\frac{3}{2}}$ is determined from Eq. (130). Equation (126) is then used to solve for the remaining angular fluxes on the ξ level. The advantage of the step-start approach is that fewer equations need to be solved for each angular quadrature set than when starting directions are used. The potential disadvantage is that the step-start is less accurate than using Eq. (129) and diamond-in-angle differencing for the starting directions. Experience has shown, however, that for reactor applications the loss in accuracy is quite small.

2. The Diamond and Weighted Diamond Spatial Differencing Schemes. We now proceed from Eq. (126) with some common methods of spatial differencing. We refer to Fig. 7, where for each cell there are five unknown angular fluxes in two dimensions for a given energy group, $\Phi_{i,j,m}$, $\Phi_{i+\frac{1}{2},j,m}$, $\Phi_{i-\frac{1}{2},j,m}$, $\Phi_{i,j-\frac{1}{2},m}$, and $\Phi_{i,j+\frac{1}{2},m}$. [In curvilinear geometries, the additional unknowns $\Phi_{i,j,m+\frac{1}{2}}$ and $\Phi_{i,j,m-\frac{1}{2}}$ have been eliminated through the use of the diamond-in-angle relationship of Eq. (125) together with the use of special starting relations for m = 1/2.] Two of the five quantities are known from boundary data, for example, $\Phi_{i-\frac{1}{2},j,m}$ and $\Phi_{i,j-\frac{1}{2},m}$ in the case of $\vec{\Omega}_m$ pointing upward and rightward in Fig. 7. Equation (126) provides one equation for the remaining three unknown fluxes. Thus, two more relations are needed. In the weighted diamond approximation, these relations are obtained by specifying a general linear relationship among the cell-edge and cell-averaged quantities:

$$\Phi_{i,j,m} = 0.5(1 + a_{i,j,m})\Phi_{i+\frac{1}{2},j,m} + 0.5(1 - a_{i,j,m})\Phi_{i-\frac{1}{2},j,m} \quad , \tag{132a}$$

and

$$\Phi_{i,j,m} = 0.5(1 + b_{i,j,m})\Phi_{i,j+\frac{1}{2},m} + 0.5(1 - b_{i,j,m})\Phi_{i,j-\frac{1}{2},m} \quad , \tag{132b}$$

where

$$-1 \leq a_{i,j,m} \leq 1 \text{ and } -1 \leq b_{i,j,m} \leq 1.$$

In general, the coefficients a and b in Eqs. (132a) and (132b) are both cell and angle dependent. Two methods, the diamond difference method and the step difference method, are special cases of Eq. (132):

(i) Diamond difference: $0 = a_{i,j,m} = b_{i,j,m}$, (133)

(ii) Step difference: $a_{i,j,m} = \frac{\mu_m}{|\mu_m|}$; $b_{i,j,m} = \frac{\xi_m}{|\xi_m|}$. (134)

With Eqs. (132a) and (132b), we can now solve Eq. (126) for the cell-averaged flux, $\Phi_{i,j,m}$, and the outgoing boundary fluxes. Substituting and rearranging leads to the following expression for $\Phi_{i,j,m}$:

$$\begin{aligned}\Phi_{i,j,m} = \Big[& (S_m V)_{i,j} + \beta_m (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z_j \Phi_{i,j,m-\frac{1}{2}} \\ & + |\mu_m| \frac{(1 - a)A_{i+\frac{1}{2}} + (1 + a)A_{i-\frac{1}{2}}}{1 \pm a} \Delta z_j \Phi_{IN,i} \\ & + |\xi_m| \frac{2\Delta B_i}{1 \pm b} \Phi_{IN,j} \Big] \\ & + \Big[|\mu_m| \frac{(1 - a)A_{i+\frac{1}{2}} + (1 + a)A_{i-\frac{1}{2}}}{1 \pm a} \Delta z_j \\ & + \beta_m (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z_j \\ & + |\xi_m| \frac{2\Delta B_i}{1 \pm b} + (\Sigma_t V)_{i,j} \Big] \quad .\end{aligned} \tag{135}$$

In Eq. (135), we have suppressed the subscripts on $a_{i,j,m}$ and $b_{i,j,m}$; and in the terms with $1 \pm a$, one uses $1 + a$ for $\mu_m > 0$ and $1 - a$ for $\mu_m < 0$. Similarly, one uses $1 + b$ for $\xi_m > 0$ and $1 - b$ for $\xi_m < 0$. Additionally, we have defined $\Phi_{IN,i}$ to be the incoming (known) angular flux $\Phi_{i-\frac{1}{2},j,m}$ for $\mu_m > 0$ and $\Phi_{i+\frac{1}{2},j,m}$ for $\mu_m < 0$; similarly, $\Phi_{IN,j}$ is the incoming (known) angular flux $\Phi_{i,j-\frac{1}{2},m}$ for $\xi_m > 0$ and $\Phi_{i,j+\frac{1}{2},m}$ for $\xi_m < 0$.

Thus, Eq. (135) gives the cell-averaged flux, and the remaining unknown outgoing fluxes are determined from Eqs. (132a) and (132b). These outgoing fluxes are then used as incoming fluxes for the adjacent cells downstream relative to the direction $\vec{\Omega}_m$ so that the entire space-angle mesh can be systematically swept and solved. We now discuss the quality of the solution effected by using the weighted diamond discretization method.

Experience has shown that the weighted diamond solution for mesh cells of dimension $\Delta r, \Delta z$ converges $0(\Delta r, \Delta z)$* in integral quantities as the mesh is refined. The diamond method, however, converges $0(\Delta r^2, \Delta z^2)$ in integral quantities.[33] This latter property, along with its simplicity is why the diamond method is the most frequently employed spatial differencing method in discrete ordinates computer codes. For large meshes ($\Delta r, \Delta z \gtrsim 2$ mean free paths), these methods do not retain positivity, as is discussed next.

Equation (135) gives a positive value for $\Phi_{i,j,m}$ for all values of a and b in the range $[-1,1]$ for positive incoming angular fluxes and for positive sources. However, the outgoing boundary fluxes, as determined from Eqs. (132a) and (132b), are guaranteed positive only for the step method, Eq. (134).** This can be shown by solving Eqs. (132a) and (132b) for the outgoing fluxes and substituting Eq. (135) for the cell average flux. Thus, for $\mu_m < 0$ and $\xi_m < 0$, the outgoing flux $\Phi_{i-\frac{1}{2},j,m}$ is, with many subscripts, omitted for simplicity:

---

*$0(h,k)$ means to order h and k in each of the independent variables respectively.

**The step method is so inaccurate for reasonable mesh sizes, however, that it is virtually never used.

$$\Phi_{i-\frac{1}{2},j} = \Big[2(SV)_{i,j} + \beta(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z_j$$

$$\times [2\Phi_{i,j,m-\frac{1}{2}} - (1 + a)\Phi_{i+\frac{1}{2},j}]$$

$$+ |\mu|[(1 - a)A_{i+\frac{1}{2}} + (1 + a)A_{i-\frac{1}{2}}]\Delta z_j \Phi_{i+\frac{1}{2},j}$$

$$+ |\xi| \frac{2\Delta B_i}{1 - b} [2\Phi_{IN,j} - (1 + a)\Phi_{i+\frac{1}{2},j}] \qquad (136)$$

$$- (1 + a)(\Sigma_t V)_{i,j}\Phi_{i+\frac{1}{2},j}\Big]$$

$$+ (1 - a)\left\{ |\mu|\left[\frac{(1 - a)A_{i+\frac{1}{2}} + (1 + a)A_{i-\frac{1}{2}}}{1 - a}\right]\Delta z_j \right.$$

$$\left. + \beta(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z_j + |\xi| \frac{2\Delta B_i}{1 - b} + (\Sigma_t V)_{i,j} \right\} .$$

This expression is negative for $(\Sigma_t V)$ large or when the mesh is highly skewed so that $\Phi_{i+\frac{1}{2},j} >> 2\Phi_{IN,j}$. Only for the step method ($a = -1$) is $\Phi_{i-\frac{1}{2},j}$ guaranteed positive.

There are two remedies for dealing with negative extrapolations from the weighted diamond method. In the first, we adjust the parameter in the weighted diamond expression ($a_{i,j,m}$ or $b_{i,j,m}$) so that the extrapolation is positive. For example, upon examination of Eq. (136), we can guarantee positivity if the numerator is positive. One of several possible ways to do this is to satisfy the following inequality:

$$(1 + a_{i,j,m}) \leq (S_m V)_{i,j} + \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z_j \Phi_{i,j,m-\frac{1}{2}}$$

$$+ \frac{1}{2} |\mu|(A_{i+\frac{1}{2}} + A_{i-\frac{1}{2}})\Delta z_j \Phi_{i+\frac{1}{2},j,m} + |\xi_m| \frac{2\Delta B_i}{1 - b} \Phi_{IN,j} \qquad (137)$$

$$+ \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z_j + |\xi_m| \frac{2\Delta B_i}{1 - b} + (\Sigma_t V)_{i,j} \Phi_{i+\frac{1}{2},j,m} .$$

Thus, using Eq. (137) with equality to define the weighting parameter in Eq. (136) and in the extrapolation Eq. (132), we obtain a positive outgoing boundary flux for the cell. In a way, this is a fixup method; that is, this is

a nonlinear adjustment of the computational algorithm to assure a positive solution.

A much simpler way to ensure non-negative fluxes is the set-to-zero fixup. In this method, which is used most frequently with the diamond method ($a = b = 0$), when an extrapolated outgoing boundary flux is negative, we set it to zero and re-solve the balance Eq. (126) for the cell-averaged flux. For example, assume that $\Phi_{i-\frac{1}{2},j,m} < 0$, whereas $\Phi_{i,j-\frac{1}{2},m} > 0$ for $\mu_m < 0$ and $\xi_m < 0$. We then set $\Phi_{i-\frac{1}{2},j,m} = 0$, and from Eqs. (126) and (132b),

$$\begin{aligned}\Phi_{i,j,m} = {} & (S_m V)_{i,j} + \beta_m (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}) \Delta z_j \Phi_{i,j,m-\frac{1}{2}} \\ & + |\mu_m| A_{i+\frac{1}{2}} \Delta z_j \Phi_{i+\frac{1}{2},j,m} + |\xi_m| \frac{2\Delta B_i}{1-b} \Phi_{i,j+\frac{1}{2},m} \\ & + \beta_m + |\mu_m| (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}) \Delta z_j \\ & + |\xi_m| \frac{2\Delta B_i}{1-b} + (\Sigma_t V)_{i,j} \quad .\end{aligned} \tag{138}$$

We then recompute $\Phi_{i,j-\frac{1}{2},m}$ using Eqs. (138) and (132b); therefore, balance is preserved. If $\Phi_{i,j-\frac{1}{2},m}$ now happens to be negative, we set it to zero and again solve Eq. (126) for the final value of $\Phi_{i,j,m}$.

If we go back to the desirable attributes of spatial discretization schemes given in Sec. III.C, we see that the motivation for the negative flux fixups is to ensure a positive solution. The cost is that we have sacrificed the diffusion limit because the only weighted diamond method with the diffusion limit is the diamond-differencing method ($a = b = 0$). Thus, any fixup scheme used in the diamond method destroys the capability of the differenced equations to have the diffusion limit.* Also, the diamond method is $O(\Delta r^2, \Delta z^2)$ in integral quantities whereas the weighted diamond is $O(\Delta r, \Delta z)$. This means that as the mesh is refined the diamond method is a great deal more accurate in integral quantities than is the weighted diamond. Therefore, any fixup also

*This may not necessarily be bad because fixup is generally required when the local system is far from the diffusion regime and not required when the system is near the diffusion regions. Thus, the impact of fixup upon the diamond-differencing method is usually small.

worsens this property of diamond differencing. Correspondingly, a method that has the diffusion limit is more accurate for large meshes in problems where diffusion theory is an accurate approximation to the transport problem. As has been mentioned, this occurs in many reactor applications - thus, our emphasis here on the diffusion limit. A more complete explanation of the diffusion limit is given in Ref. 14.

Another problem with the use of fixup is that it can have a destabilizing effect upon iteration convergence. That is, fixup is a nonlinear process dependent upon the local value of the flux, so oscillations may occur in the pointwise value of the flux that will prevent complete convergence.

In summary, the most useful weighted diamond spatial discretization method is the diamond-differencing method. It is simple, accurate for small meshes, and has the diffusion limit, all of which account for its popularity in transport codes designed for nuclear analysis. The need for fixup to ensure positivity is a drawback. To overcome this, an option is to abandon the simple linear differencing scheme of the diamond or weighted diamond method and to use other spatial differencing methods that more closely fulfill the objectives of positivity, accuracy, and having the diffusion limit. The next two methods to be described have improved positivity and higher-order accuracy in integral quantities relative to diamond or weighted diamond methods while retaining the diffusion limit.

3. The Linear Discontinuous Method. In the linear discontinuous (LD) method,[34] a linear function in space is assumed to represent the angular flux within a spatial mesh cell for each direction, but the linear function in one cell is not assumed to be continuous with the linear function used for an adjacent cell at the common boundary between the cells. This is in contrast to the diamond method, which can be considered a linear-continuous method. Using our (r,z) cylindrical geometry model as an example, we represent the angular flux for the LD method in the i,j cell as

$$\Phi_m(r,z) = \Phi_{i,j,m} + \left(\frac{r - r_i}{\Delta r}\right)\Phi_{R,i,j,m} + \left(\frac{z - z_j}{\Delta z}\right)\Phi_{Z,i,j,m} \quad , \qquad (139)$$

$$r_{i-\frac{1}{2}} < r < r_{i+\frac{1}{2}} \quad , \quad z_{j-\frac{1}{2}} < z < z_{i+\frac{1}{2}} \quad ,$$

where

$$\Delta r = r_{i+\frac{1}{2}} - r_{i-\frac{1}{2}} \quad ,$$

$$\Delta z = z_{j+\frac{1}{2}} - z_{j-\frac{1}{2}} \quad ,$$

$$\Phi_{i,j,m} = \text{cell-average flux} \quad ,$$

$$\Phi_{R,i,j,m} = \text{r-moment of the flux} \quad ,$$

$$\Phi_{Z,i,j,m} = \text{z-moment of the flux} \quad ,$$

$$r_i = \frac{r_{i+\frac{1}{2}}^3 - r_{i-\frac{1}{2}}^3}{r_{i+\frac{1}{2}}^2 - r_{i-\frac{1}{2}}^2} \quad ,$$

$$z_i = \frac{1}{2}\left(z_{j+\frac{1}{2}} + z_{j-\frac{1}{2}}\right) \quad .$$

To derive equations that preserve the spatial moments of the flux as defined in Eq. (139), we write the transport equation in which the angular derivative terms have been approximated by diamond differencing as

$$\mu_m \frac{\partial(r\Phi_m)}{\partial r} + (\beta_m - \mu_m)\Phi_m(r,z) + \xi_m r \frac{\partial \Phi_m}{\partial z} + r\Sigma_t(r,z)\Phi_m(r,z) = rS_m(r,z) + \beta_m \Phi_{m-\frac{1}{2}}(r,z) \quad . \tag{140}$$

The angular derivative term in Eq. (140) is similarly represented in cell i,j as

$$\Phi_{m-\frac{1}{2}}(r,z) = \Phi_{m-\frac{1}{2},i,j} + \frac{r - r_i}{\Delta r} \Phi_{R,m-\frac{1}{2},i,j} + \frac{z - z_j}{\Delta z} \Phi_{Z,m-\frac{1}{2},i,j} \quad , \tag{141a}$$

and because of the diamond differencing in angle, one can show that

$$\Phi_{R,i,j,m} = 0.5(\Phi_{R,i,j,m-\frac{1}{2}} + \Phi_{R,i,j,m+\frac{1}{2}}) \tag{141b}$$

and

$$\Phi_{Z,i,j,m} = 0.5(\Phi_{Z,i,j,m-\frac{1}{2}} + \Phi_{Z,i,j,m+\frac{1}{2}}) \quad . \tag{141c}$$

The procedure for deriving the requisite LD equations is to substitute Eqs. (139), (141a), (141b), and (141c) into Eq. (140) and then integrate over the mesh cell using the weighting factors 1, $(r - r_i)/\Delta r$, and $(z - z_j)/\Delta z$, respectively. This yields the following three equations, which we call the spatial moment equations of the transport equation for cell i,j.

$$\begin{aligned}
&\mu_m(A_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2},j,m} - A_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2},j,m})\Delta z \\
&\quad + (\beta_m - \mu_m)(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Phi_{i,j,m}\Delta z \\
&\quad + \xi_m \Delta B_i(\Phi_{i,j+\frac{1}{2},m} - \Phi_{i,j-\frac{1}{2},m}) + (\Sigma_t V)_{i,j}\Phi_{i,j,m} \\
&\quad = (S_m V)_{i,j} + \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Phi_{i,j,m-\frac{1}{2}}\Delta z \quad ,
\end{aligned} \tag{142}$$

$$\begin{aligned}
&\mu_m\left[(r_{i+\frac{1}{2}} - r_i)A_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2},j,m} + (r_i - r_{i-\frac{1}{2}})A_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2},j,m}\right]\Delta z \\
&\quad + (\beta_m - \mu_m)(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\left(\frac{r_i \Delta r \Delta z}{12\bar{r}_i}\right)\Phi_{R,i,j,m} \\
&\quad - (\beta_m - \mu_m)(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\left(\frac{\Delta r \Delta z}{12\bar{r}_i}\right)\Phi_{i,j,m} - \mu_m V_{i,j}\Phi_{i,j,m} \\
&\quad + \xi_m \frac{W_{i,j}}{\Delta z}(\Phi_{R,i,j+\frac{1}{2},m} - \Phi_{R,i,j-\frac{1}{2},m}) + (\Sigma_t W)_{i,j}\Phi_{R,i,j,m} \\
&\quad = (S_{R,m}W)_{i,j} - \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\left(\frac{r_i \Delta r \Delta z}{12\bar{r}_i}\right)\Phi_{R,i,j,m-\frac{1}{2}} \\
&\quad - \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\left(\frac{\Delta r \Delta z}{12\bar{r}_i}\right)\Phi_{i,j,m-\frac{1}{2}} \quad ,
\end{aligned} \tag{143}$$

$$\mu_m(A_{i+\frac{1}{2}}\Phi_{Z,i+\frac{1}{2},j,m} - A_{i-\frac{1}{2}}\Phi_{Z,i-\frac{1}{2},j,m})\Delta z$$

$$+ (\beta_m - \mu_m)(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z\Phi_{Z,i,j,m}$$

$$+ 6\xi_m(\Phi_{i,j+\frac{1}{2},m} - \Phi_{i,j-\frac{1}{2},m} - 2\Phi_{i,j,m})\Delta r\Delta z\Delta B_i \qquad (144)$$

$$+ (\Sigma_t V)_{i,j}\Phi_{Z,i,j,m}$$

$$= (S_{Z,m}V)_{i,j} + \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}})\Delta z\Phi_{Z,i,j,m-\frac{1}{2}} \quad ,$$

where some simplifications have occurred in the angular derivative terms of Eqs. (143) and (144),

$$W_{i,j} = \frac{\Delta B_i \Delta z}{\Delta r}\left[\frac{r^2_{i+\frac{1}{2}} + r^2_{i-\frac{1}{2}}}{2} - r_i^2\right] \quad , \qquad (145)$$

$$\bar{r}_i = \frac{1}{2}(r_{i+\frac{1}{2}} + r_{i-\frac{1}{2}}) \quad , \qquad (146)$$

and $S_{R,m,i,j}$ and $S_{Z,m,i,j}$ are the source spatial moments. Notice that Eq. (142) is the same as Eq. (126), the starting point for the weighted diamond method. This leads to the following interpretation of the linear discontinuous method above. Equations (143) and (144) replace the simple weighted diamond linear extrapolation Eq. (132), with a much more elaborate set, to preserve the first spatial moments of the transport equation.

To proceed with the solution, we must approximate the moments appearing in Eqs. (143) and (144). This is where we make the linear approximation; that is,

$$\Phi_{R,i,j,m} = \begin{cases} 2(\Phi_{i,j,m} - \Phi_{i-\frac{1}{2},j,m}) & \text{for } \mu_m < 0 \\ 2(\Phi_{i+\frac{1}{2},j,m} - \Phi_{i,j,m}) & \text{for } \mu_m > 0 \end{cases} \quad , \qquad (147)$$

$$\Phi_{Z,i,j,m} = \begin{cases} 2(\Phi_{i,j,m} - \Phi_{i,j-\frac{1}{2},m}) & \text{for } \xi_m < 0 \\ 2(\Phi_{i,j+\frac{1}{2},m} - \Phi_{i,j,m}) & \text{for } \xi_m > 0 \end{cases} \quad ; \qquad (148)$$

we further assume

$$\Phi_{R,i,j\pm\frac{1}{2},m} = \Phi_{R,i,j,m} \quad , \tag{149}$$

$$\Phi_{Z,i\pm\frac{1}{2},j,m} = \Phi_{Z,i,j,m} \quad . \tag{150}$$

These assumptions are then substituted into Eqs. (143) and (144) to produce the following for $\mu_m < 0$, $\xi_m < 0$:

$$\begin{aligned}
&- |\mu_m|[r_{i+\frac{1}{2}} - r_i)A_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2},j,m} \\
&\quad + (r_i - r_{i-\frac{1}{2}})A_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2},j,m}]\Delta z \\
&\quad + 2(\beta_m + |\mu_m|)(A_{i+1-\frac{1}{2}} - A_{i-\frac{1}{2}})\left(\frac{r_i \Delta r \Delta z}{12\bar{r}_i}\right) \\
&\quad \times (\Phi_{i,j,m} - \Phi_{i-\frac{1}{2},j,m}) - (\beta_m - \mu_m)(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}) \\
&\quad \times \left(\frac{\Delta r \Delta z}{12\bar{r}_i}\right)\Phi_{i,j,m} - \mu_m V_{i,j}\Phi_{i,j,m} + \xi_m \frac{W_{i,j}}{\Delta z}\Phi_{R,i,j-\frac{1}{2},m} \\
&\quad - 2\xi_m \frac{W_{i,j}}{\Delta z}(\Phi_{i,j,m} - \Phi_{i-\frac{1}{2},j,m}) + 2(\Sigma_t W)_{i,j} \\
&\quad \times (\Phi_{i,j,m} - \Phi_{i-\frac{1}{2},j,m}) = (S_{R,m}W)_{i,j} - \beta_m(A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}) \\
&\qquad\qquad \times \left(\frac{r_i \Delta r \Delta z}{12\bar{r}_i}\right)\Phi_{R,i,j,m-\frac{1}{2}} \quad ,
\end{aligned} \tag{151}$$

$$
\begin{aligned}
&- |\mu_m| A_{i+\frac{1}{2}} \Phi_{Z,i+\frac{1}{2},j,m} \Delta z + 2|\mu_m| A_{i+\frac{1}{2}} (\Phi_{i,j,m} - \Phi_{i-\frac{1}{2},j,m}) \Delta z \\
&+ 2\beta_m (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}) \Delta z (\Phi_{i,j,m} - \Phi_{i,j-\frac{1}{2},m}) \\
&+ 6\xi_m (\Phi_{i,j+\frac{1}{2},m} - \Phi_{i,j-\frac{1}{2},m} - 2\Phi_{i,j,m}) \Delta B_i + 2(\Sigma_t V)_{i,j} \qquad (152) \\
&\times (\Phi_{i,j,m} - \Phi_{i,j-\frac{1}{2},m}) = (S_{Z,m} V)_{i,j} + \beta_m (A_{i+\frac{1}{2}} - A_{i-\frac{1}{2}}) \Delta z \\
&\times \Phi_{Z,i,j,m-\frac{1}{2}} \ .
\end{aligned}
$$

Equations (151) and (152) can be solved for $\Phi_{i-\frac{1}{2},j,m}$ and $\Phi_{i,j-\frac{1}{2},m}$ in terms of $\Phi_{i,j,m}$ and the known incoming boundary data and sources, and substituted into Eq. (142) to obtain the cell-averaged flux, $\Phi_{i,j,m}$. As is readily seen, this procedure involves considerably more computational work and storage than does the weighted diamond approach. What is gained? The method has the diffusion limit, it is more accurate, and it is somewhat more positive.

Thus, we list the advantages of the linear discontinuous method over the diamond method as

a. More accurate for small meshes; $O(\Delta r^3, \Delta z^3)$, in integral quantities;
b. Less negative for large meshes

$$
\Phi_{i-\frac{1}{2},j,m} \to \frac{\mu_m}{\Sigma_t \Delta r} \text{ as } \frac{\Sigma_t \Delta r}{\mu_m} \to \infty \ ,
$$

$$
\Phi_{i,j-\frac{1}{2},m} \to \frac{\xi_m}{\Sigma_t \Delta z} \text{ as } \frac{\Sigma_t \Delta_z}{\xi_m} \to \infty \ ;
$$

c. Retains the diffusion limit.

Its disadvantage is that it is computationally more expensive in computer time (factor of 2 to 3) and storage (factor of 3) than the diamond method for a given mesh. Further, the linear discontinuous method is not fully positive and, hence, in general, requires some sort of fixup. One can devise a fixup, however, that allows the method to be positive with a decrease in accuracy, yet still satisfies the diffusion limit.

All things considered, in the authors' opinion a main overall advantage of the linear discontinuous method over the diamond method is the better stability of the LD method on problems with large meshes. The stability referred to here is in the potential flux spatial oscillations from mesh to mesh; they are much more damped in the LD method than those that have been observed with the diamond method. Therefore, for problems with large mesh ($\geq$ 3 m.f.p.), the LD method is preferred over the diamond method for reactor physics applications; its use with relatively large mesh cells is actually more efficient than that of the diamond method on a mesh small enough to reduce the spatial oscillation error.

4. The Linear Nodal Method. The failure of the linear discontinuous method to be positive arises from the linear representation of the flux and source within a spatial mesh cell. For practical methods, the linear representation of the source [for example, Eq. (139) applied to the source] is probably an essential limitation for all methods; however, it is relatively straightforward to relax the linear representation of the angular flux. A variety of methods do this, but the most promising computationally oriented method is the linear nodal method.[35,36] We illustrate the method by starting from Eq. (140) and developing two equations, the first by integrating over r and the second by integrating over z within mesh cell i,j. This yields

$$\mu_m \frac{\partial[r\bar{\Phi}_{j,m}(r)]}{\partial r} + (\beta_m - \mu_m)\bar{\Phi}_{j,m}(r) + r\Sigma_{t,i,j}\Phi_{j,m}(r)$$
$$= r\bar{S}_{j,m}(r) + \beta_m\bar{\Phi}_{j,m-\frac{1}{2}}(r) - \xi_m r(\Phi_{j+\frac{1}{2},m}(r) - \Phi_{j-\frac{1}{2},m}) \tag{153}$$

$$\eta_m \frac{\partial\Phi_{i,m}(z)}{\partial z} + \frac{\beta_m - \mu_m}{\bar{r}_i}\bar{\Phi}_{i,m}(z) + \Sigma_{t,i,j}\bar{\Phi}_{i,m}(z)$$
$$= \bar{S}_{i,m}(z) + \frac{\beta_m}{\bar{r}_i}\bar{\Phi}_{i,m-\frac{1}{2}}(z) \tag{154}$$
$$- \mu_m[r_{i+\frac{1}{2}}\Phi_{i+\frac{1}{2},m}(z) - r_{i-\frac{1}{2}}\Phi_{i-\frac{1}{2},m}(z)] \quad ,$$

where

$$\bar{\Phi}_{j,m}(r) = \int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} \Phi_m(r,z)dz \quad , \tag{155a}$$

$$\bar{\Phi}_{i,m}(z) = \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \Phi_m(r,z)rdr \quad , \tag{155b}$$

with similar definitions for $S_{j,m}$ and $S_{i,m}$. The procedure now is to represent the source and the terms $\bar{\Phi}_{j\pm\frac{1}{2},m}(z)$, $\bar{\Phi}_{i+\frac{1}{2},m}(r)$, $\bar{\Phi}_{j,m-\frac{1}{2}}(r)$, $\bar{\Phi}_{i,m-\frac{1}{2}}(z)$ as linear functions and substitute into the right side of Eqs. (153) and (154). These equations can then be solved for $\bar{\Phi}_{m,j}(r)$ and $\bar{\Phi}_{i,m}(z)$ analytically in terms of these linear representations. Because the number of unknowns exceeds the number of equations, we use the first spatial moment equations and integrate as we did to obtain Eqs. (152) and (153). This set of equations, then, is sufficient to eliminate the unknowns and to develop the entire solution for the i,j cell. The procedure is much the same as the linear discontinuous method except that the analytic solution is used in each direction. Thus, the cell boundary fluxes are not extrapolated but are evaluated from an analytic formula. In this way, assuming a positive representation of the boundary data and the sources, the average outgoing cell boundary fluxes will be positive. The details of this development for (x,y) geometry are given in Ref. 36, along with an indication of the method's computational effectiveness.

The linear nodal method requires about half again as much computational time as the LD method per mesh cell with about the same storage requirement. It is more stable with respect to spatial oscillations and requires no flux fixup. Thus, for problems with large spatial mesh cells, the linear nodal method may be the method of choice over both the diamond and linear discontinuous methods in two or three dimensions.

To illustrate the effect of spatial differencing methods, we present a simple, iron/water shielding test problem representing a shielded reactor system. This is a three energy group problem in (x,y) geometry with the geometric arrangement shown in Fig. 11. The central 10-cm × 10-cm region

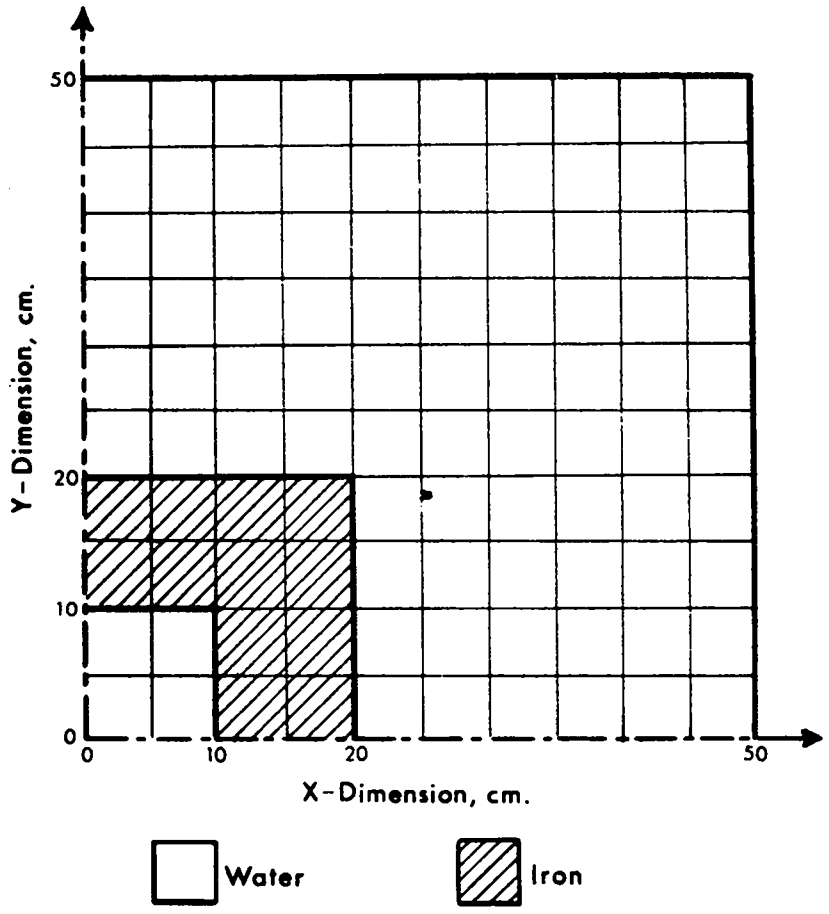


Fig. 11. Iron/water shielding problem.

contains a uniform source, and the problem is symmetric in x and y. For energy group three, the system is very large since the mean free path in the iron is about 1 cm and in the water regions about 0.3 cm. Table XII shows a summary of the calculational results as a function of spatial mesh size, which varies for group 3 from 15 mfp to 1 mfp. All calculations were performed using an S_4 quadrature set. Also presented in Table XI are the calculated groupwise leakages for the diamond difference (DD), the linear discontinuous (LD), and linear nodal (LN) methods, along with an indication of the computational time required on a CDC 7600 computer. The computer run times shown in the table are for comparison only. The calculations were performed using a prototype test

TABLE XII

IRON/WATER SHIELDING PROBLEM RESULTS
(Convergence Criterion = 10^{-5})

MESH SIZE, cm. (No. of Mesh Points)	Run Time, min.	D D NET LEAKAGE ($ Error)			Run Time min.	L D NET LEAKAGE ($ Error)			Run Time, min.	L N NET LEAKAGE ($ Error)		
		Group 1	Group 2	Group 3		Group 1	Group 2	Group 3		Group 1	Group 2	Group 3
5.0	0.2	2.337	2.864	37.39	0.3	4.407	2.750	8.557	0.6	4.745	2.483	4.822
(10x10 = 100)		(-51.1)	(53.6)	(677.2)		(-7.86	(9.34)	(77.86)		(-0.79	(-1.27)	(0.23)
2.5	0.6	4.115	2.612	13.60	1.2	4.698	2.481	4.636	1.7	4.783	2.527	4.901
(20x20 = 400)		(-14.0)	(3.86)	(182.7)		-1.78	-1.35	-3.64		(0)	(0.48)	(1.87)
1.0	3.5	4.680	2.627	5.215	7.1	4.778	2.523	4.886	10.5	4.783	2.520	4.856
(50x50 = 2500)		(-2.15)	(4.45)	(8.40)		(-0.10)	(0.32)	(1.56)		(0)	(0.20)	(0.94)
0.5	15.5	4.764	2.544	4.887	31.2	4.783	2.517	4.826	48.6	4.783	2.517	4.834
(100x100 = 10 000)		(-0.40)	(1.15)	(1.58)		(0)	(0.08)	(0.31)		(0)	(0.08)	(0.48)
0.333 ...	38.9	4.776	2.526	4.840								
(150x150 = 22 500)		(-0.15)	(0.44)	(0.60)								

code that contained no iteration acceleration methods. Actual production-type computer codes could be expected to perform the calculations in perhaps one-fifth of the times indicated in the table.)

The table shows clearly that the diamond method is inadequate for large mesh spacings, and the LD and LN methods are much more acceptable. The LN method gives good results even for the coarsest mesh.

It is generally concluded that for two-dimensional reactor core analysis problems, the diamond method is the method of choice because of efficiency since mesh sizes are generally $\leq$ 2 mfp. For shielding applications, however, the diamond method is frequently inadequate or inefficient because of the necessity of refining the mesh to acceptable sizes, and here the LD or LN method is preferable.

5. The Short Characteristic Method. We have outlined some methods used for spatial discretization of the two-dimensional (r,z) transport equation. The main thrust has been to enforce conservation of particles over a

spatial mesh cell and to obtain positivity of solution in ways that range from the simple to the elaborate. The enforcement of conservation is important for eigenvalue problems or for problems with a high within-group scattering ratio. In this section, we present a brief summary of a method in current use that does not use conservation within a mesh cell, but which is computationally simple and strictly positive. The envisioned application is for shielding problems where scattering need not be accurately resolved. The method is the short characteristic method, which has been developed and implemented by Takeuchi and Sasamoto[37,38] in the PALLAS series of codes.

We refer to Fig. 12 for the development of this method in (x,y) geometry. Given the direction $\vec{\Omega}_m$ for cell i,j, we want to determine the flux at the point D (the fluxes at points A, B, and C are known from solutions in the adjoining downstream cells). We can formally write the solution of the transport equation at point D by integrating along the characteristic $\vec{\Omega}_m$, which strikes the incoming cell boundary at point E. That is,

$$\Phi(D,\vec{\Omega}_m) = \Phi(E,\vec{\Omega}_m)e^{-\Sigma_t \ell} + \int_0^{\ell} S(t)e^{-\Sigma_t t}\, dt \quad , \tag{156}$$

where

t = the coordinate along $\vec{\Omega}_m$,

ℓ = the length of the trajectory from point E to D,

$S(t)$ = the source distribution along t.

In this method, the value of the flux at E is obtained by linear interpolation of the values at A and B. That is,

$$\Phi(E,\vec{\Omega}_m) = \rho\Phi(A,\vec{\Omega}_m) + (1 - \rho)\Phi(B,\vec{\Omega}_m) \quad , \tag{157}$$

where

$$\rho = k\mu_m/h\eta_m \quad ;$$

μ_m, η_m are the direction cosines of $\vec{\Omega}_m$ in the x and y directions, respectively, and h,k are the cell width and height, respectively.

We assume that the source is known at each of the vertex points from the previous iteration, and we assume linear representation of the source along the characteristic as

$$S(t,\vec{\Omega}_m) = \frac{t}{\ell} S(D,\vec{\Omega}_m) + \left(1 - \frac{t}{\ell}\right) S(E,\vec{\Omega}_m) \quad , \tag{158}$$

where

$$S(E,\vec{\Omega}_m) = \rho S(A,\vec{\Omega}_m) + (1 - \rho) S(B,\vec{\Omega}_m) \quad . \tag{159}$$

Thus, this solution method treats particle transport from point to point in the mesh using interpolation to obtain the required initial or incoming flux values and the sources. The accuracy of this method when the scattering ratio is small is limited by the number of angular directions, $\vec{\Omega}_m$, and the size of the spatial mesh that governs the accuracy of the interpolations.

We emphasize the weakness of this short characteristic method: no particle balance equation within a cell, such as Eq. (126), is satisfied. This restricts either the applicability or accuracy of the method, and it seems

to be most applicable to shielding problems. Indeed, applying this method in the reactor physics area has been most effective where no self-scatter iteration is performed. The reason is that eigenvalue solution techniques rely on particle balance to obtain an accurate and efficient solution. It is also true that for eigenvalue (fission) problems and problems with high scattering, the most effective iteration acceleration methods rely on particle conservation within a cell or a group of cells. Thus, we list the advantages of this method as

1) positive, yet computationally simple,
2) able to treat large absorbing regions well because it treats the streaming operator analytically,
3) flexible in its ability to treat complex geometries.

We see the disadvantages as

1) not accurate or efficient for eigenvalue problems,
2) low order of accuracy for strongly fissioning or scattering problems.

6. Summary. The most commonly used spatial discretization method in present general-purpose transport codes is the diamond-differencing method with either set-to-zero fixup or some weighted diamond fixup. If relatively few fixups are used, this method has most of the desirable properties of a spatial discretization method for reactor physics applications. The more elaborate methods incorporated into some limited-distribution codes seek to address the accuracy-positivity requirement while preserving the other desirable qualities of conservation and diffusion limit. These more elaborate methods are also useful when more accurate values of pointwise quantities on a given mesh are required because these methods greatly reduce spatial flux oscillations. Frequently such pointwise quantities can be obtained accurately with the diamond method only by use of an uneconomical mesh refinement. The last method discussed, the short-characteristic method, is a good example of a special-purpose, nonconservative discretization method. It is useful for some types of shielding problems where other methods may exhibit excessive inaccuracies in pointwise values of the solution.

C. Acceleration of the Inner Iteration

In Sec. III, we described the source inner iteration procedure for solving the transport equation. In the multigroup formulation, we note that the convergence of the inner iteration depends upon the scattering ratio, c_g, for that group where $c_g \equiv \Sigma^0_{s,g\to g}/\Sigma_{t,g}$; for a model problem, one can show that the spectral radius of convergence is the scattering ratio. When source iteration is applied to the geometrically complex problems encountered in reactor analysis, numerical experience shows that the spectral radius, ρ, of flux convergence is the highest scattering ratio of any material region in the problem in each group.

In the multigroup formulation of the transport problem, the scattering ratio for the groups depends upon the number and distribution of groups over the energy range of solution. In general, the more groups, the smaller the scattering ratio for each group (because of the higher probability for scattering out of the group). This is modified by the effects of scattering resonances in some groups or the use of a "dump" group in fast reactor applications, which covers all energies below a specified minimum. For a many group problem in fast reactor analysis (>20 groups), unaccelerated iteration for most of the groups may be satisfactory for convergence because the scattering ratio is less than 0.5, and it only takes about three iterations to reduce the error in the solution by an order of magnitude [see Eq. (95)]. However, such a fine group structure seldom occurs, so since a spectral radius of 0.8 requires 10 iterations to reduce the error in the solution by an order of magnitude, some form of iteration acceleration is usually needed for efficient computation of the transport solution. In this section, we describe three iteration acceleration methods for the inner iteration: Chebyshev, rebalance, and diffusion synthetic.

1. Chebyshev Acceleration of the Inner Iteration. The Chebyshev polynomial-based iteration acceleration method is a particular form of the general residual polynomial family of methods to accelerate the convergence of matrix iterative problems.[6] These methods are very general. They utilize the properties of the iteration matrix itself and thus are usually straightforward to implement. We start our discussion with the discrete ordinates, inner-iteration equation for energy group, g, using Eq. (93) written as

$$[\vec{\Omega} \cdot \vec{\nabla}\phi_g^{k+\frac{1}{2}}(\vec{r})]_m + \Sigma_{t,g}(\vec{r})\phi_{m,g}^{k+\frac{1}{2}}(\vec{r})$$

$$= \Sigma^0_{s,g\to g}(\vec{r})[\tilde{\phi}^k_{0,g}(\vec{r})] \qquad (160)$$

$$+ \sum_{n=2}^{NM} (2\ell + 1)\, \Sigma^{\ell}_{s,g\to g}(\vec{r}) R_n(\vec{\Omega})\tilde{\phi}^k_{n,g}(\vec{r}) + S_{m,g}(\vec{r}) \quad ,$$

where the superscripts k, $k+\frac{1}{2}$ denote an iteration index. In Eq. (160), we have used the spherical harmonics expansion form of Sec. II.D with the self-scatter (or within group) source separated out from the remainder of the scattering source. Further, the self-scatter source has been broken into two parts, the isotropic portion and the higher-order portion.

The unaccelerated iteration procedure is to set the flux moments

$$\tilde{\phi}^{k+1}_{0,g}(\vec{r}) = \sum_{m=1}^{M} w_m \phi^{k+\frac{1}{2}}_{m,g}(\vec{r}) \quad , \qquad (161a)$$

and

$$\tilde{\phi}^{k+1}_{n,g}(\vec{r}) = \sum_{m=1}^{M} w_m R_n(\vec{\Omega}_m)\phi^{k+\frac{1}{2}}_{m,g}(\vec{r}) \quad , \qquad (161b)$$

for $n > 1$.

The Chebyshev polynomial-based acceleration procedure, as it is normally implemented, assumes that the higher-moment scattering terms make a minor contribution to the source convergence compared to the zero moment isotropic term. Hence, Eq. (161b) remains the same, but Eq. (161a) is modified to

$$\phi^{k+1}_{0,g}(\vec{r}) = \alpha^k \sum_{m=1}^{M} w_m \phi^{k+\frac{1}{2}}_{m,g}(\vec{r}) + (1 - \alpha^k)\tilde{\phi}^k_{0,g}(\vec{r})$$

$$+ \beta^k(\tilde{\phi}^k_{0,g} - \tilde{\phi}^{k-1}_{0,g}) \quad , \qquad (162)$$

where

$$\alpha^{\circ} = \frac{2}{2 - \bar{\rho}} \quad , \quad \beta^{\circ} = 0 \quad ,$$

$$\alpha^{k} = \frac{4}{\bar{\rho}} \left[\frac{\cosh\,(k - 1)\gamma}{\cosh\, k\gamma}\right] \quad ,$$

$$\beta^{k} = \left(1 - \frac{\bar{\rho}}{2}\right) \alpha^{k} - 1 \quad ,$$

$$k = 1, 2, \cdots \quad ,$$

$$\gamma = \cosh^{-1}\left(\frac{2}{\bar{\rho}} - 1\right) \quad , \quad \text{and}$$

$\bar{\rho}$ = an estimate of the spectral radius of the transport iteration matrix.

The procedure involved in Eq. (162) is a straightforward application of the Chebyshev method to a problem where the eigenvalues of the iteration matrix lie in the range, $0 \leq \rho \leq \bar{\rho}$. This application implies that the iteration matrix is a positive definite matrix; that is, it has a largest eigenvalue whose eigenvector is everywhere positive. This, in turn, implies that the spatial differencing scheme as used in Eq. (160) should be positive. These restrictions can be relaxed, but numerical experience has indicated that the Chebyshev method is applied most effectively to those problems where the iteration matrix is positive definite and acceleration is important.

We also note that the Chebyshev method depends upon the spectral radius of the iteration matrix. This is estimated during the iteration process itself in general-purpose codes. Normally, one takes a few ($k \leq 5$) unaccelerated iterations and estimates $\bar{\rho}^{k}$ in some suitable manner.[6,39] Once the Chebyshev extrapolation has begun, one can update the estimate of $\bar{\rho}$ by continuing to monitor $\bar{\rho}^{k}$ and restarting the Chebyshev cycle. An alternative procedure is to use the infinite medium estimate of the spectral radius, which is simply $\max_{i,j} c_{i,j}$, where $c_{i,j}$ is the scattering ratio in spatial mesh cell (i,j).

There are many code-dependent details in the successful implementation of the Chebyshev method, but these remarks should suffice to display the basic concept and the potential of this method for the acceleration of the inner iteration of the discrete ordinates transport equation. Numerical experience has shown that the Chebyshev iteration procedure is stable and that it generally reduces the number of required iterations by a factor of two compared with unaccelerated iteration; that is, the Chebyshev spectral radius is approximately c^2, where c is the scattering ratio. However, when c is close to one, many iterations are still required, and it is well worth the search for a more effective acceleration method.

2. Coarse Mesh Rebalance. In the previous method, we described a general extrapolation procedure based upon the mathematical properties of the iteration matrix and its application to accelerate the convergence of an iteration scheme. We now describe an iteration-acceleration method based more upon the physical content of the equations. This acceleration method is called coarse mesh rebalance.

In solving the transport equation, we are usually concerned with obtaining angular moments of the flux, rather than the angular flux itself, as the primary solution. In the iterative procedure of Eq. (160), we obtain, with each successive iteration, improved estimates of the angular fluxes and, consequently, of the angular flux moments. It is plausible that the accuracy of these estimates would be improved if a balance on angular moments is enforced at each stage of the iteration because this is what should be occurring physically. For example, if we integrate Eq. (160) spatially over the entire system of interest and sum over the discrete ordinates, we obtain the balance for group g,

$$L_g^{k+\frac{1}{2}} + A_g^{k+\frac{1}{2}} = SS_g^k + S_{0,g} \quad , \tag{163}$$

where

$$L_g^{k+\frac{1}{2}} = \int_{\delta R} \vec{n} \cdot \vec{J}_{s,g}^{k+\frac{1}{2}} d^2\vec{r} \quad ,$$

$\vec{n}$ = the outward-pointing normal to the system exterior surface,

$J^{k+\frac{1}{2}}_{s,g}$ = the net current at the exterior surface,

$$A^{k+\frac{1}{2}}_{g} = \int_{R} \Sigma_{t,g}(\vec{r})\phi^{k+\frac{1}{2}}_{0,g}(\vec{r})\, d\vec{r} \quad ,$$

$$SS^{k}_{g} = \int_{R} \Sigma^{0}_{s,g\to g}(\vec{r})\phi^{k}_{0,g} \quad ,$$

$$S_{0,g} = \int_{R} S_{0,g}(\vec{r})\, d\vec{r}, \text{ the total isotropic source to group in the system,}$$

R = the region within the system,

δR = the surface bounding R.

We now rebalance; that is, we bring Eq. (163) into balance with the most recent flux information, $\phi^{k+\frac{1}{2}}_{m,g}$, by finding a factor f^{k+1} such that

$$L^{k+1}_{g} + Ab^{k+1}_{g} = S_{0,g} \quad , \tag{164}$$

where

$$Ab^{k+1}_{g} = \int_{R} (\Sigma_{t,g} - \Sigma^{0}_{s,g\to g})[\phi^{k+\frac{1}{2}}_{0,g}(\vec{r})f^{k+1}_{g}]\, d\vec{r} \quad ,$$

$$L^{k+1}_{g} = L^{k+\frac{1}{2}}_{g} f^{k+1}_{g} \quad .$$

Thus,

$$f^{k+1}_{g} = \frac{S_{0,g}}{L^{k+\frac{1}{2}}_{g} + Ab^{k+\frac{1}{2}}_{g}} \tag{165}$$

is the appropriate rebalance factor. It is then applied to the flux moments before the source is computed for the next iterate, and the iteration proceeds

as usual. That is, $\tilde{\Phi}^{k+1}_{n,g} = \tilde{\Phi}^{k+\frac{1}{2}}_{n,g} f^{k+1}_{g}$. The procedure given by Eqs. (164) and (165) is called "whole system rebalance." It is a stable (convergent) method of iteration acceleration.

This rebalance procedure is readily generalized to the situation where balance is enforced on a spatial subset of the calculational grid. To illustrate this procedure, we specialize to the spatially discretized form of Eq. (160) on an orthogonal grid in two-dimensional geometry. The spatial subset is defined from the calculational grid by taking a subset of the grid lines to form a coarse mesh, which is then itself an orthogonal mesh (as shown in Fig. 13). It is general practice and physically reasonable that all material boundaries lie on coarse mesh lines. The coarse mesh can also correspond to the original fine mesh. Again, the idea is to enforce balance only on the angle-integrated equation. Thus, starting from Eq. (160), we sum over the angles to obtain for each group g (with the group subscript suppressed, for simplicity),

$$\begin{aligned} & A_{i+\frac{1}{2}}\left(J^{+}_{i+\frac{1}{2}} + J^{-}_{i+\frac{1}{2}}\right)^{k+\frac{1}{2}} - A_{i-\frac{1}{2}}\left(J^{+}_{i-\frac{1}{2}} + J^{-}_{i-\frac{1}{2}}\right)^{k+\frac{1}{2}} \\ & \quad + \Delta B_i\left(I^{+}_{j+\frac{1}{2}} + I^{-}_{j+\frac{1}{2}} - I^{+}_{j-\frac{1}{2}} - I^{-}_{j-\frac{1}{2}}\right)^{k+\frac{1}{2}} \\ & \quad + \left(\Sigma_t V\right)_{i,j} \Phi^{k+\frac{1}{2}}_{0,i,j} = \left(\Sigma^{0}_{g\to g} V\right)_{i,j} \Phi^{k}_{0,i,j} + \left(S_0 V\right)_{i,j} \quad , \end{aligned} \tag{166}$$

with the flows

$$J^{+}_{i+\frac{1}{2}} = \sum_{\mu_m>0} w_m \mu_m \Phi_{m,i+\frac{1}{2},j} \quad , \tag{167a}$$

$$J^{-}_{i+\frac{1}{2}} = \sum_{\mu_m<0} w_m \mu_m \Phi_{m,i+\frac{1}{2},j} \quad , \tag{167b}$$

$$I^{+}_{j+\frac{1}{2}} = \sum_{\eta_m>0} w_m \eta_m \Phi_{m,i,j+\frac{1}{2}} \quad , \tag{167c}$$

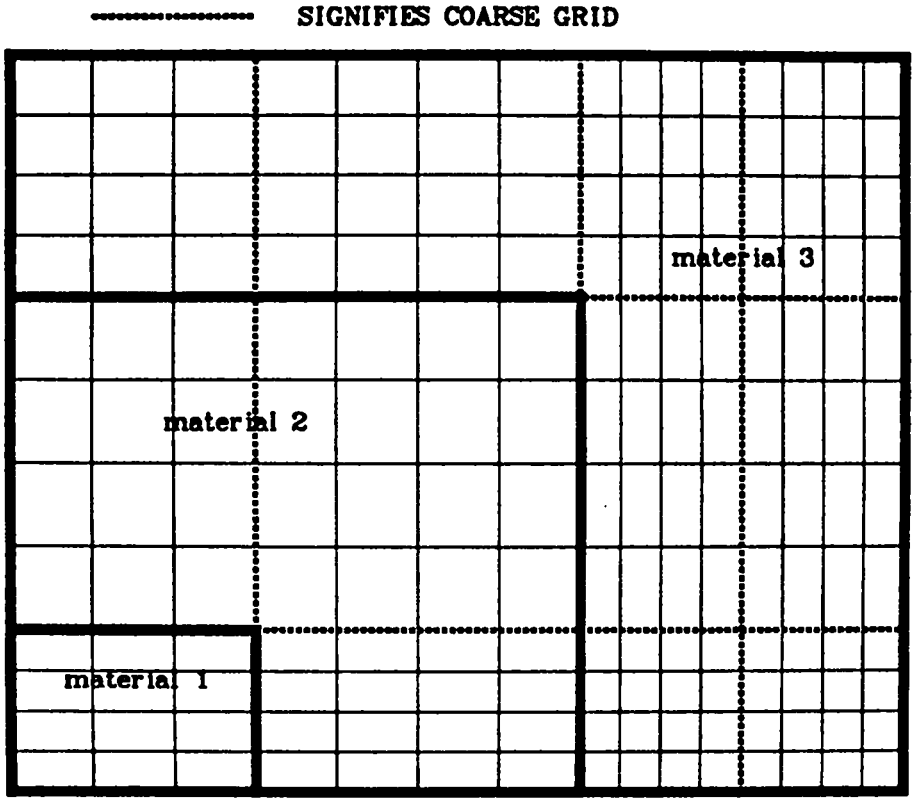


Fig. 13. Typical coarse mesh arrangement for rebalance.

$$I^{-}_{j+\frac{1}{2}2} = \sum_{\eta_m<0} w_m \eta_m \Phi_{m,i,j+\frac{1}{2}} \quad , \tag{167d}$$

and the iteration index k shows the state of the solution.

Now balance is enforced over each coarse mesh by defining a factor, $f^{k+1}_{I,J}$, for each coarse mesh (I,J) such that Eq. (166) is exactly satisfied on that coarse mesh. An equation for the $f^{k+1}_{I,J}$ is derived from Eq. (166) by summing over the coarse mesh and associating the appropriate rebalance factor with each mesh. We obtain, for each group g, the following rebalance equation:

$$FR^{k+\frac{1}{2}}_{I+\frac{1}{2},J} f^{k+1}_{I,J} - FL^{k+\frac{1}{2}}_{I+\frac{1}{2},J} f^{k+1}_{I+1,J} - FR^{k+\frac{1}{2}}_{I-\frac{1}{2},J} f^{k+1}_{I-1,J} + FL^{k+\frac{1}{2}}_{I-\frac{1}{2},J} f^{k+1}_{I,J}$$

$$+ FU^{k+\frac{1}{2}}_{I,J+\frac{1}{2}} f^{k+1}_{I,J} - FD^{k+\frac{1}{2}}_{I,J+\frac{1}{2}} f^{k+1}_{I,J+1} \qquad (168)$$

$$- FU^{k+\frac{1}{2}}_{I,J-\frac{1}{2}} f^{k+1}_{I,J-1} + FD^{k+\frac{1}{2}}_{I,J-\frac{1}{2}} f^{k+1}_{I,J} + \Sigma_R f^{k+1}_{I,J} = S_{0I,J} \quad ,$$

where

$$FL_{I+\frac{1}{2},J} = \sum_{j\epsilon J} A_{I+\frac{1}{2},j} J^{-}_{I+\frac{1}{2},j} \quad , \qquad (169a)$$

$$FR_{I+\frac{1}{2},J} = \sum_{j\epsilon J} A_{I+\frac{1}{2},j} J^{+}_{I+\frac{1}{2},j} \quad , \qquad (169b)$$

$$FU_{I,J+\frac{1}{2}} = \sum_{i\epsilon I} B_i I^{+}_{i,J+\frac{1}{2}} \quad , \qquad (169c)$$

$$FD_{I,J+\frac{1}{2}} = \sum_{i\epsilon I} B_i I^{-}_{i,J+\frac{1}{2}} \quad , \qquad (169d)$$

$$\Sigma_{R,I,J} = \sum_{i\epsilon I} \sum_{j\epsilon J} \left(\Sigma_t - \Sigma^0_{g\to g}\right)_{i,j} \Phi_{0,i,j} V_{i,j} \quad , \qquad (169e)$$

$$S_{0,I,J} = \sum_{i\epsilon I} \sum_{j\epsilon J} S_{0,i,j} V_{i,j} \quad , \qquad (169f)$$

and I,J denotes the coarse mesh interval. Equation (168) has the form of a standard five-point difference equation for the rebalance factors f and, thus, can be solved by any of a number of standard techniques. Upon solution of the rebalance equation, we define the next flux moment iterate for Eq. (160) to be

$$(\tilde{\Phi}_n)^{k+1}_{i,j} = (\tilde{\Phi}_n)^k_{i,j} f^{k+1}_{I,J} \text{ for } i\epsilon I \text{ and } j\epsilon J \quad , \qquad (170)$$

where k is the iteration index.

In practice, this coarse mesh rebalance approach is better than the whole system rebalance of Eq. (165) because balance is enforced at each coarse mesh cell rather than just for the whole system. There are two remaining questions about this method: 1) how does the choice of the coarse mesh bear upon the effectiveness of this acceleration? and 2) how does the choice of the coarse mesh bear upon the stability of the method? The first question results from the observation through numerical experiments that when it converges, this method is generally the more effective the finer the coarse mesh; that is, it is most effective when the fine and coarse mesh correspond to each other. The measure of effectiveness we are using here is the rate of convergence of the accelerated iteration; that is, if we assume that the convergence rate of the unaccelerated iteration is c_{max}, the maximum scattering ratio, then the accelerated iteration convergence rate can be characterized as ac_{max}, $0 < a < 1$. Thus, the more effective the accelerator is, the smaller the value of a. Question 2), above, results from the observation that as the coarse mesh approaches the size of the fine mesh, in some problems the acceleration is less and less stable, and, in fact, can be unstable depending on the size of the fine mesh and the kind of spatial discretization used.[40] It has been observed that for high scattering problems ($c \approx 1$) this spatial size limit for stability is 1-2 mfp; thus, in a typical multigroup reactor analysis problem, it is difficult to ensure stability by choosing the coarse mesh to be the fine mesh, which, we maintain, is required for maximum effectiveness. The art of applying the coarse mesh rebalance acceleration method is in choosing a rebalance mesh coarse enough to ensure stability and fine enough to achieve good effectiveness. This can be a very difficult, if not impossible, task in many multigroup reactor analysis problems. However, numerical experience has shown that for stable applications of coarse mesh rebalance, this acceleration method is more effective than the Chebyshev approach with the exception of whole system rebalance.

Thus, we have presented two methods, one of which is stable but of limited effectiveness and the other of varying effectiveness but not always stable. The next method goes a long way to ensure effectiveness while remaining stable for a wide range of problems.

3. The Diffusion Synthetic Acceleration Method.[41-43] In presenting the diffusion synthetic acceleration (DSA) method, we return to Eq. (160). The

basic idea is somewhat like that involved in developing the rebalance method; that is, we seek to balance the zeroth angular moment and the first angular moment of the transport equation. By doing this, we can develop an equation with some interesting properties, and in the course of this development, we can address the effectiveness and stability of the resulting acceleration method. As will be seen shortly, this acceleration method is based upon the diffusion equation, which is appealing because it is a valid approximation to the transport equation in certain limits. These limits are expressed mathematicly by assuming that the angular flux is accurately represented by a linear function of angle. Physically, this representation is valid within large, homogeneous regions (about 2-3 mfp away from material boundaries) and where the scattering ratio is close to one. It is precisely in such a situation that the source iteration of the transport equation is slowly convergent; hence, it is reasonable to expect that a judicious employment of the diffusion equation will aid the convergence process.

First, we consider Eq. (160) not differenced on a spatial mesh, and we take the first two angular moments while we drop the energy group index for simplicity; this yields

$$\vec{\nabla} \cdot \vec{J}^{k+\frac{1}{2}}(\vec{r}) + \Sigma_t \Phi_0^{k+\frac{1}{2}}(\vec{r}) = \Sigma_s^0 \Phi_0^k(\vec{r}) + S_0(\vec{r}) \quad , \tag{171}$$

$$\frac{1}{3} \nabla \Phi_0^{k+\frac{1}{2}}(\vec{r}) + \nabla \cdot \underline{\underline{P}}^{k+\frac{1}{2}}(\vec{r}) + \Sigma_t \vec{J}^{k+\frac{1}{2}}(\vec{r}) = \Sigma_s^1 \vec{J}^k(\vec{r}) + S_1(\vec{r}) \quad , \tag{172}$$

where

$$\Phi_0(\vec{r}) = \sum_{m=1}^{M} w_m \Phi_m(\vec{r}) \quad ,$$

$$\vec{J}(\vec{r}) = \sum_{m=1}^{M} w_m \vec{\Omega}_m \Phi_m(\vec{r}) \quad ,$$

$$\underline{\underline{P}}(\vec{r}) = \sum_{m=1}^{M} w_m \left(\vec{\Omega}_m \vec{\Omega}_m - \Omega_m^2 \underline{\underline{I}} \right) \Phi_m(\vec{r}) \quad ,$$

$\underline{\underline{I}}$ = the unit diad ,

$$\vec{S}_1 = \sum_{m=1}^{M} w_m \vec{\Omega}_m S_m(\vec{r}) \quad .$$

In the above, $\Phi_0(\vec{r})$ is the scalar flux, $\vec{J}(\vec{r})$ is the current, and $\underline{\underline{P}}(\vec{r})$ is a tensor of a second angular moment quantities. As in rebalance and upon examining Eqs. (171) and (172), we want the following equations to be satisfied at each iteration for each group:

$$\vec{\nabla} \cdot \vec{J}^{k+1}(\vec{r}) + \Sigma_t \Phi_0^{k+1}(\vec{r}) = \Sigma_s^0 \Phi_0^{k+1}(\vec{r}) + S_0(\vec{r}) \quad , \tag{173}$$

$$\frac{1}{3} \nabla \Phi_0^{k+1}(\vec{r}) + \nabla \cdot \underline{\underline{P}}^{k+\frac{1}{2}}(\vec{r}) + \Sigma_t \vec{J}^{k+1}(\vec{r}) = \Sigma_s^1 \vec{J}^{k+1}(\vec{r}) + S_1(\vec{r}) \quad . \tag{174}$$

Notice that we do not insist that the quantity $\underline{\underline{P}}$ be accelerated at the iteration step; instead, it retains its unaccelerated value. We also note that if the angular flux is a linear function of angle, then $\underline{\underline{P}}$ is zero and is, thus, in some sense, a second-order correction term in our acceleration method. Indeed, if $\underline{\underline{P}}$ is zero, then Eqs. (173) and (174) can be used to determine Φ_0^{k+1} and $\vec{J}^{k+1}$ in one iteration. However, we stress that $\underline{\underline{P}}$ need not be zero or small to ensure the effectiveness of the acceleration. We combine Eqs. (172) and (174) to eliminate $\underline{\underline{P}}^{k+\frac{1}{2}}$ and obtain

$$\begin{aligned} (\Sigma_t - \Sigma_s^1)\vec{J}^{k+1}(\vec{r}) = & -\frac{1}{3} \nabla \Phi_0^{k+1}(\vec{r}) + \frac{1}{3} \nabla \Phi_0^{k+\frac{1}{2}}(\vec{r}) \\ & - \Sigma_s^1 [\vec{J}^k(\vec{r}) - \vec{J}^{k+\frac{1}{2}}(\vec{r})] \\ & + (\Sigma_t - \Sigma_s^1)\vec{J}^{k+\frac{1}{2}}(\vec{r}) \quad . \end{aligned} \tag{175}$$

Substitution of Eq. (175) into Eq. (173) now yields

$$-\nabla \cdot D(\vec{r})\nabla\Phi_0^{k+1}(\vec{r}) + \Sigma_R\Phi_0^{k+1}(\vec{r})$$

$$= S_0(\vec{r}) - [\nabla \cdot D\nabla\Phi_0^{k+\frac{1}{2}}(\vec{r}) + \vec{\nabla} \cdot \vec{J}^{k+\frac{1}{2}}(\vec{r})] \qquad (176)$$

$$+ \nabla \cdot \{\frac{\Sigma_s^1}{\Sigma_{tr}} [\vec{J}^k(\vec{r}) - \vec{J}^{k+\frac{1}{2}}(\vec{r})] \quad ,$$

where

$$\Sigma_{tr} = \Sigma_t - \Sigma_s^1 \quad ,$$

$$D = \frac{1}{3\Sigma_{tr}} \quad ,$$

$$\Sigma_R = \Sigma_t - \Sigma_{s,g\to g}^0 \quad .$$

The solution of Eq. (176) thus replaces Eq. (161a) in the iteration solution of Eq. (160). Equation (176) has the form of the standard diffusion equation with the addition of two source correction terms on the right side. Also, this equation can be solved with either the conventional diffusion boundary conditions or with boundary conditions specifically tailored to yield a solution in one iteration if the transport solution is indeed a linear function of angle.[44] For weakly anisotropic scattering, the second correction term involving the difference of currents makes a small contribution to the total correction and thus is usually ignored.* The first correction term is essential, giving the necessary correction to diffusion theory so that, at convergence, the solution to Eq. (176) will yield the same scalar flux as obtained from Eq. (160). This is what makes this method an acceleration method. By assuming that the second correction term to the source on the right side of Eq. (176) is zero, we obtain the following "source correction" diffusion synthetic equation for inner iteration acceleration for each group, g:

---

*If we retain this term, then the $\vec{J}^{k+1}$ from Eq. (175) can be used to accelerate the P_1 scattering term of Eq. (166). However, in reactor applications, this is a rarely needed complication and is not included in present reactor physics codes.

$$-\nabla \cdot D(\vec{r})\nabla\phi_0^{k+1}(\vec{r}) + (\Sigma_t - \Sigma^0_{s,g\to g})\phi_0^{k+1}$$

$$= S_0(\vec{r}) - [\nabla \cdot D\nabla\phi_0^{k+\frac{1}{2}}(\vec{r}) + \vec{\nabla} \cdot \vec{J}^{k+\frac{1}{2}}(\vec{r})] \quad , \tag{177}$$

where the group subscript has been omitted for simplicity. We also note that if we replace the diffusion coefficient $D(\vec{r})$, above, with the following diagonal tensor with components,

$$[\underline{\underline{D}}]^{k+\frac{1}{2}}_{\alpha,\beta} = - \frac{J^{k+\frac{1}{2}}_{\alpha}(\vec{r})}{\nabla_\alpha \phi_0^{k+\frac{1}{2}}(\vec{r})} \delta_{\alpha,\beta} \qquad \begin{cases} \alpha = 1, 2, 3 \\ \beta = 1, 2, 3 \end{cases} \quad , \tag{178}$$

then Eq. (177) becomes

$$-\nabla \cdot \underline{\underline{D}}^{k+\frac{1}{2}} \cdot \nabla\phi_0^{k+1}(\vec{r}) + \Sigma_R(\vec{r})\phi_0^{k+1}(\vec{r}) = S_0(\vec{r}) \quad . \tag{179}$$

This form is used for the homogeneous eigenvalue problem, as will be seen in the next section, and is designated the "diffusion correction" scheme.[42] The scheme is nonlinear and, hence, not amenable to analysis. However, numerical experience indicates that DSA methods based upon Eq. (179) have the same performance as those based upon Eq. (177). An attendant practical problem with defining a diffusion coefficient from Eq. (178) is that it can numerically go negative or even infinite. A remedy has been devised to modify the removal term in these cases while using the conventional diffusion coefficent. This preserves the homogeneous nature of the diffusion equation. The details of this procedure are explained in Ref. 42.

The advantage of Eq. (176) is that one can do a stability analysis for a model problem. The model problem is an infinite homogeneous medium in Cartesian (x,y,z) geometry; thus, the cross sections are constant, allowing a Fourier analysis. We rewrite our equations for this case as

$$\vec{\Omega} \cdot \vec{\nabla}\phi_m^{k+\frac{1}{2}}(\vec{r}) + \Sigma_t\phi_m^{k+\frac{1}{2}}(\vec{r}) = c\Sigma_t\phi_0^k(\vec{r}) + S_m(\vec{r}) \quad , \tag{180}$$

$$-D\nabla^2\Phi_0^{k+1}(\vec{r}) + (1 - c)\Sigma_t\Phi_0^{k+1}(\vec{r})$$

$$= S_0(\vec{r}) - [D\nabla^2\Phi_0^{k+\frac{1}{2}}(\vec{r}) + \vec{\nabla} \cdot \vec{J}^{k+\frac{1}{2}}(\vec{r})] \quad . \tag{181}$$

If we resolve Φ by a Fourier analysis,[43,45] we write

$$\left.\begin{aligned} \Phi_m^{k+\frac{1}{2}} &= A_m^{k+\frac{1}{2}} e^{i\vec{\lambda}\cdot\vec{r}} \\ \Phi_0^{k} &= B^{k} e^{i\vec{\lambda}\cdot\vec{r}} \end{aligned}\right\} \quad i = -1 \quad , \quad -\infty < \lambda < \infty \quad , \tag{182}$$

and from Eq. (180) we obtain (with source set to zero)

$$A_m^{k+\frac{1}{2}} = \frac{c}{1 + (i\vec{\Omega}_m \cdot \vec{\lambda})/\Sigma_t} B^k \quad . \tag{183}$$

We now substitute into Eq. (181) to obtain

$$\omega(\lambda) \quad \left|\frac{B^{k+1}}{B^k}\right| = c\left[\frac{(1 + D\lambda^2/\Sigma_t)\rho(\lambda) - 1}{1 - c + D\lambda^2/\Sigma_t}\right] \quad , \tag{184}$$

where

$\lambda = |\vec{\lambda}|$,

$\omega(\lambda)$ = the eigenvalue of the accelerated iteration as a function of the parameter λ,

$$\rho(\lambda) = \sum_{m=1}^{M} \frac{w_m}{1 + (\vec{\Omega}_m \cdot \vec{\lambda}/\Sigma_t)^2} \quad .$$

In terms of the spectral radius of convergence of the iteration, $\bar{\rho}$, we note that

$\bar{\rho}_0 = \sup_{\lambda} \rho(\lambda)c$ = spectral radius of the unaccelerated iteration,

and

$$\bar{\rho} = \sup_{\lambda} \omega(\lambda) = \text{spectral radius of the accelerated iteration.}$$

In Fig. 14, plots show the iteration spectra $\omega(\lambda)/c$ and $\rho(\lambda)$ as functions of λ/Σ_t for isotropic scatter ($\Sigma_s^1 = 0$). For the worst case with $c = 1$, the unaccelerated iteration is nonconvergent since it has a spectral radius of unity; for the worst case with $c = 1$, the spectral radius for the diffusion synthetic accelerated iteration has a maximum value of 0.2237, so only six iterations are required to reduce the error in the calculated fluxes by four orders of magnitude.

This is an impressive acceleration, at least for the model problem. It now remains to see whether this holds up for nonmodel problems where the cross

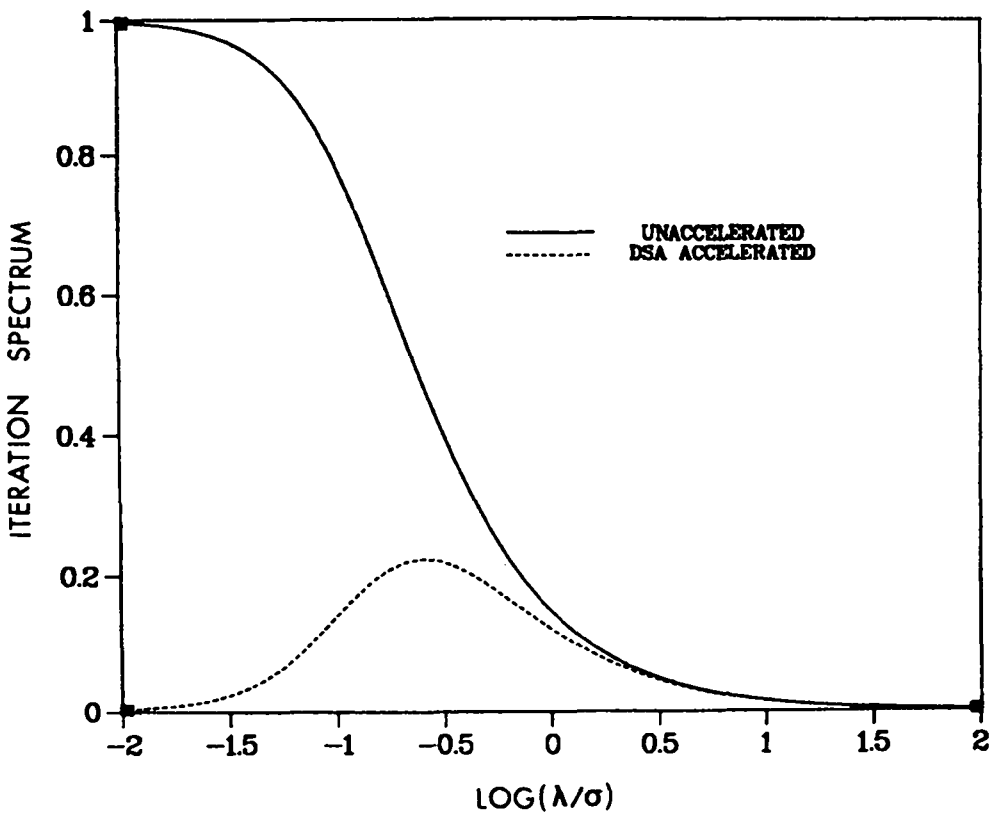


Fig. 14. Iteration spectra for unaccelerated and DSA accelerated iteration.

sections can change by orders of magnitude as a function of space. To address this issue, we first must consider the effects of spatial discretization. This is a very complicated consideration because the results depend upon the discretization chosen. We illustrate the concepts involved in deriving a DSA equation for a discretized transport equation using a one-dimensional-slab problem with diamond differencing. In this case, the relevant transport equations are (with angular and energy group subscripts suppressed):

$$\frac{\mu}{\Delta x_i}\left(\Phi_{i+\frac{1}{2}}^{k+\frac{1}{2}} - \Phi_{i-\frac{1}{2}}^{k+\frac{1}{2}}\right) + \Sigma_{t,i}\Phi_i^{k+\frac{1}{2}} = c_i\Sigma_{t,i}\Phi_{0,i}^{k} + S_i \quad , \tag{185}$$

$$\Phi_i^{k+\frac{1}{2}} = \frac{1}{2}\left(\Phi_{i+\frac{1}{2}}^{k+\frac{1}{2}} + \Phi_{i-\frac{1}{2}}^{k+\frac{1}{2}}\right) \quad . \tag{186}$$

Again, we take the first two angular moments of Eq. (185) to obtain

$$\frac{1}{\Delta x_i}\left(J_{i+\frac{1}{2}}^{k+\frac{1}{2}} - J_{i-\frac{1}{2}}^{k+\frac{1}{2}}\right) + \Sigma_{t,i}\Phi_{0,i}^{k+\frac{1}{2}} = c_i\Sigma_{t,i}\Phi_{0,i}^{k} + S_{0,i} \quad , \tag{187}$$

$$\frac{1}{3\Delta x_i}\left(\Phi_{0,i+\frac{1}{2}}^{k+\frac{1}{2}} - \Phi_{0,i-\frac{1}{2}}^{k+\frac{1}{2}}\right) + \frac{2}{3\Delta x_i}\left(\Phi_{2,i+\frac{1}{2}}^{k+\frac{1}{2}} - \Phi_{2,i-\frac{1}{2}}^{k+\frac{1}{2}}\right) + \Sigma_{t,i}J_i^{k+\frac{1}{2}} = 0 \quad , \tag{188}$$

$$\Phi_{0,i}^{k+\frac{1}{2}} = \frac{1}{2}\left(\Phi_{0,i+\frac{1}{2}} + \Phi_{0,i-\frac{1}{2}}\right) \quad , \tag{189a}$$

$$J_i^{k+\frac{1}{2}} = \frac{1}{2}\left(J_{i+\frac{1}{2}} + J_{i-\frac{1}{2}}\right) \quad . \tag{189b}$$

We again assume Eqs. (187) through (189) are satisfied at iterate k+1 except for the Φ_2 term; thus, the current equation becomes

$$J_i^{k+1} = J_i^{k+\frac{1}{2}} - \frac{1}{3(\Sigma_t \Delta x)_i}\left(\Phi_{0,i+\frac{1}{2}}^{k+1} - \Phi_{0,i-\frac{1}{2}}^{k+1}\right)$$

$$+ \frac{1}{3(\Sigma_t \Delta x)}\left(\Phi_{0,i+\frac{1}{2}}^{k+\frac{1}{2}} - \Phi_{0,i-\frac{1}{2}}^{k+\frac{1}{2}}\right) \quad . \tag{190}$$

We can combine Eqs. (187) and (190) along with the diamond relationships by first adding two adjacent cells in Eq. (187),

$$\left(J_{i+\frac{3}{2}} + J_{i+\frac{1}{2}}\right)^{k+1} - \left(J_{i+\frac{1}{2}} + J_{i-\frac{1}{2}}\right)^{k+1}$$

$$+ \left[(1 - c)\Sigma_t \Delta x\right]_{i+1} \Phi_{0,i+1}^{k+1}$$

$$+ \left[(1 - c)\Sigma_t \Delta x\right]_i \Phi_{0,i}^{k+1}$$

$$= \left(S_0 \Delta x\right)_{i+1} + \left(S_0 \Delta x\right)_i \quad ,$$

which, combined with Eq. (190), leads to our final result,

$$- \frac{1}{3(\Sigma_t \Delta x)_{i+1}}\left(\Phi_{0,i+\frac{3}{2}} - \Phi_{0,i+\frac{1}{2}}\right)^{k+1}$$

$$+ \frac{1}{3(\Sigma_t \Delta x)_i}\left(\Phi_{0,i+\frac{1}{2}} - \Phi_{0,i-\frac{1}{2}}\right)^{k+1}$$

$$+ \frac{1}{4}\left\{\left[(1 - c)\Sigma_t \Delta x\right]_{i+1}\left(\Phi_{0,i+\frac{3}{2}} + \Phi_{0,i+\frac{1}{2}}\right)^{k+1}\right.$$

$$\left. + \left[(1 - c)\Sigma_t \Delta x\right]_i \left(\Phi_{0,i+\frac{1}{2}} + \Phi_{0,i-\frac{1}{2}}\right)^{k+1}\right\} \tag{191}$$

$$= \frac{1}{2}\left[\left(S_0 \Delta x\right)_{i+1} + \left(S_0 \Delta x\right)_i\right]$$

$$- \frac{1}{3(\Sigma_t \Delta x)_{i+1}}\left(\Phi_{0,i+\frac{3}{2}} - \Phi_{0,i+\frac{1}{2}}\right)^{k+\frac{1}{2}}$$

$$+ \frac{1}{3(\Sigma_t \Delta x)_i}\left(\Phi_{0,i+\frac{1}{2}} - \Phi_{0,i-\frac{1}{2}}\right)^{k+\frac{1}{2}}$$

$$- \frac{1}{2}\left(J_{i+1}^{k+\frac{1}{2}} - J_i^{k+\frac{1}{2}}\right) \quad ,$$

and

$$\phi_{0,i}^{k+1} = \frac{1}{2}\left(\phi_{0,i+\frac{1}{2}}^{k+1} + \phi_{0,i-\frac{1}{2}}^{k+1}\right) \quad . \tag{192}$$

Equations (191) and (185), are the synthetic diffusion equations under the assumption of diamond differencing of the spatial variable in slab geometry. Similar equations are derivable for diamond differencing in other geometries. With other spatial-differencing methods such as linear discontinuous, a similar line of reasoning leads to the appropriate equations in slab geometry.[43] However, it is not at all clear that a simple diffusion-like acceleration equation is possible in more complex cases.

We now consider the stability and effectiveness of the iteration embodied in Eqs. (185) and (191). For constant cross sections, we can again perform a Fourier analysis and obtain an expression for the spectral radius of iteration. We quote the result here as

$$\omega(\lambda) = \frac{(\rho - 1)/\rho + (4/3)(\Sigma_t \Delta x)^2 \tan^2(\lambda \Delta x/2)}{1 - c + (4/3)(\Sigma_t \Delta x)^2 \tan^2(\lambda \Delta x/2)} \rho c \quad , \tag{193}$$

where

$$\rho = \sum_{m=1}^{M} \left[\frac{w_m}{1 + 4(\mu_m/\Sigma_t \Delta x)^2 \tan^2(\lambda \Delta x/2)}\right] \quad ,$$

Δx = the spatial mesh size.

Since

$$\sup_{\lambda} \omega(\lambda)\Big|_{\text{Eq. (184)}} = \sup_{\lambda} \omega(\lambda)\Big|_{\text{Eq. (193)}} \quad ,$$

the spectral radius for the continuous and discrete methods is the same, independent of mesh size. Thus, for diamond differencing, iterating with Eqs. (185) and (191) is very effective and stable with respect to spatial mesh size.

This proof of stability is for the model problem only. Naturally, since one does not solve the model problem in practice, but problems in which the mesh and cross sections are not constant, one can question whether the conclusions derived from the above analysis are valid for real problems. Through many tests and numerical experiments, it has been shown that for real problems in which the boundary conditions have been properly formulated, one obtains the above stability and convergence properties.[44] That is, one can assume that about six iterations are required to reduce the error by four orders of magnitude.

Equation (191) is used to accelerate the diamond-differenced transport Eq. (185). However, negative flux fixup is frequently used in the diamond solution of Eq. (185). If Eq. (191) is used as the acceleration equation when fixup is employed, the DSA procedure will not converge in the sense that

$$\phi_{0,i+\frac{1}{2}}^{k=\infty} \neq \sum_{m=1}^{M} w_m \phi_{m,i+\frac{1}{2}}^{k=\infty} \quad .$$

A generally accepted way to remedy this is to modify the diamond relationships of Eq. (189) to the forms,

$$\phi_{0,i}^{k+1} = \phi_{0,i}^{k+\frac{1}{2}} \left(\frac{\phi_{0,i+\frac{1}{2}}^{k+1} + \phi_{0,i-\frac{1}{2}}^{k+1}}{\phi_{0,i+\frac{1}{2}}^{k+\frac{1}{2}} + \phi_{0,i-\frac{1}{2}}^{k+\frac{1}{2}}} \right)$$

$$= \gamma_i^{k+\frac{1}{2}} \left(\phi_{0,i+\frac{1}{2}}^{k+1} + \phi_{0,i-\frac{1}{2}}^{k+1} \right) \quad , \tag{194a}$$

$$J_i^{k+1} = J_i^{k+\frac{1}{2}} \left(\frac{J_{i+\frac{1}{2}}^{k+1} + J_{i-\frac{1}{2}}^{k+1}}{J_{i+\frac{1}{2}}^{k+\frac{1}{2}} + J_{i-\frac{1}{3}}^{k+\frac{1}{2}}} \right) = \delta_i^{k+\frac{1}{2}} \left(J_{i+\frac{1}{2}}^{k+1} + J_{i-\frac{1}{2}}^{k+1} \right) \quad . \tag{194b}$$

Equation (194) is a generalization of the diamond expression relating the cell-centered quantities to the cell-edge quantities and, in fact, reduces to Eq. (189) when no fixup is used in solving Eq. (185). Now, if Eqs. (194a) and (194b) are used in combining Eqs. (187) and (188) to a single equation, we obtain

$$
\begin{aligned}
&-D_{i+1}^{k+\frac{1}{2}}\left(\Phi_{0,i+\frac{3}{2}}-\Phi_{0,i+\frac{1}{2}}\right)^{k+1}+D_{i}^{k+\frac{1}{2}}\left(\Phi_{0,i+\frac{1}{2}}-\Phi_{0,i-\frac{1}{2}}\right)^{k+1}\\
&\quad+\frac{1}{4}\left\{\left[(1+c)\,\Sigma_{t}\Delta x\gamma^{k+\frac{1}{2}}\right]_{i+1}\left(\Phi_{0,i+\frac{3}{2}}+\Phi_{0,i+\frac{1}{2}}\right)^{k+1}\right.\\
&\quad\left.+\left[(1+c)\,\Sigma_{t}\Delta x\gamma^{k+\frac{1}{2}}\right]_{i}\times\left(\Phi_{0,i+\frac{1}{2}}+\Phi_{0,i-\frac{1}{2}}\right)^{k+1}\right\} \qquad (195)\\
&\quad=\frac{1}{2}\left[(S\cdot\Delta x)_{i+1}+(S\cdot\Delta x_{i}\right]-D_{i+1}^{k+\frac{1}{2}}\left(\Phi_{0,i+\frac{3}{2}}-\Phi_{0,i+\frac{1}{2}}\right)^{k+\frac{1}{2}}\\
&\quad+D_{i}\left(\Phi_{0,i+\frac{1}{2}}-\Phi_{0,i-\frac{1}{2}}\right)^{k+\frac{1}{2}}\\
&\quad-\frac{1}{2}\left(J_{i+\frac{3}{2}}-J_{i-\frac{1}{2}}\right)^{k+\frac{1}{2}} ,
\end{aligned}
$$

where

$$
D_{i}^{k+\frac{1}{2}}=\frac{1}{3\left(\Sigma_{t}\Delta x\delta^{k+\frac{1}{2}}\right)_{i}} .
$$

Unfortunately, Eq. (195) has some deficiencies that adversely impact the acceleration procedure. Because Eq. (195) has a three-point removal term, it is possible to obtain negative solutions; however, (Eq. (185) with fixup always generates non-negative solutions. This situation leads to instabilities that are not easily corrected by some kind of coding logic. To circumvent these difficulties, one frequently resorts to converting Eq. (195) into an equation with a one-point removal term while retaining Eqs. (194a) and (194b) to ensure consistency between the transport and diffusion scalar fluxes upon convergence. We write these two equations as

$$
\begin{aligned}
&- \frac{1}{3(\Sigma_t \Delta x)_{i+1}} \left(\Phi_{0,i+\frac{1}{2}}\right)^{k+1} + \frac{1}{3(\Sigma_t \Delta x)_i} \left(\Phi_{0,i+\frac{1}{2}} - \Phi_{0,i-\frac{1}{2}}\right)^{k+1} \\
&\quad + \frac{1}{2} \left\{[(1-c)\Sigma_t \Delta x]_{i+1} + [(1-c)\Sigma_t \Delta x]_i\right\} \Phi_{0,i+\frac{1}{2}}^{k+1} \\
&\quad = \frac{1}{2} \left[(S_0 \delta x)_{i+1} + (S_0 \Delta x)_i\right] - \frac{1}{3(\Sigma_t \Delta x)_{i+1}} \left(\Phi_{0,i+\frac{3}{2}} - \Phi_{0,i+\frac{1}{2}}\right)^{k+\frac{1}{2}} \\
&\quad + \frac{1}{3(\Sigma_t \Delta x)_i} \left(\Phi_{0,i+\frac{1}{2}} - \Phi_{0,i-1/2}\right)^{k+\frac{1}{2}} \qquad (196) \\
&\quad - \frac{1}{2} \left(J_{i+3/2} - J_{i-1/2}\right)^{k+1/2} + \frac{1}{2} \left\{[(1-c)\Sigma_t \Delta x]_{i+1}\right. \\
&\quad \left. + [(1-c)\Sigma_t \Delta x]_i\right\} \Phi_{0,i+\frac{1}{2}}^{k+\frac{1}{2}} - \frac{1}{2} \left\{[(1-c)\Sigma_t \Delta x]_{i+1} \Phi_{0,i+1}^{k+\frac{1}{2}}\right. \\
&\quad \left. + [(1-c)\Sigma_t \Delta x]_i \Phi_{0,i}^{k+\frac{1}{2}}\right\} \quad ,
\end{aligned}
$$

or

$$
\begin{aligned}
&- \frac{1}{3(\Sigma_t \Delta x)_{i+1}} \left(\Phi_{0,i+\frac{3}{2}} - \Phi_{0,i+\frac{1}{2}}\right)^{k+1} \\
&\quad + \frac{1}{3(\Sigma_t \Delta x)_i} \left(\Phi_{0,i+\frac{1}{2}} - \Phi_{0,i-\frac{1}{2}}\right)^{k+1} + \tilde{\Sigma}_{R,i+\frac{1}{2}}^{k+\frac{1}{2}} \Phi_{0,i+\frac{1}{2}}^{k+1} \qquad (197) \\
&\quad = \frac{1}{2} \left[(S_0 \Delta x)_{i+1} + (S_0 \Delta x)_i\right] - \frac{1}{3(\Sigma_t \Delta x)_{i+1}} \left(\Phi_{0,i+\frac{3}{2}} - \Phi_{0,i+\frac{1}{2}}\right)^{k+\frac{1}{2}} \\
&\quad + \frac{1}{3(\Sigma_t \Delta x)_i} \left(\Phi_{0,i+\frac{1}{2}} - \Phi_{0,i-\frac{1}{2}}\right)^{k+\frac{1}{2}} - \frac{1}{2} \left(J_{i+\frac{3}{2}} - J_{i-\frac{1}{2}}\right)^{k+\frac{1}{2}} \quad ,
\end{aligned}
$$

where

$$
\tilde{\Sigma}_{R,i+\frac{1}{2}}^{k+\frac{1}{2}} = \frac{1}{2} \left\{[(1-c)\Sigma_t \Delta x]_{i+1} \Phi_{0,i+1}^{k+\frac{1}{2}} + [(1-c)\Sigma_t \Delta x]_i \Phi_i^{k+\frac{1}{2}}\right\} \div \Phi_{0,i+\frac{1}{2}}^{k+\frac{1}{2}} \quad .
$$

Equation (196) is a linear form of a diffusion synthetic equation that is stable but loses its effectiveness in acceleration for large mesh spacing.

Equation (197), a nonlinear diffusion synthetic equation, cannot be mathematically analyzed as to its stability, but numerical results indicate that it has the same stability properties as Eq. (196). The stability properties of Eq. (196) are mathematically determined assuming diamond differencing using the same procedure as that leading to Eq. (193) and do not apply when negative flux fixup is used. Thus, the impact of using diamond differencing with negative flux fixup upon this formulation of the diffusion acceleration method is to introduce some nonlinear steps (steps that depend nonlinearly upon the transport solution) into the solution procedure. These nonlinear steps, along with the nonlinear aspects of the flux fixups, have been observed to limit the effectiveness of the acceleration, but stability is not impaired for the reasonable size meshes encountered in most reactor analysis. Even with this reduced acceleration effectiveness for large meshes, numerical experience has shown that the diffusion synthetic acceleration method is generally more effective than either coarse mesh rebalance or the Chebyshev method.

For general one- and two-dimensional geometry, it is convenient to use the nonlinear form of the acceleration equation, as indicated in Eq. (197), together with the relationship of Eq. (194). For two-dimensional problems, this type of procedure leads to a five-point acceleration equation[42] rather than a nine-point equation if a truly linear and consistent procedure were followed. The advantage of the five-point form is the ease and efficiency of solution and the availability of a large arsenal of diffusion equation solution methods. In the discussion of outer iteration methods presented next, note that for the diffusion synthetic method, a large fraction of the computation time in two-dimensional problems is spent solving the diffusion equation. Thus, efficient methods are important for this purpose.

D. Acceleration of the Outer Iterations

In Sec. III.A, we introduced the concept of source outer iteration in the multigroup formulation of the transport equation. The relevant equation is Eq. (51), in which two source processes are iterated: the fission source and the upscatter source. Also included is the possibility of eigenvalue determination when the inhomogeneous source Q_g, $g = 1, \cdots G$, is absent. The eigenvalue determination does not pose an additional problem because it is done during the source iteration process by the power method. This has come to be

standard in reactor analysis codes, whether based upon diffusion or transport theory.

In this section, we present three methods used in transport theory codes to accelerate the outer iteration procedure. These are based upon the Chebyshev polynomial method, the coarse mesh rebalance method, and the diffusion acceleration method. In this presentation, we will not go into much detail but will outline each of the methods and leave the details to the references.

1. Chebyshev Polynomial-Based Outer Iteration Acceleration.[45] In transport theory, the significant outer iteration processes are on the fission source and the upscatter source in the multigroup approximations. If we ignore upscatter for this discussion and concentrate on the fission problem, the Chebyshev method can be used effectively for the outer iterations. We write this problem from the transport equation as

$$\vec{\Omega} \cdot \vec{\nabla}\Phi_{m,g}^{k+\frac{1}{2}}(\vec{r}) + \Sigma_{t,g}(\vec{r})\Phi_{m,g}^{k+\frac{1}{2}}(\vec{r}) - \sum_{g'=1}^{g} \Sigma_{s,g'\to g}^{0}(\vec{r})\Phi_{0,g'}^{k+\frac{1}{2}}(\vec{r})$$

$$= \chi_g \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'}\Phi_{0,g'}^{k}(\vec{r}) + Q_{m,g}(\vec{r}) \quad , \tag{198}$$

where k is an iteration index.

In Eq. (198), we have assumed that we have converged the inner iteration process to the k+1/2 iterate in the scattering terms. We have also assumed, for convenience, isotropic processes in the scattering term although this is not necessary. If we define the fission source as

$$F^{k}(\vec{r}) \equiv \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'}\Phi_{0,g'}^{k}(\vec{r}) \quad , \tag{199}$$

and for the next iterate,

$$F^{k+\frac{1}{2}}(\vec{r}) = \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'}\Phi_{0,g'}^{k+\frac{1}{2}}(\vec{r}) \quad , \tag{200}$$

then Eqs. (198) and (200) constitute an iterative procedure for the fission source, $F(\vec{r})$, and this procedure can be accelerated. Details of this acceleration procedure for the diffusion equation are given in Ref. 45, and we summarize the results here. Instead of simply using Eq. (200), we update the fission source by the following equation:

$$F^{k+1}(\vec{r}) = F^{k}(\vec{r}) + \alpha_k[F^{k+\frac{1}{2}}(\vec{r}) - F^{k}(\vec{r})] + \beta_k[F^{k}(\vec{r}) - F^{k-1}(\vec{r})] \quad , \tag{201}$$

where the extrapolation parameters α and β are determined from the Chebyshev polynomial-based spectral fitting procedure. That is, we assume that the iteration of Eq. (198) has a matrix representation with an eigenvalue spectrum, $\lambda\varepsilon\{\lambda_i\}$, $i = 1, \cdots, I$, where I is the rank of the iteration matrix. If the eigenvalues are ordered from largest at $i = 1$ to smallest, then the spectral distribution of λ is represented by Chebyshev polynomials in the range $\{\lambda_2/\lambda_1, \lambda_3/\lambda_1, \cdots\}$. If we define $\bar{\rho} = \lambda_2/\lambda_1$, then α and β are defined by

$$\alpha_1 = \frac{2}{2 - \bar{\rho}} \quad , \quad \beta_1 = 0 \quad , \tag{202a}$$

$$\alpha_k = \frac{4}{\bar{\rho}} \frac{\cosh(k - 1)\gamma}{\cosh k\gamma} \quad , \tag{202b}$$

$$\beta_k = \left(1 - \frac{\bar{\rho}}{2}\right)\alpha_k - 1 \quad ,0 \tag{202c}$$

with

$$\gamma = \cosh^{-1}\left(\frac{2}{\bar{\rho}} - 1\right) \quad . \tag{202d}$$

Again, as in the inner iteration application of this method, $\bar{\rho}$ is estimated during the iteration process from the quantity

$$\bar{\rho} \simeq \lim_{k\to\infty} \sqrt{\frac{(F^k - F^{k-1}, F^k - F^{k-1})}{(F^{k-1} - F^{k-2}, F^{k-1} - F^{k-2})}} \quad , \tag{203}$$

where the notation (a,b) denotes the inner product of a and b.

The details of how this is implemented in a code and the considerations necessary for good estimates of ρ, which determine the quality of the parameters α and β, are explained in Ref. 45. Also in that reference, the modifications necessary to apply the procedure to eigenvalue problems is detailed. In this latter case, we obtain an acceleration of the power iteration method.

The attractive features of the Chebyshev polynomial-based acceleration method are 1) it is easy to apply with a minimum of coding effort, and 2) it is stable as long as any inner iterations are sufficiently converged. Unfortunately, the effectiveness of this procedure is limited, especially for eigenvalue problems, as the system becomes large. The accelerated spectral radius depends upon the unaccelerated spectral radius of convergence roughly as the square. Thus, as the unaccelerated spectral radius of convergence approaches unity, so does the Chebyshev polynomial-based accelerated spectral radius. (We derive the unaccelerated spectral for a model problem in the outer diffusion synthetic acceleration section below.) To improve upon this performance for transport equation outer iterations, we must apply more of the physics of the iteration process. The next two methods discussed do this to some degree and generally result in both theoretical and computational improvement in the iteration convergence rate.

2. Coarse Mesh Rebalancing of the Outer Iterations.[31,46,47] By coarse mesh rebalancing, we refer to the general procedure outlined in the previous section on inner iterations. Below, we describe some of the many ways to apply this technique to the outer iterations. What is attempted in each of these ways is to bring the multigroup system into balance at each step of the iteration procedure.

One of the simplest methods is called whole-system, groupwise rebalance, in which we integrate each energy group equation over the entire spatial domain of the system and seek the multigroup factors that will bring the multigroup equations into balance. To develop the appropriate rebalance equations, we take the multigroup equation, Eq. (51), and integrate over angle to obtain the following equation:

$$\vec{\nabla} \cdot \vec{J}_g^{k+\frac{1}{2}}(\vec{r}) + \Sigma_{t,g}(\vec{r})\phi_{0,g}^{k+\frac{1}{2}}(\vec{r})$$

$$= \chi_g \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'}\phi_{0,g'}^{k}(\vec{r}) + \sum_{g'=1}^{g} \Sigma_{s,g'\to g}^{0}\phi_{0,g'}^{k+\frac{1}{2}}(\vec{r}) \qquad (204)$$

$$+ \sum_{g'=g+1}^{G} \Sigma_{s,g'\to g}^{0}\phi_{0,g'}^{k}(\vec{r}) + Q_{0,g}(\vec{r}) \quad .$$

We now integrate over space and employ rebalance factors to obtain

$$L_g^{k+\frac{1}{2}} f_g^{k+1} = \chi_g \sum_{g'=1}^{G} (\nu\Sigma_f\phi)_{g'}^{k+\frac{1}{2}} f_{g'}^{k+1}$$

$$+ \sum_{\substack{g'=1 \\ g'\neq g}}^{G} (\Sigma_s\phi)_{g'\to g}^{k+\frac{1}{2}} f_{g'}^{k+1} + \bar{Q}_{0,g} \quad , \qquad (205)$$

where

$$L_g^{k+\frac{1}{2}} = \int_R [\nabla \cdot J_g^{k+\frac{1}{2}}(\vec{r}) + (\Sigma_t - \Sigma_{s,g\to g}^{0})\phi_0^{k+\frac{1}{2}}(\vec{r})]d\vec{r} \quad , \qquad (206a)$$

$$\overline{(\nu\Sigma_f\phi)}_g^{k+\frac{1}{2}} = \int_R (\nu\Sigma_f)_g\phi_{0,g}^{k+\frac{1}{2}}(\vec{r})d\vec{r} \quad , \qquad (206b)$$

$$(\Sigma_s\phi)_{g'\to g}^{k+\frac{1}{2}} = \int_R \Sigma_{s,g'\to g}^{0}(\vec{r})\phi_{0,g'}^{k+\frac{1}{2}}(\vec{r})d\vec{r} \quad , \qquad (206c)$$

$$\bar{Q}_{0,g} = \int_R Q_{0,g}(\vec{r})d\vec{r} \quad , \qquad (206d)$$

and

R = the spatial domain of the system.

We can rewrite Eq. (205) as the matrix equation for the f_g as

$$Mf = Q \quad , \tag{207}$$

where

M = a G×G matrix,
f = the vector of the f_g,
Q = the vector of Q_g,

and solve it using some matrix solver routine. Once we obtain the rebalance of factors f_g, g = 1, ···, G, we define our updated scalar flux as

$$\Phi_{0,g}^{k+1}(\vec{r}) = \Phi_{0,g}^{k+\frac{1}{2}}(\vec{r}) f_g^{k+1} \tag{208}$$

and proceed. Actually, all angular moments can be updated as in Eq. (208), but this has not been found to be useful.

We note that this type of rebalance, which concentrates on the energy spectral details of the problem, is not well suited for k_{eff} eigenvalue problems. This is because we usually solve the eigenvalue problem by the power iteration method described in Sec.III.A, which treats the fission term of Eq. (200) as a given source. [Recall that Eqs. (198) and (204) can be cast in the k_{eff} eigenvalue form by replacing χ_g with χ_g/k_{eff} and by setting $Q_{m,g}(\vec{r})$ to zero.] Rebalance in this case will be effective only if there is significant upscatter since the downscatter is solved without iteration over the groups. Of course, a matrix eigenvalue problem similar to Eq. (207) can be posed where the matrix solver could obtain both the eigenvalues and eigenvectors of M. However, this is not as effective as estimating the eigenvalue from the original Eq. (204), and, except for the first few iterations, has been found to be largely wasted effort.[48]

A rebalance method that has been applied to the eigenvalue problem with some success is described next. This method, which we call group-collapsed coarse mesh outer rebalance, takes the coarse mesh rebalance equations of Sec.

IV.C, Eqs. (168) and (169), and simply sums them over the groups. Thus, we have the following:

$$
\begin{aligned}
&- \overline{FL}_{I+\frac{1}{2},J} f_{I+1,J} + \overline{FL}_{I-\frac{1}{2},J} f_{I,J} - \overline{FR}_{I-\frac{1}{2},J} f_{I-1,J} + \overline{FR}_{I+\frac{1}{2},J} f_{I,J} \\
&- \overline{FD}_{IJ+\frac{1}{2}} f_{I,J+1} + \overline{FD}_{I,J-\frac{1}{2}} f_{I,J} - \overline{FU}_{I,J-\frac{1}{2}} f_{I,J-1} + \overline{FU}_{I,J+\frac{1}{2}} f_{I,J} \qquad (209) \\
&+ \overline{ABT}_{I,J} = \frac{1}{k_{eff}} \overline{FT}_{I,J} + \overline{QQ} \quad ,
\end{aligned}
$$

where

$$
\overline{FL}_{I+\frac{1}{2},J} = \sum_{g=1}^{G} FL^{g}_{I+\frac{1}{2},J} \quad , \qquad (210a)
$$

$$
\overline{FR}_{I+\frac{1}{2},J} = \sum_{g=1}^{G} FR^{g}_{I+\frac{1}{2},J} \quad , \qquad (210b)
$$

$$
\overline{FD}_{I,J+\frac{1}{2}} = \sum_{g=1}^{G} FD^{g}_{I,J+\frac{1}{2}} \quad , \qquad (210c)
$$

$$
\overline{FU}_{I,J+\frac{1}{2}} = \sum_{g=1}^{G} FU^{g}_{I,J+\frac{1}{2}} \quad , \qquad (210d)
$$

$$
\overline{ABT}_{I,J} = \sum_{i \in I} \sum_{j \in J} \sum_{g=1}^{G} \left(\Sigma_{t,g} - \sum_{g'=1}^{G} \Sigma_{s,g' \to g} \right)_{i,j} V_{i,j} \Phi_{0,i,j,g} \quad , \qquad (210e)
$$

$$\overline{FT}_{I,J} = \sum_{i\varepsilon I}\sum_{j\varepsilon J}\sum_{g=1}^{G} (\nu\Sigma_f)_g \Phi_{0,i,j,g} V_{i,j} \quad , \tag{210f}$$

$$\overline{QQ} = \sum_{i\varepsilon I}\sum_{j\varepsilon J}\sum_{g=1}^{G} (Q_{0,g} V)_{i,j} \quad . \tag{210g}$$

We then update the flux iterate for Eq. (204) as

$$\Phi_{0,i,j,g}^{k+1} = \Phi_{0,i,j,g}^{k+\frac{1}{2}} f_{I,J}^{k+1} \quad \text{for } i\varepsilon I \quad , j\varepsilon J \quad , g = 1, \cdots, G \quad . \tag{211}$$

We note that in this method we have lost the bulk of our (energy) spectral information in the acceleration equation and we have added more spatial detail. Thus, this method is best applied to situations where the transport solution for $\Phi_{0,g}^{k+\frac{1}{2}}$ contains most of the spectral information, as is the case for purely downscatter problems. Then this outer iteration rebalance method is useful in helping to converge the fission source that appears as a sum over groups. On the other hand, this method is not very useful for problems with significant upscatter because there the spectral information is important. Returning to Eq. (209) and the fission problem, we note that when $\overline{QQ} = 0$, we can solve for the eigenvalue, k_{eff}, by power iteration. This is useful, at least in the early stages of transport iteration, in that one quickly obtains a good estimate of the system eigenvalue. It is not so useful in later outer iterations because there is a spectral effect that is more efficiently obtained from the unaccelerated outer iteration than from Eq. (209).[31]

In summary, we have presented two outer rebalance methods that are reasonably cheap computationally and, to some extent, are effective, at least in the early outer iterations. The method represented by Eq. (205) focuses on accounting for the energy spectral information in performing the rebalance; that represented by Eq. (209) focuses more on spatial variations.

One can derive rebalance equations that account for both the spectral and the spatial effects at the outer iteration stage. In developing this method, we again use the definitions of Eq. (169) of Sec. IV.C to write the following groupwise, coarse-mesh, rebalance equation:

$$
\begin{aligned}
&-FL_{g,I+\frac{1}{2},J} f_{g,I+1,J} + FL_{g,I-\frac{1}{2},J} f_{g,I,J} - FR_{g,I-\frac{1}{2},J} f_{g,I-1,J} \\
&+ FR_{g,I+\frac{1}{2},J} f_{g,I,J} - FD_{g,I,J+\frac{1}{2}} f_{g,I,J+1} + FD_{g,I,J-\frac{1}{2}} f_{g,I,J} \\
&- FU_{g,I,J-\frac{1}{2}} f_{g,I,J-1} + FU_{g,I,J+\frac{1}{2}} f_{g,I,J} + \Sigma_{R,g,I,J} f_{g,I,J} \qquad (212) \\
&= \chi_g \sum_{g'=1}^{G} (\overline{\nu\Sigma_f \Phi_0})_{g',I,J} f_{g',I,J} \\
&+ \sum_{\substack{g'=1 \\ g' \neq g}}^{G} (\overline{\Sigma_s \Phi_0})_{g' \to g,I,J} f_{g',I,J} + \bar{Q}_{0,g,I,J} \ ,
\end{aligned}
$$

where we solve this at iterate k+1.

Quantities with bars are defined in Eq. (206), with the spatial integrations performed only over the appropriate coarse mesh intervals implied by the subscripts I,J. As before, to cast Eq. (212) in the k_{eff} eigenvalue form, simply replace χ_g with χ_g/k_{eff} and set $\bar{Q}_{0,g,I,J} = 0$.

Equation (212) itself must be solved for the rebalance factors, $f_{g,I,J}$, by an iterative process before the rebalance factors can be applied to accelerate the transport outer iterations. Thus, this groupwise, coarse mesh rebalance method can be computationally much more expensive than the other two methods above. Although groupwise, coarse mesh rebalance contains more information than the other two methods and should thus be more effective, its complexity has caused it to be used little in reactor analysis codes. Further, the method is not being actively pursued because a more predictably effective diffusion synthetic method has been developed. It is described next.

3. Diffusion Synthetic Acceleration of the Outer Iterations. As with rebalance, The diffusion synthetic acceleration (DSA) method is readily applied to the outer iteration of the transport equation. The DSA method is also amenable to some analysis on a model problem that can help determine the best way to apply the method to real problems. The attractive aspect of using DSA over coarse mesh rebalance is that it uses the familiar diffusion equation. As we show, this is much more convenient in an outer iteration acceleration scheme because the solution methods for the multigroup diffusion equation are well

developed and generally well understood by the reactor analyst. To demonstrate our procedure, we write the outer iteration DSA equation from Eq. (176) as

$$-\nabla \cdot D_g \nabla \Phi_{0,g}^{k+1}(\vec{r}) + (\Sigma_t - \Sigma_{s,g\to g}^0)\Phi_{0,g}^{k+1}(\vec{r})$$

$$= \chi_g \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'} \Phi_{0,g'}^{k+1}(\vec{r}) + \sum_{\substack{g'=1 \\ g'\neq g}} \Sigma_{s,g'\to g}^0 \Phi_{0,g'}^{k+1}(\vec{r}) \qquad (213)$$

$$+ Q_{0,g}(\vec{r}) - [\nabla \cdot D_g \nabla \Phi_{0,g}^{k+\frac{1}{2}}(\vec{r}) + \vec{\nabla} \cdot \vec{J}_g^{k+\frac{1}{2}}(\vec{r})] \quad ,$$

$$g = 1, \cdots, G \quad .$$

This is a multigroup difusion equation with a source correction that comes from the previous transport solution, designated with iterate $k+\frac{1}{2}$. For eigenvalue problems [$Q_{0,g}(\vec{r}) = 0$ and χ_g replaced by χ_g/k_{eff}], this equation is modified by changing the diffusion coefficient to a diagonal tensor with the following components:

$$[\underline{\underline{D}}_g]_{\alpha,\beta}^{k+\frac{1}{2}} = -\left(\frac{J_{g,\alpha}^{k+\frac{1}{2}}}{\nabla_\alpha \Phi_0^{k+\frac{1}{2}}}\right)\delta_{\alpha,\beta} \quad , \quad \begin{cases} \alpha = 1, 2, 3 \\ \beta = 1, 2, 3 \end{cases} ; \qquad (214)$$

thus, the right-side correction is zero, and the diffusion equation is homogeneous but nonlinear.

To demonstrate how Eq. (213) is used in the outer iteration scheme, we write the companion transport equation as

$$\vec{\Omega} \cdot \vec{\nabla} \Phi_{m,g}^{k+\frac{1}{2}}(\vec{r}) + \Sigma_{t,g}(\vec{r})\Phi_{m,g}^{k+\frac{1}{2}}(\vec{r})$$

$$= \Sigma_{s,g\to g}^0(\vec{r})\Phi_{0,g}^{k}(\vec{r}) + \sum_{n=2}^{NM} (2\ell+1)\Sigma_{s,g\to g}^{\ell}(\vec{r}) R_n(\vec{\Omega}_m)\tilde{\Phi}_{n,g}^{k}(\vec{r}) \qquad (215)$$

$$+ S_{m,g}(\vec{r}) \quad ,$$

where we have used the spherical harmonics expansion form as described in Sec. II.D for the within-group scattering source, and $S_{mg}(\vec{r})$ represents the source from fission and scattering into the group g. In the above transport equation, we assume that only the scalar flux, $\Phi_{0,g}(\vec{r})$, is updated from the diffusion equation, and the anisotropic scattering components come from the previous transport angular flux. With Eqs. (213) and (215), the following iteration strategy is used:

a. At k = 0, solve the diffusion equation, Eq. (213), with the correction terms zero, for $\Phi_{0,g}(\vec{r})$ to some desired order of convergence.
b. With the $\Phi_{0,g}^{k=0}(\vec{r})$, solve the transport equation, Eq. (215), for $\Phi_{m,g}^{k=\frac{1}{2}}(\vec{r})$, using the most up-to-date information for the scattering part of $S_{m,g}$.
c. With this $\Phi_{m,g}^{k+\frac{1}{2}}$ compute the correction term (with the diffusion tensor or the source correction) and solve the one-group diffusion synthetic equation, Eq. (177), for the scalar fluxes in each group g.
d. With the same correction term, set up the multigroup diffusion equation, Eq. (213), and solve it for $\Phi_{0,g}^{k=1}(\vec{r})$, g = 1, ···, G.
e. Repeat the outer iteration process [steps (b) through (d)] until the source terms of Eq. (213) converge to a desired precision from one outer iteration to the next.
f. With this converged source, $S_{m,g}(\vec{r})$, iterate the energy group transport equation, Eq. (215), on the within-group scatter source to the precision desired for the convergence of the scalar fluxes $\Phi_{0,g}^{k+\frac{1}{2}}(\vec{r})$ using the DSA inner iteration process.
g. With this final converged correction term, set up and solve the "final" version of Eq. (213) and check to see whether the source is still converged to the desired precision; if not, go back and repeat steps (f) and (g).

The essential feature of the procedure is that one inner iteration [steps (b) and (c)] per outer iteration is performed until the multigroup source has been converged. Then the group transport scalar fluxes are converged to their convergence criterion via the DSA inner iteration of Sec. IV.C. The success of this strategy depends upon the inner iteration steps (b) and (c) being stable

and convergent and the outer iteration equation being stable and convergent. It has been shown that the first part of this is true in the discussion of Sec. IV.C. Here we will indicate the stability and effectiveness of the outer iteration DSA for a model problem.

To demonstrate the effectiveness of the outer iteration diffusion synthetic acceleration procedure, we consider a problem without inner iteration, that is, a fission problem with no scattering. This simplifies the analysis while preserving the essential features of the acceleration procedure.* Thus, our DSA procedure is reduced to the following equations:

$$\vec{\Omega} \cdot \vec{\nabla}\Phi_{m,g}^{k+\frac{1}{2}}(\vec{r}) + \Sigma_{t,g}(\vec{r})\Phi_{m,g}^{k+\frac{1}{2}}(\vec{r}) = \chi_g F^k(\vec{r}) + Q_{m,g}(\vec{r}) \quad , \tag{216}$$

$$\begin{aligned} -\nabla \cdot D_g(\vec{r})\nabla\Phi_{0,g}^{k+1}(\vec{r}) + \Sigma_{t,g}(\vec{r})\Phi_{0,g}^{k+1}(\vec{r}) \\ = \chi_g F^{k+1}(\vec{r}) + Q_{0,g}(\vec{r}) - [\nabla \cdot D_g \nabla\Phi_{0,g}^{k+\frac{1}{2}}(\vec{r}) + \vec{\nabla} \cdot \vec{J}_g^{k+\frac{1}{2}}(\vec{r})] \quad , \end{aligned} \tag{217}$$

$$F^k(\vec{r}) = \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'}\Phi_{0,g'}^k(\vec{r}) \quad . \tag{218}$$

The acceleration equation, Eq. (217), can be put into the more convenient form,

$$\begin{aligned} -\nabla \cdot D_g \nabla f_{0,g}^{k+1} + \Sigma_{t,g} f_{0,g}^{k+1} - \chi_g \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'} f_{0,g'}^{k+1} \\ = \chi_g (F^{k+\frac{1}{2}} - F^k) \quad , \end{aligned} \tag{219}$$

where

$$f_{0,g}^{k+1} = \Phi_{0,g}^{k+1} - \Phi_{0,g}^{k+\frac{1}{2}} \tag{220a}$$

*This development was accomplished by E. W. Larsen.

and where F^{k+1} is now defined as

$$F^{k+1} = F^{k+\frac{1}{2}} + \sum_{g'=1}^{G} (\nu\Sigma_f)_{g'} f_{0,g'}^{k+1} \quad . \tag{220b}$$

To do error analysis, we then take Eqs. (216), (219), (220a), and (220b) and perform a Fourier analysis assuming an infinite medium and constant cross sections. We quote the result for the iteration spectrum of convergence

$$\omega(\lambda) = \frac{\sum_{g=1}^{G} \frac{[(\lambda^2 + 3\Sigma_{t,g}^2)(\nu\Sigma_f)_g(\frac{1}{\lambda}) \tan^{-1}(\frac{\lambda}{\Sigma_{t,g}}) - 3\Sigma_{t,g}(\nu\Sigma_f)_g]\chi_g}{\lambda^2 + 3\Sigma_{t,g}^2}}{\sum_{g=1}^{G} \frac{\{\lambda^2 + 3\Sigma_{t,g}[\Sigma_{t,g} - (\nu\Sigma_f)_g]\}\chi_g}{\lambda^2 + 3\Sigma_{t,g}^2}} \quad . \tag{221}$$

The accelerated iteration spectral radius of convergence, $\bar{\rho}$, is given by

$$\bar{\rho} = \sup_{\lambda} \omega(\lambda) \quad , \tag{222}$$

and the unaccelerated iteration spectrum, $\omega_0(\lambda)$, is

$$\omega_0(\lambda) = \sum_{g=1}^{G} \chi_g \left[\frac{(\nu\Sigma_f)_g}{\Sigma_{t,g}}\right] \left[\frac{\Sigma_{t,g}}{\lambda} \tan^{-1}(\frac{\lambda}{\Sigma_{t,g}})\right] \quad , \tag{223}$$

so the spectral radius for the unaccelerated iteration, $\bar{\rho}_0$, is

$$\bar{\rho}_0 = \sup_{\lambda} \omega_0(\lambda) \quad , \tag{224}$$

where λ in the above is the Fourier variable. Thus, we have at $\lambda = 0$,

$$\bar{\rho}_0 = \sum_{g=1}^{G} \chi_g \frac{(\nu\Sigma_f)_g}{\Sigma_{t,g}} \quad ,$$

and since

$$\sum_g \chi_g = 1 \quad ,$$

we see that $(\nu\Sigma_f)_g/\Sigma_{t,g} \leq 1$ is sufficient but not necessary for convergence.

For the accelerated iteration procedure, we note that $\omega(0) = 0$ from Eq. (221). Thus, the acceleration method has suppressed the worst mode of the unaccelerated iteration. To be more specific as to the accelerated spectral radius, we have to specify the cross-section values. We present some representative data in Table XIII and in Fig. 15 and note that the diffusion synthetic acceleration of the outers is, indeed, very effective for this case; that is, we have reduced the spectral radius from 1.0 to 0.225 by the acceleration.

As in Sec. IV.C, for the inner iteration, we must ask whether this analysis is valid for "real" problems with finite spatial dimensions, discontinuous cross sections, and scattering. Numerical experience[42] has indicated that the analysis is, indeed, a good guide as to the expected convergence of the outer iteration DSA. Actually, much of the superior performance of codes using the DSA method over those using the Chebyshev polynomial-based acceleration or the rebalance algorithm is due to this excellent acceleration of the outer iterations.

To illustrate the effectiveness of the various acceleration methods, we present in Table XIV some representative one-dimensional calculations of multigroup systems. Three systems have been chosen, which we briefly describe as

1) HTGR - a high temperature gas reactor eigenvalue (k_{eff}) problem with 9 energy groups, including 3 upscatter groups, 87 spatial mesh points, and S_4 discrete ordinates quadrature in cylindrical geometry.
2) SARAF - A test reactor eigenvalue (k_{eff}) problem for fast reactor applications consisting of a thermal driver and a filtered experimental test section. This problem was run with 20 energy groups, 39 spatial mesh points, and S_8 discrete ordinates quadrature in cylindrical geometry.

TABLE XIII

THIRTY-GROUP DATA USED TO DETERMINE THE SPECTRA IN FIG. 15

Group (g)	χ_g	$\Sigma_{t,g}$	$\nu\Sigma_{f,g}/\Sigma_{t,g}$
1	3.04800E-05	8.82000E-03	1.00000E+00
2	7.94300E-05	5.15600E-03	1.00000E+00
3	2.36900E-04	5.32000E-03	1.00000E+00
4	1.16000E-03	1.01260E-02	1.00000E+00
5	5.87500E-03	7.61100E-03	1.00000E+00
6	1.75200E-02	1.05800E-02	1.00000E+00
7	1.02700E-01	1.43400E-02	1.00000E+00
8	9.06600E-02	1.44650E-02	1.00000E+00
9	1.08100E-01	1.56900E-02	1.00000E+00
10	1.14700E-01	1.61900E-02	1.00000E+00
11	1.10800E-01	1.57500E-02	1.00000E+00
12	1.81912E-01	1.56700E-02	1.00000E+00
13	1.20400E-01	1.60400E-02	1.00000E+00
14	7.06100E-02	1.74600E-02	1.00000E+00
15	3.73700E-02	2.02100E-02	1.00000E+00
16	2.88300E-02	2.48500E-02	1.00000E+00
17	6.98200E-03	3.38400E-02	1.00000E+00
18	1.56100E-03	4.40300E-02	1.00000E+00
19	3.74000E-04	6.15100E-02	1.00000E+00
20	7.61700E-05	9.27300E-02	1.00000E+00
21	1.76800E-05	1.51550E-01	1.00000E+00
22	3.81000E-06	3.10130E-01	1.00000E+00
23	8.50300E-07	4.85500E-01	1.00000E+00
24	1.89500E-07	1.00220E+00	1.00000E+00
25	4.23100E-08	1.03510E+00	1.00000E+00
26	9.45500E-09	6.43700E-01	1.00000E+00
27	2.10500E-09	3.62200E-01	1.00000E+00
28	4.74100E-10	9.94600E-01	1.00000E+00
29	1.05000E-10	2.50440E+00	1.00000E+00
30	3.03500E-11	7.47940E+00	1.00000E+00

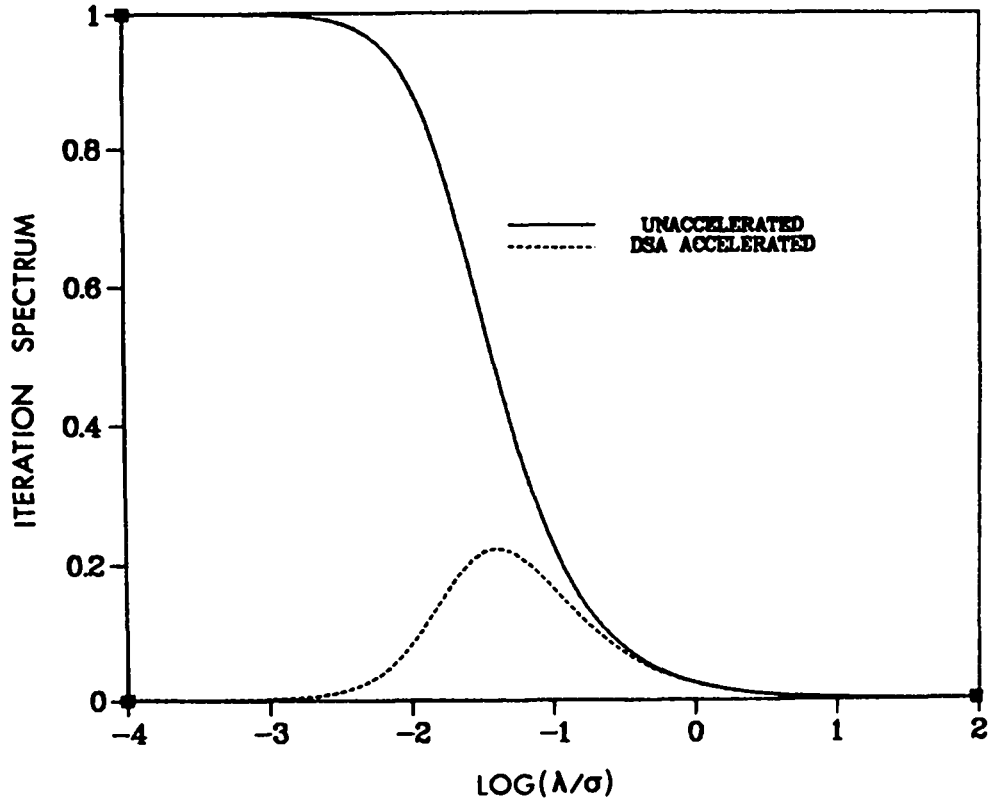


Fig. 15. Iteration spectra for 30-energy group problem.

TABLE XIV

TOTAL NUMBER OF TRANSPORT ITERATIONS TO CONVERGE THE POINTWISE FLUX FOR VARIOUS ACCELERATION METHODS

Problem	CHEBY	CMR	DSA	Time Ratio
1. (HTGR)	12000	6400	124	0.08
2. (SARAF)	5047	*	188	0.06
3. (KTST)	--	*	200	--

* - divergence.
-- - results not available.

3. KTST - A two-group critical size search problem in slab geometry with upscatter. This is a mockup of a water reactor with a water reflector and was run with S_{16} discrete ordinates quadrature.

In Table XIV, we present the number of iterations required for each of the three accelertion methods; CHEBY = Chebyshev, CMR = coarse mesh rebalance, and DSA = diffusion synthetic. The time ratio is that of the DSA time to the best achieved by either of the two other methods.

We note that in this selection of eigenvalue problems, DSA is clearly superior; also, the coarse mesh rebalance algorithm can be quite unpredictable, resulting in nonconvergence of the iteration.

Finally, we comment upon the computational effectiveness of the iteration strategy outlined in steps (a) through (g) a few pages earlier. We note that with one inner iteration per outer iteration in the transport sweep of the acceleration procedure, we are, in fact, replacing transport sweeps in the outer iteration with diffusion sweeps. Granted that the spectral radius of convergence for this is excellent, good computational effectiveness of the procedure depends upon an efficient solution of the diffusion equation. That is, in Eq. (213), we are asking for the converged diffusion solution at iterate $k + 1$; but to obtain this, we must iterate the multigroup solution for the fission source. To complicate matters further, in two or three dimensions, there is an inner iteration for each group of the diffusion solution. This inner iteration is necessary for the efficient inversion of the diffusion operator. The entire series of iterations for the diffusion solution can be computationally expensive; hence, it may abrogate the excellent theoretical performance of the transport DSA procedure. Fortunately, a great deal of work has been expended in the past upon efficient diffusion solvers. Indeed, when these are employed, good computational performance brought about by DSA procedure is generally seen.[49]

Another facet of the DSA method is that it can be employed as a diffusion improvement method for many eigenvalue calculations. In many cases, an estimate is wanted of the transport effects upon the eigenvalue and power distributions as computed using diffusion theory. This estimate can be made from the DSA iteration strategy by terminating the overall procedure, described by steps (a)-(g) above, before full transport convergence is achieved. Note that steps (b)-(d) are concerned with converging the multigroup source to its

final transport value before the final pointwise fluxes are converged. Generally in this procedure, the eigenvalue converges first, followed by convergence of the pointwise fission source, followed by the convergence of the pointwise fluxes. Thus, depending on the particular need, the iterations can be halted when the eigenvalue is well estimated but before convergence of pointwise quantities, and the user obtains a good estimate of the transport eigenvalue. In fact, even though in this example the pointwise fission source and the pointwise fluxes are not fully converged, they can still be considered as being intermediate between pure diffusion results and fully converged transport results. This feature of the DSA iteration strategy results in a very flexible computational tool that can be used to produce results ranging from pure diffusion theory, through improved (by transport corrections) diffusion, all the way to fully converged transport theory. With such a tool, users can greatly reduce computational times on problems where fully converged transport solutions are not necessary, but where "better than pure diffusion" results are desired.

In summary, although many forms of outer iteration acceleration of the transport solution have been employed in codes, the diffusion synthetic accelertion method has been the most effective, both theoretically and computationally, for a large variety of reactor analysis problems.

E. Search Capabilities in Discrete Ordinates Codes

In many general-purpose, discrete ordinates codes, it is possible to perform eigenvalue search calculations. In these search calculations, certain problem parameters, for example, dimensions or material concentrations, are automatically adjusted to values that produce a desired value of k_{eff}. The basic quantity used to alter the problem parameters is the eigenvalue of the specific calculation. It is important to note that the term eigenvalue in a search calculation takes on a meaning different from an ordinary k_{eff} calculation. In the k_{eff} calculation, the term "eigenvalue" simply refers to the quantity that approaches the value of k_{eff} as the iterative calculation is converged. In search calculations, however, the "eigenvalue" is a quantity used directly to alter the parameter (or parameters) being searched on. The specific manner in which this is done is described below.

1. Types of Searches. The most common types of eigenvalue searches are the buckling search, the time-absorption (α) search, the spatial-dimension search, and the concentration search.

In a buckling search, the parameter to be determined in the search is the geometric buckling, B^2. For example, in a two-dimensional (x,y) geometry calculation, the z-direction leakage of particles can be simulated by adding the term DB^2 as an effective absorption term in the transport equation. The quantity D is the diffusion coefficient, which is normally space- and energy-dependent. The value of B^2 is typically altered by means of the expression

$$B^2 = EV*B_i^2 \quad , \tag{225}$$

where EV is the search eigenvalue and B_i^2 is a user-input buckling value, that is, an initial geometric buckling guess.

For a time-absorption, or α, search the time-dependent angular flux is assumed to be separable in time with respect to space, energy, and angle, viz.,

$$\Phi(\vec{r},E,\vec{\Omega},t) = \Phi(\vec{r},E,\vec{\Omega})e^{\alpha t} \quad . \tag{226}$$

If this separable form is inserted into the time-dependent transport equation, the exponential time dependence can be cancelled, and a fictitious cross-section term of the form α/v_g appears as a "time-absorption" correction to the total and absorption cross sections. Here, v_g is the particle speed associated with energy group g. The exponent α is the eigenvalue, EV, sought in a time-absorption, α, search. Obviously, $\alpha = 0$ for an exactly critical system and $\alpha > 0$ corresponds to a supercritical system.

In spatial-dimension searches, the dimensions of selected portions of the problem model are adjusted to achieve the desired value of k_{eff}. Although various prescriptions for adjusting the dimensions are used, the following generic form displays the basic method. Let Δd_k denote the dimension of the k-th region in a problem model. For two-dimensional geometries, this dimension represents, in a general sense, either the height or width of a particular region. The dimensions are altered by the general formula

$$\Delta d_k = \Delta d_k^i * [1 + f_k * EV] \quad , \tag{227}$$

where Δd_k^i is the "as input" dimension of the k-th region and EV is the dimension search eigenvalue. The term f_k represents a user-supplied quantity that permits expansion, contracton, or no alteration of the dimensions for each region in the problem. For example, f_k can be selected such that one region can be expanded while an adjacent region is contracted such that the total dimension of the two regions remains constant.

With concentration searches, the user can selectively determine the nuclide concentrations that will produce the desired value of k_{eff}. Specific formulations for varying the concentrations differ among the various computer codes, but all are typified by

$$C_\ell = C_\ell^i * [1 + f_\ell * EV] \quad , \tag{228}$$

where C_ℓ is the adjusted concentration for nuclide (or material) ℓ, C_ℓ^i is the user-input value of C_ℓ, and EV is the concentration search eigenvalue. The term f_ℓ represents a user-supplied quantity that permits an increase, a decrease, or no alteration in the concentration. With such a formulation, for example, uranium enrichment searches can be performed such that as the atom density of U-235 is increased, the atom density of U-238 is decreased so the total atom density of uranium remains constant.

2. Overall Search Strategy. Regardless of the type of search being conducted or the computer code being used, the following search strategy is generally used.[50] The search is executed by performing a sequence of k_{eff}-type calculations, each for a different value of the search eigenvalue. The search is for a value of the eigenvalue that makes the value of λ unity, where λ is defined as

$$\lambda \equiv \frac{(\text{Fission Source})^j}{(\text{Fission Source})^{j-1}} \tag{229}$$

for the j^{th} outer iteration.

In the following description of the search strategy, it is helpful to refer to Fig. 16, in which the deviation of λ from unity for each outer iteration is plotted.

The user provides input values and data to define the initial system. An input value for the initial eigenvalue is included. For this initial system,

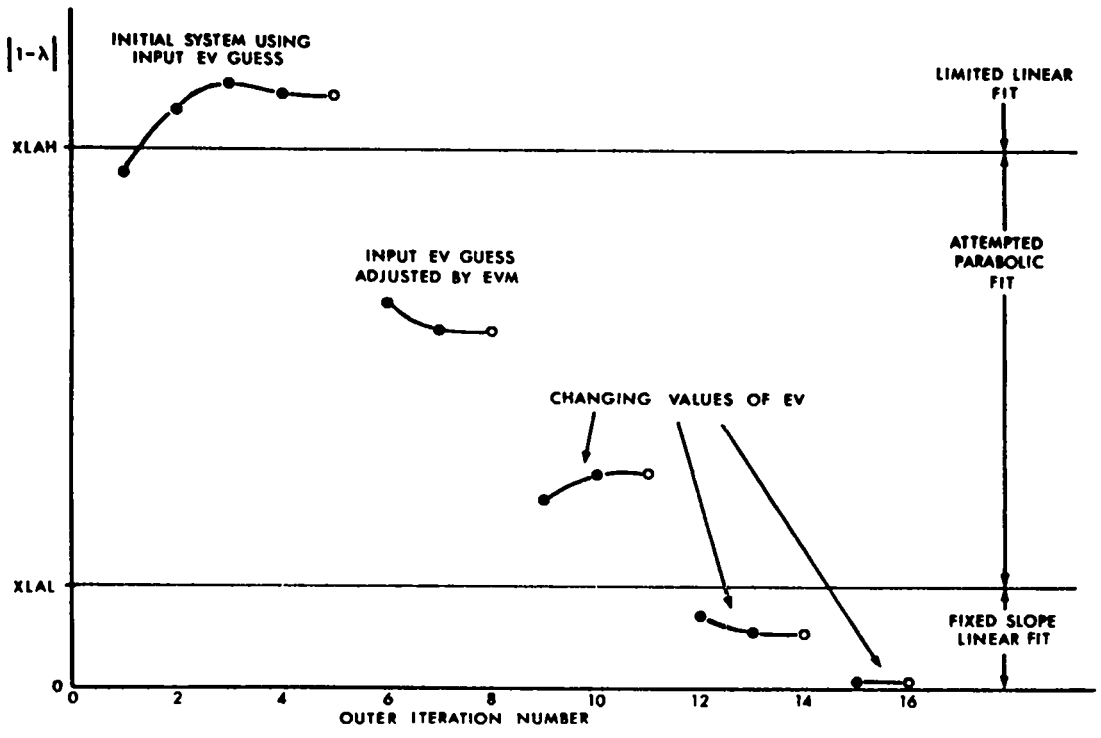


Fig. 16. Variation in λ during a hypothetical eigenvalue search.

outer iterations are performed until two successive values of λ differ by less tha= some user-controlled convergence criterion. After the first converged λ sequence is obtained, the initial value of the eigenvalue (EV) is altered by a user-supplied quantity denoted here by EVM. If $\lambda > 1$ (multiplying system), the new eigenvalue is equal to EV(initial) + EVM; if $\lambda < 1$ (decaying system), the new eigenvalue is EV(initial) + EVM. Thus, the sign and value of EVM should be chosen such that the use of EV(initial) + EVM will reduce the reactivity of the system. Conversely, the use of EV(initial) - EVM should increase the reactivity of the system.

Basically, after two converged values of λ are obtained for two values of the search eigenvalue, the computer code will effectively make a plot of the converged value of λ as a function of the search eigenvalue. A curve fit extrapolated to a value of $\lambda = 1$ is used to predict the next value of the search eigenvalue. Depending on the amount of information available and the size of $|1 - \lambda|$, this curve fit proceeds in different ways. A parabolic fit cannot be made until three converged values of λ are available for three different values of EV. Even then a parabolic fit is not attempted unless $|1 - \lambda|$ is greater than an input search lower limit (here denoted as XLAL) and

less than an input search upper limit (here denoted XLAH). If a parabolic fit is tried and the roots are imaginary, a straight line fit is used. If the roots are not imaginary, the closest root is used for the new value of EV. Once a bracket is obtained (change of sign of $\lambda - 1$), the fit procedure is not allowed to move outside the region of the bracket. Should a parabolic fit predict an eigenvalue outside the bracket region, this value is rejected and the new eigenvalue is taken as one-half the sum of the two previous eigenvalues.

Whenever a parabolic fit is not used, a linear fit is used, and the new eigenvalue is computed from

$$(EV)_{new} = (EV)_{previous} + POD*EVS*(1 - \lambda) \quad , \tag{230}$$

where POD is an input "parameter oscillation damper" that can be used to restrict the amount of change in the eigenvalue and EVS is a measure of the slope of the λ versus EV curve. When $|1 - \lambda| >$ XLAH, $(1 - \lambda)$ in Eq. (230) is replaced by XLAH (with the correct sign) to prevent too large a change in the predicted new eigenvalue. After $|1 - \lambda|$ is less than XLAL, the value of EVS is fixed and kept constant until convergence to prevent numerical difficulty in the approximation of the slope when λ is close to unity.

Because parametric search problems represent sequences of k_{eff} calculations, it is important that the user study this search strategy in order to optimize his calculations. It is also important that the user pose soluble problems. That is, there are many problems for which solutions are not possible. For example, a dimension search on the critical size of a natural uranium/ordinary water mixture is doomed to failure since even an infinitely sized mixture remains subcritical.

Convergence in time absorption (α) calculations is typically one-sided. If EV (the eigenvalue α) is negative, the effective removal cross section, $\Sigma_R + \alpha/v$ might become negative. If this happens, the automatic search procedure may fail dramatically. For this reason, a parameter oscillation damper, POD, of 0.5 or less is frequently used in such searches.

V. CONSIDERATIONS IN CHOOSING A CODE

Numerous discrete ordinates computer codes have been written over the years and can be acquired by potential users. If you do not already have a discrete ordinates code capability at your facility or if you wish to acquire new or additional codes, you will be faced with several choices. Should you import someone else's code or should you try to write your own? Since the writing of all but the simplest discrete ordinates codes is an expensive and time-consuming task that requires a somewhat experienced programmer, it is normally best to try to import an already-developed computer code. If you decide to do this, which computer code should you choose? In this section we present some of the considerations that should be addressed in answering this question. Some of the considerations are quite obvious; others are more subtle but are frequently just as important.

A. Code Capabilities

The first and most obvious consideration to be given to a candidate discrete ordinates code is whether the code will solve the types of problems that you need solved. Is the code one-, two-, or three-dimensional? Does it treat the geometries you need? Is it time-independent or time-dependent? Will it solve both forward and adjoint problems? Will the code perform the kind of calculation you need, for example, k_{eff}, inhomogeneous source, eigenvalue searches? Does the code permit arbitrary anisotropic order of scattering? Do the boundary conditions treated in the code match your needs? The basic computational capabilities of a candidate code should first be compared with your needs.

In addition to the comparisons of needs versus basic calculational capabilities in the candidate code, there are several other considerations. Discrete ordinates computer codes can generally be classified as either general-purpose codes or specific-application codes. General-purpose codes, as their names imply, contain a broad range of problem-type calculational capabilities. For example, a two-dimensional, time-independent, general-purpose, discrete ordinates code usually solves the multigroup, discrete ordinates equations in (x,y), (r,z), and (r,θ) geomtries. Such codes also usually solve k_{eff} (criticality), inhomogeneous source, and several types of searches. They normally allow arbitrary anisotropic scattering order. These codes are quite flexible; they usually contain good, reliable, numerical

methods; they are normally well tested and documented. Because of their flexibility, general-purpose codes tend to be quite large and may not be computationally optimal for a specific type of problem. On the other hand, specific-application computer codes are generally developed to solve a limited class of problems specifically suited to the facility authoring the code. For example, a specific-application code might solve only (x,y) geometry, time-independent, shielding problems. Such codes are usually much smaller and may be more computationally optimal than general-purpose codes; but, of course, they lack the flexibility of the latter.

Another very important factor to be considered in choosing a code is whether it is essentially free standing and self-contained or whether it requires other auxiliary codes. A modern free-standing, discrete ordinates code will contain a library of angular quadrature sets built into the code. It will accept multigroup cross sections in several generally acceptable and commonly used forms. It can be quite frustrating to acquire and implement a computer code only to learn that one or more additional codes must be acquired and implemented to generate or preprocess data before the main code can be used.

Variable dimensioning and flexible data management are other highly desirable features in a discrete ordinates code. Fixed dimensions on vectors and arrays can unnecessarily restrict the range of problem sizes that can be analyzed, as can inflexible data management.

B. Computing Environment

In choosing a computer code for implementation at your facility, it is very important to consider both your computing environment and the computing environment in which the desired code is operational. There are great differences in computer architecture and computer operating systems, and these differences can cause great difficulties in implementing an imported computer code. In some cases known to the authors, the implementation of a discrete ordinates code into a different computing environment has required man-months or even man-years of effort, whereas other codes have successfully been implemented into new environments in a few man-days.

Some computers, most notably IBM machines, operate with 32 bits per word. Such machines are commonly called "short word" machines. Other computers, for example, CDC and CRAY computers, operate with 48 or more bits per word. These

are called "long word" machines. On short-word computers, six-character hollerith words must be "double precisioned"; on long-word machines such hollerith words can be treated as single precision. Since 32-bit short words with hexadecimal roundoff carry only the equivalence of some six decimal digits,[51] calculations requiring greater precision must be double precisioned. On long-word computers, double precisioning is seldom needed. Because of these word length differences, computer codes written for a long-word computer may cause great problems when implementation is attempted on a short-word computer. A well-written code, designed for exchange, will minimize or eliminate these problems.

The memory hierarchy of computers varies with manufacturer, and computer codes written for one form of hierarchy may not be compatible on a computer with a different memory hierarchy. The two most notable memory hierarchies are the two-level hierarchy and the single-level hierarchy. Two-level memory hierarchy computers normally possess both small, fast core central memory and a separate rapid access peripheral storage, which we shall refer to as extended core. Control Data Corporation (CDC) computers, such as the CDC-7600 and Cyber 720 computers, are the most common two-level hierarchy machines. Fast core central memory on two-level machines is typically limited to about $60\ 000_{10}$ words; extended core memory may accommodate several hundred thousands of words of storage. Single-level memory computers do not possess a separate extended core, but instead have a single, large, fast central memory commonly accommodating several hundred thousand words. IBM computers and CRAY-1 computers are widely used single-level memory hierarchy machines. It should be quite clear that the data management/transfer procedures in a code written for a single-level memory computer can be quite different from those in a code written for a two-level memory computer. The memory hierarchy assumed in a computer code must be considered in deciding whether to import a discrete ordinates code.

In addition to central memory and extended core storage, virtually all computer codes require the use of auxiliary storage devices or units. The number and type of auxiliary storage devices required by computer codes can be quite different, however. Nearly all codes will require the availability of several sequential access storage units, and such units exist in most computing environments. The number of sequential access units required by some computer codes can be quite large, however, and some computing installations may not

allow as many units as a particular code needs. In addition to sequential access storage, many modern codes require the use of direct, or random, access storage. Some computing installations do not support direct access data storage devices. Even for those installations that do permit direct access data storage, the rules for effecting direct access data transfers are both manufacturer dependent and computing-installation dependent. Such local dependencies make the interchange of codes between installations somewhat more difficult.

These and other differences in computer architecture and computing environments impact the portability of discrete ordinates codes. Although the differences in computing environments are significant, it is by no means a hopeless task to import codes. Well-written codes exist whose authors, aware of computing environment differences, have written in such a manner as to minimize problems involved in code implementation into different computing environments.

C. Programming Language

In choosing a discrete ordinates code, it is imperative that the code be programmed in a language usable at your installation. Fortunately, most codes are written in FORTRAN. Currently, most codes are written in FORTRAN-IV, but codes that use FORTRAN-77 are appearing. Some incompatibilities exist between the two languages, so it is important that your facility has a compiler that supports the language used in the code. Many codes, although written predominately in FORTRAN, still contain subroutines written in assembly language or require system library routines provided by the computer manufacturer. These routines may have to be replaced with equivalent subroutines for your computing environment. Be aware of the potential problems associated with such local systems-dependent routines.

D. Efficiency and Accuracy

Both computational efficiency and numerical method accuracy are important factors to be considered in choosing a discrete ordinates computer code.

Computational efficiency refers not only to the speed at which a given calculation can be performed, but also to the data storage requirements and data transfer/management techniques.

The speed at which a calculation is performed is strongly influenced by the manner in which instructions are programmed. A single IF-test in an innermost loop of an iterative code can increase computer running times by perhaps 5% or more. Good, clean programming can often be executed at least twice as fast as a poorly programmed code. Structuring a code to take advantage of vector operations on vector-processing computers can produce factors of 2, 4, or even 10 in computational speed. Since most discrete ordinates codes are iterative, iteration acceleration schemes are a virtual necessity. A typical iterative discrete ordinates code with an effective acceleration method will frequently run 10 to a 100 times faster than an unaccelerated but otherwise identical code.

A discrete ordinates code should also manage and transfer data efficiently. Large, multigroup discrete ordinates problems can require an enormous quantity of data and information that must be stored, managed, and transferred. The quantity of information managed can be minimized by clever use of temporary, or scratch, data, by proper construction of phase-space sweeping loops, and by storing only needed information. Time spent in data transfers can likewise be minimized by constructing the program such that the number of data transfers is minimized, by indexing vectors/arrays so they can be transferred sequentially, by using record or block transfers instead of transferring words individually, etc.

Unfortunately, computational and data storage/transfer efficiency in discrete ordinates codes can conflict with the accuracy of the solution. For example, the so-called "step" spatial differencing scheme for spatial discretization requires a minimal amount of flux information to be computed and stored, it is computationally simple and fast, and it is a positive scheme so that there is no need for negative flux checks or fixups. For efficiency, the step scheme looks excellent. As a spatial discretization scheme, however, it is generally unacceptably inaccurate. In other words, for the step scheme to provide an accurate solution to the transport equation, the spatial mesh size must be made so small and the number of spatial mesh cells so large that the net computational effort and the data storage requirements become unacceptably large. Therefore, the step scheme is seldom used in discrete ordinates codes. More accurate, or higher-order, difference schemes, such as the diamond scheme and other more recent schemes, although more costly in terms of calculational effort and/or data storage per mesh cell, may permit

acceptable accuracy on sufficiently few mesh cells that the overall net calculational effort and data storge required is quite good. Thus, an acceptable compromise between efficiency and accuracy should be sought in a discrete ordinates code.

E. User-Oriented Features

The acceptance and effective use of a discrete ordinates code is greatly influenced by the extent to which user-oriented features exist in the code. Included in user-oriented features are problem specification and data input, interpretation of results, postprocessing or edit features, and documentation. An otherwise excellent code can be used seldom if it is not user oriented. Some of the desirable attributes of a user-oriented code are described below.

The input of problem specifications and data should be clear, easily supplied, and easily checked. Free-field, card-image input is much easier and less prone to error than fixed-field, card-image input. The use of hollerith words to identify input parameters or data arrays is very helpful to the user. For example, if a problem is to use 28 energy groups, it is much clearer for the user to enter this parameter in a form like NGROUP=28 than it is to enter the digits 28 as the fifth entry on the second card-image of the input "deck." The ability of a code to use acceptable built-in default values for parameters can reduce the amount of input required of the user but can still allow users to override these defaulted values if they wish to provide the parameter in their input. Cross-section data in several forms should be accepted to permit flexibility. Redundant input should be minimized. Quadrature sets should be provided as built-in libraries within the code, but users should still be permitted to provide their own quadrature sets if they so desire. The code should have the option of processing the user-supplied input and halting before executing the solution of the problem. This option allows users to thoroughly check their problem specifications before effecting the full, perhaps time-consuming, transport calculation.

Results of a calculation are usually provided as printed output, remote terminal output (printed or display), or both. The results provided by the code should be easy to read and easy to interpret. Liberal use of descriptive headings is a must. Results should be formatted to fit nicely on the output page or screen. The user should have the option of supressing or displaying input parameters/data. The user should also be able to control the printing of

certain calculational results, for example, scalar fluxes, angular fluxes, macroscopic cross sections, etc.

Editing or postprocessing of results is best done by an essentially independent editing module or code. With such a construction, the actual transport calculation can be performed once, with fluxes and other quantities saved as files or tapes. Different edit calculations can then be run independently without having to rerun the transport calculations. The ability to present results in graphical form instead of simple tables of numbers is also highly desirable. Since graphics capabilities and instructions vary widely among computing environments, a graphics output capability is generally not directly available in an as-imported code.

Adequate documentation is an essential part of user-oriented features. Intelligent use of a computer code requires a clear and thorough users manual. With an inadequate manual, the code will either be seldom used or, even worse, it could be used incorrectly. The effective use of a code requires far more than a "black box" treatment, and a good users manual will minimize this. In addition to documentation in a users manual, it is important that the code itself provide documentation in the form of error-diagnostic messages in the code output. Comprehensive error checks and clear error messages are extremely valuable features in a computer code.

F. Availability of Computer Codes

A major consideration in choosing a computer code is its availability. Is the code available and, if so, from where? Several computer codes have been developed but are considered proprietary and are unavailable for external use. Codes developed for military applications or by private companies are frequently in this category. Other codes are available only to a limited or restricted community of users. An example of the latter are codes developed by national governmental agencies or laboratories, which may be available only for external distribution and use within that nation; export of the codes to a foreign country may be prohibited. Fortunately, most discrete ordinates codes are available for general export, distribution, and use.

If the desired computer code is available for external distribution, one can acquire the code in two ways. The first is to obtain the code directly from an installation where the code is operational. This installation may be that of the code's author(s), or it may be that of a known user of the code.

The second source for obtaining a code is through one of the major code centers. The three principal centers are the Radiation Shielding Information Center, the National Energy Software Center (both in the United States), and the NEA Data Bank in France. These centers, in addition to providing computer codes to interested users, offer additional related services. When they acquire a computer code from an author, they compile and execute the code to ensure its proper operation. They ensure that the code is adequately documented, and they provide copies of code documents on request. These code centers are, thus, a valuable resource to the interested user, both for obtaining information on available computer codes and for obtaining the code. The addresses of the three major code centers are provided below:

Radiation Shielding Information Center (RSIC)
Oak Ridge National Laboratory
P. O. Box X
Oak Ridge, TN 37830
U.S.A.

National Energy Software Center (NESC)
Argonne National Laboratory
8700 South Cass Ave.
Argonne, IL 60439
U.S.A.

NEA Data Bank
Nuclear Energy Agency
Organization for Economic Cooperation and Development
91191 Gif-sur-Yvette CEDEX
FRANCE

G. Test Problems

A code imported and made operational in a particular computing environment must be validated. Unfortunately, validating a version of a code on one computer under one operating system does not mean that a different version operating in a different environment is validated. Inclusion of test problems with a computer code package that is being imported provides some degree of code validation at the receiving installation. Normally, several such test problems should be included. For each problem, the input specifications and data should be provided, as well as a copy of the calculation results from the sending installation.

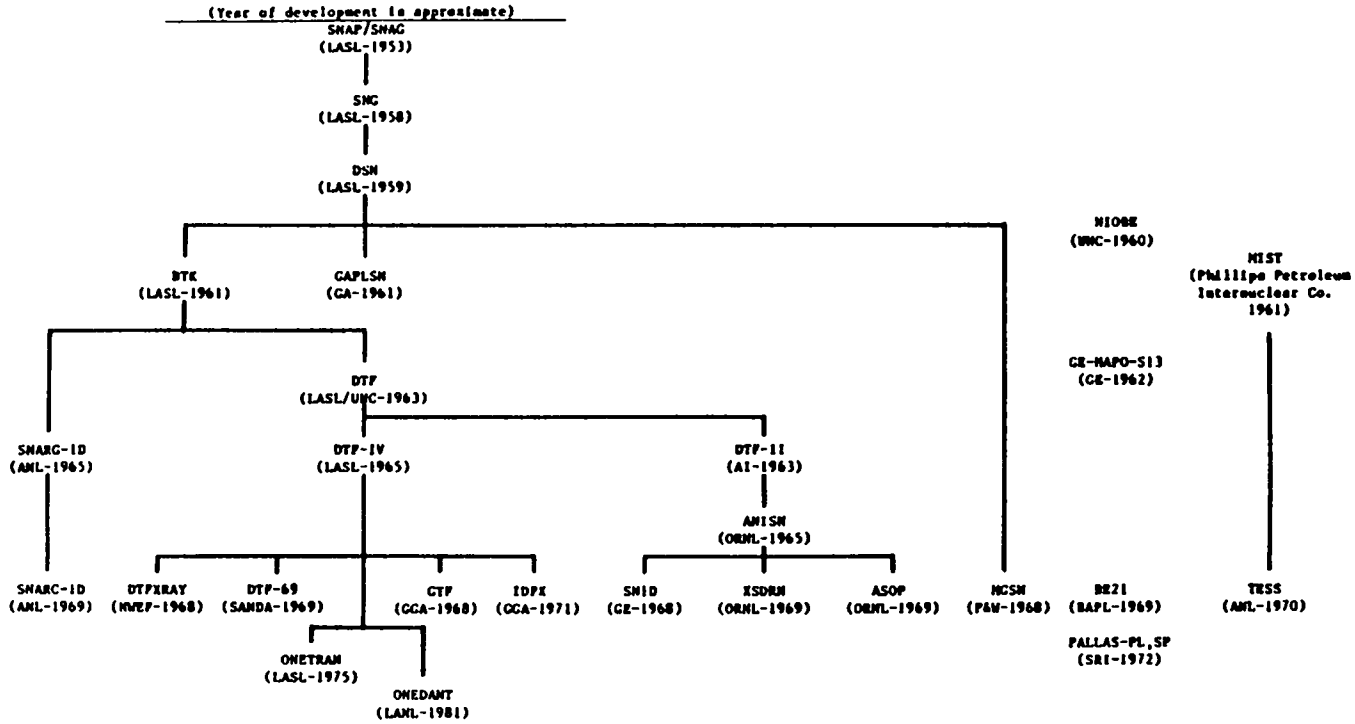
VI. TYPICAL DISCRETE ORDINATES TRANSPORT CODES

Many different discrete ordinates transport computer codes have been developed since the discrete ordinates method was originated at Los Alamos in the 1950s. To differentiate between discrete ordinates codes, it is helpful to understand their origins. Figures 17 and 18 chart the genealogy of most of the one- and two-dimensional, time-independent, discrete ordinates codes. These charts are reasonably complete but are not intended to be exhaustive.

A. One-Dimensional, Time-Independent Codes

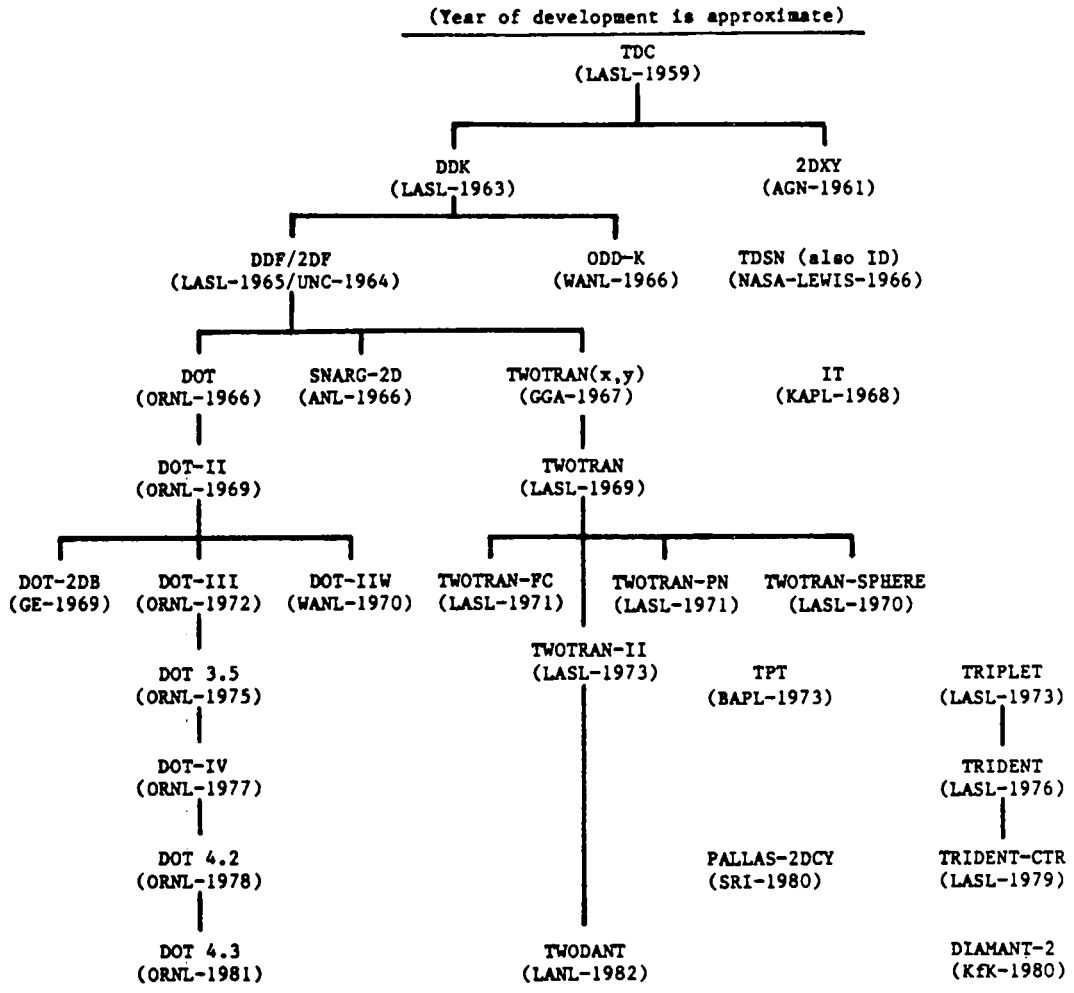
By far, the most common one-dimensional, time-independent, discrete ordinates code used today is ANISN.[52] To a lesser degree, DTF-IV[50] is also widely used. Both of these codes, developed in the mid-1960s, treat plane, cylinder, and sphere geometries; they solve inhomogeneous source, k_{eff}, and several eigenvalue search problems with several boundary condition options. Both codes perform both direct (forward) and adjoint calculations, and both allow general-order anisotropic scattering. In other words, both are general-purpose codes. Both have been used at installations around the world, and it is common that a given installation may have its own special version of the code. The popularity of these codes resulted in the creation of several offspring codes in the late 1960s and early 1970s. The SN1D code[53] offers the same general options as ANISN but allows free format input and group and zone buckling options and contains other special features not available in the basic ANISN code. The IDFX[54] code represents an improved, more sophisticated version of DTF-IV, especially in its improved iteration acceleration techniques. Both the XSDRN[55] (ANISN) and GTF[56] (DTF-IV) codes are programs for calculating space-dependent flux spectra for use in forming multigroup cross sections. The ASOP code[57] is an extension of ANISN to perform shield optimization computations by solving sequences of transport calculations and dose constraint equations. DTF69[58] is a specialized version of DTF-IV for solving photon (gamma-ray) transport calculations only. All of these computer codes use the diamond difference spatial discretization scheme.

The PALLAS-PL,SP[13,59] and ONETRAN[46] discrete ordinates codes represent departures from their diamond-differenced forebearers. PALLAS-PL,SP uses short characteristics for determining the spatial variation of the angular particle flux, as discussed in Sec. IV.B. The code is limited to radiation shielding



AI	Atomics International	LANL	Los Alamos National Laboratory
ANL	Argonne National Laboratory	NWEF	Naval Weapons Evaluation Facility
BAPL	Bettis Atomic Power Laboratory	ORNL	Oak Ridge National Laboratory
GA	General Atomic	P&W	Pratt and Whitney
GE	General Electric	SANDIA	Sandia National Laboratories, Albuquerque
GGA	Gulf General Atomic	SRI	Ship Research Institute, Japan
LASL	Los Alamos Scientific Laboratory		

Fig. 17. Genealogy of one-dimensional, time-independent, discrete ordinates codes.



AGN	Aerojet General Nuclear
ANL	Argonne National Laboratory
BAPL	Bettis Atomic Power Laboratory
GE	General Electric
GGA	Gulf General Atomic
KAPL	Knolls Atomic Power Laboratory
KfK	Kernforschungszentrum Karlsruhe
LASL	Los Alamos Scientific Laboratory
LANL	Los Alamos National Laboratory
ORNL	Oak Ridge National Laboratory
SRI	Ship Research Institute, Japan
WANL	Westinghouse Astronuclear Laboratory
UNC	United Nuclear Corporation

Fig. 18. Genealogy of two-dimensional, time-independent, discrete ordinates codes.

calculations in plane and spherical geometries and is, thus, not a general-purpose discrete ordinates code. ONETRAN is a general-purpose, one-dimensional code that employs the linear-discontinuous spatial discretization scheme described in Sec. IV.B. An additional capability in ONETRAN is its two-angle plane geometry option. {Recall that, in standard plane geometry, the angular flux is independent of the azimuthal angle, ϕ, so the angular dependence is reduced to the μ interval $[-1,1]$.} The two-angle option in plane geometry allows the angular flux to be dependent on both μ and the azimuthal angle, a very useful feature for some applications.

The ONEDANT code[60] is a general-purpose code using the diamond difference scheme. Its chief advantages are its flexible, free-format input capability, its use of the diffusion synthetic method for iteration acceleration as described in Secs. IV.C and IV.D, and its portability. The code is currently finding increasingly widespread use throughout the world.

B. Two-Dimensional, Time-Independent Codes

For general-purpose, two-dimensional, time-independent, discrete ordinates computer codes, the choice is basically between one of the DOT series of codes or a version of the TWOTRAN series. These general-purpose codes treat (x,y), (r,z), and (r,θ) geometries; they solve inhomogeneous source, k_{eff}, and eigenvalue search problems with several boundary condition options; both operate in the direct (forward) and adjoint modes; and both accept arbitrary anisotropic order of scattering. The DOT codes have been developed primarily for radiation shielding analysis. The TWOTRAN codes have been developed in a reactor/core physics environment. Thus, although similar, the two families of codes embody somewhat different emphases and flavors.

In the DOT family of codes (Fig. 18), DOT-III[61] and its improved offspring DOT 3.5 have been widely used throughout the world. In the late 1970s, a significantly improved, new version in the DOT series was developed and denoted DOT-IV[28] although the first generally available form was released under the name DOT 4.2. Subsequent changes and more efficient programming led to the release of an improved version called DOT 4.3[4] in the early 1980s. For radiation shielding, the DOT codes have been without peer in terms of worldwide acceptance and usage.

For reactor core physics application and, to a much lesser degree, for shielding applications, the TWOTRAN codes have been widely used. The TWOTRAN

code,[47] developed in the late 1960s, included several iteration acceleration features. Such acceleraton is virtually essential for the inner and outer iteration used in core physics applications. The TWOTRAN-II[31] code represented an improved, more easily exported version of the code and for nearly a decade represented the standard for core physics applications. In the early 1980s, a totally new offspring in the TWOTRAN family, TWODANT,[62] was developed. Two major improvements in TWODANT are its free field, easily understood input and its use of the very effective diffusion synthetic iteration acceleration method. TWODANT has replaced TWOTRAN in many institutions.

In addition to the DOT and TWOTRAN codes, several somewhat independent, more specialized discrete ordinates codes were developed in the late 1970s. The TRIDENT[63] and TRIDENT-CTR[29] codes represent discrete ordinates codes using the linear-discontinuous, finite element method for spatial discretization on somewhat arbitrary triangular mesh cells in (x,y) and (r,z) geometries. With triangular mesh cells, rather general geometrical shapes, including toroidal geometries, can be closely approximated. The DIAMANT2[64] and TWOHEX[65] codes also use a triangular mesh, but they are restricted to uniform equilateral triangles in (x,y) geometry. They are designed specially for modeling fast breeder reactor cores with hexagonal fuel assemblies.

The PALLAS-2DCY[13,66] code is a special-purpose, shielding-analysis, discrete ordinates code. It is restricted to (r,z) cylindrical geometry and shielding (inhomogeneous source) problems. It employs the method of short characteristics described in Sec. IV.B.

C. Three-Dimensional, Time-Independent Codes

The extension of discrete ordinates codes to three dimensions has proceeded somewhat slowly. The first known demonstration code was THREETRAN,[67] which was written for (x,y,z) geometry. It was quite limited in its generality, and its iteration procedures were virtually unaccelerated. The code was not intended to be a practical production tool, but instead was a demonstration-in-principle that data management strategies and procedures could be devised and made to function efficiently on the computers and data storage capabilities extant in the 1970s. In this latter respect, THREETRAN was a successful demonstration code. Although a modest number of improvements were made to the code at Los Alamos, and the code was subsequently renamed THREETRAN (x,y,z) in the late 1970s, little work was done on iteration acceleration. As

a result, calculations required so much computing time that when work on the code was suspended, it could still be considered only a demonstration code.

The next three-dimensional code was THREETRAN (hex,z),[68] developed for the hexagonal-z geometries characteristic of fast-breeder reactor cores. This code uses an equilateral triangular, right prism, mesh cell suitable for describing hexagonal-z geometries. As with THREETRAN (x,y,z), THREETRAN (hex,z) contained minimal iteration acceleration and, thus, it too can be considered only a demonstration code.

The ENSEMBLE[69] code, developed in Japan, is an (x,y,z) geometry discrete ordinates code for radiation shielding applications only. The code permits anisotropic scattering and uses the coarse mesh rebalancing scheme for iteration acceleration. The code also contains a negative flux fixup routine that uses a variable weight diamond difference scheme. A ray effect mitigation option is also available. Although used locally by its authors, the ENSEMBLE code has not received widespread attention or usage.

During the late 1970s, therefore, the technical feasibility of three-dimensional, discrete ordinates codes was demonstrated. Their practical usage, however, was not well established because of the expense involved in both computer execution time and computer storage requirements. Many mesh cells are required to adequately model most full-size, three-dimensional problems using the diamond difference spatial discretization scheme. It is likely that more accurate schemes will have to be developed and implemented so larger mesh cells can be used to reduce the total number of mesh cells to more practical levels. In addition, more modern and effective iteration acceleration schemes are needed to reduce the time for iterative convergence.

D. Time-Dependent Discrete Ordinates Codes

Reasonably little work has been done in developing time-dependent, discrete ordinates computer codes. In about 1970, a one-dimensional, time-dependent version of the ANISN code was developed jointly by Oak Ridge National Laboratory and Los Alamos National Laboratory and named TDA[70] (Time Dependent ANISN). This code used a weighted diamond difference scheme in space and time with an optional exponential time-differencing scheme. As originally formulated, the code did not treat delayed neutrons; however, more recent versions of the code permit delayed neutron treatment.

At approximately the same time as TDA was being developed, TRANZIT, a two-dimensional, (r,z) cylindrical geometry, time-dependent code was developed at Los Alamos.[71] It employs a weighted diamond difference approximation in all variables. TRANZIT does not provide for including a fission source, but a time-dependent, inhomogeneous, distributed source separable in space and time can be used. It also contains a first-collision source option. Essentially no further development has been devoted to TRANZIT since it was first developed in 1970.

In 1976, the general one-dimensional, time-dependent code TIMEX[72] was developed. TIMEX is quite similar to the time-independent code ONETRAN in its geometry and boundary condition capabilities. Like ONETRAN, it employs a linear discontinuous finite element scheme for spatial discretization. The time variable in TIMEX is differenced by an explicit, unconditionally stable technique. Delayed neutrons are treated.

The above three computer codes are the only known, generally available, time-dependent, discrete ordinates codes in use. Numerous other time-dependent, discrete ordinates codes exist at local installations around the world, for example, the French code EFD,[73] but these codes are generally not available for public use.

VII. GUIDANCE FOR THE USER

The ultimate reason for a discrete ordinates computer code is that it be used. The work of the code developer, the considerations made in choosing a particular code for use at a given facility, and the steps that were taken to make the code operational at that facility all become history. It is now the user's job to make effective use of this code as a tool in design or analysis. How can this tool be used most effectively? How does the user know that the code is giving good answers? Where does the user begin? This section provides some guidance that should help answer these questions.

A. Familiarity with the Code

One of the first things you should do as a user is to get familiar with the code. Three essentials are 1) the code users manual, 2) familiarity with the actual execution of the code and the information it provides, and 3) familiarity and understanding of the physics of particle transport.

1. Code Users Manual. The code users manual is a valuable resource. Use it! At a minimum, it should tell you how to provide the input to the code and how to interpret the code's output. Many manuals provide much more information. They will describe the numerical methods and approximations contained in the code or will provide references where this information can be found. The manual will give specific advice, cautions, and recommendations for proper usage of the code. It will describe the error diagnostics and error messages provided by the code. It is the authors' experience that most of the mistakes and much of the misuse of a code could have been avoided had the user only become familiar with the contents of the manual. Accordingly, our advice to you, the user, is to take the time and effort at the outset to get familiar with the code users manual. Remember, however, that a code users manual is a guide and not absolute truth. It likely contains general recommendations and observations covering a broad range of problem types, computers, and/or applications. For your particular situation, these recommendations and observations may not apply. Thus, you must acquire hands-on experience with the code - the second area of familiarization.

2. Run the Code. Familiarity with the actual execution of the code requires your running problems. You probably have already used other computer codes and know how they perform; running several problems with your discrete ordinates code will give you a feel for the relative running times. It is quite easy to estimate the execution times of subsequent runs once a problem of a particular type has been run. The execution time of most codes is proportional to the total number of spatial mesh intervals or angular directions. As a user, you should verify this proportionality and get a good feel for the amount of time, hence the cost, of calculations. Similarly, the number of energy groups will affect running times. If upscatter or fission coupling is not significantly altered by changing the number of energy groups, execution times are usually proportional to the number of groups.

In addition to acquiring a feel for the required execution times, get a feel for the accuracy of the results. Vary the number of spatial mesh intervals and the angular quadrature order to see if the results change significantly. There is no sense in running a two- dimensional S_{16} calculation with a 50 × 100 spatial mesh if you can get the same results with an S_8 calculation on a 25 × 50 mesh. Similarly, although you find that a particular mesh structure and quadrature order is required to yield acceptable accuracy in

a given parameter, say k_{eff}, accurate differences in the parameter caused by changes in problem dimensions or material compositions can often be obtained with a coarser (hence, less expensive to calculate) mesh structure and quadrature order. As a responsible user, you should confirm this behavior by running the code.

Code execution times and accuracy of results are affected by iteration convergence precisions. Convergence precisions that are too tight will simply waste computer time. The users manual will usually provide guidance on acceptable convergence precisions, but you still should familiarize yourself with the behavior of the code as a function of varying the convergence criteria.

While you are becoming familiar with running the code and observing how the code responds to various changes, you will likely find yourself wondering why the code responds and behaves as it does. This leads us to the third area of familiarization.

3. Learn About the Physics of Particle Transport. Familiarity with and understanding of some of the physics of particle transport are important for the knowledgeable user. An understanding of some of the whys enables the user to better know the hows of correct problem modeling and execution. The value of understanding some of the physics of particle transport can be illustrated by considering the problem of a monoenergetic point isotropic source of particles at the center of a uniform 10 mean free path (mfp) sphere of nonscattering material. For such a problem, the scalar flux for source-energy particles varies as $[\exp(-r)]/r^2$, where r, the radial distance from the center, is measured in mean free paths. To model the point source as a finite sphere for analysis with a discrete ordinates code, this source-containing sphere could be quite tiny, say of radius 0.0001 mfp. clearly, such a choice models a "point" source quite well, but such a choice is likely to cause significnt problems in modeling the rest of the problem. The reason for this is that from $r = 0.0001$ mfp to $r = 1$ mfp, the scalar flux will decrease by at least eight orders of magnitude because of the $1/r^2$ behavior of the flux. An extremely-fine-mesh spacing with a very large number of mesh cells will be required near the origin. By simply expanding the source sphere to a radius of, say, 0.05 or 0.1 mfp, much of the $1/r^2$ singularity is removed with little loss in accuracy even at distances as close as 1 mfp from the sphere origin. Such modeling can reduce dramatically the number of mesh cells. The "physics" of this problem

also indicates that logarithmic variation in mesh spacing may be preferred over uniform mesh spacing, especially near the source where rather fine mesh spacing is necessary. Logarithmic interpolation options are available in several general-purpose, discrete ordinates codes. Further, the "physics" of angular redistribution in this problem dictates that a rather high S_N order is probably required, perhaps S_{32} or S_{48}. In addition, although a "pure absorbing" material may seem somewhat academic, an understanding of the "physics" of multigroup cross sections shows that such materials do very nearly exist. For high-energy neutron groups in which the energy group is fairly narrow, within-group, or self-scatter can be quite small. Since any reaction other than self-scatter removes particles from the energy group, materials will appear as nearly pure absorbers (removers) for such groups. The above example shows that there is a great deal of "physics" that, when understood, can be applied by you, the user. There are many other areas where an understanding of the physics of numerical particle transport is valuable. The unaccelerated inner-iteration process used in most transport codes consists of generating the flux solution by adding together the flux resulting from no self-scatter collisions (first inner iteration), the flux resulting from one self-scattering collision (second iteration), the flux resulting from two self-scattering collisions (third iteration), etc. Understanding this process explains why in large regions for energy groups in which the material is nearly a pure self-scatterer, unaccelerated inner iterations converge very slowly and why an effective inner iteration acceleration scheme is so important. Understand that applying a negative flux fixup forces more particles to leak from the affected mesh cell and determine whether such fixups are important to your calculation. You should learn about the probable need for placing more than one mesh cell in voids in curvilinear geometries where particle streaming occurs. Learn that diamond differencing yields very accurate integral results, even when local angular fluxes are poorly predicted. Learn to apply the correct boundary conditions, especially in curved geometries. Far too frequently, cylindrical cell calculations are performed using specular reflection as an outer boundary condition instead of the "white" (isotropic return) boundary condition that was developed for curved geometry cell calculations.

Remember that a computer code is inanimate and cannot analyze and correct your input so that you will correctly solve a problem. It will solve, or attempt to solve, only the problem that you have given it. Being familiar with

the code and its users manual and understanding the physics of particle transport will help you provide the code with meaningful, well-posed problems. As a user with such familiarity and understanding, most of your problems should be run acceptably. There will be times, however, when your problem is ill-posed in that it causes trouble when the code attempts to effect a solution. Below are some of the common indications of trouble and their causes.

B. Indications and Causes of Trouble

There are many causes for an unsatisfactory run of a problem by a discrete ordinates code. Simple user input errors, bad or inappropriate nuclear data, poor geometric modeling, an inappropriate angular quadrature, or simply a very tough problem are examples of such causes. Indications of trouble or unsatisfactory execution range from the glaring, obvious fatal error through the more subtle, not-so-obvious indications, all the way to no indication at all.

1. Obvious Indications of Trouble. The most obvious indication that something is wrong in a computer code run is the fatal error in which problem execution is abruptly halted, usually with a highly visible message to the user. Fatal errors are normally caused either by input errors or by deficiencies in the computer code.

The most common fatal errors are those caused by user input error. These are either detected by the computer code or by the computing system. The most user-friendly codes will have extensive input data checking capabilities, error diagnostics, and clearly understood error messages. Such codes make the correcting of input errors quite simple. At the other extreme are computer codes with virtually no built-in error diagnostics. Input errors in these codes are usually discovered by the computing system, if discovered at all. System error messages tend to be somewhat general, and pinpointing the input error is significantly more difficult than with a user-friendly code.

Fatal errors caused by errors or deficiencies in the code usually manifest themselves as computing system fatal errors. Examples of these are system overflows where perhaps the code has attempted to divide by zero or address-out-of-range errors. Errors of this sort usually occur because your particular problem is exercising an option or a logical flow path that has not been exercised before. In such a case, you normally must either fix the coding error or deficiency, or the code author must be contacted for assistance. If

the code being used is somewhat new, it is fairly likely that you may encounter such a situation - we hope that you do not, but do not be surprised when this happens. Large computer codes have many options, and the number of combinations and permutations in the logic of the code can be very large. Only through the running of hundreds or thousands of problems by many different users will a code be thoroughly checked out. Keep this in mind, and be a friendly - or at least an understanding - user when you encounter an error in the code.

2. Not-So-Obvious Indications of Trouble. Many computer code runs do not result in fatal errors but are, nevertheless, unsatisfactory. Such runs commonly provide the user with indications that the run may be unsatisfactory, but these indications are not as obvious as the fatal error. These not-so-obvious indications of trouble usually appear in one or more of the following forms: warning messages, iteration convergence problems, negative scalar fluxes, and poor particle balance. User-controlled output and edit prints and/or graphical displays also frequently indicate trouble.

The well-written, user-friendly production code will frequently provide relatively visible warning messages when it detects conditions that may cause the results to be questionable or unsatisfactory. One situation in which a warning message might be provided is when a specular reflection boundary condition instead of the white boundary condition is being used in a cylindrical cell calculation. Another example might be the detection of inconsistent nuclear cross sections in which the total cross section is not equal to the sum of absorption and scattering cross sections. Still another might indicate that the code reached the user-input limit on allowable run time before full convergence was achieved. Such warning messages can be of great value to you in your deciding whether the code results are satisfactory. If warning messages are provided by your code, look for them. They have been provided by the code author to help you.

Discrete ordinates codes virtually always use an inner- and an outer-iteration method for effecting a solution. Convergence of these iterations is usually considered a requirement for a satisfactory solution. Most codes provide an iteration monitor print in their output to present a record of the iteration convergence. You should always review this iteration monitor print before accepting the results of a calculation. This monitor print will indicate failure to achieve convergence, an important indicator of trouble.

Before giving some of the reasons why a problem may fail to converge, we should make some general remarks about discrete ordinates code iterations. The more leaky or absorbing a problem is, the faster the iterations will converge. Conversely, in diffusion-like problems with large, low-leakage, weakly absorbing regions, the unaccelerated iteration convergence rate can be very slow. Eigenvalue problems and search calculations require more than one outer iteration. Inhomogeneous source problems with isotropic scatter and no upscatter should converge in one outer iteration if the inner iterations for each energy group are allowed to achieve full convergence.

Because of the iterative procedures used in discrete ordinates codes, virtually all such codes employ techniques to accelerate the convergence rate of both the inner and outer iterations. Earlier we described the Chebyshev, rebalance, and diffusion synthetic acceleration schemes commonly used. The great majority of the time, these schemes will perform quite adequately, and full convergence wil be achieved. In some problems, however, neither Chebyshev nor rebalance acceleration will work very effectively, and convergence will be achieved quite slowly. It is inherent with certain problems that neither method will perform well, and there is little that you, the user, can do to prevent it. What you must do is be aware of the difficulty of predicting the final, fully converged result when the iterations are approaching the result asymptotically. Even though the difference in results between successive outer iterations will be quite small, the results may still be unacceptably far from the true asymptotic result.

Occasionally you may encounter a problem in which the acceleration method itself delays or prevents convergence. Fine mesh rebalance is most likely to do this. If divergence of the iterative procedure occurs for a problem using fine mesh rebalance, making the rebalance mesh somewhat coarser may correct the situation. If divergence still occurs with coarse mesh rebalance, the trouble probably lies elsewhere. The Chebyshev acceleration method also, on rare occasions, can prevent convergence, in which case you should simply turn off the acceleration and try rerunning the calculation.

Most discrete ordinates codes permit the user to specify the desired convergence precision through user-input convergence criteria. Generally, the more local the convergence criterion, the more slowly convergence will be achieved. Thus, pointwise flux convergence from one inner iteration to the next may be more difficult to achieve than convergence of a global quantity

such as k_{eff}. Since inner iteration convergence criteria are commonly applied to pointwise quantities, it is customary to limit the number of inner iterations in any given outer iteration. This practice prevents endless iterations in the event that convergence is not possible. It also saves computation time. In a problem requiring several outer iterations, time can be wasted doing many inner iterations during the early outer iterations. As the outer iterations proceed toward convergence, inner iterations will converge concurrently. Failure to reach inner iteration convergence can be caused by the acceleration method, as previously mentioned, or it can be caused by too tight a convergence criterion. It can also be caused by allowing too small a limit on the allowable number of inner iterations. The appearance of negative fluxes or the application of negative flux fixups can sometimes occur in an oscillating "on-off" pattern between iterations for a given mesh cell. This oscillatory behavior may prevent the pointwise fluxes from achieving full convergence. Frequently, such oscillations occur in unimportant mesh cells, and they have little effect on the overall results. In such cases, failure to reach full inner-iteration convergence is not important.

Just as with inner iterations, outer-iteration convergence can be prevented by too tight a convergence criterion. On some computers, a criterion of 10^{-8} will fail simply because the result of subtracting two nearly equal numbers may differ by more than this. In selecting convergence criteria for your problem, you must understand the manner in which the criteria are used and the effectiveness of the convergence tests in the code. Different codes use different criteria. What may be a necessary convergence criterion in one code may be gross overkill in another.

The third not-so-obvious indicator of trouble is the existence of negative fluxes in the output from a run. Negative fluxes can occur because the scattering source, as computed by the code, is negative. This can happen when the scattering processes are highly anisotropic and/or the particle flow is highly anisotropic. If the spherical harmonics expansion for the scattering source is truncated at too low an order and the expansion is evaluated at the discrete directions of the problem at hand, it may be negative. Normally, such occurrences result in a few negative angular fluxes, but the angle-integrated scalar flux remains positive. Such situations can usually be tolerated. If, however, the situation is such that the scalar flux becomes negative, some remedial action is probably required. Increasing the order of the scattering

approximation may suffice if higher-order scattering data are available. The most common cause of negative scattering sources lies with the higher-order scattering cross-section data. Referring to Eq. (34) in Sec. II, the scattering expansion for higher-order (anisotropic) terms contains a $(2\ell + 1)$ expansion coefficient that multiplies the higher-order cross sections, Σ_s^ℓ. Some computer codes explicitly perform this multiplication; other codes require that the $(2\ell + 1)$ factor be included in the Σ_s^ℓ data. If the latter form of Σ_s^ℓ is used in the former type of code, the result is an erroneous increasing of the higher-order contributions to the scatter source. This commonly causes the resulting negative fluxes. It is, therefore, imperative that when performing calculations with anisotropic scattering, you make certain that the $(2\ell + 1)$ expansion factor is being treated correctly.

If negative scattering sources are not the cause of negative fluxes in your output, the probable cause is mesh spacing that is too large. The common spatial-differencing schemes employed in discrete ordinates codes are the diamond differencing and the linear discontinuous schemes described earlier. Both of these schemes can produce negative angular fluxes if the mesh spacing is too large. Most codes employ a negative flux fixup scheme to eliminate these negative fluxes. Occasionally, however, a multidimensional problem is so coarsely meshed that even the negative flux fixup will fail, and negative fluxes will remain. The only recourse you have for eliminating these problems is to refine the mesh in the regions where the offending negative fluxes occur.

The final indication of trouble in a calculation appears in particle balance and balance table summary prints in those codes that provide such output. Particle balance is normally a global measure of the equality between losses and sources. Particle balance suffers when one or more energy groups are not fully converged. It also suffers when a low-order angular quadrature is used in conjunction with high-order anisotropic scatter, in which case the quadrature may not correctly integrate the spherical harmonic flux moments used in the scattering source and may, thus, cause an erroneous addition to the sources in the particle balance equation. Balance tables typically provide certain space-angle integrated quantities for each energy group and for the sum of the groups. Quantities typically provided are fission source rate, absorption rate, inscatter, self-scatter, and outscatter rates, leakage rates from each surface, etc. These tables provide valuable clues to the proper or improper performance of a calculation. For example, they are a sensitive

measure of cross-section inconsistencies. If the group sum of the outscattering rates is not equal to the group sum of the inscattering rates while particle balance is satisfied, there is either an error in the cross sections or in the code itself.

In summary, there are many clues and indications of the satisfactory or unsatisfactory performance of a calculation provided in the output of most codes. Warning messages, the iteration monitor print, and the convergence information it contains, flux prints or plots, particle balance and balance tables are commonly available. While not as glaringly obvious as a fatal error, they are there nevertheless, and they provide indications of trouble. Get into the habit of using them.

3. Nonobvious Errors. There are, unfortunately, many nonobvious errors that occur in running problems. The cause of many of these errors is the code user. User inexperience, haste, simple carelessness, and honest mistakes often result in invalid calculations, wasted computer time, and erroneous conclusions. The wrong problem is often solved, and the error is not recognized. Mistakenly entering the geometry option flag for (x,y) geometry instead of the desired (r,z) geometry can easily go unnoticed. Using a fast reactor multigroup cross-section set for a thermal neutron system can lead to grossly incorrect conclusions. Failing to check that mesh spacing or quadrature variations do not significantly affect calculational results can be dangerous. Simply ignoring or not noticing the indications of trouble discussed above can have serious consequences. Remember that the code will attempt to solve the problem you give it. It cannot read your mind to determine whether input specifications describe the problem desired. The responsibility for minimizing such errors rests squarely on the user. Be careful. Check the input carefully; review the output. Use discrete ordinates codes intelligently. Recognize that a powerful tool is at the user's disposal but that correct use of this tool is the user's responsibility alone.

REFERENCES

1. K. D. Lathrop and F. W. Brinkley, Jr., "TWOTRAN SPHERE: A FORTRAN Program to Solve the Multigroup Transport Equation in Two-Dimensional Spherical Geometry," Los Alamos Scientific Laboratory report LA-4567 (November 1970).

2. E. M. Gelbard, "Spherical Harmonics Methods: P_L and Double-P_L Approximations," in Computing Methods in Reactor Physics, H. Greenspan, C. N. Kelber, and D. Okrent, Eds., Gordon and Breach, New York, 1968, p. 329.

3. G. I. Bell and S. Glasstone, Nuclear Reactor Theory, VanNostrand Reinhold Co., New York, 1970, p. 163.

4. W. A. Rhodes and R. L. Childs, "An Updated Version of the DOT 4 One- and Two-Dimensional Neutron/Photon Transport Code, Oak Ridge National Laboratory report ORNL-5851 (July 1982).

5. G. I. Bell and S. Glasstone, Nuclear Reactor Theory, Chap. 6, VanNostrand Reinhold Co., New York, 1970.

6. E. L. Wachspress, "Iterative Solution of Elliptic systems and Applications to the Neutron Diffusion Equation of Reactor Physics," Prentice Hall, Englewood Cliffs, NJ, 1966.

7. J. K. Fletcher, "The Solution of the Multigroup Neutron Transport Using Equation Spherical Harmonics," Nucl. Sci. Eng. 84, 33, (1983).

8. A. M. Weinberg and E. P. Wigner, "The Physical Theory of Neutron Chain Reactors," The University of Chicago Press, Chicago, IL, 246, 1958.

9. B. G. Carlson and K. D. Lathrop, "Transport Theory - The Method of Discrete Ordinates," Los Alamos Scientific Laboratory report LA-3251-MS, Rev. (October 1965).

10. W. F. Miller, Jr. and W. H. Reed, "Ray-Effect Mitigation Methods for Two-Dimensional Neutron Transport Theory," Nucl. Sci. Eng. 62, 391 (1977).

11. K. D. Lathrop, "Remedies for Ray Effects," Nucl. Sci. Eng. 45, 255 (1971).

12. E. E. Lewis, "Progress in Multidimensional Neutron Transport Computation," Nucl. Sci. Eng. 64, 279 (1977).

13. N. Sasamoto and K. Takeuchi, "An Improvement in the PALLAS Discrete Ordinates Transport Code," Nucl Sci. Eng. 71, 330 (1979).

14. E. W. Larsen, "On Numerical Solutions of Transport Problems in the Diffusion Limit," Nucl. Sci. Eng. 83, 90 (1983).

15. B. G. Carlson and C. E. Lee, "Mechanical Quadrature and the Transport Equation," Los Alamos Scientific Laboratory report LA-2573-MS (August 1961).

16. K. D. Lathrop and B. G. Carlson, "Discrete Ordinates Angular Quadrature of the Neutron Transport Equation," Los Alamos Scientific Laboratory report LA-3186 (February 1965).

17. C. E. Lee, "Discrete S_N Approximation to Transport Theory," Los Alamos Scientific Laboratory report LA-2595-MS (March 1962).

18. B. G. Carlson, "Transport Theory: Discrete Ordinates Quadrature Over the Unit Sphere," Los Alamos Scientific Laboratory report LA-4554 (December 1970).

19. B. G. Carlson, "Tables of Equal Weight Quadrature EQ_N Over the Unit Sphere," Los Alamos Scientific Laboratory report LA-4734 (July 1971).

20. E. M. Gelbard, "Arrangement of Ordinates in Deep-Penetration Cylindrical and X-Y S_N Calculations," in Proc. Conf. New Developments in Reactor Mathematics and Applications, CONF-710302, Vol. 2, U.S. Atomic Energy Commission, 1971.

21. T. J. Seed and R. W. Albrecht, "Application of Walsh Functions to Neutron Transport Problems -- I and II," Nucl. Sci. Eng. 60, 337-356, 1976.

22. I. K. Abu-Shumays, "Compatible Product Angular Quadrature for Neutron Transport in x-y Geometry," Nucl. Sci. Eng. 64, 299-316, 1977.

23. J. E. Morel, "Tables of Lobatto Quadrature Sets for S_N Calculations in One-Dimensional Cylindrical Geometry," Sandia National Laboratories report SAND80-2720, January 1981.

24. P. J. Davis and I. Polonsky, "Numerical Interpolation, Differentiation, and Integration," in Handbook of Mathematical Functions, M. Abramowitz and I. A. Stegun, Eds., National Bureau of Standards, Washington, D.C., 1965, p. 916.

25. J. Yvon, "La Diffusion Macroscopique des Neutrons une Methode d'Approximation," J. Nucl. Energy I, Vol. 4, 1957, pp. 305-318.

26. J. P. Jenal, P. J. Erickson, W. A. Rhodes, D. B. Simpson, and M. L. Williams, "The Generation of a Computer Library for Discrete Ordinates Quadrature Sets," Oak Ridge National Laboratory report ORNL/TM-6023 (October 1977).

27. G. E. Hansen and W. H. Roach, "Six and Sixteen Group Cross Sections for Fast and Intermediate Critical Assemblies," Los Alamos Scientific Laboratory report LA-2543-MS (December 1961).

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

29. T. J. Seed, "TRIDENT-CTR User's Manual," Los Alamos Scientific Laboratory report LA-7835-M (May 1979).

30. R. J. Cerbone and K. D. Lathrop, "S_N Calculation of Highly Forward Peaked Neutron Angular Fluxes Using Asymmetrical Quadrature Sets," Nucl. Sci. Eng. 35, 139-141 (Jan. 1969).

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

32. E. T. Tomlinson, W. A. Rhoades, and W. W. Engle, Jr., "Flux Extrapolation Models Used in the DOT-IV Discrete Ordinates Neutron Transport Code," Oak Ridge National Laboratory report ORNL/TM-7033 (May 1980).

33. E. W. Larsen, "Spatial Convergence Properties of the Diamond Difference Method in (x,y) Geometry," Nucl. Sci. Eng. 80, 710 (1982).

34. M. Mordant, "Some Efficient Lagrangian Mesh, Finite Elements Encoded in ZEPHYR for Two Dimensional Transport Calculations," Annals of Nucl. Energ. 8, 657 (1981).

35. W. F. Walters and R. D. O'Dell, "Nodal Methods for Discrete-Ordinates Transport problems in (x,y) Geometry," Proc. International Top. Meeting on Advances in Math. Meth. for Solution of Nucl. Eng. Problems, April 27-29, 1981.

36. W. F. Walters, "Recent Developments in Nodal and Characteristic Methods in Transport Theory," Trans. Am. Nucl. Soc. 43, 611 (1982).

37. K. Takeuchi and N. Sasamoto, "Fundamental Theory of the Direct Integration Method for Solving the Steady-State Integral Transport Equation for Radiation Shielding Calculations," Nucl. Sci. Eng. 80, 536 (1982).

38. K. Takeuchi and N. Sasamoto, "Direct Integration Method for Solving the Multigroup Neutron Transport Equation in Three-Dimensional Geometry," Nucl. Sci. Eng. 80, 554 (1982).

39. R. S. Varga, "Matrix Iterative Analysis," Prentice-Hall, Englewood Cliffs, NJ, 1962.

40. W. H. Reed, "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory," Nucl. Sci. Eng. 45, 245 (1971).

41. E. M. Gelbard and L. A. Hageman, "The Synthetic Method as Applied to the S_N Equations," Nucl. Sci. Eng. 37, 288 (1969).

42. R. E. Alcouffe, "Diffusion Synthetic Acceleration Method for the Diamond Differenced Discrete Ordinates Equations," Nucl. Sci. Eng. 64, 344 (1977).

43. E. W. Larsen, "Diffusion Synthetic Acceleration Methods for the Discrete Ordinates Equations," Proc. of Top. Meeting on Adv. in Reactor Computations, Amer. Nucl. Soc., Salt Lake City, March 28-31, 1983, p. 705.

44. E. W. Larsen, "Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete Ordinates Equations, Part 1: Theory," Nucl. Sci. Eng. 82, 47 (1982).

45. D. R. Ferguson and K. L. Derstine, "Optimized Iteration Strategies and Data Management Considerations for Fast Reactor Finite Difference Diffusion Theory Codes," Nucl. Sci. Eng. 64, 593 (1977).

46. T. R. Hill, "ONETRAN: A Discrete Ordinates Finite Element Code for the Solution of the One-Dimensional Multigroup Transport Equation," Los Alamos Scientific Laboratory report LA-5990-MS (April 1975).

47. K. D. Lathrop and F. W. Brinkley, Jr., "Theory and Use of the General-Geometry TWOTRAN Program," USAEC Report LA-4432, Los Alamos Scientific Laboratory (May 1970).

48. T. R. Hill, "Efficient Methods for Time Absorption (α) Eigenvalue Calculations," Proc. of Top. Meeting on Adv. in Reactor Computations, Amer. Nucl. Soc., Salt Lake City, March 28-31, 1983, p. 724a.

49. R. E. Alcouffe, "The Multigrid Method for Solving the Two-Dimensional Diffusion Equation," Proc. of Top. Meeting on Adv. in Reactor Computations, Amer. Nucl. Soc., Salt Lake City, March 28-31, 1983, p. 340.

50. K. D. Lathrop, "DTF-IV, A FORTRAN-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering," USAEC report LA-3373, Los Alamos Scientific Laboratory (November 1965).

51. D. R. Vondy, "Programming Practices and Computer Code Development," Oak Ridge National Laboratory report ORNL-TM-5065 (Dec. 1975).

52. W. W. Engle, Jr., "A User's Manual for ANISN, A One Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering," USAEC report K-1693, Union Carbide Corporation, 1967.

53. R. Protsik, S. C. Crick, and J. M. Kelley, "SN1D: A One-Dimensional Discrete ordiantes Transport Code with General Anisotropic Scattering," report GEAO-0064 (Rev. 1), General Electric Company, 1970.

54. R. Archibald, K. D. Lathrop, and D. Mathews, "IDFX: A Revised Version of the IDF (DTF-IV) S_N Transport Theory Code," report Gulf GA-B-10820, Gulf General Atomic, 1971.

55. N. M. Green and C. W. Craven, Jr., "XSDRN: A Discrete Ordinates Spectral Averaging Code," USAEC report ORNL-TM-2500, Oak Ridge National Laboratory, 1969.

56. R. J. Archibald and K. D. Lathrop, "GTF: A Space Dependent Thermal Spectrum Code," USAEC report GA-8775, Gulf General Atomic, 1968.

57. W. W. Engle, Jr., "A User's Manual for ASOP, ANISN Shield Optimization Program," USAEC Report CTC-INF-941, Union Carbide Corporation, Computing Technology Center, 1969.

58. J. H. Renken and K. G. Adams, "An Improved Capability for Solution of Photon Transport problems by the Method of Discrete Ordinates," USAEC report SC-RR-69-739, Sandia Laboratories, 1969.

59. K. Takeuchi, "PALLAS-PL,SP, A One-Dimensional Transport Code," in Papers of Ship Research Institute No. 42, Ship Research Institute, Tokyo, Japan (1973).

60. R. D. O'Dell, F. W. Brinkley, Jr., and D. R. Marr, "User's Manual for ONEDANT: A Code Package for One-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory report LA-9184-M (February 1982).

61. W. A. Rhoades and F. R. Mynatt, "The DOT III Two-Dimensional Discrete Ordinates Transport Code," Oak Ridge National Laboratory report ORNL-TM-4280 (Sept. 1973).

62. R. E. Alcouffe, F. W. Brinkley, Jr.; D. R. Marr, and R. D. O'Dell, "User's Manual for TWODANT: A Code Package for Two-Dimensional, Diffusion-Accelerated, Neutral-Particle Transport," Los Alamos National Laboratory report LA-10049-M, Rev. (October 1984).

63. T. J. Seed, W. F. Miller, Jr., and F. W. Brinkley, Jr., "TRIDENT: A Two-Dimensional, Multigroup, Triangular Mesh Discrete-ordinates, Explicit Neutron Transport Code," Los Alamos Scientific Laboratory report LA-6735-M (March 1977).

64. K. Kufner and R. Heger, "DIAMANT2: A Multigroup Neutron-Transport program for Triangular and Hexagonal Geometry," Kernforschungzentrum Karlsruhe report KfK 3033 (Sept. 1980) (in German); Oak Ridge National Laboratory report ORNL-TR-4797 (English translation).

65. W. F. Walters, F. W. Brinkley, Jr., and D. R. Marr, "User's Guide for TWOHEX: A Code Package for Two-Dimensional, Neutral-Particle Transport in Equilateral Triangular Meshes," Los Alamos National Laboratory report LA-10258-M (October 1984).

66. K. Takeuchi, "PALLAS-2DCY-FC, A Calculational Method and Radiation Transport Code in Two-Dimensional (R,Z) Geometry," in Papers of Ship Research Institute No. 57, Ship Research Institute, Tokyo, Japan (July 1979).

67. K. D. Lathrop, "THREETRAN: A Program to Solve the Multigroup Discrete Ordinates Transport Equation in (x,y,z) Geometry," Los Alamos Scientific Laboratory report LA-6333-MS (May 1976).

68. W. F. Walters, R. D. O'Dell, and F. W. Brinkley, Jr., "THREETRAN (hex,z) Users' Manual," Los Alamos Scientific Laboratory report LA-8089-M (October 1979).

69. T. Nishimura, K. Tada, H. Tokobori, and A. Sugawara, "Development of Discrete Ordinates S_N Code in Three-Dimensional (X,Y,Z) Geometry for Shielding Design," Journal of Nucl. Sci. and Tech. 17, 539-558 (July 1980).

70. S. A. Dupree, H. A. Sandmeier, G. E. Hansen, W. W. Engle, Jr., and F. R. Mynatt, "Time-Dependent Neutron and Photon Transport Calculations Using the Method of Discrete Ordinates," USAEC report LA-4557, Los Alamos Scientific Laboratory (May 1971).

71. K. D. Lathrop, R. E. Anderson, and F. W. Brinkley, Jr., "TRANZIT: A Program for Multigroup Time-Dependent Transport in (ρ,z) Cylindrical Geometry," USAEC report LA-4575, Los Alamos Scientific Laboratory (February 1971).

72. T. R. Hill and W. H. Reed, "TIMEX: A Time-Dependent Explicit Discrete Ordinates Program for the Solution of Multigroup Transport Equations with Delayed Neutrons," Los Alamos Scientific Laboratory report LA-6201-MS (February 1976).

73. P. Daronian and Y. Chauvet, "Specifications du Code EFD," Commissariat A L'Energie Atomique report W/MA DO-168, Centre D'Etudes de Limeil, France (in French) (Oct. 1977).

APPENDIX A

CONVERTING SPHERICAL HARMONICS EXPANSION INTO COMPUTATIONAL FORM

In Sec. II, we derived the spherical harmonics expression for the scattering source as Eq. (30), which we reproduce here.

$$S_s(\vec{r},E,\vec{\Omega}) = \int_0^\infty dE' \sum_{\ell=0}^{L} \Sigma_s^\ell(\vec{r},E'\to E) \sum_{m=-\ell}^{\ell} Y_{\ell m}(\mu,\phi)\Phi_\ell^m(\vec{r},E') \quad . \tag{A1}$$

Although this form is general and succinct, it is not computationally convenient because the spherical harmonics and the flux moments Φ_ℓ^m are complex quantities. To convert Eq. (A1) into a more computationally convenient form, we expand it as

$$\begin{aligned} S_s(\vec{r},E,\vec{\Omega}) = \int_0^\infty dE' \sum_{\ell=0}^{L} \Sigma_s^\ell(\vec{r},E'\to E)\{&Y_{\ell,0}(\mu,\phi)\Phi_\ell^0(\vec{r},E') \\ &+ \sum_{m=1}^{\ell} [Y_{\ell,m}(\mu,\phi)\Phi_\ell^m(\vec{r},E') + Y_{\ell,-m}(\mu,\phi)\Phi_\ell^{-m}(\vec{r},E')]\} \quad . \end{aligned} \tag{A2}$$

We note that

$$Y_{\ell,-m}(\mu,\phi) = (-1)^m Y^*_{\ell,m}(\mu,\phi) \quad ,$$

$$\Phi_\ell^m = (-1)^m \Phi_\ell^{m*} \quad ,$$

and thus the m summation in Eq. (A2) is converted to

$$\sum_{m=1}^{\ell} [Y_{\ell,m}(\mu,\phi)\Phi_\ell^m(\vec{r},E') + Y^*_{\ell,m}(\mu,\phi)\Phi_\ell^{m*}(\vec{r},E')] \quad ,$$

which is a real quantity. If we define real and imaginary parts of the flux moments as

$$\Phi_{\ell}^{m} = \Phi_{R,\ell}^{m} - i\Phi_{I,\ell}^{m} \quad ,$$

$$Y_{\ell,m} = R_{\ell,m} + iI_{\ell,m} \quad ,$$

then the above summation may be written as

$$2\sum_{m=1}^{\ell}\left[\Phi_{R,\ell}^{m}(\vec{r},E)R_{\ell,m}(\mu,\phi) + \Phi_{I,\ell}^{m}(\vec{r},E)I_{\ell,m}(\mu,\phi)\right] \quad .$$

Further, if we write the spherical harmonics in the form

$$Y_{\ell,m}(\mu,\phi) = W_{\ell,m}(\mu)e^{im\phi} \quad ,$$

then the scattering source can be written in the following form:

$$S_s(\vec{r},E,\vec{\Omega}) = \int_0^{\infty} dE' \sum_{\ell=0}^{L} \Sigma_s^{\ell}(\vec{r},E'\rightarrow E)\{W_{\ell,0}(\mu)\Phi_{\ell}^{0}(\vec{r},E')$$

$$+ 2\sum_{m=1}^{\ell} W_{\ell,m}(\mu)[\Phi_{R,\ell}^{m}(\vec{r},E')\cos m\phi + \Phi_{I,\ell}^{m}(\vec{r},E')\sin m\phi]\} \quad ,$$

where

$$\Phi_{R,\ell}^{m}(\vec{r},E') = \int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\phi' \; \Phi(\vec{r},E',\vec{\Omega})W_{\ell,m}(\mu')\cos m\phi' \quad ,$$

$$\Phi_{I,\ell}^{m}(\vec{r},E') = \int_{-1}^{1} d\mu' \int_{0}^{2\pi} d\phi' \; \Phi(\vec{r},E',\vec{\Omega}')W_{\ell,m}(\mu')\sin m\phi \quad , \qquad \text{(A3)}$$

and the m summation is evaluated only when $\ell \geq 1$. To convert to our final computational form, we define an index, n, which counts the number of separate moments appearing in Eq. (A3). We then write the computationally compact form

$$S_s(\vec{r},E,\vec{\Omega}) = \int_0^{\infty} dE' \sum_{n=1}^{NM} (2\ell + 1)\, \Sigma_s^{\ell}(\vec{r},E' \rightarrow E) R_n(\vec{\Omega}) \tilde{\Phi}_n(\vec{r},E') \quad , \qquad (A4)$$

where NM is the number of moments.

To define the Σ_s^{ℓ}, R_n, and $\tilde{\Phi}_n$ terms, we make a one-to-one correspondence between the terms in Eqs. (A4) and (A3). Clearly, for special geometrical symmetries, the scattering source can still be reduced to Eq. (A4) with different definitions of each of the quantities therein (including the number of moments NM). Some of these are presented in Table II. Note that we have also included a $2\ell + 1$ factor explicitly in Eq. (A4) to be compatible with the forms that are incorporated in existing transport codes. Note also that the zeroth flux moment, denoted as Φ_0^0 using the notation of Eq. (A1) and as $\tilde{\Phi}_1$ using the notation of Eq. (A4), is the scalar flux that we normally denote simply as Φ_0.

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche (A01)

Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code	Page Range	NTIS Price Code
001-025	A02	151-175	A08	301-325	A14	451-475	A20
026-050	A03	176-200	A09	326-350	A15	476-500	A21
051-075	A04	201-225	A10	351-375	A16	501-525	A22
076-100	A05	226-250	A11	376-400	A17	526-550	A23
101-125	A06	251-275	A12	401-425	A18	551-575	A24
126-150	A07	276-300	A13	426-450	A19	576-600	A25
						601-up*	A99

*Contact NTIS for a price quote.